Guida per l'utente

# AWS Glue

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

# AWS Glue: Guida per l’utente

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

I marchi e l’immagine commerciale di Amazon non possono essere utilizzati in relazione a prodotti o servizi che non siano di Amazon, in una qualsiasi modalità che possa causare confusione tra i clienti o in una qualsiasi modalità che denigri o discrediti Amazon. Tutti gli altri marchi non di proprietà di Amazon sono di proprietà dei rispettivi proprietari, che possono o meno essere affiliati, collegati o sponsorizzati da Amazon.

# Table of Contents

Che cos'è AWS Glue? ..... 1
Funzionalità di AWS Glue ..... 2
Scopri le innovazioni in AWS Glue ..... 3
Nozioni di base su AWS Glue ..... 4
Accesso a AWS Glue ..... 4
Servizi correlati ..... 4
AWS Glue per la fine del supporto di Ray ..... 5
Migrazione di un lavoro Ray su Amazon Elastic Kubernetes Service ..... 5
Come funziona ..... 17
Processi ETL serverless eseguiti in isolamento ..... 18
Concetti ..... 19
AWS Glue terminologia ..... 21
componenti ..... 24
AWS Glue console ..... 24
AWS Glue Data Catalog ..... 25
AWS Glue crawler e classificatori ..... 26
AWS Glue Operazioni ETL ..... 26
Streaming di ETL in AWS Glue ..... 26
Il sistema dei lavori AWS Glue ..... 27
Componenti ETL visivi ..... 27
AWS Glue per Spark e AWS Glue per Ray ..... 33
Cosa c'è AWS Glue per Ray? ..... 34
Conversione da schemi semistrutturati a schemi relazionali ..... 35
AWS Tipi di colla ..... 37
AWS Tipi di Glue Data Catalog ..... 37
Tipi negli script AWS Glue with Spark ..... 37
AWS Tipi di Glue Crawler ..... 38
Nozioni di base ..... 39
Panoramica dell'utilizzo di AWS Glue ..... 39
Impostazione delle autorizzazioni IAM ..... 41
Fasi successive ..... 46
Configurazione di AWS Glue Studio ..... 46
Nozioni di base sui notebook in AWS Glue Studio ..... 58
Configurazione dei profili di utilizzo ..... 61

Gestione dei profili di utilizzo ... 62
Profili di utilizzo e processi ... 75
Nozioni di base su AWS Glue Data Catalog ... 76
Panoramica di ... 76
Fase 1: crea un database ... 76
Passaggio 2. Creare una tabella ... 78
Fasi successive ... 79
Impostazione dell'accesso di rete agli archivi di dati ... 82
Configurazione di un VPC per la connessione a PyPI per AWS Glue ... 84
Configurazione di DNS nel VPC ... 86
Configurazione della crittografia ... 86
Creazione di reti per lo sviluppo ... 90
Impostazione della rete per un endpoint di sviluppo ... 90
Impostazione di Amazon EC2 per un server notebook ... 92
Rilevamento e catalogazione dei dati ... 94
Popolare il catalogo dati ... 97
Usando un Crawler di AWS Glue ... 98
Definizione manuale dei metadati ... 188
Integrazione con le tabelle Amazon S3 ... 208
Integrazione con altri servizi AWS ... 218
Impostazioni del Catalogo dati ... 220
Popolamento e gestione delle tabelle transazionali ... 223
Creazione di tabelle Iceberg ... 223
Ottimizzazione delle tabelle Iceberg ... 227
Ottimizzazione delle prestazioni delle query per le tabelle Iceberg ... 269
Gestione del Catalogo dati ... 271
Aggiornamento dello schema e aggiunta di nuove partizioni ... 272
Ottimizzazione delle prestazioni delle query con le statistiche delle colonne ... 280
Crittografia del catalogo dati ... 305
Protezione del catalogo dati con Lake Formation ... 305
Lavorare con le AWS Glue Data Catalog viste ... 306
Accesso al Catalogo dati ... 310
Connessione al Data Catalog utilizzando l'endpoint REST AWS Glue Iceberg ... 311
Connessione al Data Catalog utilizzando l'endpoint di estensione AWS Glue Iceberg REST ... 314
AWS Glue REST APIs per Apache Iceberg ... 315

Connessione al Catalogo dati da un'applicazione Spark standalone ..... 334
Mappatura di dati tra Amazon Redshift e Apache Iceberg ..... 336
Considerazioni e limitazioni sull'utilizzo di AWS Glue Iceberg REST Catalog APIs ..... 337
Best practice per Catalogo dati ..... 338
Monitoraggio delle metriche di utilizzo del Data Catalog in Amazon CloudWatch ..... 339
Panoramica delle metriche del Catalogo dati ..... 339
Aggiungere metriche alla dashboard CloudWatch ..... 340
Registro dello schema di AWS Glue ..... 342
Schemi ..... 343
Registri ..... 345
Controllo delle versioni e compatibilità degli schemi ..... 346
Librerie Serde open source ..... 352
Quote del registro degli schemi ..... 352
Come funziona ..... 353
Nozioni di base ..... 355
Connessione ai dati ..... 410
Panoramica sull'utilizzo di connettori e connessioni ..... 410
Connessioni unificate ..... 411
Tipi di autenticazione di supportati ..... 415
Considerazioni ..... 416
Connessioni disponibili ..... 416
Limitazioni ..... 419
Connessioni API REST ..... 420
Creazione di un tipo di connessione REST in AWS Glue ..... 420
AWS Glue proprietà di connessione ..... 422
Proprietà di connessione richieste ..... 423
Connessione Amazon DocumentDB ..... 424
OpenSearch Connessione al servizio ..... 425
Connessione Amazon Redshift ..... 426
BigQuery Connessione Google ..... 428
Proprietà di connessione MySQL ..... 429
proprietà di connessione Oracle ..... 430
Proprietà di connessione PostgreSQL ..... 432
Connessione SAP HANA ..... 433
Connessione Snowflake ..... 435
proprietà di connessione di SQL Server ..... 437

Connessione Teradata Vantage ..... 438
Connessione Vertica ..... 439
Connessione Azure Cosmos ..... 440
Connessione Azure SQL ..... 441
Proprietà della connessione Salesforce ..... 441
Proprietà della connessione JDBC ..... 443
Proprietà di connessione MongoDB e MongoDB Atlas ..... 448
Proprietà della connessione SSL ..... 449
Proprietà della connessione Kafka per l'autenticazione ..... 452
Memorizzazione delle credenziali di connessione in Gestione dei segreti AWS ..... 453
Aggiungere una AWS Glue connessione ..... 454
Connessione ad Adobe Analytics ..... 456
Connessione ad Adobe Marketo Engage ..... 465
Connessione ad Amazon Redshift ..... 476
Connessione ad Asana ..... 481
Connessione ad Azure Cosmos DB ..... 491
Connessione ad Azure SQL ..... 494
Connessione a Edge NXT di Blackbaud Raiser ..... 497
Connessione a CircleCI ..... 507
Connessione a Datadog ..... 520
Connessione a Docusign Monitor ..... 527
Connessione a Domo ..... 535
Connessione a Dynatrace ..... 543
Connessione ad Annunci su Facebook ..... 550
Connessione a Facebook Page Insights ..... 558
Connessione a Freshdesk ..... 580
Connessione a Freshsales ..... 590
Connessione a Google Ads ..... 597
Connessione a Google Analytics 4 ..... 624
Connessione a Google BigQuery ..... 636
Connessione a Google Search Console ..... 641
Connessione a Google Sheets ..... 650
Connessione a HubSpot ..... 657
Connessione a Instagram Ads ..... 686
Connessione a Intercom ..... 694
Connessione a Jira Cloud ..... 715

Connessione a Kustomer ..... 741
Connessione a LinkedIn ..... 769
Connessione a Mailchimp ..... 777
Connessione a Microsoft Dynamics 365 CRM ..... 787
Connessione a Microsoft Teams ..... 795
Connessione a Mixpanel ..... 806
Connessione al Monday ..... 821
Connessione a MongoDB ..... 830
Connessione a Oracle NetSuite ..... 834
Connessione al OpenSearch servizio ..... 846
Connessione a Okta ..... 849
Connessione a PayPal ..... 857
Connessione a Pendo ..... 865
Connessione a Pipedrive ..... 873
Connessione a Productboard ..... 882
Connessione a QuickBooks ..... 891
Connessione a un'API REST ..... 899
Connessione a Salesforce ..... 906
Connessione a Salesforce Marketing Cloud ..... 919
Connessione a Salesforce Commerce Cloud ..... 945
Connessione a Salesforce Marketing Cloud Account Engagement ..... 955
Connessione a SAP HANA ..... 968
Connessione a SAP OData ..... 972
Connessione a SendGrid ..... 995
Connessione a ServiceNow ..... 1005
Connessione a Slack ..... 1016
Connessione a Smartsheet ..... 1028
Connessione a Snapchat Ads ..... 1039
Connessione a Snowflake ..... 1048
Connessione a Stripe ..... 1053
Connessione a Teradata ..... 1101
Connessione a Twilio ..... 1104
Connessione a Vertica ..... 1123
Connessione a WooCommerce ..... 1127
Connessione a Zendesk ..... 1155
Connessione a Zoho CRM ..... 1184

Connessione a Zoom Meetings ..... 1195
Aggiunta di una connessione JDBC utilizzando i propri driver JDBC ..... 1205
Utilizzo di connettori e connessioni personalizzati ..... 1210
Creazione di connettori personalizzati ..... 1212
Creazione di connessioni per i connettori ..... 1216
Creazione di processi con connettori personalizzati ..... 1225
Gestione di connettori e connessioni ..... 1233
Sviluppo di connettori personalizzati ..... 1236
Restrizioni per l'utilizzo di connettori e connessioni in AWS Glue Studio ..... 1238
Test di una connessione AWS Glue ..... 1239
Configurazione delle AWS chiamate in modo che passino attraverso il tuo VPC ..... 1240
La connessione a un archivio dati JDBC in un VPC ..... 1241
Accesso a dati VPC mediante interfacce di rete elastiche ..... 1241
Proprietà dell'interfaccia di rete elastica ..... 1242
Utilizzo di una connessione MongoDB o MongoDB Atlas ..... 1243
Crawling di un archivio di dati Amazon S3 utilizzando un endpoint VPC ..... 1243
Prerequisiti ..... 1244
Creazione della connessione ad Amazon S3 ..... 1245
Test della connessione ad Amazon S3 ..... 1247
Creazione di un crawler per un archivio di dati Amazon S3 ..... 1248
Creazione di un crawler per le tabelle del Catalogo dati supportate da Amazon S3 ..... 1250
Esecuzione di un crawler ..... 1251
Risoluzione dei problemi ..... 1252
Lavorare con AWS Lake Formation dati protetti ..... 1252
Accesso completo alla tabella ..... 1252
Risoluzione dei problemi di connessione ..... 1253
Tutorial: utilizzo del AWS Glue connettore per Elasticsearch ..... 1254
Prerequisiti ..... 1255
Passaggio 1: (Facoltativo) Crea un AWS segreto per le informazioni sul OpenSearch cluster ..... 1255
Fase 2: sottoscrizione al connettore ..... 1256
Fase 3: Attivare il connettore AWS Glue Studio e creare una connessione ..... 1257
Fase 4: configurazione di un ruolo IAM per il processo ETL ..... 1258
Passaggio 5: Creare un lavoro che utilizzi la connessione OpenSearch ..... 1258
Fase 6: esecuzione del processo ..... 1260
Creazione AWS Glue di posti di lavoro con sessioni interattive ..... 1261

Panoramica delle sessioni interattive in AWS Glue ..... 1261
Limitazioni ..... 1262
Nozioni di base sulle sessioni interattive AWS Glue ..... 1262
Prerequisiti per impostare le sessioni interattive a livello locale ..... 1262
Installazione di Jupyter e sessioni AWS Glue interattive, kernel Jupyter. ..... 1262
Esecuzione di Jupyter ..... 1263
Configurazione delle credenziali di sessione e della regione ..... 1263
Aggiornamento dall'anteprima delle sessioni interattive ..... 1265
Utilizzo di sessioni interattive con SageMaker AI Studio ..... 1265
Utilizzo di sessioni interattive con codice Microsoft Visual Studio ..... 1265
Sessioni Interattive con IAM ..... 1269
Configurazione delle sessioni interattive di AWS Glue per Jupyter e notebook AWS Glue Studio ..... 1277
Introduzione ai magic di Jupyter ..... 1277
Magic supportati dalle sessioni interattive di AWS Glue per Jupyter ..... 1278
Sessioni di denominazione ..... 1296
Specifica di un ruolo IAM per le sessioni interattive ..... 1297
Configurazione di sessioni con profili denominati ..... 1297
Convertire uno script o un taccuino in un lavoro AWS Glue ..... 1299
Utilizzo delle operazioni di streaming in sessioni interattive di AWS Glue ..... 1299
Commutazione del tipo di sessione streaming ..... 1299
Flusso di input di campionamento per lo sviluppo interattivo ..... 1299
Esecuzione di applicazioni di streaming in sessioni interattive ..... 1301
Prezzi delle sessioni interattive di AWS Glue ..... 1302
Configura le sessioni interattive di AWS Glue ..... 1302
Sviluppo e test a livello locale ..... 1302
Sviluppo con AWS Glue Studio ..... 1303
Sviluppo tramite le sessioni interattive ..... 1304
Sviluppare AWS Glue lavori localmente con Docker ..... 1304
Endpoint dev ..... 1314
Migrazione dagli endpoint di sviluppo alle sessioni interattive ..... 1316
Utilizzo di endpoint per lo sviluppo di script ..... 1317
Gestione di notebook ..... 1346
Creazione di processi ETL visivi ..... 1348
Crea lavori ETL visivi con AWS Glue Studio ..... 1348
Accesso alla console ..... 1348

Passaggi successivi per la creazione di un processo in AWS Glue Studio ..... 1349
Crea flussi ETL visivi con Amazon SageMaker ..... 1349
Avvio di processi ETL visivi in AWS Glue Studio ..... 1349
Avvio di processi in AWS Glue Studio ..... 1350
Caratteristiche dell'editor dei processi ..... 1351
Trasformare i dati con AWS Glue trasformazioni gestite ..... 1358
Trasforma i dati con le trasformazioni visive personalizzate ..... 1433
Utilizzo dei framework Data Lake con AWS Glue Studio ..... 1451
Connessione alle origini dati ..... 1463
Configurazione dei nodi di destinazione dati ..... 1472
Modifica o caricamento di uno script del processo ..... 1477
Modifica dei nodi padre per un nodo nel diagramma del processo ..... 1482
Eliminazione di nodi dal diagramma del processo ..... 1483
Aggiungendo parametri di origine e destinazione al nodo AWS Glue Data Catalog ..... 1487
Utilizzo dei sistemi di controllo delle versioni Git in AWS Glue ..... 1489
Creazione di codice con notebook AWS Glue Studio ..... 1498
Limitazioni ..... 1498
Panoramica sull'utilizzo dei notebook ..... 1498
Creazione di un processo ETL utilizzando i notebook in AWS Glue Studio ..... 1499
Componenti per l'editor del notebook ..... 1500
Salvataggio del notebook e dello script del processo ..... 1501
Gestione delle sessioni di notebook ..... 1502
Utilizzo di Amazon Q Developer con AWS Glue Studio notebook ..... 1504
Monitoraggi delle esecuzioni dei processi ..... 1504
Accesso al pannello di controllo di monitoraggio dei processi ..... 1505
Panoramica del pannello di controllo di monitoraggio dei processi ..... 1505
Visualizzazione esecuzioni dei processi ..... 1505
Visualizzazione dei log di esecuzione del processo ..... 1510
Visualizzazione dei dettagli di un'esecuzione di un processo ..... 1510
Visualizzazione delle Amazon CloudWatch metriche relative all'esecuzione di un job Spark ..... 1514
Visualizzazione delle Amazon CloudWatch metriche per l'esecuzione di un job Ray ..... 1515
Rileva ed elabora dati sensibili ..... 1517
Come scegliere il modo in cui desideri che vengano scansionati i dati ..... 1517
Scelta delle entità PII da rilevare ..... 1519
Specificazione del livello di distinzione di rilevamento ..... 1523

Come scegliere cosa fare con i dati PII identificati ..... 1524
Aggiungere sostituzioni di operazioni granulari ..... 1525
Gestione dei processi ..... 1525
Avviare un'esecuzione del processo ..... 1526
Pianificazione delle esecuzioni dei processi ..... 1526
Gestione delle pianificazioni dei processi ..... 1528
Interruzione dei processi ..... 1529
Visualizzazione dei processi ..... 1529
Visualizzare le informazioni sulle esecuzioni dei processi recenti ..... 1530
Visualizzare lo script del processo ..... 1531
Modificare le proprietà del processo ..... 1532
Salvare il lavoro ..... 1535
Clonazione di un processo ..... 1537
Eliminazione dei processi ..... 1537
Utilizzo dei processi ..... 1539
AWS Glue versioni ..... 1539
AWS Glue versioni ..... 1539
Policy di supporto versione AWS Glue ..... 1559
Migrazione AWS Glue per i job Spark alla versione 5.1 AWS Glue ..... 1562
Migrazione AWS Glue per i job Spark alla versione 5.0 AWS Glue ..... 1569
Migrazione AWS Glue per i job Spark alla versione 4.0 AWS Glue ..... 1584
Analisi degli aggiornamenti con l'IA ..... 1599
Utilizzo dei processi Spark ..... 1615
Parametri Job ..... 1616
Spark e lavori PySpark ..... 1626
Tipi di worker ..... 1794
Aggiunta di processi di streaming ETL ..... 1799
Registra la corrispondenza con FindMatches ..... 1814
Migrare i programmi Spark ..... 1851
Utilizzo dei processi Ray ..... 1859
Guida introduttiva AWS Glue per Ray ..... 1860
Ambienti di runtime Ray supportati ..... 1861
Contabilità per i worker nei processi Ray ..... 1862
Parametri dei processi Ray ..... 1863
Parametri dei processi Ray ..... 1866
Configurazione delle proprietà dei processi shell di Python ..... 1868

Limitazioni ... 1868
Ambiente di esecuzione ... 1869
Definire le proprietà del processo per i processi shell di Python ... 1869
Librerie supportate dai processi shell di Python ... 1871
Fornire la propria libreria Python ... 1873
Utilizzabile AWS CloudFormation con i lavori della shell Python in AWS Glue ... 1877
Migrazione dalla shell 3.6 di Python alla 3.9 ... 1877
Migrazione dai job della AWS shell Glue Python ... 1877
Monitoraggio ... 1879
AWS tag ... 1880
Automatizzare con EventBridge ... 1886
Monitoraggio delle risorse ... 1889
Registrazione utilizzando CloudTrail ... 1891
AWS Glue Streaming ... 1896
Casi d'uso per lo streaming ... 1896
Quali sono i vantaggi dell'utilizzo AWS Glue dello streaming? ... 1897
Quando usare lo streaming? AWS Glue ... 1898
Origini dati supportate ... 1899
Destinazioni di dati supportate ... 1899
Tutorial: crea il tuo primo carico di lavoro in streaming con Studio AWS Glue ... 1900
Prerequisiti ... 1900
Utilizzo dei dati in streaming da Amazon Kinesis ... 1900
Tutorial: crea il tuo primo carico di lavoro in streaming utilizzando i notebook AWS Glue Studio ... 1911
Prerequisiti ... 1912
Utilizzo dei dati in streaming da Amazon Kinesis ... 1912
Concetti relativi allo streaming ... 1919
Anatomia di un lavoro AWS Glue in streaming ... 1919
Connessioni streaming ... 1923
Connessioni Kafka ... 1923
Connessioni Kinesis ... 1930
AWS Glue scalabilità automatica in streaming ... 1937
Attivazione dell'Auto Scaling in AWS Glue Studio ... 1937
Abilitazione dell'Auto Scaling con CLI o SDK AWS ... 1938
Come funziona ... 1939
Finestre di manutenzione ... 1941

Impostazione della finestra di manutenzione ..... 1941
Comportamento della finestra di manutenzione ..... 1942
Monitoraggio del lavoro ..... 1943
Gestione della perdita di dati ..... 1945
Concetti avanzati relativi allo streaming in AWS Glue ..... 1946
Considerazioni di carattere temporale relative all'elaborazione dei flussi ..... 1946
Raggruppamenti in finestre ..... 1947
Gestione di dati in ritardo e filigrane ..... 1953
Monitoraggio dei lavori di AWS Glue streaming ..... 1954
Visualizzazione delle metriche di streaming AWS Glue ..... 1955
Utilizzo delle metriche di AWS Glue streaming ..... 1957
Come ottenere le prestazioni migliori ..... 1962
Zero-ETL integrazioni ..... 1964
Zero-ETL capacità in AWS Glue ..... 1964
Configurazione di un'origine ..... 1965
Origine Amazon DynamoDB ..... 1966
Fonte di connessione SaaS ..... 1967
Oracle Database presso AWS ..... 1982
Configurazione di una destinazione ..... 1982
Target S3 per uso generico ..... 1983
Destinazione di Tabelle S3 ..... 1991
Amazon Redshift Obiettivo di storage gestito ..... 1996
Amazon Redshift destinazione del data warehouse ..... 1997
Configurazione della destinazione dell'integrazione ..... 1999
Configurazione di un'integrazione ..... 2000
Intervallo di aggiornamento ..... 2000
Istantanea su richiesta ..... 2002
Modifica dell'intervallo di aggiornamento ..... 2003
Creazione e gestione di integrazioni ..... 2003
Creazione di un'integrazione ..... 2003
Modifica di un'integrazione ..... 2005
Eliminazione di un'integrazione ..... 2006
Stati di un'integrazione ..... 2006
Monitoraggio di un'integrazione ..... 2007
Visualizzazione dei log di ..... 2007
Visualizzazione delle metriche ..... 2010

Notifiche degli eventi ..... 2010
Unnesting dello schema e partizionamento dei dati ..... 2011
Comportamento predefinito di disnidificazione e partizionamento ..... 2012
Disnidificazione dello schema ..... 2012
Partizionamento dei dati ..... 2017
Limitazioni ..... 2024
Usando APIs ..... 2025
Limitazioni ..... 2026
Service Quotas ..... 2027
Regioni supportate ..... 2027
Risoluzione dei problemi ..... 2029
Fonte DynamoDB ..... 2030
Origine SaaS ..... 2030
Obiettivo S3 ..... 2030
Obiettivo S3-Table ..... 2031
Risoluzione dei problemi generali ..... 2031
AWS Glue Qualità dei dati ..... 2032
Vantaggi e funzionalità principali ..... 2032
Come funziona ..... 2033
Qualità dei dati per AWS Glue Data Catalog ..... 2033
Qualità dei dati per AWS Glue lavori ETL ..... 2034
Confronto AWS Glue dei punti di ingresso relativi alla qualità dei dati ..... 2034
Considerazioni ..... 2036
Terminologia ..... 2036
Limits ..... 2037
Note di rilascio per AWS Glue Data Quality ..... 2038
Disponibilità generale: nuove funzionalità ..... 2038
27 novembre 2023 (anteprima) ..... 2038
12 marzo 2024 ..... 2038
26 giugno 2024 ..... 2039
7 agosto 2024 ..... 2039
22 novembre 2024 ..... 2039
6 dicembre 2024 ..... 2039
7 luglio 2025 ..... 2039
21 novembre 2025 ..... 2040
Rilevamento delle anomalie in Qualità dei dati di AWS Glue ..... 2040

Come funziona ..... 2041
Dettagli dell'algoritmo di rilevamento delle anomalie ..... 2045
Autorizzazioni IAM per AWS Glue Data Quality ..... 2046
autorizzazioni IAM ..... 2047
Configurazione IAM richiesta per la pianificazione delle esecuzioni di valutazione ..... 2050
Policy IAM di esempio ..... 2051
Guida introduttiva a AWS Glue Data Quality per Data Catalog ..... 2056
Prerequisiti ..... 2057
Step-by-step esempio ..... 2057
Generazione di raccomandazioni di regole ..... 2057
Monitoraggio dei suggerimenti di regole ..... 2059
Modifica dei set di regole suggeriti ..... 2059
Creazione di un nuovo set di regole ..... 2061
Esecuzione di un set di regole per valutare la qualità dei dati ..... 2062
Visualizzazione del punteggio e dei risultati della qualità dei dati ..... 2064
Utilizzo di interrogazioni di preelaborazione ..... 2064
Tipi di origini supportate ..... 2068
Argomenti correlati ..... 2069
Valutazione della qualità dei dati con AWS Glue Studio ..... 2069
Vantaggi ..... 2070
Valutazione della qualità dei dati per i processi ETL in AWS Glue Studio ..... 2070
Generatore di regole di qualità dei dati ..... 2082
Configurazione del rilevamento delle anomalie nei job AWS Glue ETL ..... 2087
Visualizzazione dei punteggi e delle anomalie relativi alla qualità dei dati ..... 2090
Qualità dei dati per lavori ETL nei notebook AWS Glue Studio ..... 2094
Prerequisiti ..... 2095
Creazione di un processo ETL in AWS Glue Studio ..... 2095
Riferimento a Data Quality Definition Language (DQDL) ..... 2100
Sintassi ..... 2102
Documentazione di riferimento del tipo di regola ..... 2122
Utilizzo APIs per misurare e gestire la qualità dei dati ..... 2186
Prerequisiti ..... 2187
Utilizzo dei consigli di AWS Glue Data Quality ..... 2187
Utilizzo dei set di regole AWS Glue Data Quality ..... 2190
L'utilizzo di AWS Glue Data Quality funziona ..... 2193
Utilizzo dei risultati di AWS Glue Data Quality ..... 2198

Configurazione di avvisi, implementazioni e pianificazioni ..... 2203
Configurazione di avvisi e notifiche nell'integrazione con Amazon EventBridge ..... 2204
Imposta avvisi e notifiche nell'integrazione CloudWatch ..... 2212
Esecuzione di query sui risultati di qualità dei dati ..... 2214
Implementazione di regole di qualità dei dati ..... 2218
Pianificazione delle regole di qualità dei dati ..... 2218
Crittografia dei dati a riposo per AWS Glue Data Quality ..... 2218
AWS chiavi di proprietà ..... 2218
Chiavi gestite dal cliente ..... 2219
Creazione di una chiave gestita dal cliente ..... 2221
Creazione di una configurazione di sicurezza ..... 2224
AWS Contesto di crittografia Glue Data Quality ..... 2225
Monitoraggio delle chiavi di crittografia per AWS Glue Data Quality ..... 2226
Ulteriori informazioni ..... 2230
Risoluzione degli errori di AWS Glue Data Quality ..... 2231
Errore: modulo assente ..... 2232
Errore: autorizzazioni insufficienti ..... 2232
Errore: set di regole non univoci ..... 2232
Errore: tabelle con caratteri speciali ..... 2232
Errore: overflow con un set di regole di grandi dimensioni ..... 2232
Errore: lo stato della regola è non riuscito ..... 2233
AnalysisException: impossibile verificare l'esistenza del database predefinito ..... 2233
La mappa delle chiavi fornita non è adatta a determinati frame di dati ..... 2234
java.lang. RuntimeException : Impossibile recuperare i dati. ..... 2234
ERRORE DI AVVIO: errore durante il download da S3 per il bucket ..... 2234
InvalidInputException (status: 400): DataQuality le regole non possono essere analizzate . 2235
Errore: EventBridge non attiva i processi Qualità dei dati di Glue in base alla pianificazione che ho impostato ..... 2235
Errori CustomSQL ..... 2235
Regole dinamiche ..... 2236
Eccezione nella classe utente: org.apache.spark.sql. AnalysisException: org.apache.hadoop.hive.ql.metadata. HiveException ..... 2238
UNCLASSIFIED_ERROR; IllegalArgumentException: Errore di analisi: non sono state fornite regole o analizzatori., nessuna valida alternativa in fase di input ..... 2239
Integrazione dei dati di Amazon Q in AWS Glue ..... 2240
Che cos'è Amazon Q? ..... 2240

Integrazione dei dati di Amazon Q in AWS Glue ..... 2240
Utilizzo dell'integrazione dei dati di Amazon Q ..... 2241
Best practice ..... 2243
Miglioramento del servizio ..... 2243
Considerazioni ..... 2244
Configurazione dell'integrazione dei dati di Amazon Q ..... 2244
Configurazione delle autorizzazioni IAM ..... 2244
Generazione di codice supportata ..... 2247
Interazioni di esempio ..... 2248
Interazioni con la chat di Amazon Q ..... 2248
AWS Glue Interazioni con i notebook da studio ..... 2249
Utilizzo della consapevolezza del contesto ..... 2253
Interazioni di esempio ..... 2253
Limitazioni ..... 2257
Orchestrazione ..... 2258
Avvio di lavori e crawler utilizzando i trigger ..... 2258
Trigger di AWS Glue ..... 2258
Aggiunta di trigger ..... 2261
Attivazione e disattivazione dei trigger ..... 2265
Esecuzione di attività ETL complesse utilizzando gli schemi e i flussi di lavoro ..... 2266
Panoramica di flussi di lavoro ..... 2267
Creazione e costruzione manuale di un flusso di lavoro ..... 2271
Avvio di un flusso di lavoro con un EventBridge evento ..... 2276
Visualizzazione degli EventBridge eventi che hanno avviato un flusso di lavoro ..... 2283
Esecuzione e monitoraggio di un flusso di lavoro ..... 2284
Arresto dell'esecuzione di un flusso di lavoro ..... 2287
Ripresa e ripristino dell'esecuzione di un flusso di lavoro ..... 2288
Recupero e impostazione delle proprietà di esecuzione del flusso di lavoro ..... 2294
Interrogazione dei flussi di lavoro utilizzando AWS Glue API ..... 2295
Restrizioni dei flussi di lavoro e degli schemi ..... 2300
Risoluzione degli errori relativi agli schemi ..... 2301
Autorizzazioni per utenti e ruoli per gli schemi ..... 2306
Sviluppo di schemi ..... 2310
Panoramica degli schemi ..... 2311
Sviluppo di schemi ..... 2314
Registrazione di uno schema ..... 2339

Visualizzazione degli schemi ... 2341
Aggiornamento di uno schema ... 2343
Creazione di un flusso di lavoro da uno schema ... 2345
Visualizzazione delle esecuzioni dello schema ... 2347
AWS CloudFormation per AWS Glue ... 2349
Database di esempio ... 2351
Esempio di database, tabella e partizioni ... 2352
Classificatore Grock di esempio ... 2356
Classificatore JSON di esempio ... 2358
Classificatore XML di esempio ... 2359
Esempio di crawler Amazon S3 ... 2360
Esempio di connessione ... 2362
Esempio di connessione ... 2363
Esempio di connessione ... 2364
Esempio di crawler JDBC ... 2366
Esempio di processo da Amazon S3 ad Amazon S3 ... 2369
Esempio di processo per il trasferimento da JDBC ad Amazon S3 ... 2370
Esempio di trigger on demand ... 2372
Esempio di trigger pianificato ... 2373
Esempio di trigger condizionale ... 2374
Esempio di trasformazione basata su machine learning ... 2376
Set di regole di qualità dei dati di esempio ... 2377
Esempio di set di regole sulla qualità dei dati con scheduler EventBridge ... 2378
Esempio di endpoint di sviluppo ... 2381
AWS Glue guida alla programmazione ... 2383
Fornire i propri script personalizzati ... 2383
AWS Glue per Spark ... 2384
Tutorial: scrittura di uno script Spark ... 2384
ETL in PySpark ... 2398
ETL in Scala ... 2648
Caratteristiche e ottimizzazioni ... 2734
AWS Glue per Ray ... 3012
Tutorial: scrittura di uno script Ray ... 3012
Utilizzo di Ray Core e Ray Data in AWS Glue for Ray ... 3019
Fornitura di file e librerie Python ... 3021
Connessione ai dati ... 3026

Lavorare con AWS gli SDK ..... 3029
AWS Glue API ..... 3031
Sicurezza ..... 3059
- tipi di dati - ..... 3059
DataCatalogEncryptionSettings ..... 3059
EncryptionAtRest ..... 3060
ConnectionPasswordEncryption ..... 3060
EncryptionConfiguration ..... 3061
S3Encryption ..... 3062
CloudWatchEncryption ..... 3062
JobBookmarksEncryption ..... 3062
SecurityConfiguration ..... 3063
GluePolicy ..... 3063
DataQualityEncryption ..... 3064
- operazioni - ..... 3064
GetDataCatalogEncryptionSettings (get_data_catalog_encryption_settings) ..... 3064
PutDataCatalogEncryptionSettings (put_data_catalog_encryption_settings) ..... 3065
PutResourcePolicy (put_resource_policy) ..... 3066
GetResourcePolicy (get_resource_policy) ..... 3067
DeleteResourcePolicy (delete_resource_policy) ..... 3068
CreateSecurityConfiguration (create_security_configuration) ..... 3069
DeleteSecurityConfiguration (delete_security_configuration) ..... 3070
GetSecurityConfiguration (get_security_configuration) ..... 3070
GetSecurityConfigurations (get_security_configurations) ..... 3071
GetResourcePolicies (get_resource_policies) ..... 3072
Oggetti del catalogo ..... 3073
Cataloghi ..... 3073
Database ..... 3088
Tabelle ..... 3098
Partizioni ..... 3147
Connessioni ..... 3173
Funzioni definite dall'utente ..... 3217
Importazione di un catalogo Athena ..... 3224
Ottimizzatore di tabelle ..... 3226
- tipi di dati - ..... 3226
TableOptimizer ..... 3226

TableOptimizerConfiguration ..... 3227
TableOptimizerVpcConfiguration ..... 3228
CompactionConfiguration ..... 3228
IcebergCompactionConfiguration ..... 3229
TableOptimizerRun ..... 3230
BatchGetTableOptimizerEntry ..... 3231
BatchTableOptimizer ..... 3232
BatchGetTableOptimizerError ..... 3233
RetentionConfiguration ..... 3233
IcebergRetentionConfiguration ..... 3233
OrphanFileDeletionConfiguration ..... 3234
IcebergOrphanFileDeletionConfiguration ..... 3234
CompactionMetrics ..... 3235
RetentionMetrics ..... 3235
OrphanFileDeletionMetrics ..... 3236
IcebergCompactionMetrics ..... 3236
IcebergRetentionMetrics ..... 3236
IcebergOrphanFileDeletionMetrics ..... 3237
RunMetrics ..... 3237
- operazioni - ..... 3238
GetTableOptimizer (get_table_optimizer) ..... 3238
BatchGetTableOptimizer (batch_get_table_optimizer) ..... 3239
ListTableOptimizerRuns (list_table_optimizer_runs) ..... 3240
CreateTableOptimizer (create_table_optimizer) ..... 3242
DeleteTableOptimizer (delete_table_optimizer) ..... 3243
UpdateTableOptimizer (update_table_optimizer) ..... 3244
Crawler e classificatori ..... 3245
Classificatori ..... 3245
Crawler ..... 3259
Statistiche delle colonne ..... 3287
Pianificatore ..... 3304
Script ETL auto-generanti ..... 3306
- tipi di dati - ..... 3307
CodeGenNode ..... 3307
CodeGenNodeArg ..... 3307
CodeGenEdge ..... 3308

Location (Ubicazione) ..... 3308
CatalogEntry ..... 3309
MappingEntry ..... 3309
- operazioni - ..... 3310
CreateScript (create_script) ..... 3310
GetDataflowGraph (get_dataflow_graph) ..... 3311
GetMapping (get_mapping) ..... 3311
GetPlan (get_plan) ..... 3312
API processo visuale ..... 3314
- tipi di dati - ..... 3314
CodeGenConfigurationNode ..... 3318
JDBCConnectorOpzioni ..... 3325
StreamingDataPreviewOptions ..... 3327
AthenaConnectorSource ..... 3327
JDBCConnectorFonte ..... 3328
SparkConnectorSource ..... 3329
CatalogSource ..... 3329
La mia SQLCatalog fonte ..... 3330
Fonte Postgre SQLCatalog ..... 3330
SQLCatalogFonte Oracle ..... 3331
Microsoft SQLServer CatalogSource ..... 3331
CatalogKinesisSource ..... 3332
DirectKinesisSource ..... 3332
KinesisStreamingSourceOptions ..... 3333
CatalogKafkaSource ..... 3336
DirectKafkaSource ..... 3336
KafkaStreamingSourceOptions ..... 3337
RedshiftSource ..... 3339
AmazonRedshiftSource ..... 3340
AmazonRedshiftNodeData ..... 3340
AmazonRedshiftAdvancedOption ..... 3343
Opzione ..... 3343
S3 CatalogSource ..... 3344
S3 SourceAdditionalOptions ..... 3344
S3 CsvSource ..... 3345
Diretto JDBCSource ..... 3347

S3 DirectSourceAdditionalOptions ..... 3348
S3 JsonSource ..... 3348
S3 ParquetSource ..... 3350
S3 DeltaSource ..... 3351
S3 CatalogDeltaSource ..... 3352
CatalogDeltaSource ..... 3353
S3 HudiSource ..... 3353
S3 CatalogHudiSource ..... 3354
S3 ExcelSource ..... 3355
CatalogHudiSource ..... 3356
Sorgente Dynamo DBCatalog ..... 3357
RelationalCatalogSource ..... 3357
JDBCConnectorObiettivo ..... 3358
SparkConnectorTarget ..... 3359
BasicCatalogTarget ..... 3360
Il mio SQLCatalog obiettivo ..... 3360
Postgre Target SQLCatalog ..... 3361
Oracle SQLCatalog Target ..... 3361
Microsoft SQLServer CatalogTarget ..... 3362
RedshiftTarget ..... 3362
AmazonRedshiftTarget ..... 3363
UpsertRedshiftTargetOptions ..... 3363
S3 CatalogTarget ..... 3364
S3 GlueParquetTarget ..... 3365
CatalogSchemaChangePolicy ..... 3366
S3 DirectTarget ..... 3366
S3 HudiCatalogTarget ..... 3367
S3 HudiDirectTarget ..... 3368
S3 DeltaCatalogTarget ..... 3369
S3 DeltaDirectTarget ..... 3370
S3 HyperDirectTarget ..... 3372
S3 IcebergDirectTarget ..... 3373
DirectSchemaChangePolicy ..... 3374
ApplyMapping ..... 3375
Mapping ..... 3375
SelectFields ..... 3376

DropFields ..... 3376
RenameField ..... 3377
Spigot ..... 3377
Join ..... 3378
JoinColumn ..... 3379
SplitFields ..... 3379
SelectFromCollection ..... 3379
FillMissingValues ..... 3380
Filtro ..... 3380
FilterExpression ..... 3381
FilterValue ..... 3381
CustomCode ..... 3382
SparkSQL ..... 3382
SqlAlias ..... 3383
DropNullFields ..... 3383
NullCheckBoxList ..... 3384
NullValueField ..... 3384
DataType ..... 3385
Unione ..... 3385
Union ..... 3386
PIIDetection ..... 3386
Aggregazione ..... 3388
DropDuplicates ..... 3389
GovernedCatalogTarget ..... 3389
GovernedCatalogSource ..... 3390
AggregateOperation ..... 3390
GlueSchema ..... 3391
GlueStudioSchemaColumn ..... 3391
GlueStudioColumn ..... 3391
DynamicTransform ..... 3393
TransformConfigParameter ..... 3394
EvaluateDataQuality ..... 3394
DQResultsPublishingOptions ..... 3395
DQStopJobOnFailureOptions ..... 3396
EvaluateDataQualityMultiFrame ..... 3396
Recipe ..... 3397

RecipeReference ... 3397
SnowflakeNodeData ... 3398
SnowflakeSource ... 3400
SnowflakeTarget ... 3401
ConnectorDataSource ... 3401
ConnectorDataTarget ... 3402
RecipeStep ... 3403
RecipeAction ... 3403
ConditionExpression ... 3404
S3 CatalogIcebergSource ... 3404
CatalogIcebergSource ... 3405
S3 IcebergCatalogTarget ... 3405
Sorgente Dynamo DBELTConnector ... 3406
DDBELTConnectionOpzioni ... 3407
DDBELTCatalogAdditionalOptions ... 3408
Route ... 3408
GroupFilters ... 3409
AutoDataQuality ... 3409
Jobs ... 3410
Jobs ... 3410
Esecuzioni di processo ... 3437
Triggers ... 3458
Integrazione APIs ... 3473
- tipi di dati - ... 3473
Integrazione ... 3473
IntegrationConfig ... 3475
IntegrationPartition ... 3475
IntegrationError ... 3477
IntegrationFilter ... 3477
InboundIntegration ... 3477
SourceProcessingProperties ... 3478
TargetProcessingProperties ... 3479
SourceTableConfig ... 3479
TargetTableConfig ... 3480
- operazioni - ... 3480
CreateIntegration (create_integration) ... 3481

ModifyIntegration (modify_integration) ..... 3484
DescribeIntegrations (describe_integrazioni) ..... 3487
DeleteIntegration (delete_integration) ..... 3488
DescribeInboundIntegrations (describe_inbound_integrazioni) ..... 3490
CreateIntegrationTableProperties (create_integration_table_properties) ..... 3491
UpdateIntegrationTableProperties (update_integration_table_properties) ..... 3492
GetIntegrationTableProperties (get_integration_table_properties) ..... 3494
DeleteIntegrationTableProperties (delete_integration_table_properties) ..... 3495
CreateIntegrationResourceProperty (create_integration_resource_property) ..... 3496
UpdateIntegrationResourceProperty (update_integration_resource_property) ..... 3497
GetIntegrationResourceProperty (get_integration_resource_property) ..... 3498
UntagResource (untag_resource) ..... 3499
ListTagsForResource (list_tags_per_resource) ..... 3500
- eccezioni - ..... 3500
ResourceNotFoundException ..... 3501
InternalServerException ..... 3501
IntegrationAlreadyExistsFault ..... 3501
IntegrationConflictOperationFault ..... 3502
IntegrationQuotaExceededFault ..... 3502
KMSKeyNotAccessibleFault ..... 3502
IntegrationNotFoundFault ..... 3502
TargetResourceNotFound ..... 3503
InvalidIntegrationStateFault ..... 3503
Sessioni interattive ..... 3503
- tipi di dati - ..... 3503
Sessione ..... 3503
SessionCommand ..... 3506
Dichiarazione ..... 3506
StatementOutput ..... 3507
StatementOutputData ..... 3508
ConnectionsList ..... 3508
- operazioni - ..... 3508
CreateSession (crea_sessione) ..... 3509
StopSession (stop_session) ..... 3512
DeleteSession (delete_session) ..... 3513
GetSession (get_session) ..... 3514

ListSessions (lista_sessioni) ..... 3515
RunStatement (run_statement) ..... 3516
CancelStatement (dichiarazione di annullamento) ..... 3517
GetStatement (get_statement) ..... 3518
ListStatements (list_statements) ..... 3519
GetGlueIdentityCenterConfiguration (get_glue_identity_center_configuration) ..... 3520
UpdateGlueIdentityCenterConfiguration (update_glue_identity_center_configuration) ..... 3521
CreateGlueIdentityCenterConfiguration (create_glue_identity_center_configuration) ..... 3521
DeleteGlueIdentityCenterConfiguration (delete_glue_identity_center_configuration) ..... 3522
DevEndpoints ..... 3523
- tipi di dati - ..... 3523
DevEndpoint ..... 3523
DevEndpointCustomLibraries ..... 3527
- operazioni - ..... 3528
CreateDevEndpoint (create_dev_endpoint) ..... 3528
UpdateDevEndpoint (update_dev_endpoint) ..... 3534
DeleteDevEndpoint (delete_dev_endpoint) ..... 3535
GetDevEndpoint (get_dev_endpoint) ..... 3536
GetDevEndpoints (get_dev_endpoints) ..... 3537
BatchGetDevEndpoints (batch_get_dev_endpoints) ..... 3538
ListDevEndpoints (list_dev_endpoints) ..... 3539
Registro degli schemi ..... 3540
- tipi di dati - ..... 3540
RegistryId ..... 3540
RegistryListItem ..... 3541
MetadataInfo ..... 3542
OtherMetadataValueListItem ..... 3542
SchemaListItem ..... 3542
SchemaVersionListItem ..... 3543
MetadataKeyValuePair ..... 3544
SchemaVersionErrorItem ..... 3544
ErrorDetails ..... 3545
SchemaVersionNumber ..... 3545
SchemaId ..... 3545
- operazioni - ..... 3546
CreateRegistry (create_registry) ..... 3547

CreateSchema (crea_schema) ..... 3548
GetSchema (get_schema) ..... 3552
ListSchemaVersions (list_schema_versions) ..... 3554
GetSchemaVersion (get_schema_version) ..... 3556
GetSchemaVersionsDiff (get_schema_versions_diff) ..... 3557
ListRegistries (list_registries) ..... 3558
ListSchemas (list_schemas) ..... 3559
RegisterSchemaVersion (register_schema_version) ..... 3560
UpdateSchema (update_schema) ..... 3562
CheckSchemaVersionValidity (check_schema_version_idity) ..... 3563
UpdateRegistry (update_registry) ..... 3564
GetSchemaByDefinition (get_schema_by_definition) ..... 3565
GetRegistry (get_registry) ..... 3567
PutSchemaVersionMetadata (put_schema_version_metadata) ..... 3568
QuerySchemaVersionMetadata (query_schema_version_metadata) ..... 3570
RemoveSchemaVersionMetadata (remove_schema_version_metadata) ..... 3571
DeleteRegistry (delete_registry) ..... 3573
DeleteSchema (delete_schema) ..... 3574
DeleteSchemaVersions (delete_schema_versions) ..... 3575
Flussi di lavoro ..... 3576
- tipi di dati - ..... 3576
JobNodeDetails ..... 3577
CrawlerNodeDetails ..... 3577
TriggerNodeDetails ..... 3577
Crawl ..... 3577
Nodo ..... 3578
Edge ..... 3579
Flusso di lavoro ..... 3579
WorkflowGraph ..... 3581
WorkflowRun ..... 3581
WorkflowRunStatistics ..... 3582
StartingEventBatchCondition ..... 3583
Blueprint ..... 3584
BlueprintDetails ..... 3585
LastActiveDefinition ..... 3585
BlueprintRun ..... 3586

- operazioni - ..... 3587
CreateWorkflow (create_workflow) ..... 3588
UpdateWorkflow (update_workflow) ..... 3589
DeleteWorkflow (delete_workflow) ..... 3591
GetWorkflow (get_workflow) ..... 3591
ListWorkflows (list_workflows) ..... 3592
BatchGetWorkflows (batch_get_workflows) ..... 3593
GetWorkflowRun (get_workflow_run) ..... 3594
GetWorkflowRuns (get_workflow_runs) ..... 3595
GetWorkflowRunProperties (get_workflow_run_properties) ..... 3596
PutWorkflowRunProperties (put_workflow_run_properties) ..... 3597
CreateBlueprint (create_blueprint) ..... 3598
UpdateBlueprint (update_blueprint) ..... 3599
DeleteBlueprint (delete_blueprint) ..... 3600
ListBlueprints (list_blueprints) ..... 3600
BatchGetBlueprints (batch_get_blueprints) ..... 3601
StartBlueprintRun (start_blueprint_run) ..... 3602
GetBlueprintRun (get_blueprint_run) ..... 3603
GetBlueprintRuns (get_blueprint_runs) ..... 3604
StartWorkflowRun (avvio_workflow_run) ..... 3605
StopWorkflowRun (stop_workflow_run) ..... 3606
ResumeWorkflowRun (resume_workflow_run) ..... 3606
Profili di utilizzo ..... 3608
- tipi di dati - ..... 3608
ProfileConfiguration ..... 3608
ConfigurationObject ..... 3608
UsageProfileDefinition ..... 3609
- operazioni - ..... 3610
CreateUsageProfile (create_usage_profile) ..... 3610
GetUsageProfile (get_usage_profile) ..... 3611
UpdateUsageProfile (update_usage_profile) ..... 3612
DeleteUsageProfile (delete_usage_profile) ..... 3613
ListUsageProfiles (list_usage_profiles) ..... 3613
Machine learning ..... 3614
- tipi di dati - ..... 3614
TransformParameters ..... 3615

EvaluationMetrics ... 3616
MLTransform ... 3616
FindMatchesParameters ... 3619
FindMatchesMetrics ... 3621
ConfusionMatrix ... 3622
GlueTable ... 3623
TaskRun ... 3624
TransformFilterCriteria ... 3625
TransformSortCriteria ... 3626
TaskRunFilterCriteria ... 3626
TaskRunSortCriteria ... 3627
TaskRunProperties ... 3627
FindMatchesTaskRunProperties ... 3628
ImportLabelsTaskRunProperties ... 3629
ExportLabelsTaskRunProperties ... 3629
LabelingSetGenerationTaskRunProperties ... 3629
SchemaColumn ... 3630
TransformEncryption ... 3630
MLUserDataEncryption ... 3631
ColumnImportance ... 3631
- operazioni - ... 3631
MLTransform Crea (create_ml_transform) ... 3632
MLTransform Aggiorna (update_ml_transform) ... 3636
MLTransform Elimina (delete_ml_transform) ... 3638
MLTransform Ottieni (get_ml_transform) ... 3639
MLTransforms Ottieni (get_ml_transforms) ... 3642
MLTransforms Elenco (list_ml_transforms) ... 3643
MLEvaluationTaskRun Avvio (start_ml_evaluation_task_run) ... 3644
Avvia MLLabeling SetGenerationTaskRun (start_ml_labeling_set_generation_task_run) ... 3645
Get MLTask Run (get_ml_task_run) ... 3647
Get MLTask Runs (get_ml_task_runs) ... 3648
Annulla MLTask esecuzione (cancel_ml_task_run) ... 3650
StartExportLabelsTaskRun (start_export_labels_task_run) ... 3651
StartImportLabelsTaskRun (start_import_labels_task_run) ... 3652
Qualità dei dati ... 3653
- tipi di dati - ... 3653

DataSource ..... 3654
DataQualityRulesetListDetails ..... 3655
DataQualityTargetTable ..... 3655
DataQualityRulesetEvaluationRunDescription ..... 3656
DataQualityRulesetEvaluationRunFilter ..... 3657
DataQualityEvaluationRunAdditionalRunOptions ..... 3657
DataQualityRuleRecommendationRunDescription ..... 3658
DataQualityRuleRecommendationRunFilter ..... 3658
DataQualityResult ..... 3659
DataQualityAnalyzerResult ..... 3660
DataQualityObservation ..... 3661
MetricBasedObservation ..... 3662
DataQualityMetricValues ..... 3662
DataQualityRuleResult ..... 3663
DataQualityResultDescription ..... 3664
DataQualityResultFilterCriteria ..... 3664
DataQualityRulesetFilterCriteria ..... 3665
DataQualityAggregatedMetrics ..... 3666
StatisticAnnotation ..... 3667
TimestampedInclusionAnnotation ..... 3667
AnnotationError ..... 3667
DatapointInclusionAnnotation ..... 3668
StatisticSummaryList ..... 3668
StatisticSummary ..... 3669
RunIdentifier ..... 3670
StatisticModelResult ..... 3670
DataQualityGlueTable ..... 3671
- operazioni - ..... 3672
StartDataQualityRulesetEvaluationRun (start_data_quality_ruleset_evaluation_run) ..... 3673
CancelDataQualityRulesetEvaluationRun (cancel_data_quality_ruleset_evaluation_run) .... 3675
GetDataQualityRulesetEvaluationRun (get_data_quality_ruleset_evaluation_run) ..... 3676
ListDataQualityRulesetEvaluationRuns (list_data_quality_ruleset_evaluation_runs) ..... 3678
StartDataQualityRuleRecommendationRun (start_data_quality_rule_recommendation_run) 3679
CancelDataQualityRuleRecommendationRun
(cancel_data_quality_rule_recommendation_run) ..... 3680
GetDataQualityRuleRecommendationRun (get_data_quality_rule_recommendation_run) .. 3681

ListDataQualityRuleRecommendationRuns (list_data_quality_rule_recommendation_runs) 3683
GetDataQualityResult (get_data_quality_result) ... 3684
BatchGetDataQualityResult (batch_get_data_quality_result) ... 3686
ListDataQualityResults (list_data_quality_results) ... 3686
CreateDataQualityRuleset (create_data_quality_ruleset) ... 3687
DeleteDataQualityRuleset (delete_data_quality_ruleset) ... 3689
GetDataQualityRuleset (get_data_quality_ruleset) ... 3690
ListDataQualityRulesets (list_data_quality_rulesets) ... 3691
UpdateDataQualityRuleset (update_data_quality_ruleset) ... 3692
ListDataQualityStatistics (list_data_quality_statistics) ... 3693
TimestampFilter ... 3694
CreateDataQualityRulesetRequest ... 3695
GetDataQualityRulesetResponse ... 3696
GetDataQualityResultResponse ... 3697
StartDataQualityRuleRecommendationRunRequest ... 3698
GetDataQualityRuleRecommendationRunResponse ... 3699
BatchPutDataQualityStatisticAnnotation (batch_put_data_quality_statistic_annotation) ... 3701
GetDataQualityModel (get_data_quality_model) ... 3702
GetDataQualityModelResult (get_data_quality_model_result) ... 3703
ListDataQualityStatisticAnnotations (list_data_quality_statistic_annotations) ... 3704
PutDataQualityProfileAnnotation (put_data_quality_profile_annotation) ... 3705
Dati sensibili ... 3705
- tipi di dati - ... 3705
CustomEntityType ... 3706
- operazioni - ... 3706
CreateCustomEntityType (create_custom_entity_type) ... 3706
DeleteCustomEntityType (delete_custom_entity_type) ... 3708
GetCustomEntityType (get_custom_entity_type) ... 3708
BatchGetCustomEntityTypes (batch_get_custom_entity_types) ... 3709
ListCustomEntityTypes (list_custom_entity_types) ... 3710
Etichettatura APIs ... 3711
— tipi di dati — ... 3711
Tag ... 3711
- operazioni - ... 3712
TagResource (tag_resource) ... 3712
UntagResource (untag_resource) ... 3713

GetTags (get_tags) ..... 3713
Tipi di dati comuni ..... 3714
Tag ..... 3714
DecimalNumber ..... 3715
ErrorDetail ..... 3715
PropertyPredicate ..... 3715
ResourceUri ..... 3716
ColumnStatistics ..... 3716
ColumnStatisticsError ..... 3717
ColumnError ..... 3717
ColumnStatisticsData ..... 3717
BooleanColumnStatisticsData ..... 3718
DateColumnStatisticsData ..... 3719
DecimalColumnStatisticsData ..... 3719
DoubleColumnStatisticsData ..... 3720
LongColumnStatisticsData ..... 3720
StringColumnStatisticsData ..... 3721
BinaryColumnStatisticsData ..... 3721
Modelli di stringa ..... 3722
Eccezioni ..... 3724
AccessDeniedException ..... 3725
AlreadyExistsException ..... 3725
ConcurrentModificationException ..... 3725
ConcurrentRunsExceededException ..... 3725
CrawlerNotRunningException ..... 3726
CrawlerRunningException ..... 3726
CrawlerStoppingException ..... 3726
EntityNotFoundException ..... 3726
FederationSourceException ..... 3727
FederationSourceRetryableException ..... 3727
GlueEncryptionException ..... 3727
IdempotentParameterMismatchException ..... 3728
IllegalWorkflowStateException ..... 3728
InternalServiceException ..... 3728
InvalidExecutionEngineException ..... 3728
InvalidInputException ..... 3729

InvalidStateException ... 3729
InvalidTaskStatusTransitionException ... 3729
JobDefinitionErrorException ... 3729
JobRunInTerminalStateException ... 3730
JobRunInvalidStateTransitionException ... 3730
JobRunNotInTerminalStateException ... 3730
LateRunnerException ... 3731
NoScheduleException ... 3731
OperationTimeoutException ... 3731
ResourceNotReadyException ... 3731
ResourceNumberLimitExceededException ... 3732
SchedulerNotRunningException ... 3732
SchedulerRunningException ... 3732
SchedulerTransitioningException ... 3732
UnrecognizedRunnerException ... 3733
ValidationException ... 3733
VersionMismatchException ... 3733
AWS Glue Esempi di codice API ... 3734
Nozioni di base ... 3735
ciao AWS Glue ... 3736
Informazioni di base ... 3745
Azioni ... 3884
Scenari ... 4038
Guida introduttiva al Glue Data Catalog ... 4039
Sicurezza ... 4048
Protezione dei dati ... 4049
Crittografia dei dati a riposo ... 4049
Crittografia dei dati in transito ... 4067
Conformità a FIPS ... 4068
Gestione delle chiavi ... 4068
AWS Gluedipendenza da altri servizi AWS ... 4069
Endpoint di sviluppo ... 4069
Gestione dell'identità e degli accessi ... 4070
Destinatari ... 4071
Autenticazione con identità ... 4071
Gestione dell'accesso tramite policy ... 4073

Come funziona AWS Glue con IAM ... 4074
Configurazione delle autorizzazioni IAM per AWS Glue ... 4081
AWS Esempi di politiche di controllo degli accessi di Glue ... 4116
Concessione di politiche AWS gestite ... 4142
Concessione di policy con ambito dinamico ... 4151
AWS GlueSpecificare la risorsa ARNs ... 4152
Come concedere l'accesso multi-account ... 4162
Risoluzione dei problemi ... 4169
AWS Lake Formation modelli di controllo degli accessi ... 4171
Utilizzo di AWS Glue with AWS Lake Formation per l'accesso completo alla tabella ... 4172
Lake Formation per FGAC ... 4184
Utilizzo di Amazon S3 Access Grants con AWS Glue ... 4197
Come funziona con S3 Access Grants AWS Glue ... 4197
S3 Access concede considerazioni con AWS Glue ... 4198
Configura S3 Access Grants con AWS Glue ... 4198
Propagazione attendibile delle identità ... 4201
Panoramica di ... 4201
Funzionalità e vantaggi ... 4201
Casi d'uso ... 4202
Come funziona ... 4203
Integrazioni ... 4203
Guida introduttiva alla propagazione affidabile delle identità in ETL AWS Glue ... 4204
Considerazioni e limitazioni per l'integrazione di AWS Glue ETL Trusted Identity Propagation ... 4210
Sessioni in background degli utenti ... 4212
Registrazione e monitoraggio ... 4216
Convalida della conformità ... 4217
Resilienza ... 4217
Sicurezza dell'infrastruttura ... 4217
Configurazione degli endpoint VPC (AWS PrivateLink) per AWS Glue ... 4218
Configurazione di Amazon condiviso VPCs ... 4220
Risoluzione dei problemi relativi a AWS Glue ... 4222
Raccolta di informazioni AWS Glue sulla risoluzione dei problemi ... 4222
Risoluzione degli errori di configurazione comuni di Glue ... 4223
Errore: risorsa non disponibile ... 4224
Errore: impossibile trovare l'endpoint S3 o il gateway NAT per il subnetId in VPC ... 4225

Errore: regola in entrata obbligatoria nel gruppo di sicurezza ..... 4225
Errore: regola in uscita obbligatoria nel gruppo di sicurezza ..... 4225
Errore: esecuzione del Job non riuscita perché al ruolo passato devono essere assegnate (presupponiamo le autorizzazioni di ruolo per il AWS Glue servizio) ..... 4226
Errore: DescribeVpcEndpoints azione non autorizzata. impossibile convalidare l'ID VPC vpc-id ..... 4226
Errore: DescribeRouteTables azione non autorizzata. impossibile convalidare l'id di sottorete: Subnet-ID in VPC id: vpc-id ..... 4226
Errore: chiamata a ec2 non riuscita: DescribeSubnets ..... 4226
Errore: chiamata a ec2 non riuscita: DescribeSecurityGroups ..... 4226
Errore: impossibile trovare la sottorete per la zona di disponibilità ..... 4226
Errore: eccezione dell'esecuzione del processo durante la scrittura in una destinazione JDBC ..... 4227
Errore: Amazon S3: l'operazione non è valida per la classe di archiviazione dell'oggetto ... 4227
Errore: timeout di Amazon S3 ..... 4228
Errore: accesso ad Amazon S3 negato ..... 4228
Errore: l'ID chiave di accesso Amazon S3 non esiste ..... 4228
Errore: l'esecuzione del processo restituisce un errore durante l'accesso ad Amazon S3 con un URI `s3a://` ..... 4229
Errore: token di servizio Amazon S3 scaduto ..... 4231
Errore: non è stato trovato alcun DNS privato per l'interfaccia di rete ..... 4231
Errore: provisioning dell'endpoint di sviluppo non riuscito ..... 4231
Errore: server notebook CREATE_FAILED ..... 4231
Errore: impossibile avviare il notebook locale ..... 4232
Errore: esecuzione del crawler non riuscita ..... 4232
Errore: le partizioni non sono state aggiornate ..... 4232
Errore: aggiornamento del segnalibro del processo non riuscito a causa della mancata corrispondenza delle versioni ..... 4233
Errore: un processo sta rielaborando i dati mentre i segnalibri del processo sono abilitati .. 4233
Errore: comportamento di failover tra VPCs AWS Glue ..... 4235
Errori del crawler quando il crawler usa le autorizzazioni di Lake Formation ..... 4235
Errore: la posizione S3: s3://examplepath non è registrata ..... 4236
Errore: non User/Role è autorizzato a eseguire: lakeformation: on resource GetDataAccess ..... 4236
Errore: autorizzazioni Lake Formation insufficienti su (nome del database: exampleDatabase, nome tabella: exampleTable) ..... 4236

Errore: autorizzazioni di Lake Formation insufficienti su s3://examplepath ..... 4236
Domande frequenti sulla configurazione del crawler utilizzando le credenziali di Lake Formation ..... 4237
Risoluzione degli errori relativi ai processi Ray ..... 4238
Ispezione dei log dei processi Ray ..... 4238
Risoluzione degli errori relativi ai processi Ray ..... 4239
Eccezioni di machine learning AWS Glue ..... 4241
Annulla MLTask RunActivity ..... 4241
Crea MLTask RunActivity ..... 4241
Elimina MLTransform attività ..... 4242
Ottenere MLTask RunActivity ..... 4243
Ottenere MLTask RunsActivity ..... 4243
Ottieni MLTransform attività ..... 4243
Ottieni MLTransforms attività ..... 4243
GetSaveLocationForTransformArtifactActivity ..... 4244
GetTaskRunArtifactActivity ..... 4244
Pubblica MLTransform ModelActivity ..... 4245
PullLatestMLTransformModelActivity ..... 4245
PutJobMetadataForMLTransformAttività ..... 4246
StartExportLabelsTaskRunActivity ..... 4246
StartImportLabelsTaskRunActivity ..... 4246
Inizio MLEvaluation TaskRunActivity ..... 4248
Inizio MLLabeling SetGenerationTaskRunActivity ..... 4248
MLTransformAttività di aggiornamento ..... 4249
AWS Glue quote ..... 4250
Migliorare AWS Glue le prestazioni ..... 4251
Strategie di ottimizzazione per il tipo di processo ..... 4251
Miglioramento delle prestazioni di Spark ..... 4251
Ottimizzazione delle letture con pushdown ..... 4252
Il predicato pushdown sui file archiviati su Amazon S3 ..... 4252
Esecuzione del pushdown quando si utilizzano origini JDBC ..... 4253
Note e limitazioni per il pushdown in AWS Glue ..... 4256
Utilizzo di Auto Scaling per AWS Glue ..... 4257
Requisiti ..... 4258
Abilitazione dell'Auto Scaling in AWS Glue Studio ..... 1937
Abilitazione dell'Auto Scaling con CLI o SDK AWS ..... 1938

Abilitazione del dimensionamento automatico con sessioni interattive ..... 4260
Suggerimenti e considerazioni ..... 4260
Monitoraggio dell'Auto Scaling con i parametri di Amazon CloudWatch ..... 4261
Monitoraggio dell'Auto Scaling con Amazon Logs CloudWatch ..... 4262
Monitoraggio di Auto Scaling con interfaccia utente di Spark ..... 4263
Monitoraggio dell'utilizzo della DPU di esecuzione del processo Auto Scaling ..... 4264
Limitazioni ..... 4264
Partizionamento del carico di lavoro con esecuzione delimitata ..... 4264
Abilitazione del partizionamento del carico di lavoro ..... 4265
Configurazione di un trigger AWS Glue per l'esecuzione automatica del processo ..... 4266
Problemi noti ..... 4267
Prevenzione dell'accesso ai dati tra processi ..... 4267
Cronologia della documentazione ..... 4270
Aggiornamenti precedenti ..... 4337
AWS Glossario ..... 4339
..... mmmmcccxl

# Che cos'è AWS Glue?

AWS Glue è un servizio di integrazione dati serverless che semplifica agli utenti analitici il rilevamento, la preparazione, lo spostamento e l'integrazione di dati da più origini. Puoi usarlo per analisi, machine learning e sviluppo di applicazioni. Include anche strumenti aggiuntivi di produttività e gestione dei dati per la creazione, l'esecuzione di processi e l'implementazione di flussi di lavoro aziendali.

Con AWS Glue puoi rilevare e collegarti a oltre 70 diverse origini di dati e gestire i tuoi dati in un catalogo dati centralizzato. Puoi creare, eseguire e monitorare visivamente pipeline di estrazione, trasformazione e caricamento (ETL) per caricare dati nei data lake. Inoltre, puoi eseguire ricerche e query immediatamente nei dati catalogati utilizzando Amazon Athena, Amazon EMR e Amazon Redshift Spectrum.

AWS Glue consolida le principali funzionalità di integrazione dei dati in un singolo servizio. Tali funzionalità includono rilevamento dati, ETL moderno, pulizia, trasformazione e catalogazione a livello centralizzato. È anche serverless, per cui non esiste alcuna infrastruttura da gestire. Con un supporto flessibile per tutti i carichi di lavoro come ETL, ELT e streaming in un unico servizio, AWS Glue supporta gli utenti tra vari carichi di lavoro e tipi di utenti.

AWS Glue, inoltre, semplifica l'integrazione dei dati nell'architettura. Si integra con i servizi AWS di analisi e i data lake Amazon S3. AWS Gluedispone di interfacce di integrazione e strumenti per la creazione di lavori facili da usare per tutti gli utenti, dagli sviluppatori agli utenti aziendali, con soluzioni su misura per diverse competenze tecniche.

Grazie alla scalabilità on demand, AWS Glue è utile per concentrarsi su attività di elevato valore che massimizzano il valore dei dati. È scalabile per qualunque dimensione di dati e supporta tutti i tipi di dati e varianti di schemi. Per aumentare l'agilità e ottimizzare i costi, AWS Glue offre disponibilità e fatturazione integrate elevate. pay-as-you-go

Per informazioni sui prezzi, consulta Prezzi di AWS Glue.

AWS Glue Studio

AWS Glue Studio è un'interfaccia grafica che facilita la creazione, l'esecuzione e il monitoraggio di processi di integrazione dati in AWS Glue. Puoi comporre visivamente flussi di lavoro di trasformazione dei dati ed eseguirli con facilità sul motore ETL serverless basato su Apache Spark di AWS Glue.

Con AWS Glue Studio, puoi creare e gestire processi di raccolta, trasformazione e pulizia di dati. Puoi utilizzare AWS Glue Studio anche per risolvere problemi e modificare script di processi.

Argomenti


- Funzionalità di AWS Glue
- Scopri le innovazioni in AWS Glue
- Nozioni di base su AWS Glue
- Accesso a AWS Glue
- Servizi correlati
- AWS Glue per la fine del supporto di Ray


# Funzionalità di AWS Glue

Le funzioni di AWS Glue si dividono in tre categorie principali:

- Rilevamento e organizzazione dei dati
- Trasformazione, preparazione e pulizia dei dati per l'analisi
- Creazione e monitoraggio di pipeline di dati

Rilevamento e organizzazione dei dati

- Unifica e cerca in più archivi di dati: archivia, indicizza e cerca su più fonti di dati e sink catalogando tutti i tuoi dati. AWS
- Rilevamento automatico dei dati: utilizzo dei crawler AWS Glue per la deduzione automatica delle informazioni degli schemi e l'integrazione di tali informazioni in AWS Glue Data Catalog.
- Gestione di schemi e autorizzazioni: convalida e controllo dell'accesso a database e tabelle.
- Connettiti a un'ampia varietà di fonti di dati: accedi a più fonti di dati, sia in locale che in locale AWS, utilizzando AWS Glue le connessioni per creare il tuo data lake.

Trasformazione, preparazione e pulizia dei dati per l'analisi

- Trasformazione visuale dei dati con un'interfaccia di canvas del processo: definizione di un processo ETL nell'editor dei processi visivi e generazione automatica del codice per l'estrazione, la trasformazione e il caricamento dei dati.

- Creazione di complesse pipeline ETL con una semplice pianificazione del processo: richiamo di processi AWS Glue in base a un programma, on demand o in base a un evento.
- Pulizia e trasformazione dei dati in streaming in transito: possibilità di consumo dati continuo e pulizia e trasformazione dei dati in transito. In tal modo, i dati sono disponibili per l'analisi in pochi secondi nell'archivio dei dati di destinazione.
- Deduplicazione e pulizia dei dati con machine learning integrato: pulizia e preparazione dei dati per l'analisi senza diventare esperti di machine learning, utilizzando la funzione `FindMatches`. Questa funzione deduplica e trova registri non perfettamente corrispondenti tra loro.
- Notebook di processo integrati: i notebook di processo AWS Glue forniscono notebook serverless con una configurazione minima in AWS Glue, per poter cominciare rapidamente.
- Modifica, debug e verifica del codice ETL: con le sessioni interattive di AWS Glue, puoi esplorare e preparare i dati in modo interattivo. Puoi esplorare, sperimentare ed elaborare i dati in modo interattivo utilizzando l'IDE o il notebook di tua scelta.
- Definizione, rilevamento e correzione di dati sensibili: il rilevamento dei dati sensibili di AWS Glue consente di definire, identificare ed elaborare dati sensibili nella pipeline di dati e nel data lake.

Creazione e monitoraggio di pipeline di dati

- Scalabilità automatica in base al carico di lavoro: aumento o riduzione delle risorse in modo dinamico in base al carico di lavoro. In tal modo, i processi vengono assegnati agli operatori solo quando necessario.
- Automatizzazione di processi con trigger basati su eventi: avvio di crawler o processi AWS Glue con trigger basati su eventi e progettazione di una catena di percorsi e crawler dipendenti.
- Esegui e monitora i processi: esegui i processi AWS Glue con il motore che preferisci, Spark o Ray. Monitorali con strumenti di monitoraggio automatizzati, approfondimenti sull'esecuzione dei processi AWS Glue e AWS CloudTrail. Migliora il monitoraggio dei processi supportati da Spark con l'interfaccia utente di Apache Spark.
- Definizione di flussi di lavoro per attività ETL e di integrazione: definizione di flussi di lavoro per ETL e attività di integrazione per più crawler, processi e trigger.

# Scopri le innovazioni in AWS Glue

Scopri le ultime innovazioni AWS Glue e scopri in che modo i clienti utilizzano AWS Glue per consentire la preparazione dei dati in modalità self-service in tutta l'organizzazione.

Scopri come i clienti AWS Glue vanno oltre la configurazione tradizionale e come si configurano AWS Glue per il monitoraggio del lavoro e delle prestazioni.

## Nozioni di base su AWS Glue

Ti consigliamo di iniziare con le sezioni seguenti:

- Panoramica sull'utilizzo di AWS Glue
- Concetti di AWS Glue
- Configurazione di autorizzazioni IAM per AWS Glue
- Nozioni di base su AWS Glue Data Catalog
- Creazione di processi in AWS Glue
- Nozioni di base sulle sessioni interattive di AWS Glue
- Orchestrazione in AWS Glue

## Accesso a AWS Glue

Puoi creare, visualizzare e gestire i processi AWS Glue utilizzando una qualunque delle interfacce seguenti:

- Console AWS Glue: fornisce un'interfaccia web per la creazione, la visualizzazione e la gestione di processi AWS Glue. Per accedere alla console, consulta AWS Glue.
- AWS Glue Studio: fornisce un'interfaccia grafica per la creazione e la modifica dei processi AWS Glue in modo visivo. Per ulteriori informazioni, consulta Creazione di processi ETL visivi.
- AWS Gluesezione della Guida di AWS CLI riferimento: fornisce AWS CLI comandi utilizzabili conAWS Glue. Per ulteriori informazioni, consulta la AWS CLI Documentazione di riferimento per AWS Glue.
- AWS GlueAPI: fornisce una documentazione di riferimento dell'API completa per gli sviluppatori. Per ulteriori informazioni, consulta API AWS Glue.

## Servizi correlati

Gli utenti di AWS Glue utilizzano anche:

- AWS Lake Formation : un servizio costituito da un livello di autorizzazione che fornisce un controllo granulare fine dell'accesso alle risorse in AWS Glue Data Catalog.
- AWS Glue DataBrew : uno strumento visivo di preparazione dei dati che è possibile utilizzare per la pulizia e la normalizzazione dei dati senza scrivere codice.

# AWS Glue per la fine del supporto di Ray

> **Important**
>
> AWS Glue for Ray non è più aperto a nuovi clienti. I clienti esistenti possono continuare a utilizzare il servizio normalmente. Per ulteriori informazioni, consulta AWS Glue Ray End of Support.

Dopo un'attenta valutazione, abbiamo deciso di chiudere AWS Glue Ray a nuovi clienti a partire dal 30 aprile 2026. Se desideri utilizzarlo AWS Glue per Ray, registrati prima di tale data. I clienti esistenti possono continuare a utilizzare il servizio normalmente.

AWS continua a investire in miglioramenti della sicurezza e della disponibilità di AWS Glue For Ray. Tieni presente che non abbiamo intenzione di introdurre nuove funzionalità in AWS Glue for Ray, ad eccezione dei miglioramenti in termini di sicurezza e disponibilità.

In alternativa a AWS Glue for Ray, consigliamo di utilizzare Amazon Elastic Kubernetes Service. Amazon Elastic Kubernetes Service è un servizio conforme a Kubernetes completamente gestito e certificato che semplifica il processo di creazione, protezione, funzionamento e manutenzione dei cluster Kubernetes. AWSÈ un'opzione altamente personalizzabile che si affida a KubeRay Operator open source per distribuire e gestire i cluster Ray su Kubernetes, che offre un migliore utilizzo delle risorse, una gestione semplificata dell'infrastruttura e il supporto completo per le funzionalità di Ray.

## Migrazione di un lavoro Ray su Amazon Elastic Kubernetes Service

Questa sezione fornisce i passaggi per la migrazione da AWS Glue for Ray a Ray su Amazon Elastic Kubernetes Service. Questi passaggi sono utili per due scenari di migrazione:

- Migrazione standard (x86/amd64): per questi casi d'uso, la strategia di migrazione utilizza il contenitore OpenSource Ray per le implementazioni di base ed esegue gli script direttamente sul contenitore di base.

- ARM64 Migrazione: per questi casi d'uso, la strategia di migrazione supporta configurazioni di container personalizzate per dipendenze e requisiti di architettura specifici. ARM64

## Prerequisiti per la migrazione

Installa i seguenti strumenti CLI:aws,, kubectleksctl, helm Python 3.9+. Questi strumenti CLI sono necessari per il provisioning e la gestione dell'ambiente Ray on EKS. eksctlsemplifica la creazione e la gestione dei cluster EKS. kubectlè la CLI Kubernetes standard per l'implementazione e la risoluzione dei problemi dei carichi di lavoro sul cluster. helmviene utilizzato per l'installazione e la gestione KubeRay (l'operatore che esegue Ray su Kubernetes). Python 3.9+ è necessario per Ray stesso e per eseguire gli script di invio dei lavori localmente.

Installa eksctl

Segui le istruzioni sulle opzioni di installazione per Eksctl o usa le istruzioni seguenti per l'installazione.

Per macOS:

```
brew tap weaveworks/tap
brew install weaveworks/tap/eksctl
```

Per Linux:

```
curl --silent --location "https://github.com/weaveworks/eksctl/releases/latest/
download/eksctl_$(uname -s)_amd64.tar.gz" | tar xz -C /tmp

# Move the extracted binary to /usr/local/bin
sudo mv /tmp/eksctl /usr/local/bin

# Test the installation
eksctl version
```

Installa kubectl

Segui le istruzioni su Configurare kubectl ed eksctl o usa le istruzioni seguenti per l'installazione.

Per macOS:

```
brew install kubectl
```

Per Linux:

```
curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/
bin/linux/amd64/kubectl"
chmod +x kubectl
sudo mv kubectl /usr/local/bin/
```

Installa helm

Segui le istruzioni sull'installazione di Helm o usa le istruzioni seguenti per l'installazione.

Per macOS:

```
brew install helm
```

Per Linux:

```
curl https://raw.githubusercontent.com/helm/helm/main/scripts/get-helm-3 | bash
```

## Passaggio 1. Crea o scegli un'immagine Docker per Ray

Opzione 1: usa l'immagine Ray ufficiale (non è richiesta la build)

Questa opzione utilizza l'immagine ufficiale di Ray Docker su Docker Hub, ad esempio`rayproject/ray:2.4.0-py39`, gestita dal progetto Ray.

> **Note**
>
> Questa immagine è solo per amd64. Usala se le tue dipendenze sono compatibili con amd64 e non hai bisogno di build specifiche per ARM.

Opzione 2: crea e pubblica la tua immagine arm64 Ray 2.4.0

Questa opzione è utile quando si utilizzano i nodi Graviton (ARM), coerentemente con quelli utilizzati internamente da AWS Glue For Ray. È possibile creare un'immagine personalizzata associata alle stesse versioni di dipendenza di Ray, AWS Glue per ridurre le discrepanze di compatibilità.

Crea un Dockerfile localmente:

```
# Build an ARM64 image
```

```
FROM --platform=linux/arm64 python:3.9-slim-bullseye
# Handy tools: wget for KubeRay probes; CA certs; keep image small
RUN apt-get update && apt-get install -y --no-install-recommends \
    wget ca-certificates \
    && rm -rf /var/lib/apt/lists/*

# Keep pip/setuptools modern enough for wheels resolution
RUN python -m pip install -U "pip<24" "setuptools<70" wheel

# ---- Install Ray 2.4.0 (ARM64 / Py3.9) and Glue-like dependencies ----
# 1) Download the exact Ray 2.4.0 wheel for aarch64 (no network at runtime)
RUN python -m pip download --only-binary=:all: --no-deps --dest /tmp/wheels ray==2.4.0

# 2) Core libs used in Glue (pin to Glue-era versions)
#    + the dashboard & jobs API dependencies compatible with Ray 2.4.0.
#    (Pins matter: newer major versions break 2.4.0's dashboard.)
RUN python -m pip install --no-cache-dir \
    /tmp/wheels/ray-2.4.0-*.whl \
    "pyarrow==11.0.0" \
    "pandas==1.5.3" \
    "boto3==1.26.133" \
    "botocore==1.29.133" \
    "numpy==1.24.3" \
    "fsspec==2023.4.0" \
    "protobuf<4" \
    # --- dashboard / jobs server deps ---
    "aiohttp==3.8.5" \
    "aiohttp-cors==0.7.0" \
    "yarl<1.10" "multidict<7.0" "frozenlist<1.4" "aiosignal<1.4" "async_timeout<5" \
    "pydantic<2" \
    "opencensus<0.12" \
    "prometheus_client<0.17" \
    # --- needed if using py_modules ---
    "smart_open[s3]==6.4.0"

# Optional: prove Ray & arch at container start
ENV PYTHONUNBUFFERED=1
WORKDIR /app

# KubeRay overrides the start command; this is just a harmless default
CMD ["python","-c","import ray,platform; print('Ray', ray.__version__, 'on',
 platform.machine())"]
```

```
# Set environment variables
export AWS_REGION=us-east-1
export AWS_ACCOUNT=$(aws sts get-caller-identity --query Account --output text)
export REPO=ray-2-4-arm64
export IMAGE=${AWS_ACCOUNT}.dkr.ecr.${AWS_REGION}.amazonaws.com/${REPO}:v1

# Create repository and login
aws ecr create-repository --repository-name $REPO >/dev/null 2>&1 || true
aws ecr get-login-password --region $AWS_REGION \
  | docker login --username AWS --password-stdin ${AWS_ACCOUNT}.dkr.ecr.
${AWS_REGION}.amazonaws.com

# Enable Buildx (for cross-builds on non-ARM hosts)
docker buildx create --name multi --driver docker-container --use 2>/dev/null || true

# Build & push ARM64 image
docker buildx build \
  --platform linux/arm64 \
  -t "$IMAGE" \
  . --push

# Verify the image architecture remotely
aws ecr batch-get-image \
  --repository-name $REPO \
  --image-ids imageTag=v1 \
  --accepted-media-types application/vnd.docker.distribution.manifest.v2+json \
  | jq -r '.images[0].imageManifest' \
  | jq -r 'fromjson.config.digest'
```

Una volta terminato, fai riferimento a questa ARM64 immagine nelle RayCluster specifiche con. `nodeSelector: { kubernetes.io/arch: arm64 }`

```
spec:
  rayVersion: "2.4.0"
  headGroupSpec:
    template:
      spec:
        containers:
        - name: ray-head
          image: <your ECR image>
```

## Passaggio 2. Conversione AWS Glue da Ray Job Configuration a Ray su Amazon Elastic Kubernetes Service

AWS Glue for Ray i job supportano una serie di argomenti di lavoro che configurano i lavoratori, le dipendenze, la memoria e la registrazione. Quando si esegue la migrazione ad Amazon Elastic Kubernetes KubeRay Service con, questi argomenti devono essere RayCluster tradotti in campi di specifiche o impostazioni dell'ambiente di runtime Ray Job.

Mappatura degli argomenti del lavoro

Mappatura AWS Glue di Ray Arguments su Ray su EKS Equivalents

| AWS Glue per l'argomento Ray | Cosa fa in AWS Glue cambio di Ray | Equivalente a Ray su Amazon Elastic Kubernetes Service |
|---|---|---|
| `--min-workers` | Numero minimo di lavoratori che la mansione deve assegnare. | `workerGroupSpecs[].minReplicas` nel tuo RayCluster |
| `--working-dir` | Distribuisce uno zip (URI S3) a tutti i nodi. | Usa Ray runtime env: `working_dir` se esegui l'invio da file locali; `py_modules` affinché gli zip di S3 puntino all'artefatto S3 |
| `--s3-py-modules` | Aggiunge Python wheels/dists da S3. | Usa Ray runtime env: `py_modules: ["s3://.../xxx.whl", ...]` |
| `--pip-install` | Installa pacchetti PyPI aggiuntivi per il lavoro. | Ray runtime env: `pip: ["pkg==ver", ...]` (Ray Job `--runtime-env-json` CLI RayJob `runtimeEnvYAML` o). |
| `--object_store_memory_head` | % di memoria per l'archivio Plasma del nodo principale. | `headGroupSpec[].rayStartParams.object-store-memory` nel tuo RayCluster. Nota che dovrebbe essere in byte. AWS Glue usa la percentuale, mentre Ray usa i byte. |

| AWS Glue per l'argomento Ray | Cosa fa in AWS Glue cambio di Ray | Equivalente a Ray su Amazon Elastic Kubernetes Service |
|---|---|---|
| `--object_store_memory_worker` | % di memoria per l'archivio Plasma dei nodi di lavoro. | Come sopra ma impostato in byte `rayStartParams.object-store-memory` (byte) di ogni gruppo di lavoro. |
| `--object_spilling_config` | Configura Ray object spilling. | `headGroupSpec[].rayStartParams.object-spilling-config` |
| `--logging_configuration` | AWS Glue-log gestiti (CloudWatch, S3). | Check pod stdout/stderr: usa. `kubectl -n ray logs <pod-name> --follow` Controlla i log da Ray Dashboard (port-forward to:8265), dove puoi anche vedere i log delle attività e dei lavori. |

Mappatura della configurazione del Job

Mappatura delle configurazioni AWS Glue di Ray Job su Ray on EKS Equivalents

| Configurazione | Cosa fa in cambio di Ray AWS Glue | Ray sull'equivalente di EKS |
|---|---|---|
| Tipo di lavoratore | Imposta il tipo di lavoratore predefinito consentito durante l'esecuzione di un lavoro. L'impostazione predefinita è Z 2X (8 vCPU, 64 GB RAM). | Tipo di istanza Nodegroup in EKS (ad esempio, r7g.2xlarge ≈ 8 vCPU/64 GB per ARM, r7a.2xlarge per x86). |
| Numero massimo di lavoratori | Il numero di lavoratori che desideri AWS Glue assegnare a questo lavoro. | Imposta `workerGroupSpecs[].maxReplicas` lo stesso numero di quello in AWS Glue cui hai usato. Questo è il limite superiore per la scalabilità automatica. Analogamente impostato `minReplicas` come limite |

| Configurazione | Cosa fa in cambio di Ray AWS Glue | Ray sull'equivalente di EKS |
| --- | --- | --- |
| | | inferiore. Puoi iniziare con`replicas: 0`,`minReplicas: 0` . |

## Fase 3. Configurazione del servizio Amazon Elastic Kubernetes

Puoi creare un nuovo cluster Amazon Elastic Kubernetes Service o riutilizzare un cluster Amazon Elastic Kubernetes Service esistente. Se utilizzi un cluster esistente, salta i comandi di creazione del cluster e passa a Aggiungi un gruppo di nodi, IRSA e installa. KubeRay

Crea un cluster Amazon Elastic Kubernetes Service

> **Note**
>
> Se disponi di un cluster Amazon Elastic Kubernetes Service esistente, salta i comandi per creare un nuovo cluster e aggiungi semplicemente un gruppo di nodi.

```
# Environment Variables
export AWS_REGION=us-east-1
export CLUSTER=ray-eks
export NS=ray # namespace for your Ray jobs (you can reuse another if you like)

# Create a cluster (OIDC is required for IRSA)
eksctl create cluster \
  --name $CLUSTER \
  --region $AWS_REGION \
  --with-oidc \
  --managed
```

Aggiungi un gruppo di nodi

```
# ARM/Graviton (matches Glue's typical runtime):
eksctl create nodegroup \
  --cluster $CLUSTER \
  --region $AWS_REGION \
  --name arm64-ng \
```

```
  --node-type m7g.large \
  --nodes 2 --nodes-min 1 --nodes-max 5 \
  --managed \
  --node-labels "workload=ray"

# x86/amd64 (use if your image is amd64-only):
eksctl create nodegroup \
  --cluster $CLUSTER \
  --region $AWS_REGION \
  --name amd64-ng \
  --node-type m5.large \
  --nodes 2 --nodes-min 1 --nodes-max 5 \
  --managed \
  --node-labels "workload=ray"
```

> **Note**
>
> Se utilizzi un cluster Amazon Elastic Kubernetes Service esistente, `--with-oidc` utilizzalo per abilitare OIDC quando aggiungi un gruppo di nodi.

Crea namespace e ruolo IAM per Service Accounts (IRSA) per S3

Un namespace Kubernetes è un raggruppamento logico di risorse (pod, servizi, ruoli, ecc.). Puoi creare o riutilizzare uno spazio dei nomi esistente. Dovrai anche creare una policy IAM per S3 che rispecchi l'accesso del tuo job. AWS Glue Usa le stesse autorizzazioni personalizzate del tuo ruolo AWS Glue lavorativo (in genere S3 read/write per bucket specifici). Per concedere autorizzazioni ad Amazon Elastic Kubernetes Service simili AWSGlue ServiceRole alle, crea un Service Account (IRSA) associato a questa policy IAM. Per istruzioni su come configurare questo account di servizio, consulta IAM Roles for Service Accounts.

```
# Create (or reuse) namespace
kubectl create namespace $NS || true
```

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": ["s3:PutObject", "s3:GetObject", "s3:ListBucket"],
    "Resource": [
```

```
      "arn:aws:s3:::YOUR-BUCKET",
      "arn:aws:s3:::YOUR-BUCKET/*"
    ]
  }]
}
```

```
# Create the IAM policy and wire IRSA:
aws iam create-policy \
  --policy-name RayS3Policy \
  --policy-document file://example.json || true

# Create a service account (IRSA) bound to that policy.
eksctl create iamserviceaccount \
  --cluster $CLUSTER \
  --region $AWS_REGION \
  --namespace $NS \
  --name ray-s3-access \
  --attach-policy-arn arn:aws:iam::${AWS_ACCOUNT}:policy/RayS3Policy \
  --approve \
  --override-existing-serviceaccounts
```

KubeRay Operatore di installazione (controller che esegue Ray su K8s)

```
helm repo add kuberay https://ray-project.github.io/kuberay-helm/
helm repo update
helm upgrade --install kuberay-operator kuberay/kuberay-operator \
  --namespace kuberay-system \
  --create-namespace

# Validate the operator pod Running
kubectl -n kuberay-system get pods
```

## Passaggio 4. Avvia un cluster Ray

Crea un file YAML per definire un ray cluster. Di seguito è riportato un esempio di configurazione (raycluster.yaml):

```
apiVersion: ray.io/v1
kind: RayCluster
metadata:
  name: glue-like
  namespace: ray
```

```
spec:
  rayVersion: "2.4.0"
  headGroupSpec:
    template:
      spec:
        nodeSelector:
          kubernetes.io/arch: amd64
        serviceAccountName: ray-s3-access
        containers:
        - name: ray-head
          image: rayproject/ray:2.4.0-py39
          imagePullPolicy: Always
          resources:
            requests: { cpu: "1", memory: "2Gi" }
            limits:   { cpu: "1", memory: "2Gi" }
  workerGroupSpecs:
  - groupName: workers
    replicas: 0 # start with just a head (like small Glue dev job) and turn number of
 replicas later
    minReplicas: 0
    maxReplicas: 5
    template:
      spec:
        nodeSelector:
          kubernetes.io/arch: amd64
        serviceAccountName: ray-s3-access
        containers:
        - name: ray-worker
          image: rayproject/ray:2.4.0-py39
          imagePullPolicy: Always
          resources:
            requests: { cpu: "1", memory: "2Gi" }
            limits:   { cpu: "1", memory: "2Gi" }
```

Implementa il cluster Ray sul cluster Amazon Elastic Kubernetes Service

```
kubectl apply -n $NS -f raycluster.yaml

# Validate that the head pod turns to READY/ RUNNING state
kubectl -n $NS get pods -l ray.io/cluster=glue-like -w
```

Se è necessario modificare lo yaml distribuito, elimina prima il cluster e poi riapplica lo yaml aggiornato:

```
kubectl -n $NS delete raycluster glue-like
kubectl -n $NS apply -f raycluster.yaml
```

Accesso alla Ray Dashboard

Puoi accedere alla dashboard di Ray abilitando il port forwarding usando kubectl:

```
# Get service
SVC=$(kubectl -n $NS get svc -l ray.io/cluster=glue-like,ray.io/node-type=head -o
 jsonpath='{.items[0].metadata.name}')

# Make the Ray dashboard accessible at http://localhost:8265 on your local machine.
kubectl -n $NS port-forward svc/$SVC 8265:8265
```

## Fase 5. Invia Ray Job

Per inviare un lavoro Ray, usa la CLI di Ray jobs. La versione CLI può essere più recente del cluster, è retrocompatibile. Come prerequisito, memorizza lo script di lavoro localmente in un file, ad es. `job.py`

```
python3 -m venv ~/raycli && source ~/raycli/bin/activate
pip install "ray[default]==2.49.2"

# Submit your ray job by supplying all python dependencies that was added to your Glue
 job
ray job submit --address http://127.0.0.1:8265 --working-dir . \
  --runtime-env-json '{
    "pip": ["boto3==1.28.*","pyarrow==12.*","pandas==2.0.*"]
  }' \
  -- python job.py
```

Il lavoro può essere monitorato sulla dashboard di Ray.

# AWS Glue: Come funziona

AWS Glue utilizza altri AWS servizi per orchestrare i processi ETL (estrazione, trasformazione e caricamento) per creare data warehouse e data lake e generare flussi di output. AWS Gluechiama le operazioni API per trasformare i dati, creare log di runtime, archiviare la logica dei processi e creare notifiche per aiutarti a monitorare le esecuzioni dei processi. La console AWS Glue si connette a questi servizi in un'applicazione gestita, in modo che tu possa concentrarti sulla creazione e sul monitoraggio del lavoro ETL. La console esegue le operazioni amministrative e di sviluppo del processo per tuo conto. Devi fornire le credenziali e altre proprietà a AWS Glue per accedere alle origini dati e scrivere nelle destinazioni dati.

AWS Glue si occupa di effettuare il provisioning delle risorse necessarie per l'esecuzione del carico di lavoro e di gestire tali risorse. Non devi creare l'infrastruttura per uno strumento ETL, perché l'operazione viene eseguita da AWS Glue. Quando sono necessarie risorse, per ridurre i tempi di avvio, AWS Glue usa un'istanza del pool di istanze attivo per eseguire il carico di lavoro.

Con AWS Glue crei i processi usando le definizioni di tabella nel catalogo dati. I lavori sono costituiti da script che contengono le istruzioni per eseguire le attività di trasformazione dei dati desiderate. Per avviare i processi, in base a una pianificazione o come risultato di un evento specificato, potrai utilizzare i trigger. Puoi decidere dove conservare i dati dell'obiettivo e quale origine dati popola l'obiettivo. In base agli input, AWS Glue trasforma i dati dal formato di origine a quello di destinazione. In alternativa, puoi anche fornire script personalizzati nella console AWS Glue o nell'API per elaborare i dati in base alle tue esigenze specifiche.

Origini dati e destinazioni

AWS Glue per Spark consente di leggere e scrivere dati da più sistemi e database, tra cui:

- Simple Storage Service (Amazon S3)
- Amazon DynamoDB
- Amazon Redshift
- Amazon Relational Database Service (Amazon RDS)
- Database accessibili da JDBC di terze parti
- MongoDB e Amazon DocumentDB (compatibile con MongoDB)
- Altri connettori del marketplace e plug-in Apache Spark

Flussi dei dati

AWS Glue per Spark può trasmettere dati dai seguenti sistemi:

- Flusso di dati Amazon Kinesis
- Apache Kafka

AWS Glue è disponibile in diverse AWS regioni. Per ulteriori informazioni, consulta la sezione relativa a regioni ed endpoint AWS nella Riferimenti generali di Amazon Web Services.

Argomenti


- Processi ETL serverless eseguiti in isolamento
- AWS Glue concetti
- AWS Glue componenti
- AWS Glue per Spark e AWS Glue per Ray
- Conversione da schemi semistrutturati a schemi relazionali con AWS Glue
- AWS Sistemi tipo Glue


# Processi ETL serverless eseguiti in isolamento

AWS Glue esegue i processi ETL in un ambiente serverless con un motore a scelta tra Spark e Ray. AWS Glue esegue questi processi su risorse virtuali di cui effettua il provisioning e che gestisce nel proprio account di servizio.

AWS Glue ha gli scopi seguenti:

- Isolare i dati dei clienti.
- Proteggere i dati dei clienti in transito e quelli memorizzati.
- Accedere ai dati dei clienti solo in risposta alle richieste dei clienti, utilizzando le credenziali contestuali e temporanee o con il consenso del cliente ai ruoli IAM nel suo account.

Durante il provisioning di un processo ETL, fornisci origini dati di input e destinazioni dati di output nel Virtual Private Cloud (VPC). Inoltre, puoi fornire il ruolo IAM, l'ID VPC, l'ID sottorete e il gruppo di sicurezza che sono necessari per accedere alle origini dati e alle destinazioni. Per ogni tupla (ID account del cliente, ruolo IAM, ID di sottorete e gruppo di sicurezza), AWS Glue crea un nuovo

ambiente isolato a livello di rete e di gestione da tutti gli altri ambienti all'interno del tuo account di servizio AWS Glue.

Puoi creare e configurare AWS Glue risorse, come cataloghi di dati, lavori e crawler, all'interno del tuo account. AWS Queste risorse vengono quindi associate al ruolo IAM e alle impostazioni di rete (sottorete e gruppo di sicurezza) specificate durante il processo di creazione.

AWS Glue crea interfacce di rete elastiche nella sottorete usando indirizzi IP privati. I processi utilizzano queste interfacce di rete elastiche per accedere alle origini dati e alle destinazioni dati. Il traffico in uscita e all'interno dell'ambiente di esecuzione del processo è regolato dal VPC e dalle policy di rete con un'eccezione: le chiamate effettuate alle librerie AWS Glue possono indirizzare il traffico verso operazioni API AWS Glue tramite il VPC AWS Glue. Tutte le chiamate API AWS Glue vengono registrate, pertanto i proprietari dei dati possono controllare l'accesso API abilitando AWS CloudTrail, che fornisce i log di controllo all'account.

AWS Gluegli ambienti gestiti che eseguono i processi ETL sono protetti con le stesse pratiche di sicurezza seguite da altri servizi. AWS Per una panoramica delle pratiche e delle responsabilità condivise in materia di sicurezza, consultate il white paper Introduzione ai processi AWS di sicurezza.

# AWS Glue concetti

AWS Glue è un servizio ETL (extract, transform, load) completamente gestito che consente di spostare facilmente i dati tra diverse fonti di dati e destinazioni. I componenti principali sono:

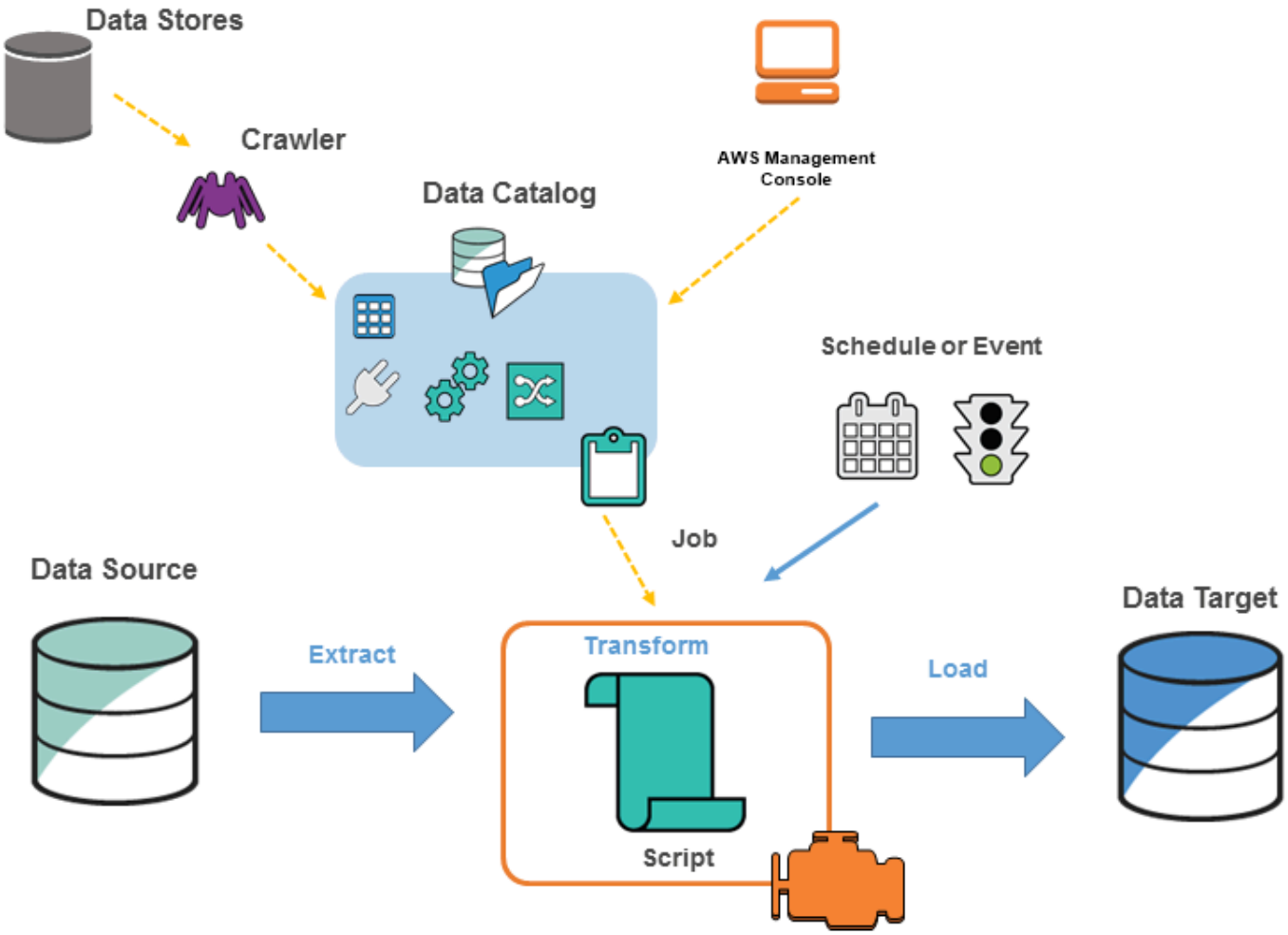
- Catalogo dati: un archivio di metadati che contiene definizioni di tabelle, definizioni di processi e altre informazioni di controllo per i flussi di lavoro ETL.
- Crawler: programmi che si connettono alle origini dati, deducono gli schemi per i dati e creano definizioni di tabelle di metadati nel Catalogo dati.
- Processi ETL: la logica aziendale per estrarre i dati dalle origini dati, trasformarli utilizzando script Apache Spark e caricarli nelle destinazioni.
- Trigger: meccanismi per avviare l'esecuzione dei processi in base a pianificazioni o eventi.

Il flusso di lavoro tipico consiste nel:

1. Definire origini dati e destinazioni dati nel Catalogo dati.
2. Usare i crawler per popolare il Catalogo dati con i metadati delle tabelle provenienti dalle origini dati.

3. Definire i processi ETL con script di trasformazione per spostare ed elaborare i dati.
4. Eseguire processi on-demand o basati su trigger.
5. Monitorare le prestazioni lavorative utilizzando i pannelli di controllo.

Il diagramma seguente mostra l'architettura di un AWS Glue ambiente.




Si definiscono i lavori AWS Glue per eseguire il lavoro necessario per estrarre, trasformare e caricare i dati (ETL) da un'origine dati a una destinazione dati. In genere, si svolgono le azioni seguenti:

- Per le origini dei datastore, si definisce un crawler per popolare il AWS Glue Data Catalog con definizioni di tabelle di metadati. Si punta il crawler a un datastore e il crawler crea le definizioni di tabelle nel catalogo dati. Per le origini di streaming, è possibile definire manualmente le tabelle del catalogo dati e specificare le proprietà del flusso dei dati.

Oltre alle definizioni delle tabelle, AWS Glue Data Catalog contiene altri metadati necessari per definire i job ETL. Si utilizzano questi metadati quando si definisce un processo per trasformare i dati.

- AWS Glue può generare uno script per trasformare i dati. In alternativa, puoi fornire lo script nella AWS Glue console o nell'API.
- È possibile eseguire il processo on demand oppure configurarlo affinché si avvii al verificarsi di un trigger specifico. Il trigger può essere una pianificazione basata sul tempo o un evento.

Durante l'esecuzione del processo, uno script estrae i dati dall'origine dati, li trasforma e li carica sulla destinazioni dati. Lo script viene eseguito in un ambiente Apache Spark in AWS Glue.

> **Important**
>
> Le tabelle e i database in AWS Glue sono oggetti in AWS Glue Data Catalog. Essi contengono metadati, non dati provenienti da un datastore.

I dati basati su testo, ad esempio CSVs, devono essere codificati **`UTF-8`** per poterli AWS Glue elaborare correttamente. Per ulteriori informazioni, consulta UTF-8 in Wikipedia.

## AWS Glue terminologia

AWS Glue si basa sull'interazione di diversi componenti per creare e gestire il flusso di lavoro di estrazione, trasformazione e caricamento (ETL).

### AWS Glue Data Catalog

L'archivio persistente di metadati in. AWS Glue Contiene definizioni di tabelle, definizioni di processi e altre informazioni di controllo per la gestione AWS Glue dell'ambiente. Ogni AWS account ne ha uno AWS Glue Data Catalog per regione.

### Classificatore

Determina lo schema dei tuoi dati. AWS Glue fornisce classificatori per tipi di file comuni, come CSV, JSON, AVRO, XML e altri. Fornisce inoltre classificatori per i più comuni sistemi di gestione

di database relazionali utilizzando una connessione JDBC. Puoi scrivere un classificatore personalizzato usando un pattern grok o specificando un tag di riga in un documento XML.

### Connessione

Un oggetto del catalogo dati che contiene le proprietà necessarie per connettersi a un particolare archivio dati.

### Crawler

Programma che si connette a un datastore (di origine o di destinazione), avanza attraverso un elenco di classificatori ordinato per priorità per determinare lo schema dei dati e quindi crea tabelle di metadati nel AWS Glue Data Catalog.

### Database

Set di definizioni di tabelle del catalogo dati associate organizzate in un gruppo logico.

### Datastore, origine dati, target dati

Un datastore è un repository per archiviare i dati in modo permanente. Alcuni esempi includono bucket Amazon S3 e database relazionali. Un'origine dati è un datastore utilizzato come input per un processo o una trasformazione. Un target dati è un datastore su cui scrive un processo o una trasformazione.

### Endpoint di sviluppo

Un ambiente che puoi usare per sviluppare e testare i tuoi script ETL. AWS Glue

### Frame dinamico

Una tabella distribuita che supporta dati nidificati come strutture e array. Ogni record è auto descrittivo, progettato per la flessibilità dello schema con dati semi-strutturati. Ogni record contiene sia i dati, sia lo schema che li descrive. È possibile utilizzare sia frame dinamici che Apache Spark DataFrames negli script ETL e convertirli da uno all'altro. I frame dinamici forniscono una serie di trasformazioni avanzate per la pulizia dei dati e l'ETL.

### Processo

Logica di business necessaria per eseguire il lavoro ETL. È composto da uno script di trasformazione, da origini dati e da destinazioni dati. Le esecuzioni dei processi sono avviate tramite trigger pianificabili o attivabili tramite eventi.

## Pannello di controllo per le prestazioni del processo

AWS Glue fornisce una dashboard di esecuzione completa per i lavori ETL. Nel pannello di controllo vengono visualizzate informazioni sulle esecuzioni dei processi in un intervallo di tempo specifico.

## Interfaccia notebook

Un'esperienza notebook migliorata con una configurazione in un solo clic per semplificare la creazione di processi e l'esplorazione dei dati. Il notebook e le connessioni vengono configurati automaticamente per te. È possibile utilizzare l'interfaccia per notebook basata su Jupyter Notebook per sviluppare, eseguire il debug e distribuire in modo interattivo script e flussi di lavoro utilizzando l'infrastruttura ETL Apache Spark senza server. AWS Glue Nell'ambiente notebook, è possibile anche eseguire query ad hoc, analisi dei dati e visualizzazione (ad esempio tabelle e grafici).

## Script

Codice che estrae i dati dalle fonti, li trasforma e li carica in obiettivi. AWS Glue genera PySpark o script Scala.

## Tabella

Definizione di metadati che rappresenta i tuoi dati. Sia che i dati siano in un file Amazon Simple Storage Service (Amazon S3), in una tabella Amazon Relational Database Service (Amazon RDS) o in un altro set di dati, una tabella definisce lo schema dei dati. Una tabella in AWS Glue Data Catalog è composta dai nomi delle colonne, dalle definizioni dei tipi di dati, dalle informazioni sulle partizioni e da altri metadati relativi a un set di dati di base. Lo schema dei dati è rappresentato nella definizione della tabella. AWS Glue I dati effettivi rimangono nell'archivio dati originale, che si tratti di un file o di una tabella di database relazionale. AWS Glue cataloga i file e le tabelle del database relazionale in. AWS Glue Data Catalog Questi fungono da origini e destinazioni quando si crea un processo ETL.

## Trasformazione

Logica di codice utilizzata per modificare i dati in un formato diverso.

## Trigger

Avvia un processo ETL. È possibile definire i trigger sulla base di un orario programmato o di un evento.

### Editor visivo dei processi

L'editor visivo del processo è un'interfaccia grafica che consente di creare, eseguire e monitorare in modo semplice processi di estrazione, trasformazione e caricamento (ETL) in AWS Glue. Puoi comporre visivamente flussi di lavoro di trasformazione dei dati, eseguirli senza problemi sul motore ETL serverless basato su AWS Glue Apache Spark e ispezionare lo schema e i risultati dei dati in ogni fase del lavoro.

### Worker

Con AWS Glue, paghi solo per il tempo necessario all'esecuzione del processo ETL. Non ci sono risorse da gestire, quindi non ti vengono addebitati costi anticipati e tariffe per il tempo di avvio o di arresto. Ti viene addebitata una tariffa oraria basata sul numero di unità di elaborazione dati (o DPUs) utilizzate per eseguire il processo ETL. Una singola unità di elaborazione dati (DPU) viene anche definita lavoratore. AWS Glue viene fornito con diversi tipi di worker per aiutarvi a selezionare la configurazione più adatta ai vostri requisiti di latenza lavorativa e costi. I worker sono disponibili nelle configurazioni Standard, G.1X, G.2X, G.4X, G.8X, G.12X, G.16X, G.025X e nelle configurazioni R.1X, R.2X, R.8X e R.8X ottimizzate per la memoria.

## AWS Glue componenti

AWS Glue fornisce una console e operazioni API per configurare e gestire il carico di lavoro di estrazione, trasformazione e caricamento (ETL). È possibile utilizzare le operazioni API tramite diversi linguaggi specifici e il comando ( SDKs). AWS Command Line Interface AWS CLIPer informazioni sull'utilizzo di AWS CLI, vedere AWS CLI Command Reference.

AWS Glue utilizza il AWS Glue Data Catalog per archiviare i metadati relativi a fonti di dati, trasformazioni e destinazioni. Il catalogo dati sostituisce il metastore Apache Hive. AWS Glue Jobs system Fornisce un'infrastruttura gestita per la definizione, la pianificazione e l'esecuzione delle operazioni ETL sui dati. Per ulteriori informazioni sull' AWS Glue API, consulta. AWS Glue API

### AWS Glue console

La AWS Glue console viene utilizzata per definire e orchestrare il flusso di lavoro ETL. La console richiama diverse operazioni API nel AWS Glue Data Catalog e AWS Glue Jobs system per eseguire le seguenti attività:

- Definisci AWS Glue oggetti come lavori, tabelle, crawler e connessioni.

- Pianificare l'esecuzione dei crawler.
- Definire eventi o programmi per i trigger di processo.
- Cerca e filtra gli elenchi di oggetti. AWS Glue
- Modificare gli script di trasformazione.

## AWS Glue Data Catalog

AWS Glue Data Catalog È il tuo archivio persistente di metadati tecnici nel AWS Cloud.

Ogni AWS account ne ha uno AWS Glue Data Catalog per AWS regione. Ogni catalogo dati è una raccolta altamente scalabile di tabelle organizzate in database. Una tabella è la rappresentazione dei metadati di una raccolta di dati strutturati o semistrutturati archiviati in fonti come Amazon RDS, Apache Hadoop Distributed File System, Amazon Service e altre. OpenSearch AWS Glue Data Catalog Fornisce un repository uniforme in cui diversi sistemi possono archiviare e trovare metadati per tenere traccia dei dati nei silos di dati. È quindi possibile utilizzare i metadati per eseguire query e trasformare i dati in modo coerente su un'ampia varietà di applicazioni.

Utilizzi il Data Catalog insieme alle AWS Identity and Access Management policy e a Lake Formation per controllare l'accesso alle tabelle e ai database. In questo modo, consenti a diversi gruppi nella tua azienda di pubblicare in modo sicuro i dati per la più ampia organizzazione proteggendo allo stesso tempo le informazioni sensibili in modo altamente granulare.

Il Data Catalog, insieme CloudTrail a Lake Formation, offre anche funzionalità complete di audit e governance, con tracciamento delle modifiche allo schema e controlli di accesso ai dati. Questo contribuisce a garantire che i dati non vengono modificati impropriamente o condivisi inavvertitamente.

Per informazioni su come proteggere e controllare ila AWS Glue Data Catalog, consulta:

- AWS Lake Formation – Per ulteriori informazioni, consulta Cos'è AWS Lake Formation? nella Guida per gli sviluppatori di AWS Lake Formation .
- CloudTrail— Per ulteriori informazioni, consulta What Is CloudTrail? nella Guida AWS CloudTrail per l'utente.

Di seguito sono riportati altri AWS servizi e progetti open source che utilizzano: AWS Glue Data Catalog

- Amazon Athena – Per ulteriori informazioni, consulta Comprensione di tabelle, database e catalogo dati nella Guida per l'utente di Amazon Athena.
- Amazon Redshift Spectrum – Per ulteriori informazioni, consulta Utilizzo di Amazon Redshift Spectrum per eseguire query su dati esterni nella Guida per gli sviluppatori di Amazon Redshift.
- Amazon EMR – Per ulteriori informazioni, consulta Utilizzo di policy basate su risorse per l'accesso Amazon EMR ad AWS Glue Data Catalog nella Guida alla gestione di Amazon EMR.
- AWS Glue Data Catalog client per Apache Hive metastore — Per ulteriori informazioni su questo GitHub progetto, consulta AWS Glue Data Catalog Client for Apache Hive Metastore.

## AWS Glue crawler e classificatori

AWS Glue consente inoltre di configurare crawler in grado di scansionare i dati in tutti i tipi di repository, classificarli, estrarne le informazioni sullo schema e archiviare automaticamente i metadati in. AWS Glue Data Catalog AWS Glue Data Catalog Possono quindi essere utilizzati per guidare le operazioni ETL.

Per informazioni su come configurare i crawler e i classificatori, consulta l'articolo Utilizzo dei crawler per compilare il Catalogo dati. Per informazioni su come programmare crawler e classificatori utilizzando l'API, consulta. AWS Glue API crawler e classificatori

## AWS Glue Operazioni ETL

Utilizzando i metadati nel Data Catalog, AWS Glue puoi generare automaticamente script Scala o PySpark (l'API Python per Apache Spark) AWS Glue con estensioni che puoi usare e modificare per eseguire varie operazioni ETL. Ad esempio, puoi estrarre, pulire e trasformare dati grezzi, quindi memorizzare il risultato in un diverso archivio, dove può essere interrogato e analizzato. Tale script potrebbe convertire un file CSV in una struttura dati relazionale e salvarlo in Amazon Redshift.

Per ulteriori informazioni su come utilizzare le funzionalità ETL, consulta. AWS Glue Script di programmazione Spark

## Streaming di ETL in AWS Glue

AWS Glue consente di eseguire operazioni ETL sullo streaming di dati utilizzando processi in esecuzione continua. AWS Glue streaming ETL è basato sul motore di streaming strutturato Apache Spark e può importare flussi da Amazon Kinesis Data Streams, Apache Kafka e Amazon Managed

Streaming for Apache Kafka (Amazon MSK). Streaming ETL può pulire e trasformare i dati di streaming e caricarli in Amazon S3 o in archivi dati JDBC. Usa Streaming ETL in AWS Glue per elaborare dati di eventi come flussi IoT, clickstream e log di rete.

Se si conosce lo schema dell'origine dati di streaming, è possibile specificarlo in una tabella del catalogo dati. In caso contrario, è possibile abilitare il rilevamento dello schema nel processo ETL di streaming. Il processo determina automaticamente lo schema dai dati in entrata.

Il job ETL di streaming può utilizzare sia trasformazioni AWS Glue integrate che trasformazioni native di Apache Spark Structured Streaming. Per ulteriori informazioni, consulta Operazioni sullo streaming DataFrames /Datasets sul sito Web di Apache Spark.

Per ulteriori informazioni, consulta the section called "Aggiunta di processi di streaming ETL".

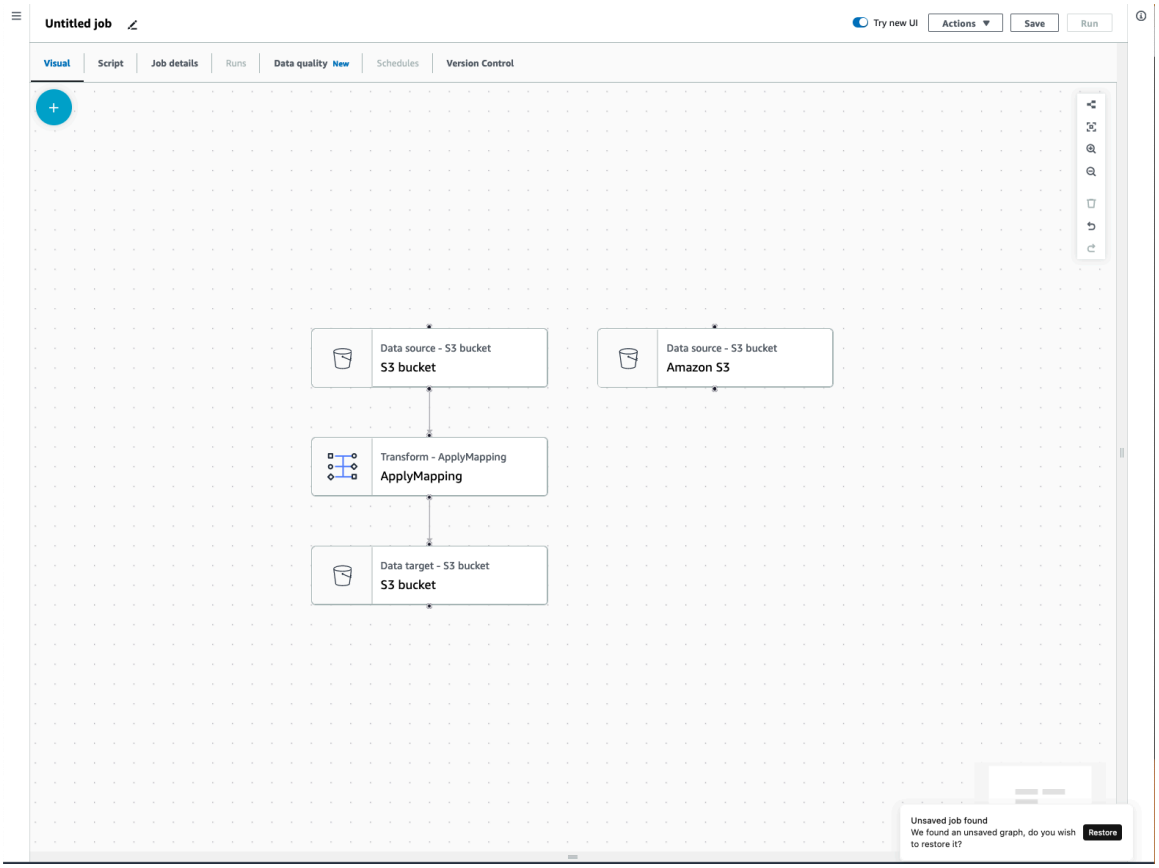
## Il sistema dei lavori AWS Glue

AWS Glue Jobs system Fornisce un'infrastruttura gestita per orchestrare il flusso di lavoro ETL. È possibile creare lavori AWS Glue che automatizzino gli script utilizzati per estrarre, trasformare e trasferire dati in posizioni diverse. I processi possono essere programmati e concatenati oppure possono essere attivati da eventi quali l'arrivo di nuovi dati.

Per ulteriori informazioni sull'utilizzo di AWS Glue Jobs system, vedere. Monitoraggio AWS Glue Per informazioni sulla programmazione tramite l'API AWS Glue Jobs system , consulta API dei processi.

## Componenti ETL visivi

AWS Glue ti consente di creare lavori ETL attraverso una tela visiva che puoi manipolare.




## Menu dei processi ETL

Le opzioni di menu nella parte superiore del canvas consentono di accedere alle varie visualizzazioni e ai dettagli di configurazione relativi al processo.

- Visivo: il canvas dell'editor di processo visivo. Da qui è possibile aggiungere nodi per creare un processo.
- Script: la rappresentazione tramite script del tuo lavoro ETL. AWS Glue genera lo script in base alla rappresentazione visiva del tuo lavoro. È inoltre possibile modificare lo script o scaricarlo.

> **Note**
>
> Se scegli di modificare lo script, l'esperienza di creazione del processo viene convertita in modo permanente in modalità di solo script. Successivamente, non è più possibile utilizzare l'editor visivo per modificare il processo. È necessario aggiungere tutte le origini, le trasformazioni e le destinazioni del processo e apportare tutte le modifiche necessarie con l'editor visivo prima di scegliere di modificare lo script.

- Dettagli del processo: la scheda Dettagli del processo consente di configurare il processo impostandone le proprietà. Sono disponibili proprietà di base, come nome e descrizione del processo, ruolo IAM, tipo di processo, versione di AWS Glue, lingua, tipo di worker, numero di worker, segnalibro del processo, esecuzione flessibile, numero di ritirati e timeout del processo, così come alcune proprietà avanzate, come connessioni, librerie, parametri di processo e tag.
- Esecuzioni: dopo l'esecuzione del processo, è possibile accedere a questa scheda per visualizzare i processi eseguiti in passato.
- Qualità dei dati: la qualità dei dati consente di valutare e monitorare la qualità delle risorse di dati. Puoi saperne di più su come utilizzare la qualità dei dati in questa scheda e aggiungere una trasformazione della qualità dei dati al tuo processo.
- Pianificazioni: i processi che hai pianificato vengono visualizzati in questa scheda. Se non esistono pianificazioni collegate a questo processo, questa scheda non è accessibile.
- Controllo della versione: puoi utilizzare Git con il tuo processo configurandolo in un repository Git.

## Pannelli ETL visivi

Quando lavori nel canvas, sono disponibili diversi pannelli che ti aiutano a configurare i nodi o a visualizzare l'anteprima dei dati e visualizzare lo schema di output.

- Proprietà: il pannello Proprietà viene visualizzato quando si sceglie un nodo nel canvas.
- Anteprima dei dati: il pannello di anteprima dei dati fornisce un'anteprima dell'output dei dati in modo da poter prendere decisioni prima di eseguire il processo ed esaminare l'output.
- Schema di output: la scheda Schema di output consente di visualizzare e modificare lo schema dei nodi di trasformazione.

### Ridimensionamento dei pannelli

È possibile ridimensionare il pannello Proprietà sul lato destro dello schermo e il pannello inferiore che contiene le schede Anteprima dati e Schema di output facendo clic sul bordo del pannello e trascinandolo a sinistra e a destra o su e giù.

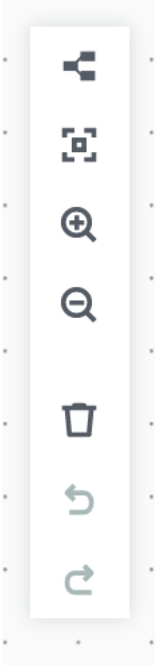
- Pannello delle proprietà: ridimensiona il pannello delle proprietà facendo clic sul bordo del canvas sul lato destro dello schermo e trascinandolo verso sinistra per aumentarne la larghezza. Per impostazione predefinita, il pannello è compresso, mentre quando viene selezionato un nodo il pannello delle proprietà si apre alla dimensione predefinita.

- Pannello Anteprima dei dati e Schema di output: ridimensiona il pannello inferiore facendo clic sul bordo inferiore del canvas nella parte inferiore dello schermo e trascinandolo verso l'alto per aumentarne l'altezza. Per impostazione predefinita, il pannello è compresso, mentre quando viene selezionato un nodo il pannello inferiore si apre alla dimensione predefinita.

## Canvas del processo

È possibile aggiungere, rimuovere e move/reorder nodi direttamente nell'area di disegno di Visual ETL. Puoi immaginarlo come uno spazio di lavoro per creare un processo ETL completamente funzionale, a partire da un'origine dati fino alla destinazione dati.

Quando lavori con i nodi sul canvas, hai a disposizione una barra degli strumenti che può aiutarti a ingrandire e ridurre le dimensioni, rimuovere nodi, creare o modificare connessioni tra i nodi, cambiare l'orientamento del flusso di processo e annullare o ripetere un'operazione.

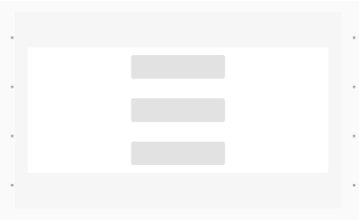


La barra degli strumenti mobile è ancorata al bordo in alto a destra del canvas e contiene diverse immagini che eseguono altrettante operazioni:

- Icona del layout: la prima icona nella barra degli strumenti è l'icona del layout. Per impostazione predefinita, la direzione dei processi visivi è dall'alto verso il basso. Riorganizza la direzione del processo visivo disponendo i nodi orizzontalmente da sinistra a destra. Facendo nuovamente clic sull'icona del layout, la direzione torna dall'alto verso il basso.

- Icona Ricentra: questa icona consente di modificare la visualizzazione del canvas centrandola. È possibile utilizzarla con processi di grandi dimensioni per tornare alla posizione centrale.
- Icona Ingrandisci: questa icona consente di aumentare la dimensione dei nodi sul canvas.
- Icona Riduci: questa icona consente di ridurre la dimensione dei nodi sul canvas.
- Icona del cestino: l'icona del cestino rimuove un nodo dal processo visivo. Prima è necessario selezionare un nodo.
- Icona Annulla: questa icona consente di annullare l'ultima operazione eseguita sul processo visivo.
- Icona Ripeti: questa icona consente di ripetere l'ultima operazione eseguita sul processo visivo.

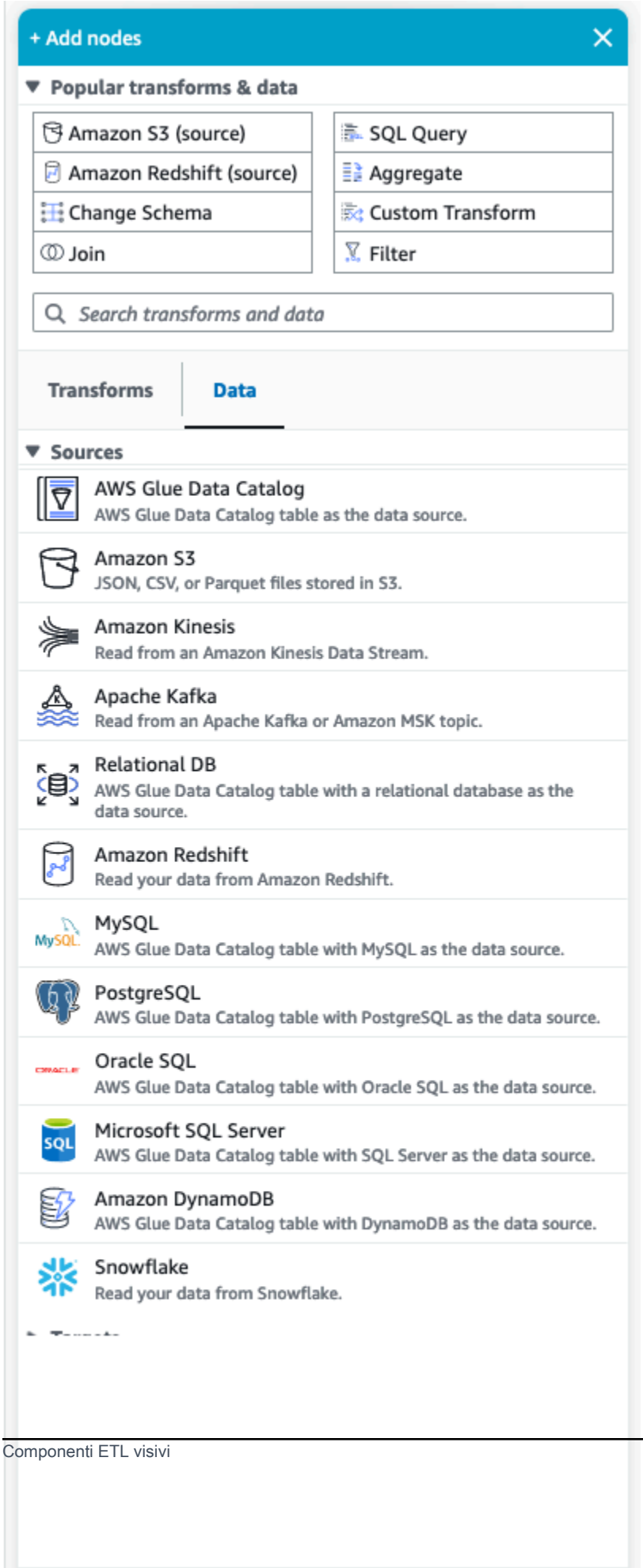
Utilizzo della minimappa



## Pannello delle risorse

Il pannello delle risorse contiene tutte le origini dati, le operazioni di trasformazione e le connessioni disponibili. Apri il pannello delle risorse sul canvas facendo clic sull'icona "+". Si aprirà il pannello delle risorse.

Per chiudere il pannello delle risorse, fai clic sulla X nell'angolo in alto a destra del pannello delle risorse. In questo modo il pannello rimarrà nascosto fino a quando non lo riaprirai.


+ Add nodes
Popular transforms & data
Amazon S3 (source)
Amazon Redshift (source)
Change Schema
Join
SQL Query
Aggregate
Custom Transform
Filter
Search transforms and data
Transforms
Data
Sources
AWS Glue Data Catalog
AWS Glue Data Catalog table as the data source.
Amazon S3
JSON, CSV, or Parquet files stored in S3.
Amazon Kinesis
Read from an Amazon Kinesis Data Stream.
Apache Kafka
Read from an Apache Kafka or Amazon MSK topic.
Relational DB
AWS Glue Data Catalog table with a relational database as the data source.
Amazon Redshift
Read your data from Amazon Redshift.
MySQL
AWS Glue Data Catalog table with MySQL as the data source.
PostgreSQL
AWS Glue Data Catalog table with PostgreSQL as the data source.
Oracle SQL
AWS Glue Data Catalog table with Oracle SQL as the data source.
Microsoft SQL Server
AWS Glue Data Catalog table with SQL Server as the data source.
Amazon DynamoDB
AWS Glue Data Catalog table with DynamoDB as the data source.
Snowflake
Read your data from Snowflake.

Trasformazioni e dati comuni

Nella parte superiore del pannello è presente una raccolta di Trasformazioni e dati comuni. Questi nodi sono comunemente usati in AWS Glue. Scegline uno per aggiungerlo al canvas. Puoi anche nascondere Trasformazioni e dati comuni facendo clic sul triangolo accanto all'intestazione Trasformazioni e dati comuni.

Nella sezione Trasformazioni e dati comuni, puoi cercare trasformazioni e nodi di origini dati. I risultati vengono visualizzati durante la digitazione. Più lettere aggiungi alla tua query di ricerca, più l'elenco dei risultati si ridurrà. I risultati della ricerca vengono compilati in base alla and/or descrizione del nome del nodo. Scegli il nodo per aggiungerlo al canvas.

Trasformazioni e dati

Esistono due schede che organizzano i nodi in Trasformazioni e Dati.

Trasformazioni: quando si sceglie la scheda Trasformazioni, è possibile selezionare tutte le trasformazioni disponibili. Scegli una trasformazione per aggiungerla al canvas. Puoi anche scegliere Aggiungi trasformazione nella parte inferiore dell'elenco Trasformazioni; questa operazione aprirà una nuova pagina alla documentazione per la creazione di Trasformazioni visive personalizzate. Seguendo i passaggi potrai creare trasformazioni personalizzate. Le trasformazioni verranno quindi visualizzate nell'elenco delle trasformazioni disponibili.

Dati: la scheda dati contiene tutti i nodi per Origini e Destinazioni. È possibile nascondere le origini e le destinazioni facendo clic sul triangolo accanto all'intestazione Origini o Destinazioni. È possibile visualizzare le origini e le destinazioni facendo nuovamente clic sul triangolo. Scegli un nodo di origine o di destinazione per aggiungerlo al canvas. È inoltre possibile scegliere Gestisci connessioni per aggiungere una nuova connessione. Si aprirà la pagina Connettori nella console.

# AWS Glue per Spark e AWS Glue per Ray

AWS Glue In Apache Spark (AWS Glue ETL), puoi usarlo per PySpark scrivere codice Python per gestire i dati su larga scala. Spark è una soluzione comune per questo problema, ma i data engineer con background incentrati su Python possono trovare la transizione poco intuitiva. Il DataFrame modello Spark non è perfettamente «Python», il che riflette il linguaggio Scala e il runtime Java su cui è basato.

In AWS Glue, puoi usare i job della shell Python per eseguire integrazioni di dati Python native. Questi processi vengono eseguiti su una singola EC2 istanza Amazon e sono limitati dalla capacità di

tale istanza. Ciò limita la velocità di trasmissione effettiva dei dati che è possibile elaborare e diventa costoso da mantenere quando si tratta di Big Data.

AWS Glue for Ray ti consente di scalare i carichi di lavoro in Python senza investimenti sostanziali nell'apprendimento di Spark. È possibile sfruttare alcuni scenari in cui Ray si comporta meglio. Offrendoti una scelta, puoi utilizzare i punti di forza di Spark e Ray in base ai casi.

AWS Glue ETL e AWS Glue for Ray sono fondamentalmente diversi, quindi supportano funzionalità diverse. Controlla le documentazione per determinare le funzionalità supportate.

## Cosa c'è AWS Glue per Ray?

Ray è un framework di calcolo distribuito open source che può essere utilizzato per scalare i carichi di lavoro, con particolare attenzione a Python. Per ulteriori informazioni su Ray, consulta il sito Web di Ray. AWS Glue I lavori Ray e le sessioni interattive ti consentono di utilizzare Ray all'interno AWS Glue.

Puoi usare AWS Glue for Ray per scrivere script Python per calcoli che verranno eseguiti in parallelo su più macchine. Nei processi e nelle sessioni interattive di Ray, è possibile utilizzare le librerie Python comuni come pandas per facilitare la scrittura e l'esecuzione dei flussi di lavoro. Per ulteriori informazioni sui set di dati di Ray, consulta Set di dati di Ray nella documentazione di Ray. Per ulteriori informazioni su Pandas, consulta il sito Web di Pandas.

Quando usi AWS Glue For Ray, puoi eseguire i flussi di lavoro di Pandas su Big Data su scala aziendale, con solo poche righe di codice. Puoi creare un job Ray dalla console o dall'SDK. AWS Glue AWS Puoi anche aprire una sessione AWS Glue interattiva per eseguire il codice in un ambiente Ray senza server. I lavori visivi in non AWS Glue Studio sono ancora supportati.

AWS Glue for Ray jobs ti consente di eseguire uno script in base a una pianificazione o in risposta a un evento di Amazon EventBridge. Jobs archivia le informazioni di registro e le statistiche di monitoraggio in CloudWatch modo da consentirti di comprendere lo stato e l'affidabilità dello script. Per ulteriori informazioni sul sistema dei AWS Glue job, vederethe section called "Utilizzo dei processi Ray".

Ray automatizza il lavoro di dimensionamento del codice Python distribuendo l'elaborazione su un cluster di macchine che riconfigura in tempo reale, in base al carico. Ciò può portare a un miglioramento delle prestazioni per dollaro di determinati carichi di lavoro. Con Ray jobs, abbiamo integrato la scalabilità automatica in modo nativo nel modello di AWS Glue lavoro, in modo da poter sfruttare appieno questa funzionalità. I lavori Ray vengono eseguiti su AWS Graviton, con conseguente aumento delle prestazioni complessive in termini di prezzo.

Oltre ai risparmi sui costi, è possibile utilizzare la scalabilità automatica nativa per eseguire i carichi di lavoro Ray senza investire tempo in operazioni di manutenzione, ottimizzazione e amministrazione del cluster. Puoi usare librerie open source familiari pronte all'uso, come pandas, e l'SDK per Pandas. AWS Questi migliorano la velocità di iterazione durante lo sviluppo su AWS Glue per Ray. Quando usi AWS Glue for Ray, sarai in grado di sviluppare ed eseguire rapidamente carichi di lavoro di integrazione dei dati a costi contenuti.

# Conversione da schemi semistrutturati a schemi relazionali con AWS Glue

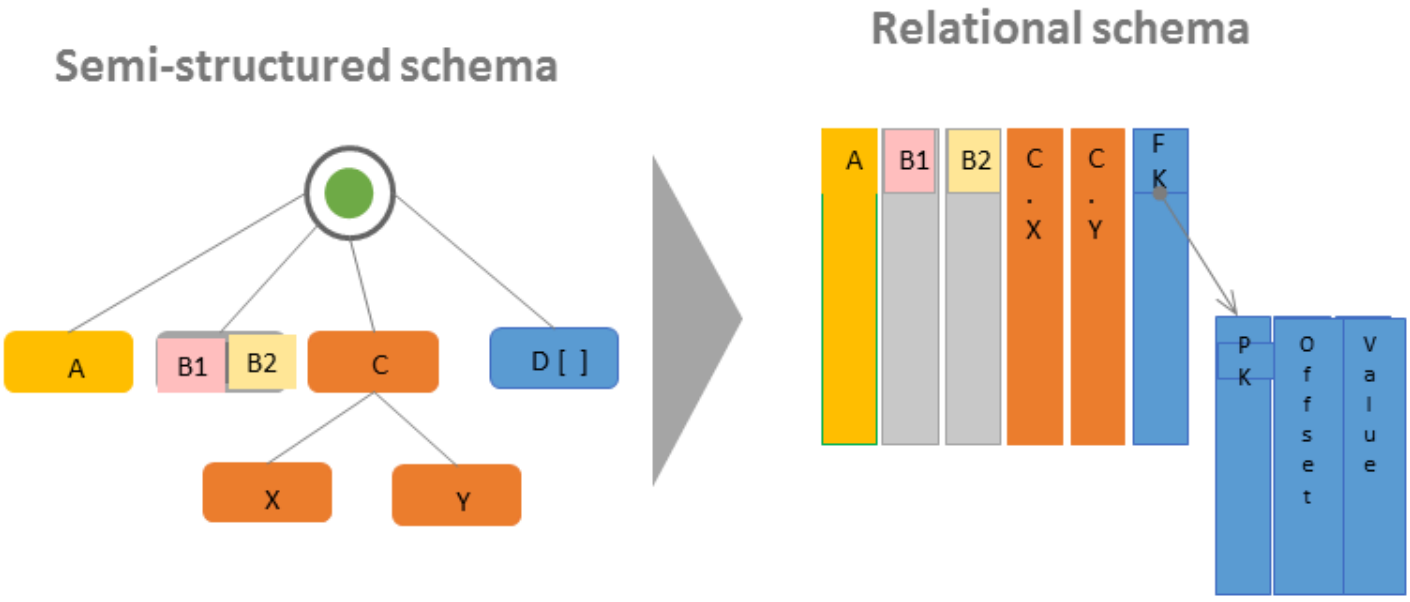
La conversione dei dati semistrutturati in tabelle relazionali è piuttosto comune. Concettualmente, stai appiattendo uno schema gerarchico a uno schema relazionale. AWS Gluepuò eseguire questa conversione per te. on-the-fly

I dati semistrutturati in genere contengono mark-up per identificare le entità all'interno dei dati. Si possono avere strutture di dati annidate senza schema fisso. Per ulteriori informazioni sui dati semistrutturati, consulta Dati semistrutturati in Wikipedia.

I dati relazionali sono rappresentati da tabelle che contengono righe e colonne. Le relazioni tra tabelle possono essere rappresentate da una relazione chiave primaria (PK) su chiave esterna (FK). Per ulteriori informazioni, consulta Database relazionale in Wikipedia.

AWS Glue usa i crawler per dedurre gli schemi per i dati semistrutturati. Quindi trasforma i dati in uno schema relazionale utilizzando un processo ETL (estrarre, trasformare e caricare). Ad esempio, è possibile analizzare i dati JSON da file di origine Amazon Simple Storage Service (Amazon S3) a tabelle Amazon Relational Database Service (Amazon RDS). Comprendere come AWS Glue gestisce le differenze tra gli schemi può aiutare a capire il processo di trasformazione.

Questo diagramma mostra come AWS Glue trasforma uno schema semistrutturato in uno schema relazionale.




Il diagramma illustra quanto segue:

- Il singolo valore `A` converte direttamente in una colonna relazionale.
- La coppia di valori `B1` e `B2` converte in due colonne relazionali.
- Struttura `C`, con figli `X` e `Y`, converte in due colonne relazionali.
- L'array `D[]` converte in una colonna relazionale con una chiave esterna (FK) che punta a un'altra tabella relazionale. Oltre a una chiave primaria (PK), la seconda tabella relazionale dispone di colonne che contengono l'offset e il valore degli oggetti nell'array.

# AWS Sistemi tipo Glue

AWS Glue utilizza sistemi di tipo multiplo per fornire un'interfaccia versatile su sistemi di dati che archiviano i dati in modi molto diversi. Questo documento chiarisce le ambiguità dei sistemi e degli standard di dati di tipo AWS Glue.

## AWS Tipi di Glue Data Catalog

Il catalogo dati è un registro di tabelle e campi archiviati in vari sistemi di dati, un metastore. Quando i componenti AWS Glue, come AWS i crawler AWS Glue e i job Glue with Spark, scrivono nel Data Catalog, lo fanno con un sistema di tipi interno per tracciare i tipi di campi. Questi valori sono mostrati nella colonna Tipo di dati dello schema della tabella nella AWS Glue Console. Questo sistema dei tipi è basato sul sistema dei tipi di Apache Hive. Per ulteriori informazioni sul sistema dei tipi di Apache Hive, consulta la sezione Tipi nella wiki di Apache Hive. Per ulteriori informazioni su tipi e supporti specifici, gli esempi sono forniti nella AWS Glue Console, come parte di Schema Builder.

### Convalida, compatibilità e altri usi

Il catalogo dati non convalida i tipi scritti nei campi del tipo. Quando i componenti AWS Glue leggono e scrivono nel Data Catalog, saranno compatibili tra loro. AWS I componenti Glue mirano inoltre a preservare un alto grado di compatibilità con i tipi Hive. Tuttavia, i componenti AWS Glue non garantiscono la compatibilità con tutti i tipi di Hive. Ciò consente l'interoperabilità con strumenti come Athena DDL quando si lavora con le tabelle nel catalogo dati.

Poiché il catalogo dati non convalida i tipi, altri servizi possono utilizzare il catalogo dati per tenere traccia dei tipi utilizzando sistemi strettamente conformi al sistema dei tipi di Hive o a qualsiasi altro sistema.

## Tipi negli script AWS Glue with Spark

Quando uno script AWS Glue with Spark interpreta o trasforma un set di dati`DynamicFrame`, forniamo una rappresentazione in memoria del set di dati così come viene utilizzato nello script. L'obiettivo di un `DynamicFrame` è simile a quello del `DataFrame` di Spark: modella il set di dati in modo che Spark possa pianificare ed eseguire trasformazioni sui dati. Garantiamo che la rappresentazione del tipo di `DynamicFrame` sia intercompatibile con il `DataFrame` fornendo i metodi `toDF` e `fromDF`.

Se le informazioni sul tipo possono essere inferite o fornite a un `DataFrame`, possono essere inferite o fornite a un `DynamicFrame`, se non diversamente documentato. Quando forniamo lettori o scrittori

ottimizzati per formati di dati specifici, se Spark è in grado di leggere o scrivere i tuoi dati, i nostri lettori e scrittori forniti saranno in grado di farlo, ad esclusione delle limitazioni documentate. Per ulteriori informazioni su lettori e scrittori, consulta the section called "Opzioni del formato dei dati".

### Il tipo di scelta

Il `DynamicFrames` fornisce un meccanismo per modellare i campi in un set di dati il cui valore può avere tipi incoerenti su disco tra le righe. Ad esempio, un campo può contenere un numero memorizzato come stringa in alcune righe e un numero intero in altre. Questo meccanismo è un tipo in memoria denominato `Choice`. Forniamo trasformazioni, come il `ResolveChoice` metodo, per risolvere le colonne Choice in un tipo concreto. AWS Glue ETL non scriverà il tipo Choice nel Data Catalog durante il normale funzionamento; i tipi Choice esistono solo nel contesto dei modelli di DynamicFrame memoria dei set di dati. Per un esempio di utilizzo del tipo Choice, consulta the section called "Esempio di preparazione dei dati".

## AWS Tipi di Glue Crawler

I crawler mirano a produrre uno schema coerente e utilizzabile per il set di dati, quindi a memorizzarlo in Data Catalog per utilizzarlo in altri componenti AWS Glue e in Athena. I crawler gestiscono i tipi come descritto nella sezione precedente sul catalogo dati, the section called "AWS Tipi di Glue Data Catalog". Per produrre un tipo utilizzabile negli scenari di tipo "Choice", in cui una colonna contiene valori di due o più tipi, i crawler creeranno un tipo `struct` che modella i tipi potenziali.

# Nozioni di base su AWS Glue

Nelle sezioni seguenti vengono fornite informazioni sulla configurazione di AWS Glue. Non tutte le sezioni per la configurazione sono necessarie per iniziare a utilizzare AWS Glue. È possibile utilizzare le istruzioni in base alle necessità per configurare le autorizzazioni IAM, la crittografia, il DNS se si utilizza un VPC, l'ambiente per accedere agli archivi dati o se si utilizzano le sessioni interattive.

Argomenti


- Panoramica dell'utilizzo di AWS Glue
- Configurazione delle autorizzazioni IAM per AWS Glue
- Configurazione dei profili AWS Glue di utilizzo
- Nozioni di base su AWS Glue Data Catalog
- Impostazione dell'accesso di rete agli archivi di dati
- Configurazione della crittografia in AWS Glue
- Creazione di reti per lo sviluppo per AWS Glue


## Panoramica dell'utilizzo di AWS Glue

Con AWS Glue, memorizzi i metadati in. AWS Glue Data Catalog Puoi utilizzare questi metadati per orchestrare i processi ETL che trasformano le origini dati e caricano il data warehouse o il data lake. Le fasi seguenti descrivono il flusso di lavoro generale e alcune delle scelte che effettui quando usi AWS Glue.

> **Note**
>
> È possibile seguire i passaggi riportati di seguito oppure creare un flusso di lavoro che esegua automaticamente i passaggi da 1 a 3. Per ulteriori informazioni, consulta the section called "Esecuzione di attività ETL complesse utilizzando gli schemi e i flussi di lavoro".

1. Compila il file AWS Glue Data Catalog con le definizioni delle tabelle.

   Nella console, per gli archivi dati persistenti, è possibile aggiungere un crawler per popolare AWS Glue Data Catalog. Puoi avviare la procedura guidata Add crawler (Aggiungi crawler) dall'elenco delle tabelle o dall'elenco dei crawler. Puoi scegliere uno o più datastore a cui accede il tuo

crawler. Puoi anche creare una pianificazione per determinare la frequenza di esecuzione del crawler. Per i flussi di dati, è possibile creare manualmente la definizione della tabella e definire le proprietà del flusso.

Facoltativamente, puoi fornire un classificatore personalizzato che ricava lo schema dei dati. Puoi creare classificatori personalizzati usando un pattern grok. AWS Glue fornisce tuttavia classificatori integrati che vengono usati automaticamente dai crawler se un classificatore personalizzato non riconosce i dati. Quando definisci un crawler, non devi necessariamente selezionare un classificatore. Per ulteriori informazioni sui classificatori in AWS Glue, consulta Definizione e gestione dei classificatori.

Il crawling di alcuni tipi di datastore richiede una connessione che fornisce l'autenticazione e le informazioni sull'ubicazione. Se necessario, puoi creare una connessione che fornisce queste informazioni obbligatorie nella console AWS Glue.

Il crawler legge il datastore e crea definizioni dei dati e tabelle denominate nel AWS Glue Data Catalog. Queste tabelle sono organizzate in un database di tua scelta. Puoi inoltre popolare il catalogo dati con tabelle create manualmente. Con questo metodo, fornisci lo schema e altri metadati per creare le definizioni di tabelle nel catalogo dati. Poiché questo metodo può essere un po' noioso e suscettibile di errori, spesso è meglio avere un crawler che crea le definizioni di tabella.

Per ulteriori informazioni sulla compilazione delle definizioni delle AWS Glue Data Catalog tabelle, vedere. Creazione di tabelle

2. Definisci un processo che descrive la trasformazione dei dati dall'origine alla destinazione.

   Di solito, per creare un processo, devi effettuare le seguenti scelte:

   - Scegliete una tabella tra AWS Glue Data Catalog quelle da utilizzare come fonte del lavoro. Il processo utilizza questa definizione di tabella per accedere alla tua origine dati e interpretare il formato dei dati.
   - Scegli una tabella o una posizione tra AWS Glue Data Catalog quelle da utilizzare come destinazione del lavoro. Il processo utilizza queste informazioni per accedere al tuo datastore.
   - Indica a AWS Glue di generare uno script per trasformare l'origine nella destinazione. AWS Glue genera il codice per chiamare le trasformazioni integrate e convertire i dati dal formato dello schema di origine a quello dello schema di destinazione. Queste trasformazioni eseguono le operazioni quali copiare i dati, rinominare le colonne e filtrare i dati per trasformare i dati in base alle esigenze. Puoi modificare lo script nella console AWS Glue.

Per ulteriori informazioni sulla definizione dei processi in AWS Glue, consulta Creazione di processi ETL visivi.

3. Esegui il tuo processo per trasformare i dati.

   Puoi eseguire il processo on demand oppure avviarlo in base a uno dei seguenti tipi di trigger:

   - Trigger basato su una pianificazione cron.
   - Trigger basato su un evento: ad esempio il completamento di un altro processo può avviare un processo AWS Glue.
   - Trigger che avvia un processo on demand.

   Per ulteriori informazioni sui trigger in AWS Glue, consulta Avvio di lavori e crawler utilizzando i trigger.

4. Monitora i crawler pianificati e i processi attivati.

   Usa la console AWS Glue per visualizzare gli elementi seguenti:

   - Dettagli ed errori dell'esecuzione del processo.
   - Dettagli ed errori dell'esecuzione del crawler.
   - Notifiche sulle attività AWS Glue

   Per ulteriori informazioni sul monitoraggio di crawler e processi in AWS Glue, consulta Monitoraggio AWS Glue.

# Configurazione delle autorizzazioni IAM per AWS Glue

Le istruzioni contenute in questo argomento consentono di configurare rapidamente le autorizzazioni AWS Identity and Access Management (IAM) per. AWS Glue Completa le seguenti operazioni:

- Concedi alle tue identità IAM l'accesso alle risorse. AWS Glue
- Crea un ruolo di servizio per eseguire lavori, accedere ai dati ed eseguire attività di AWS Glue Data Quality.

Per istruzioni dettagliate che puoi utilizzare per personalizzare le autorizzazioni IAM AWS Glue, consultaConfigurazione delle autorizzazioni IAM per AWS Glue.

Per configurare le autorizzazioni IAM per AWS Glue in Console di gestione AWS

1. Accedi a Console di gestione AWS e apri la AWS Glue console all'indirizzo https://console.aws.amazon.com/glue/.
2. Selezionare Getting started (Nozioni di base).
3. In Prepara il tuo account per AWS Glue, scegli Configura le autorizzazioni IAM.
4. Scegli le identità IAM (ruoli o utenti) a cui vuoi concedere le AWS Glue autorizzazioni. AWS Glue allega la policy `AWSGlueConsoleFullAccess` gestita a queste identità. Puoi saltare questo passaggio se desideri impostare queste autorizzazioni manualmente o impostare solo un ruolo di servizio predefinito.
5. Scegliere Successivo.
6. Scegliere il livello di accesso ad Amazon S3 di cui hanno bisogno i tuoi ruoli e i tuoi utenti. Le opzioni scelte in questo passaggio vengono applicate a tutte le identità selezionate.

   a. In Scegli le posizioni S3, scegliere le posizioni Amazon S3 a cui desideri concedere l'accesso.

   b. Quindi, seleziona se le tue identità devono avere accesso di sola lettura (consigliato) o di lettura e scrittura alle posizioni che hai selezionato in precedenza. AWS Glue aggiunge politiche di autorizzazione alle tue identità in base alla combinazione di posizioni e autorizzazioni di lettura o scrittura selezionate.

      La tabella seguente mostra le autorizzazioni associate per l' AWS Glue accesso ad Amazon S3.

| Se si sceglie... | AWS Glue allega... |
| --- | --- |
| Nessuna modifica | Nessuna autorizzazione. AWS Glue non apporterà alcuna modifica alle autorizzazioni della tua identità. |
| Concedi l'accesso a posizioni Amazon S3 specifiche (solo lettura) | La policy gestita dal cliente, `AWSGlueConsole-S3-read-only-policy`, concede l'accesso a specifiche sedi Amazon S3 con autorizzazioni di sola lettura. |

| Se si sceglie... | AWS Glue allega... |
| --- | --- |
| | JSON |

```
{
  "Version":"2012-10-17",

  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::amzn
-s3-demo-bucket1/*",
        "arn:aws:s3:::amzn
-s3-demo-bucket2/*",
        "arn:aws:s3:::amzn
-s3-demo-bucket3",
        "arn:aws:s3:::amzn
-s3-demo-bucket"
      ],
      "Condition": {
        "StringEquals": {
          "aws:Reso
urceAccount": "00000000
00"
        }
      }
    }
  ]
}
```

| Se si sceglie... | AWS Glue allega... |
| --- | --- |
| Concedi l'accesso a posizioni Amazon S3 specifiche (lettura e scrittura) | La `AWSGlueConsole-S3-read-and-write-policy` concede l'accesso a posizioni Amazon S3 specifiche con autorizzazioni di lettura e scrittura.<br><br>JSON<br><br>**{**<br>**"Version":"2012-10-17",**<br><br>**"Statement": [**<br>**{**<br>**"Effect": "Allow",**<br>**"Action": [**<br>**"s3:GetObject",**<br>**"s3:PutObject"**<br>**],**<br>**"Resource": [**<br>**"arn:aws:s3:::aes-siem-00000000000-log/*",**<br>**"arn:aws:s3:::aes-siem-00000000000-snapshot/*",**<br>**"arn:aws:s3:::aes-siem-00000000000-log",**<br>**"arn:aws:s3:::aes-siem-00000000000-snapshot"**<br>**],**<br>**"Condition": {**<br>**"StringEquals": {**<br>**"aws:ResourceAccount": "000000000000"**<br>**}**<br>**}**<br>**}**<br>**]** |

| Se si sceglie... | AWS Glue allega... |
| --- | --- |
| | `}` |

7. Scegli Next (Successivo).
8. Scegli un ruolo di AWS Glue servizio predefinito per il tuo account. Un ruolo di servizio è un ruolo IAM che viene AWS Glue utilizzato per accedere alle risorse di altri AWS servizi per tuo conto. Per ulteriori informazioni, consulta Ruoli di servizio per AWS Glue.

   - Quando scegli il ruolo di AWS Glue servizio standard, AWS Glue crea un nuovo ruolo IAM a tuo Account AWS nome `AWSGlueServiceRole` con le seguenti politiche gestite allegate. Se il tuo account ha già un ruolo IAM denominato `AWSGlueServiceRole` AWS Glue , associa queste politiche al ruolo esistente.
     - AWSGlueServiceRole— Questa policy gestita è necessaria per AWS Glue accedere e gestire le risorse per tuo conto. Consente di AWS Glue creare, aggiornare ed eliminare varie risorse come AWS Glue lavori, crawler e connessioni. Questa politica concede anche le autorizzazioni per accedere ai Amazon CloudWatch log AWS Glue a scopo di registrazione. Per iniziare, consigliamo di utilizzare questa policy per imparare a usare AWS Glue. Man mano che acquisisci maggiore dimestichezza AWS Glue, puoi creare politiche che ti consentano di ottimizzare l'accesso alle risorse in base alle esigenze.
     - AWSGlueConsoleFullAccess— Questa politica gestita garantisce l'accesso completo al AWS Glue servizio tramite. Console di gestione AWS Questa politica concede le autorizzazioni per eseguire qualsiasi operazione all'interno AWS Glue, consentendo all'utente di creare, modificare ed eliminare qualsiasi AWS Glue risorsa in base alle esigenze. Tuttavia, è importante notare che questa politica non concede le autorizzazioni per accedere agli archivi di dati sottostanti o ad altri AWS servizi che potrebbero essere coinvolti nel processo ETL. A causa dell'ampia gamma di autorizzazioni concesse dalla policy `AWSGlueConsoleFullAccess`, è necessario assegnarle con cautela e seguendo il principio del privilegio minimo. In genere si consiglia di creare e utilizzare policy più granulari personalizzate in base a casi d'uso e requisiti specifici, ove possibile.
     - AWSGlueConsole-S3- read-only-policy — Questa policy consente di AWS Glue leggere i dati da bucket Amazon S3 specifici, ma non concede le autorizzazioni per scrivere o modificare i dati in Amazon S3 o

       AWSGlueConsole-S3- read-and-write — Questa policy consente di AWS Glue leggere e scrivere dati su bucket Amazon S3 specifici come parte del processo ETL.

- Quando scegli un ruolo IAM esistente, AWS Glue imposta il ruolo come predefinito, ma non aggiunge `AWSGlueServiceRole` autorizzazioni. Assicurati di aver configurato il ruolo da utilizzare come ruolo di AWS Glue servizio. Per ulteriori informazioni, consultare Fase 1: creare una policy IAM per il servizio AWS Glue e Fase 2: creare un ruolo IAM per AWS Glue.

9. Scegliere Successivo.
10. Infine, rivedere le autorizzazioni che sono state selezionato, quindi scegliere Applica le modifiche. Quando applichi le modifiche, AWS Glue aggiunge le autorizzazioni IAM alle identità che hai selezionato. Puoi visualizzare o modificare le nuove autorizzazioni nella console IAM all'indirizzo. https://console.aws.amazon.com/iam/

Ora hai completato la configurazione minima delle autorizzazioni IAM per. AWS Glue In un ambiente di produzione, ti consigliamo di acquisire familiarità con Sicurezza in AWS Glue e di aiutarti Gestione delle identità e degli accessi per AWS Glue a proteggere AWS le risorse per il tuo caso d'uso.

## Fasi successive

Ora che sono state configurate le autorizzazioni IAM, è possibile esplorare i seguenti argomenti per iniziare a utilizzare AWS Glue:

- Guida introduttiva a Skill AWS GlueAWS Builder
- Nozioni di base su AWS Glue Data Catalog

## Configurazione di AWS Glue Studio

Completa i processi in questa sezione quando utilizzi AWS Glue per l'ETL visivo la prima volta:

Argomenti

- Esaminare le autorizzazioni IAM necessarie per l'utente AWS Glue Studio
- Esaminare le autorizzazioni IAM necessarie per i processi ETL
- Impostazione delle autorizzazioni IAM per AWS Glue Studio
- Configurazione di un VPC per il tuo processo ETL

## Esaminare le autorizzazioni IAM necessarie per l'utente AWS Glue Studio

Per utilizzareAWS Glue Studio, l'utente deve avere accesso a varie AWS risorse. L'utente deve essere in grado di visualizzare e selezionare i bucket Amazon S3, le policy e i ruoli IAM e gli oggetti AWS Glue Data Catalog.

### Autorizzazioni di servizio AWS Glue

AWS Glue Studio utilizza le operazioni e le risorse del servizio AWS Glue. Per utilizzare in modo efficace AWS Glue Studio, l'utente ha bisogno delle autorizzazioni per tali operazioni e risorse. È possibile concedere all'utente di AWS Glue Studio la policy gestita da `AWSGlueConsoleFullAccess` oppure creare una policy personalizzata con un set di autorizzazioni più piccolo.

> **Important**
>
> In base alle best practice di sicurezza, si consiglia di limitare l'accesso rafforzando le policy per limitare ulteriormente l'accesso al bucket Amazon S3 e ai gruppi di log Amazon CloudWatch . Per un esempio di policy Amazon S3, consulta la pagina relativa alla scrittura di policy IAM per concedere l'accesso a un bucket Amazon S3.

### Creazione di criteri IAM personalizzati per AWS Glue Studio

È possibile creare una policy personalizzata con un set di autorizzazioni più piccolo per AWS Glue Studio. La policy può concedere autorizzazioni per un sottoinsieme di oggetti o operazioni. Durante la creazione di una policy personalizzata, utilizza le seguenti informazioni.

Per utilizzare il AWS Glue Studio APIs, includi `glue:UseGlueStudio` nella policy d'azione le tue autorizzazioni IAM. L'utilizzo di `glue:UseGlueStudio` ti permetterà di accedere a tutte le operazioni di AWS Glue Studio anche quando più operazioni vengono aggiunte all'API nel tempo.

Per ulteriori informazioni sulle azioni definite da AWS Glue, consulta Azioni definite da AWS Glue.

#### Azioni di creazione e preparazione dei dati

- SendRecipeAction
- GetRecipeAction

#### Operazioni del grafo aciclico orientato (DAG)

- CreateDag
- UpdateDag
- GetDag
- DeleteDag

Operazioni di processo

- SaveJob
- GetJob
- CreateJob
- DeleteJob
- GetJobs
- UpdateJob

Opzione di esecuzione del processo

- StartJobRun
- GetJobRuns
- BatchStopJobRun
- GetJobRun
- QueryJobRuns
- QueryJobs
- QueryJobRunsAggregated

Operazioni dello schema

- GetSchema
- GetInferredSchema

Operazioni del database

- GetDatabases

Operazioni del piano

- GetPlan

Operazioni della tabella

- SearchTables
- GetTables
- GetTable

Operazioni di connessione

- CreateConnection
- DeleteConnection
- UpdateConnection
- GetConnections
- GetConnection

Operazioni di mappatura

- GetMapping

Operazioni proxy S3

- ListBuckets
- ListObjectsV2
- GetBucketLocation

Operazioni di configurazione di sicurezza

- GetSecurityConfigurations

Operazioni di script

- CreateScript (diverso dall'API con lo stesso nome inAWS Glue)

### Accesso a AWS Glue Studio APIs

Per accedere a AWS Glue Studio, aggiungi `glue:UseGlueStudio` nell'elenco delle policy delle operazioni nelle autorizzazioni IAM.

Nell'esempio seguente, `glue:UseGlueStudio` è inclusa nella politica d'azione, ma non è identificata individualmente. AWS Glue Studio APIs Questo perché quando lo includi`glue:UseGlueStudio`, ti viene automaticamente concesso l'accesso all'area interna APIs senza dover specificare l'individuo AWS Glue Studio APIs nelle autorizzazioni IAM.

Nell'esempio, le politiche d'azione aggiuntive elencate (ad esempio,`glue:SearchTables`) non lo sono AWS Glue Studio APIs, quindi dovranno essere incluse nelle autorizzazioni IAM come richiesto. Potresti inoltre includere operazioni Amazon S3 Proxy per specificare il livello di accesso Amazon S3 da concedere. La policy di esempio riportata di seguito fornisce l'accesso a openAWS Glue Studio, la creazione di un job visivo e, save/run se il ruolo IAM selezionato dispone di un accesso sufficiente.

### Autorizzazioni per notebook e anteprima dati

Le anteprime dei dati e i notebook consentono di visualizzare un campione dei dati in qualsiasi fase del processo (lettura, trasformazione, scrittura), senza doverlo eseguire. È necessario specificare un ruolo AWS Identity and Access Management (IAM) AWS Glue Studio da utilizzare per l'accesso ai dati. I ruoli IAM sono destinati ad essere prevedibili e non hanno credenziali standard a lungo termine come una password o chiavi d'accesso associate ad esso. Invece, quando AWS Glue Studio assume il ruolo, IAM fornisce credenziali di sicurezza temporanee.

Per garantire che le anteprime dei dati e i comandi del notebook funzionino correttamente, usa un ruolo con un nome che inizia con la stringa `AWSGlueServiceRole`. Se decidi di usare un altro nome per il ruolo, dovrai aggiungere l'autorizzazione `iam:passrole` e configurare una policy per il ruolo in IAM. Per ulteriori informazioni, consulta Crea una policy IAM per i ruoli non denominati "AWSGlueServiceRole*».

> **Warning**
>
> Se un ruolo concede l'autorizzazione `iam:passrole` per un notebook e tu implementi il concatenamento dei ruoli, un utente potrebbe ottenere involontariamente l'accesso al notebook. Al momento non è implementato alcun controllo che permetta di monitorare a quali utenti sia stato concesso l'accesso al notebook.

Se desideri negare a un'identità IAM la possibilità di creare sessioni di anteprima dei dati, consulta l'esempio di the section called "Negare a un'identità la possibilità di creare sessioni di anteprima dei dati" seguente.

Amazon CloudWatch autorizzazioni

Puoi monitorare i tuoi AWS Glue Studio lavori utilizzando Amazon CloudWatch, che raccoglie ed elabora i dati grezzi AWS Glue trasformandoli in metriche leggibili. near-real-time Per impostazione predefinita, i dati AWS Glue delle metriche vengono inviati automaticamente a. CloudWatch Per ulteriori informazioni, consulta What Is Amazon CloudWatch? nella Amazon CloudWatch User Guide e AWS GlueMetrics nella AWS Glue Developer Guide.

Per accedere alle CloudWatch dashboard, l'utente che accede AWS Glue Studio deve disporre di uno dei seguenti elementi:

- La policy `AdministratorAccess`
- La policy `CloudWatchFullAccess`
- Una policy personalizzata che includa una o più di queste autorizzazioni specifiche:
  - `cloudwatch:GetDashboard` e `cloudwatch:ListDashboards` per visualizzare i pannelli di controllo
  - `cloudwatch:PutDashboard` per creare o modificare i pannelli di controllo
  - `cloudwatch:DeleteDashboards` per eliminare i pannelli di controllo

Per ulteriori informazioni sulla modifica delle autorizzazioni per un utente IAM utilizzando le policy, consulta Modifica delle autorizzazioni per un utente IAM nella Guida per l'utente IAM.

## Esaminare le autorizzazioni IAM necessarie per i processi ETL

Quando si crea un processo utilizzando AWS Glue Studio, il processo assume le autorizzazioni del ruolo IAM specificate al momento della creazione. Questo ruolo IAM deve avere l'autorizzazione per estrarre dati dalla tua fonte di dati, scrivere dati sulla tua destinazione e accedere alle AWS Glue risorse.

Il nome del ruolo creato per il processo deve iniziare con la stringa `AWSGlueServiceRole` per poter essere usato correttamente da AWS Glue Studio. È ad esempio possibile denominare il proprio ruolo `AWSGlueServiceRole-FlightDataJob`.

### Autorizzazioni origine dati e destinazione dati

Un processo AWS Glue Studio deve avere accesso ad Amazon S3 per le origini, le destinazioni, gli script e le directory temporanee che usi nel processo. Puoi creare una policy per fornire un accesso granulare a risorse Amazon S3 specifiche.

- Le origini dati richiedono le autorizzazioni `s3:ListBucket` e `s3:GetObject`.
- Le destinazioni dati richiedono le autorizzazioni `s3:ListBucket`, `s3:PutObject` e `s3:DeleteObject`.

> **Note**
>
> La tua policy IAM deve tenere conto `s3:GetObject` dei bucket specifici utilizzati per ospitare le AWS Glue trasformazioni.
> I seguenti bucket sono di proprietà dell'account del AWS servizio e sono leggibili in tutto il mondo. Questi bucket fungono da archivio per il codice sorgente pertinente a un sottoinsieme di trasformazioni accessibile tramite l'editor visuale. AWS Glue Studio Le autorizzazioni sul bucket sono configurate per negare qualsiasi altra azione dell'API sul bucket. Chiunque può leggere gli script che forniamo per le trasformazioni, ma nessuno al di fuori del nostro team di assistenza può "inserirvi" nulla. Quando il AWS Glue processo viene eseguito, il file viene importato in locale in modo che venga scaricato nel contenitore locale. Dopodiché, non ci sono ulteriori comunicazioni con quell'account.

Regione: nome del bucket

- af-south-1: -762339736633- aws-glue-studio-transforms -1 prod-af-south
- ap-east-1: -125979764932- aws-glue-studio-transforms -1 prod-ap-east
- ap-northeast-2: -673535381443- -2 aws-glue-studio-transforms prod-ap-northeast
- ap-northeast-3: -149976050262- -3 aws-glue-studio-transforms prod-ap-northeast
- ap-south-1: -584702181950- aws-glue-studio-transforms -1 prod-ap-south
- ap-south-2: -380279651983- aws-glue-studio-transforms -2 prod-ap-south
- ap-southeast-1: -737106620487- -1 aws-glue-studio-transforms prod-ap-southeast
- ap-southeast-2: -234881715811- -2 aws-glue-studio-transforms prod-ap-southeast
- ap-southeast-3: -151265630221- -3 aws-glue-studio-transforms prod-ap-southeast

- ap-southeast-4: -052235663858- -4 aws-glue-studio-transforms prod-ap-southeast
- ca-central-1: -622716468547- -1 aws-glue-studio-transforms prod-ca-central
- ca-west-1: -915795495192- aws-glue-studio-transforms -1 prod-ca-west
- aws-glue-studio-transformseu-central-1: -560373232017- -1 prod-eu-central
- aws-glue-studio-transformseu-central-2: -907358657121- -2 prod-eu-central
- eu-north-1: -312557305497- -1 aws-glue-studio-transforms prod-eu-north
- eu-south-1: -939684186351- -1 aws-glue-studio-transforms prod-eu-south
- eu-south-2: -239737454084- -2 aws-glue-studio-transforms prod-eu-south
- eu-west-1: -244479516193- -1 aws-glue-studio-transforms prod-eu-west
- eu-west-2: -804222392271- -2 aws-glue-studio-transforms prod-eu-west
- eu-west-3: -371299348807- -3 aws-glue-studio-transforms prod-eu-west
- il-central-1: -806964611811- -1 aws-glue-studio-transforms prod-il-central
- me-central-1: -733304270342- -1 aws-glue-studio-transforms prod-me-central
- me-south-1: -112120182341- aws-glue-studio-transforms -1 prod-me-south
- sa-east-1: -881619130292- aws-glue-studio-transforms -1 prod-sa-east
- us-east-1: -510798373988- -1 aws-glue-studio-transforms prod-us-east
- us-east-2: -251189692203- -2 aws-glue-studio-transforms prod-us-east
- us-west-1: -593230150239- -1 aws-glue-studio-transforms prod-us-west
- us-west-2: -818035625594- -2 aws-glue-studio-transforms prod-us-west
- ap-northeast-1: -200493242866- -1 aws-glue-studio-transforms prod-ap-northeast
- cn-nord-1: aws-glue-studio-transforms -071033555442- -1 prod-cn-north
- cn-nord-ovest-1: aws-glue-studio-transforms -070947029561- -1 prod-cn-northwest
- us-gov-west-1: aws-glue-studio-transforms -227493901923- -1-2604 prod-us-gov-west
- eusc-de-east-1: aws-glue-studio-transforms -780995497573- prod-eusc-de-east -1-555

Se scegli Amazon Redshift come origine dati, puoi fornire un ruolo per le autorizzazioni del cluster. I lavori eseguiti su un Amazon Redshift cluster emettono comandi che accedono ad Amazon S3 per lo storage temporaneo utilizzando credenziali temporanee. Se il processo viene eseguito per più di un'ora, queste credenziali scadranno causando l'esito negativo del processo. Per evitare questo problema, puoi assegnare un ruolo al cluster Amazon Redshift stesso che concede le autorizzazioni

necessarie ai processi utilizzando le credenziali temporanee. Per ulteriori informazioni, consulta Spostamento di dati da e verso Amazon Redshift nella Guida per gli sviluppatori di AWS Glue .

Se il processo utilizza origini dati o destinazioni diverse da Amazon S3, devi allegare le autorizzazioni necessarie al ruolo IAM utilizzato dal processo per accedere a tali origini dati e destinazioni. Per ulteriori informazioni, consulta Impostazione dell'ambiente per accedere a archivio dati nella Guida per gli sviluppatori di AWS Glue .

Se si utilizzano connettori e connessioni per l'archivio dati, è necessario disporre di autorizzazioni aggiuntive, come descritto in the section called "Autorizzazioni richieste per l'utilizzo dei connettori".

Autorizzazioni necessarie per l'eliminazione dei processi

In AWS Glue Studio, è possibile selezionare nella console più processi da eliminare. Per eseguire questa azione, è necessario disporre delle autorizzazioni `glue:BatchDeleteJob`. Diversamente, la console AWS Glue richiede l'autorizzazione `glue:DeleteJob` per l'eliminazione dei processi.

AWS Key Management Service autorizzazioni

Se prevedi di accedere a sorgenti e destinazioni Amazon S3 che utilizzano la crittografia lato server con AWS Key Management Service (AWS KMS), allega una policy al AWS Glue Studio ruolo utilizzato dal job che consenta al job di decrittografare i dati. Il ruolo del processo necessita delle autorizzazioni `kms:ReEncrypt`, `kms:GenerateDataKey` e `kms:DescribeKey`. Inoltre, il ruolo professionale richiede l'`kms:Decrypt`autorizzazione per caricare o scaricare un oggetto Amazon S3 crittografato con una chiave master AWS KMS del cliente (CMK).

Sono previsti costi aggiuntivi per l'utilizzo. AWS KMS CMKs Per ulteriori informazioni, consulta AWS Key Management Service Concepts - Customer Master Keys (CMKs) and AWS Key Management Service Pricing nella AWS Key Management Service Developer Guide.

Autorizzazioni richieste per l'utilizzo dei connettori

Se usi una connessione e un connettore personalizzato AWS Glue per accedere a un archivio dati, il ruolo utilizzato per eseguire il processo ETL AWS Glue necessita di autorizzazioni aggiuntive allegate:

- La policy gestita da AWS `AmazonEC2ContainerRegistryReadOnly` per l'accesso ai connettori acquistati Marketplace AWS.
- Le autorizzazioni `glue:GetJob` e `glue:GetJobs`.

- Gestione dei segreti AWS autorizzazioni per l'accesso ai segreti utilizzati con le connessioni. Per le policy IAM di esempio, consulta Esempio: Autorizzazione per recuperare valori segreti.

Se il processo ETL AWS Glue viene eseguito all'interno di un VPC che esegue Amazon VPC, il VPC deve essere configurato come descritto in the section called "Configurazione di un VPC per il tuo processo ETL".

## Impostazione delle autorizzazioni IAM per AWS Glue Studio

È possibile creare i ruoli e assegnare policy agli utenti e ai ruoli del processo utilizzando l'utente amministratore AWS .

È possibile utilizzare la policy AWSGlueConsoleFullAccess AWS gestita per fornire le autorizzazioni necessarie per l'utilizzo della AWS Glue Studio console.

Per creare una policy personalizzata, segui la procedura descritta in Creare una policy IAM per il servizio AWS Glue nella Guida per gli sviluppatori di AWS Glue . Includi le autorizzazioni IAM descritte in precedenza in Esaminare le autorizzazioni IAM necessarie per l'utente AWS Glue Studio.

Argomenti

- Allega le policy all'utente AWS Glue Studio
- Crea una policy IAM per i ruoli non denominati "AWSGlueServiceRole*»

### Allega le policy all'utente AWS Glue Studio

Qualsiasi AWS utente che accede alla AWS Glue Studio console deve disporre delle autorizzazioni per accedere a risorse specifiche. Puoi fornire queste autorizzazioni usando l'assegnazione delle policy IAM all'utente.

Come allegare la policy gestita AWSGlueConsoleFullAccess all'utente

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, scegli Policy.
3. Nell'elenco delle politiche, seleziona la casella di controllo accanto a AWSGlueConsoleFullAccess. Puoi utilizzare il menu Filtro e la casella di ricerca per filtrare l'elenco di policy.

4. Scegli Operazioni di policy, quindi Collega.
5. Sceglie l'utente a cui collegare la policy. Puoi usare il menu Filtro e la casella di ricerca per filtrare l'elenco delle entità principali. Dopo aver scelto l'utente a cui collegare la policy, scegli Attach policy (Collega policy).
6. Ripeti i passaggi precedenti per allegare ulteriori policy all'utente, in base alle esigenze.

Crea una policy IAM per i ruoli non denominati "AWSGlueServiceRole*»

Come configurare una policy IAM per i ruoli utilizzati da AWS Glue Studio

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Aggiunta di una nuova policy IAM. È possibile aggiungere a una policy esistente o creare una nuova policy IAM inline. Per creare una policy IAM

    1. Seleziona Policy, quindi scegli Create Policy (Crea policy). Se viene visualizzato il pulsante Get Started (Inizia), sceglilo, quindi scegli Create Policy (Crea policy)
    2. Accanto a Create Your Own Policy (Crea la tua policy) scegli Select (Seleziona).
    3. In Policy Name (Nome policy) digita un nome facile da ricordare. Facoltativamente, digita un testo descrittivo in Description (Descrizione).
    4. In Policy Document (Documento policy) digita un'istruzione di policy nel formato seguente, quindi scegli Create Policy (Crea policy):
3. Copia e incolla i seguenti blocchi nella policy sotto l'array «Statement», sostituendoli *my-interactive-session-role-prefix* con il prefisso per tutti i ruoli comuni per cui associarli alle autorizzazioni. AWS Glue

```
{
    "Action": [
        "iam:PassRole"
    ],
    "Effect": "Allow",
    "Resource": "arn:aws:iam::*:role/my-interactive-session-role-prefix*",
    "Condition": {
        "StringLike": {
            "iam:PassedToService": [
                "glue.amazonaws.com "
            ]
        }
```

```
    }
}
```

Ecco l'esempio completo degli array Version (versione) e Statement (istruzione) inclusi nella policy

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Action": [
        "iam:PassRole"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:iam::*:role/my-interactive-session-role-prefix*",
      "Condition": {
        "StringLike": {
          "iam:PassedToService": [
            "glue.amazonaws.com "
          ]
        }
      }
    }
  ]
}
```

4. Per abilitare la policy per un utente, scegli Users (Utenti).
5. Seleziona l'utente a cui desideri collegare la policy.

## Configurazione di un VPC per il tuo processo ETL

Puoi utilizzare Amazon Virtual Private Cloud (Amazon VPC) per definire una rete virtuale nella tua area logicamente isolata all'interno del Cloud AWS, noto come cloud privato virtuale (VPC). È possibile avviare le risorse AWS , ad esempio le istanze, nel VPC. Il VPC è molto simile a una rete tradizionale gestibile nel data center locale, ma con i vantaggi legati all'utilizzo dell'infrastruttura scalabile di AWS. È·possibile configurare il VPC, selezionare l'intervallo di indirizzi IP, creare sottoreti e configurare tabelle di routing, gateway di rete e impostazioni di sicurezza. È possibile connettere le istanze nel VPC a Internet. Puoi connettere il tuo VPC al tuo data center aziendale, rendendolo Cloud

AWS un'estensione del tuo data center. Per proteggere le risorse in ciascuna sottorete, è possibile usare diversi livelli di sicurezza, compresi gruppi di sicurezza e liste di controllo accessi alla rete. Per ulteriori informazioni, consulta la Guida per l'utente di Amazon VPC.

È possibile configurare i processi ETL AWS Glue da eseguire all'interno di un VPC quando si utilizzano connettori. È necessario configurare il VPC per quanto segue, in base alle esigenze:

- Accesso alla rete pubblica per archivi di dati non presenti AWS. Tutti gli archivi dati JDBC ai quali il processo accede devono essere disponibili dalla sottorete VPC.
- Se il processo deve accedere sia alle risorse VPC che alla rete Internet pubblica, il VPC deve disporre di un gateway NAT (Network Address Translation) al suo interno.

  Per ulteriori informazioni, consulta Impostazione dell'ambiente per accedere agli archivi dati nella Guida per gli sviluppatori di AWS Glue .

## Nozioni di base sui notebook in AWS Glue Studio

All'avvio di un notebook tramite AWS Glue Studio, tutti i passaggi di configurazione vengono eseguiti, in modo che tu possa esplorare i dati e iniziare a sviluppare lo script del processo dopo pochi secondi.

Nelle seguenti sezioni viene descritto come creare un ruolo e concedere le autorizzazioni appropriate per utilizzare i notebook in AWS Glue Studio per i processi ETL.

Per ulteriori informazioni sulle azioni definite da AWS Glue, vedere Actions defined by AWS Glue.

Argomenti

- Concessione di autorizzazioni per il ruolo IAM

### Concessione di autorizzazioni per il ruolo IAM

Configurare AWS Glue Studio è un prerequisito per l'utilizzo di notebook.

Per utilizzare i notebook in AWS Glue, il tuo ruolo richiede quanto segue:

- Una relazione di fiducia con AWS Glue per l'operazione `sts:AssumeRole` e, se vuoi, taggare `sts:TagSession`.
- Una policy IAM contenente tutte le autorizzazioni per notebook, AWS Glue e sessioni interattive.

- Una policy IAM per un ruolo pass poiché il ruolo deve essere in grado di passare da notebook alle sessioni interattive.

Ad esempio, quando si crea un nuovo ruolo, è possibile aggiungere una politica AWS gestita standard simile `AWSGlueConsoleFullAccessRole` al ruolo, quindi aggiungere una nuova politica per le operazioni del notebook e un'altra per la PassRole politica IAM.

Azioni necessarie per una relazione di fiducia con AWS Glue

Quando si avvia una sessione di notebook, è necessario aggiungere `sts:AssumeRole` alla relazione di fiducia del ruolo che viene trasmesso a notebook. Se la tua sessione include tag, devi anche passare l'operazione `sts:TagSession`. Senza queste azioni, la sessione di notebook non può essere avviata.

Esempio:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "glue.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Policy contenenti autorizzazioni IAM per notebook

La seguente policy di esempio descrive le autorizzazioni AWS IAM richieste per i notebook. Se stai creando un nuovo ruolo, crea una policy che contenga quanto segue:

JSON

```
{
```

```
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:StartNotebook",
        "glue:TerminateNotebook",
        "glue:GlueNotebookRefreshCredentials",
        "glue:DeregisterDataPreview",
        "glue:GetNotebookInstanceStatus",
        "glue:GlueNotebookAuthorize"
      ],
      "Resource": "*"
    }
  ]
}
```

È possibile utilizzare le seguenti policy IAM per consentire l'accesso a risorse specifiche:

- AwsGlueSessionUserRestrictedNotebookServiceRole: fornisce l'accesso completo a tutte le AWS Glue risorse ad eccezione delle sessioni. Permette agli utenti di creare e utilizzare solo le sessioni notebook associate all'utente. Questa politica include anche altre autorizzazioni necessarie AWS Glue per gestire AWS Glue le risorse in altri AWS servizi.
- AwsGlueSessionUserRestrictedNotebookPolicy: Fornisce le autorizzazioni che consentono agli utenti di creare e utilizzare solo le sessioni di notebook associate all'utente. Questa policy include anche le autorizzazioni per consentire esplicitamente agli utenti di passare un ruolo di sessione AWS Glue limitato.

Policy IAM per trasmettere un ruolo

Quando crei un notebook con un ruolo, tale ruolo viene passato alle sessioni interattive in modo che lo stesso ruolo possa essere utilizzato in entrambe le posizioni. Come tale, il permesso `iam:PassRole` deve essere parte della policy del ruolo.

Crea una nuova policy per il tuo ruolo utilizzando l'esempio seguente. Sostituisci il numero di account con il tuo e il nome del ruolo.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::111122223333:role/<role_name>"
    }
  ]
}
```

# Configurazione dei profili AWS Glue di utilizzo

Uno dei principali vantaggi dell'utilizzo di una piattaforma cloud è la sua flessibilità. Tuttavia, questa facilità di creazione di risorse di calcolo comporta il rischio di far salire vertiginosamente i costi del cloud quando viene lasciato incustodito e senza protezioni. Di conseguenza, gli amministratori devono bilanciare la necessità di evitare costi elevati di infrastruttura e allo stesso tempo consentire agli utenti di lavorare senza inutili attriti.

Con i profili di AWS Glue utilizzo, gli amministratori possono creare profili diversi per varie classi di utenti all'interno dell'account, come sviluppatori, tester e team di prodotto. Ogni profilo è un insieme unico di parametri che possono essere assegnati a diversi tipi di utenti. Ad esempio, gli sviluppatori potrebbero aver bisogno di più lavoratori e possono avere un numero massimo di lavoratori più elevato, mentre i team di prodotto potrebbero aver bisogno di meno lavoratori e un valore di timeout o di inattività inferiore.

Esempio di comportamento di processi ed esecuzione dei processi

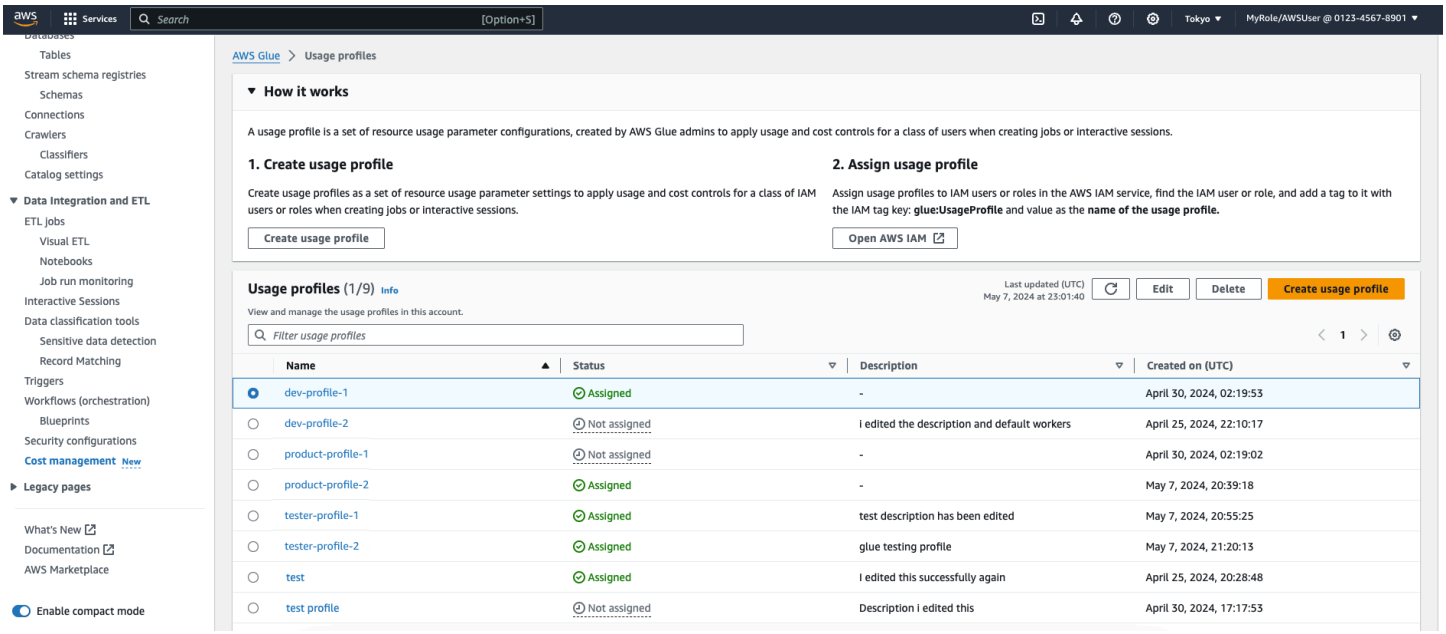
Supponiamo che un processo venga creato dall'utente A con il profilo A. Il processo viene salvato con determinati valori di parametro. L'utente B con profilo B proverà a eseguire il processo.

Quando l'utente A ha creato il processo, se non ha impostato un numero specifico di worker, è stato applicato il valore predefinito impostato nel profilo dell'utente A che è stato salvato con le definizioni del processo.

Quando l'utente B esegue il processo, questo viene eseguito con i valori salvati per esso. Se il profilo dell'utente B è più restrittivo e non può essere eseguito con un numero così elevato di worker, l'esecuzione del processo avrà esito negativo.

Profilo di utilizzo come risorsa

Un profilo di AWS Glue utilizzo è una risorsa identificata da un Amazon Resource Name (ARN). Si applicano tutti i controlli IAM (Identity and Access Management) predefiniti, inclusa l'autorizzazione basata sulle azioni e sulle risorse. Gli amministratori devono aggiornare la policy IAM degli utenti che creano AWS Glue risorse, concedendo loro l'accesso per utilizzare i profili.




Argomenti

- Creazione e gestione dei profili di utilizzo
- Profili di utilizzo e processi

# Creazione e gestione dei profili di utilizzo

## Creazione di un profilo di AWS Glue utilizzo

Gli amministratori devono creare profili di utilizzo e quindi assegnarli ai vari utenti. Durante la creazione di un profilo di utilizzo, si specificano valori predefiniti, nonché una gamma di valori consentiti per vari parametri di processo e sessione. È necessario configurare almeno un parametro per i processi o le sessioni interattive. È possibile personalizzare il valore predefinito da utilizzare quando non viene fornito un valore di parametro per il lavoro, and/or impostare un limite di intervallo

o un insieme di valori consentiti per la convalida se un utente fornisce un valore di parametro quando utilizza questo profilo.

I valori predefiniti sono una best practice impostata dall'amministratore per assistere gli autori dei processi. Quando un utente crea un nuovo processo e non imposta un valore di timeout, viene applicato il timeout predefinito del profilo di utilizzo. Se l'autore non dispone di un profilo, verranno applicate le impostazioni predefinite del AWS Glue servizio e verranno salvate nella definizione del lavoro. In fase di esecuzione, AWS Glue applica i limiti impostati nel profilo (min, max, lavoratori consentiti).

Una volta configurato un parametro, tutti gli altri parametri sono opzionali. I parametri che possono essere personalizzati per processi o sessioni interattive sono:

- Numero di lavoratori: limita il numero di lavoratori per evitare un uso eccessivo delle risorse di elaborazione. È possibile impostare un valore predefinito, minimo e massimo. Il valore minimo è 1.
- Tipo di lavoratore: limita i tipi di lavoratori pertinenti per i tuoi carichi di lavoro. È possibile impostare un tipo predefinito e consentire i tipi di lavoratori per un profilo utente.
- Timeout: definisce il tempo massimo in cui un processo o una sessione interattiva può essere eseguito/a e consumare risorse prima che venga terminato/a. Impostare i valori di timeout per evitare processi di lunga durata.

  È possibile impostare un valore predefinito, minimo e massimo in minuti. Il minimo è di 1 (minuto). Sebbene il AWS Glue timeout predefinito sia 2880 minuti, è possibile impostare qualsiasi valore predefinito nel profilo di utilizzo.

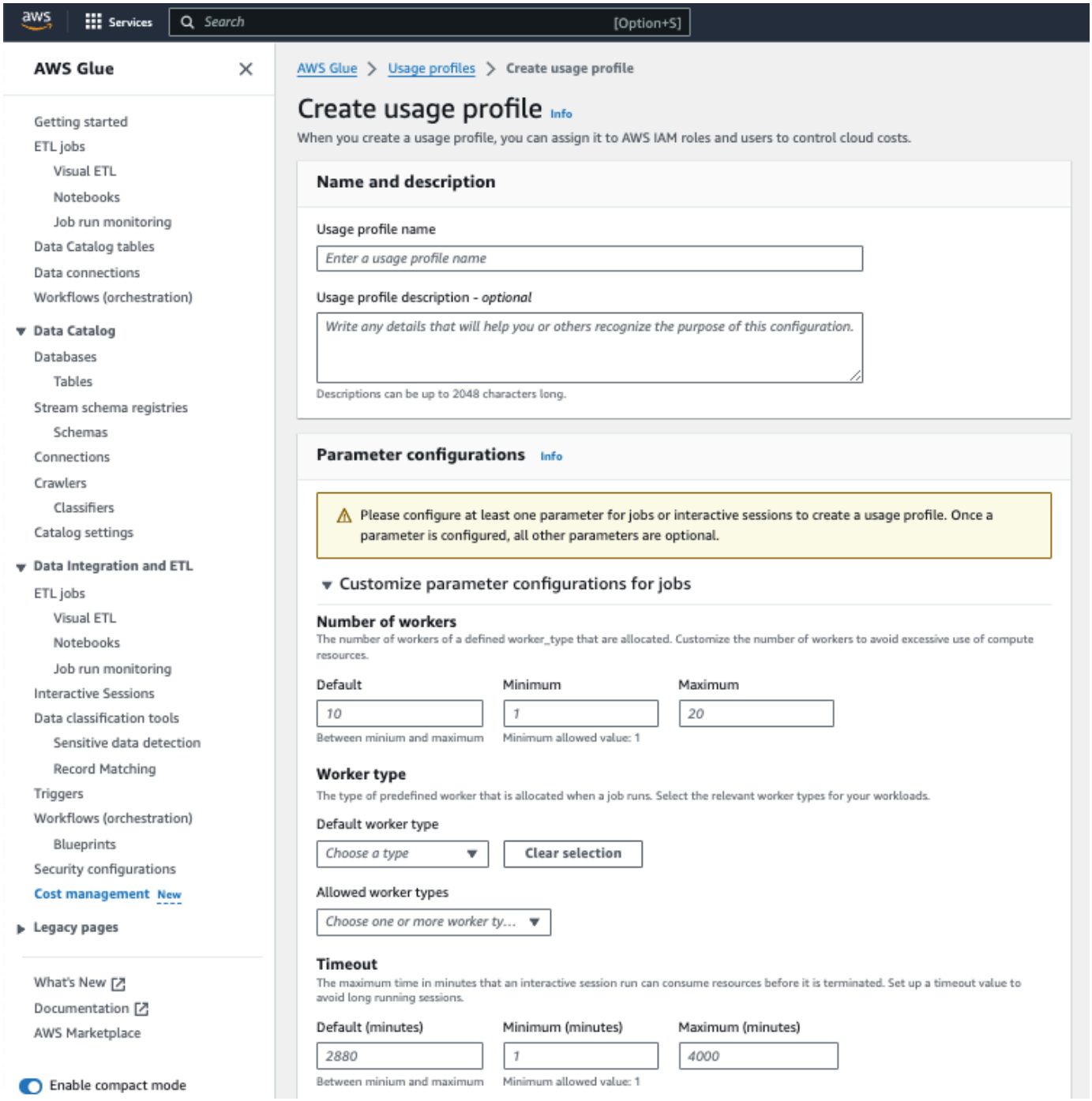
  È consigliabile impostare un valore per "default". Questo valore verrà utilizzato per la creazione del processo o della sessione se l'utente non ha impostato alcun valore.
- Timeout di inattività: definisce il numero di minuti in cui una sessione interattiva è inattiva prima del timeout dopo l'esecuzione di una cella. Definire il timeout di inattività per le sessioni interattive da terminare dopo il completamento del processo. L'intervallo di timeout di inattività deve rientrare nel limite del timeout.

  È possibile impostare un valore predefinito, minimo e massimo in minuti. Il minimo è di 1 (minuto). Sebbene il AWS Glue timeout predefinito sia 2880 minuti, è possibile impostare qualsiasi valore predefinito nel profilo di utilizzo.

  È consigliabile impostare un valore per "default". Questo valore verrà utilizzato per la creazione della sessione se l'utente non ha impostato alcun valore.

Per creare un profilo di AWS Glue utilizzo come amministratore (console)

1. Nel menu di navigazione a sinistra, scegliere Gestione dei costi.
2. Scegliere Crea profilo.
3. Immettere il Nome del profilo di utilizzo per il profilo di utilizzo.
4. Immettere una descrizione opzionale che aiuterà gli altri utenti a riconoscere lo scopo del profilo di utilizzo.
5. Definire almeno un parametro nel profilo. Qualsiasi campo del modulo è un parametro. Ad esempio, il timeout minimo di inattività della sessione.
6. Definire eventuali tag opzionali che si applicano al profilo di utilizzo.
7. Scegli Save (Salva).




Per creare un profilo di utilizzo (AWS CLI)

1. Inserire il seguente comando.

```
aws glue create-usage-profile --name profile-name --configuration file://
config.json --tags list-of-tags
```

Dove config.json può definire i valori dei parametri per le sessioni interattive (`SessionConfiguration`) e i processi (`JobConfiguration`):

```
//config.json (There is a separate blob for session/job configuration
{
    "SessionConfiguration": {
        "timeout": {
            "DefaultValue": "2880",
            "MinValue": "100",
            "MaxValue": "4000"
        },
        "idleTimeout": {
            "DefaultValue": "30",
            "MinValue": "10",
            "MaxValue": "4000"
        },
        "workerType": {
            "DefaultValue": "G.2X",
            "AllowedValues": [
                "G.1X",
                "G.2X",
                "G.4X",
                "G.8X",
                "G.12X",
                "G.16X",
                "R.1X",
                "R.2X",
                "R.4X",
                "R.8X"
            ]
        },
        "numberOfWorkers": {
            "DefaultValue": "10",
            "MinValue": "1",
            "MaxValue": "10"
        }
    },
    "JobConfiguration": {
        "timeout": {
            "DefaultValue": "2880",
            "MinValue": "100",
            "MaxValue": "4000"
```

```
        },
        "workerType": {
            "DefaultValue": "G.2X",
            "AllowedValues": [
                "G.1X",
                "G.2X",
                "G.4X",
                "G.8X",
                "G.12X",
                "G.16X",
                "R.1X",
                "R.2X",
                "R.4X",
                "R.8X"
            ]
        },
        "numberOfWorkers": {
            "DefaultValue": "10",
            "MinValue": "1",
            "MaxValue": "10"
        }
    }
}
```

2. Inserire il seguente comando per vedere il profilo di utilizzo creato:

```
aws glue get-usage-profile --name profile-name
```

La risposta:

```
{
    "ProfileName": "foo",
    "Configuration": {
        "SessionConfiguration": {
            "numberOfWorkers": {
                "DefaultValue": "10",
                "MinValue": "1",
                "MaxValue": "10"
            },
            "workerType": {
                "DefaultValue": "G.2X",
                "AllowedValues": [
                    "G.1X",
```

```
                    "G.2X",
                    "G.4X",
                    "G.8X",
                    "G.12X",
                    "G.16X",
                    "R.1X",
                    "R.2X",
                    "R.4X",
                    "R.8X"
                ]
            },
            "timeout": {
                "DefaultValue": "2880",
                "MinValue": "100",
                "MaxValue": "4000"
            },
            "idleTimeout": {
                "DefaultValue": "30",
                "MinValue": "10",
                "MaxValue": "4000"
            }
        },
        "JobConfiguration": {
            "numberOfWorkers": {
                "DefaultValue": "10",
                "MinValue": "1",
                "MaxValue": "10"
            },
            "workerType": {
                "DefaultValue": "G.2X",
                "AllowedValues": [
                    "G.1X",
                    "G.2X",
                    "G.4X",
                    "G.8X",
                    "G.12X",
                    "G.16X",
                    "R.1X",
                    "R.2X",
                    "R.4X",
                    "R.8X"
                ]
            },
            "timeout": {
```

```
                "DefaultValue": "2880",
                "MinValue": "100",
                "MaxValue": "4000"
            }
        }
    },
    "CreatedOn": "2024-01-19T23:15:24.542000+00:00"
}
```

Comandi CLI aggiuntivi utilizzati per gestire i profili di utilizzo:

- Come colla list-usage-profiles
- aws glue update-usage-profile --name --configuration *profile-name file://config.json*
- aws glue --name delete-usage-profile *profile-name*

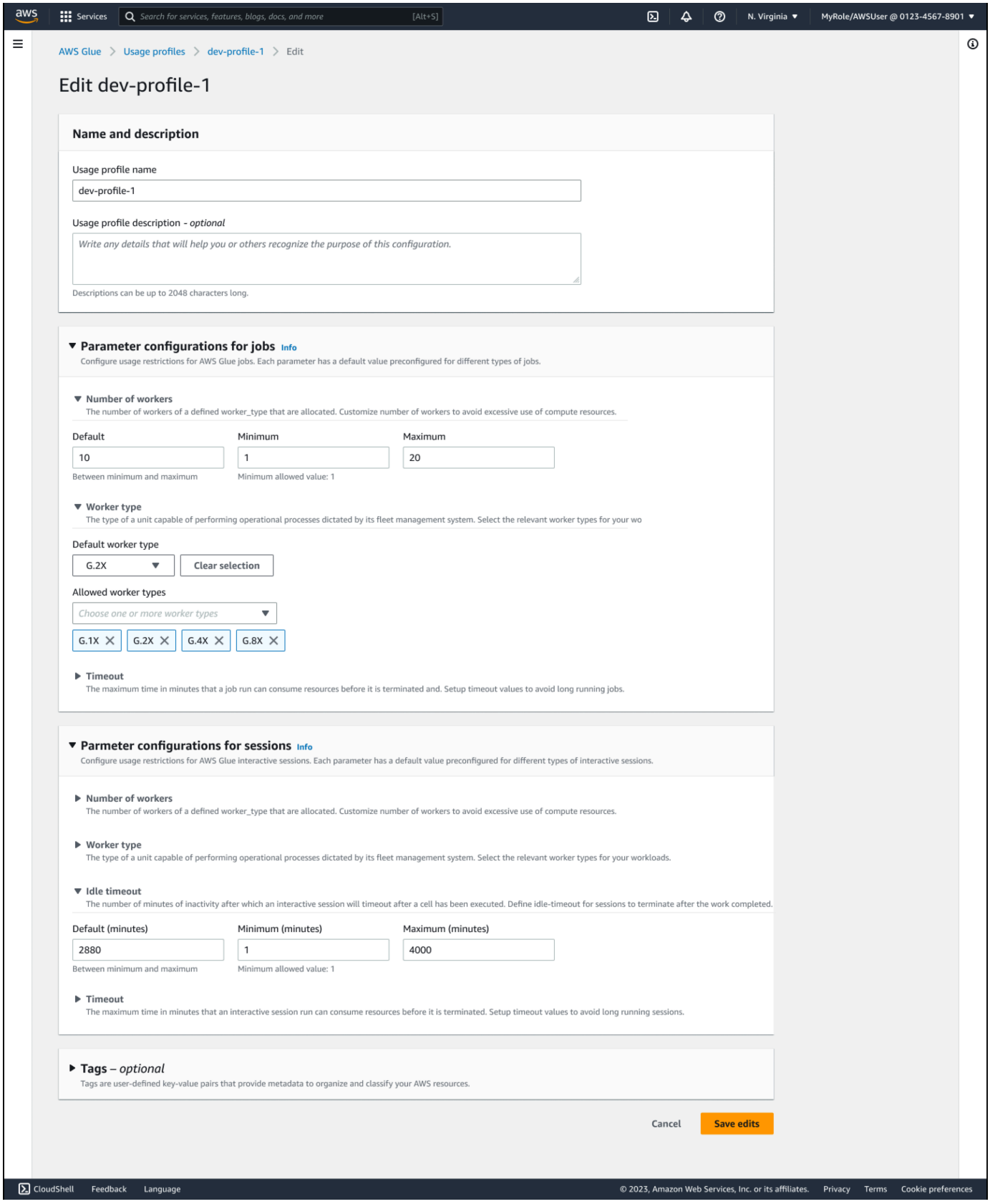
## Modifica di un profilo di utilizzo

Gli amministratori possono modificare i profili di utilizzo che hanno creato, per cambiare i valori di parametro del profilo per i processi e le sessioni interattive.

Per modificare un profilo di utilizzo:

Per modificare un profilo di AWS Glue utilizzo come amministratore (console)

1. Nel menu di navigazione a sinistra, scegliere Gestione dei costi.
2. Scegliere un profilo di utilizzo per il quale si dispone delle autorizzazioni necessarie per la modifica e scegliere Modifica.
3. Apportare le modifiche necessarie al profilo. Per impostazione predefinita, i parametri che hanno già dei valori vengono espansi.
4. Scegliere Salva modifiche.


Services
Search for services, features, blogs, docs, and more [Alt+S]
N. Virginia
MyRole/AWSUser @ 0123-4567-8901
AWS Glue > Usage profiles > dev-profile-1 > Edit
Edit dev-profile-1
Name and description
Usage profile name
dev-profile-1
Usage profile description - optional
Write any details that will help you or others recognize the purpose of this configuration.
Descriptions can be up to 2048 characters long.
Parameter configurations for jobs Info
Configure usage restrictions for AWS Glue jobs. Each parameter has a default value preconfigured for different types of jobs.
Number of workers
The number of workers of a defined worker_type that are allocated. Customize number of workers to avoid excessive use of compute resources.
Default
10
Between minimum and maximum
Minimum
1
Minimum allowed value: 1
Maximum
20
Worker type
The type of a unit capable of performing operational processes dictated by its fleet management system. Select the relevant worker types for your wo
Default worker type
G.2X
Clear selection
Allowed worker types
Choose one or more worker types
G.1X
G.2X
G.4X
G.8X
Timeout
The maximum time in minutes that a job run can consume resources before it is terminated and. Setup timeout values to avoid long running jobs.
Parmeter configurations for sessions Info
Configure usage restrictions for AWS Glue interactive sessions. Each parameter has a default value preconfigured for different types of interactive sessions.
Number of workers
The number of workers of a defined worker_type that are allocated. Customize number of workers to avoid excessive use of compute resources.
Worker type
The type of a unit capable of performing operational processes dictated by its fleet management system. Select the relevant worker types for your workloads.
Idle timeout
The number of minutes of inactivity after which an interactive session will timeout after a cell has been executed. Define idle-timeout for sessions to terminate after the work completed.
Default (minutes)
2880
Between minimum and maximum
Minimum (minutes)
1
Minimum allowed value: 1
Maximum (minutes)
4000
Timeout
The maximum time in minutes that an interactive session run can consume resources before it is terminated. Setup timeout values to avoid long running sessions.
Tags – optional
Tags are user-defined key-value pairs that provide metadata to organize and classify your AWS resources.
Cancel
Save edits
CloudShell
Feedback
Language
© 2023, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

Per modificare un profilo di utilizzo (AWS CLI)

- Inserire il seguente comando. La stessa sintassi di file `--configuration` viene utilizzata come illustrato sopra nel comando "Crea".

  ```
  aws glue update-usage-profile --name profile-name --configuration file://
  config.json
  ```

  Dove config.json definisce i valori dei parametri per le sessioni interattive (`SessionConfiguration`) e i processi (`JobConfiguration`):

## Assegnazione di un profilo di utilizzo

La colonna Stato di utilizzo nella pagina Profili di utilizzo mostra se un profilo di utilizzo è assegnato agli utenti. Il passaggio del mouse sullo stato mostra le entità IAM assegnate.

L'amministratore può assegnare un profilo di AWS Glue utilizzo a users/roles chi crea AWS Glue le risorse. L'assegnazione di un profilo è una combinazione di due azioni:

- Aggiornando il user/role tag IAM con la `glue:UsageProfile` chiave, quindi
- Aggiornamento della policy IAM dell'utente/ruolo.

Per gli utenti che utilizzano AWS Glue Studio per creare jobs/interactive sessioni, l'amministratore assegna un tag ai seguenti ruoli:

- Per le restrizioni sui processi, l'amministratore tagga il ruolo della console in cui è stato effettuato l'accesso
- Per le restrizioni sulle sessioni interattive, l'amministratore tagga il ruolo assegnato dall'utente al momento della creazione del notebook

Di seguito è riportato un esempio di policy che l'amministratore deve aggiornare su IAM users/roles che crea AWS Glue risorse:

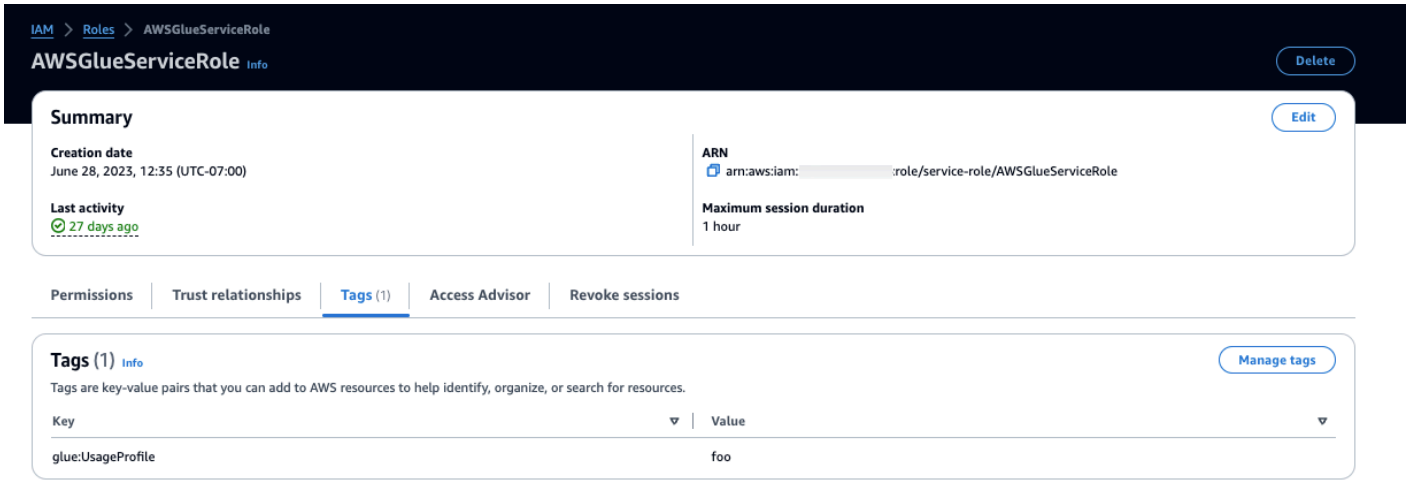
```
{
    "Effect": "Allow",
    "Action": [
        "glue:GetUsageProfile"
    ],
```

```
    "Resource": [
        "arn:aws:glue:us-east-1:123456789012:usageProfile/foo"
    ]
}
```

AWS Glue convalida le richieste di job, job run e session in base ai valori specificati nel profilo di AWS Glue utilizzo e solleva un'eccezione se la richiesta non è consentita. Per la modalità sincrona APIs, all'utente verrà generato un errore. Per i percorsi asincroni, viene creata un'esecuzione di processo non riuscita con il messaggio di errore che indica che il parametro di input non rientra nell'intervallo consentito per il profilo assegnato all'utente/ruolo.

Per assegnare un profilo di utilizzo a un utente/ruolo:

1. Aprire la console IAM (Identity and Access Management).
2. Nel pannello di navigazione, scegliere Ruoli o Utenti.
3. Scegliere un utente o un ruolo.
4. Selezionare la scheda Tag.
5. Scegliere Aggiungi nuovo tag.
6. Aggiungere un tag con la chiave di `glue:UsageProfile` e il valore del nome del tuo profilo di utilizzo.
7. Scegliere Salva modifiche.



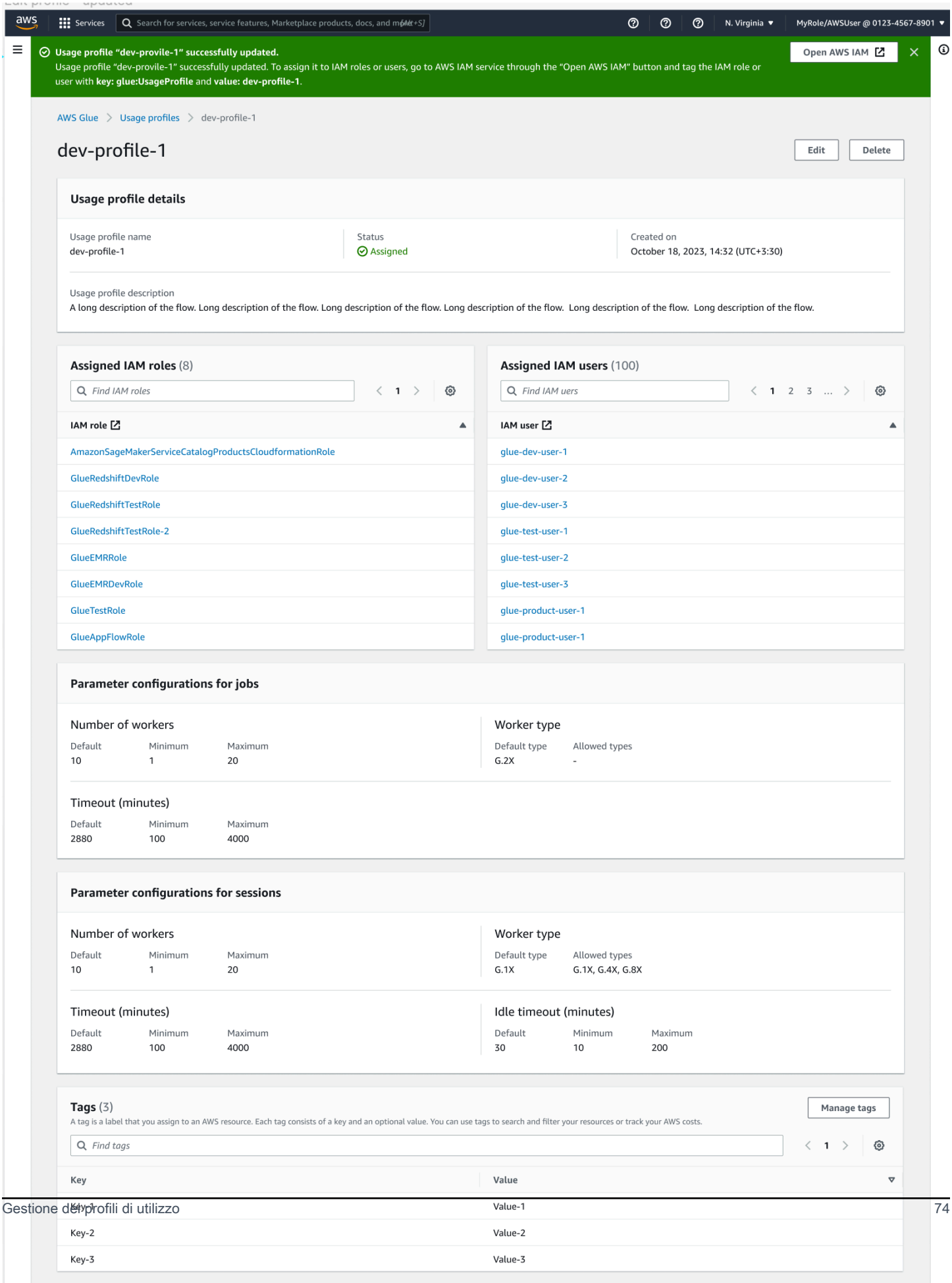
## Visualizzazione del profilo di utilizzo assegnato

Gli utenti possono visualizzare i profili di utilizzo assegnati e utilizzarli quando effettuano chiamate API per creare risorse di AWS Glue lavoro e sessione o per avviare un lavoro.

Le autorizzazioni dei profili sono fornite nelle policy IAM. Finché la politica del chiamante dispone dell'autorizzazione `glue:UsageProfile`, un utente può vedere il profilo. Altrimenti, si otterrà un errore di accesso negato.

Per visualizzare un profilo di utilizzo assegnato:

1. Nel menu di navigazione a sinistra, scegliere Gestione dei costi.
2. Scegliere un profilo di utilizzo per il quale si dispone delle autorizzazioni necessarie per la visualizzazione.


Services
Search for services, service features, Marketplace products, docs, and more [Alt+S]
N. Virginia
MyRole/AWSUser @ 0123-4567-8901
Usage profile "dev-provile-1" successfully updated.
Usage profile "dev-provile-1" successfully updated. To assign it to IAM roles or users, go to AWS IAM service through the "Open AWS IAM" button and tag the IAM role or user with key: glue:UsageProfile and value: dev-profile-1.
Open AWS IAM
AWS Glue > Usage profiles > dev-profile-1
dev-profile-1
Edit
Delete
Usage profile details
Usage profile name
dev-profile-1
Status
Assigned
Created on
October 18, 2023, 14:32 (UTC+3:30)
Usage profile description
A long description of the flow. Long description of the flow. Long description of the flow. Long description of the flow. Long description of the flow. Long description of the flow.
Assigned IAM roles (8)
Find IAM roles
IAM role
AmazonSageMakerServiceCatalogProductsCloudformationRole
GlueRedshiftDevRole
GlueRedshiftTestRole
GlueRedshiftTestRole-2
GlueEMRRole
GlueEMRDevRole
GlueTestRole
GlueAppFlowRole
Assigned IAM users (100)
Find IAM uers
IAM user
glue-dev-user-1
glue-dev-user-2
glue-dev-user-3
glue-test-user-1
glue-test-user-2
glue-test-user-3
glue-product-user-1
glue-product-user-1
Parameter configurations for jobs
Number of workers
Default 10
Minimum 1
Maximum 20
Worker type
Default type G.2X
Allowed types -
Timeout (minutes)
Default 2880
Minimum 100
Maximum 4000
Parameter configurations for sessions
Number of workers
Default 10
Minimum 1
Maximum 20
Worker type
Default type G.1X
Allowed types G.1X, G.4X, G.8X
Timeout (minutes)
Default 2880
Minimum 100
Maximum 4000
Idle timeout (minutes)
Default 30
Minimum 10
Maximum 200
Tags (3)
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.
Manage tags
Find tags
Key
Value
Key-1
Value-1
Key-2
Value-2
Key-3
Value-3

# Profili di utilizzo e processi

## Creazione di processi con profili di utilizzo

Durante la creazione di processi, verranno applicati i limiti e le impostazioni predefinite impostati nel profilo di utilizzo. Il profilo verrà assegnato al processo al momento del salvataggio.

## Esecuzione di processi con profili di utilizzo

Quando avvii un'operazione, AWS Glue applica i limiti impostati nel profilo del chiamante. Se non vi è un chiamante diretto, Glue applicherà i limiti del profilo assegnato al processo da parte del suo autore.

> **Note**
>
> Quando un lavoro viene eseguito in base a una pianificazione (tramite AWS Glue flussi di lavoro o AWS Glue trigger), verrà applicato il profilo assegnato al lavoro assegnato dall'autore.
> Quando un processo viene eseguito da un servizio esterno (Step Functions, MWAA) o da un'API `StartJobRun`, verrà applicato il limite del profilo del chiamante.

Per i AWS Glue flussi di lavoro o i AWS Glue trigger: i lavori preesistenti devono essere aggiornati per salvare il nuovo nome del profilo in modo che i limiti del profilo (minimo, massimo e lavoratori consentiti) vengano applicati in fase di esecuzione per le esecuzioni pianificate.

## Visualizzazione di un profilo di utilizzo assegnato ai processi

Per visualizzare il profilo assegnato ai tuoi lavori (che verrà utilizzato in fase di esecuzione con AWS Glue flussi di lavoro o AWS Glue trigger pianificati), puoi consultare la scheda Dettagli del lavoro. Si può anche analizzare il profilo utilizzato nelle esecuzioni precedenti nella scheda dei dettagli delle esecuzioni del processo.

## Aggiornamento o eliminazione di un profilo di utilizzo associato a un processo

Il profilo assegnato a un processo viene modificato al momento dell'aggiornamento. Se all'autore non viene assegnato un profilo di utilizzo, qualsiasi profilo precedentemente associato al processo verrà rimosso da esso.

# Nozioni di base su AWS Glue Data Catalog

AWS Glue Data Catalog È il tuo archivio di metadati tecnici persistente. È un servizio gestito che puoi utilizzare per archiviare, annotare e condividere i metadati nel Cloud. AWS Per ulteriori informazioni, consulta AWS Glue Data Catalog.

La console AWS Glue e alcune interfacce utente sono state recentemente aggiornate.

## Panoramica di

Puoi utilizzare questo tutorial per creare il tuo primo catalogo dati AWS Glue, che utilizza un bucket Amazon S3 come origine dati.

In questo tutorial, verranno eseguite le operazioni seguenti tramite la console AWS Glue:

1. Creare un database
2. Creare una tabella
3. Utilizza un bucket Amazon S3 come origine dei dati

Dopo aver completato questi passaggi, avrai utilizzato correttamente un bucket Amazon S3 come origine dati per popolare il catalogo dati AWS Glue.

## Fase 1: crea un database

Per iniziare, accedi Console di gestione AWS e apri la AWS Glue console.

Per creare un database utilizzando la console AWS Glue:

1. Nella console AWS Glue, scegli Databases (Database) dal menu a sinistra Data catalog (Catalogo dati).
2. Scegli Aggiungi database.
3. Nella pagina Crea database, immetti un nome per il database. Nella sezione Posizione - facoltativa, imposta la posizione dell'URI che i client di Catalogo dati devono utilizzare. Se la ignori, puoi continuare con la creazione del database.
4. (Facoltativo). Inserisci una descrizione per il database.
5. Scegliere Crea database.

Congratulazioni, hai appena configurato il tuo primo database usando la console AWS Glue. Il nuovo database verrà visualizzato nell'elenco dei database disponibili. È possibile modificare il database scegliendo il nome del database dal pannello di controllo Databases (Database).

Fasi successive

Altri modi per creare un database:

Hai appena creato un database utilizzando la console AWS Glue, ma esistono altri modi per creare un database:

- È possibile utilizzare i crawler per creare automaticamente un database e delle tabelle. Per configurare un database utilizzando i crawler, consulta Uso di crawler nella console AWS Glue.
- È possibile utilizzare CloudFormation i modelli. Vedi Creazione di AWS Glue risorse mediante AWS Glue Data Catalog modelli.
- Puoi creare un database anche utilizzando il plugin Operazioni API del database AWS Glue.

  Per creare un database utilizzando il plugin dell'operazione `create`, strutturare la richiesta includendo i parametri `DatabaseInput` (obbligatori).

  Ad esempio:

  Di seguito sono riportati esempi di come è possibile utilizzare la CLI, Boto3 o DDL per definire una tabella basata sullo stesso file flights_data.csv dal bucket S3 utilizzato nel tutorial.

  CLI

  ```
  aws glue create-database --database-input "{\"Name\":\"clidb\"}"
  ```

  Boto3

  ```
  glueClient = boto3.client('glue')

  response = glueClient.create_database(
      DatabaseInput={
          'Name': 'boto3db'
      }
  )
  ```

Per ulteriori informazioni sui tipi di dati, sulla struttura e sulle operazioni API del database, consulta API database.

Fasi successive

Nella sezione successiva, creerai una tabella e la aggiungerai al database.

Puoi anche esplorare le impostazioni e le autorizzazioni per il catalogo dati. Consulta Uso delle impostazioni dei cataloghi dati nella console AWS Glue.

## Passaggio 2. Creare una tabella

In questa fase, utilizzerai la console AWS Glue per creare una tabella.

1. Nella console AWS Glue, scegli Tables (Tabelle) dal menu a sinistra.
2. Scegli Aggiungi tabella.
3. Imposta le proprietà della tabella inserendo un nome per la tabella in Table details (Dettagli della tabella).
4. Nella sezione Databases (Database), scegli il database creato nella fase 1 dal menu a discesa.
5. Nella sezione Add a data store (Aggiungi un datastore), per impostazione predefinita il tipo di origine sarà S3.
6. Per Data is located in (I dati si trovano in), scegli Specified path in another account (Percorso specificato in un altro account).
7. Copia e incolla il percorso per il campo di input Include path (Percorso di inclusione):

   ```
   s3://crawler-public-us-west-2/flight/2016/csv/
   ```
8. Nella sezione Data format (Formato dei dati), per Classification (Classificazione), scegli CSV e per Delimiter (Delimitatore), scegli comma (,) (virgola [,]). Scegli Next (Successivo).
9. Ti viene chiesto di definire uno schema. Uno schema definisce la struttura e il formato di un registro di dati. Scegli Add column (Aggiungi colonna). (Per ulteriori informazioni, consulta Registri degli schemi).
10. Specifica le proprietà della colonna:

    a. Inserisci un nome per la colonna.
    b. Per Column type (Tipo di colonna), 'string' è già selezionata per impostazione predefinita.
    c. Per Column number (Numero di colonna), '1' è già selezionato per impostazione predefinita.
    d. Scegliere Aggiungi.

11. Ti viene richiesto di aggiungere indici di partizione. Questa operazione è facoltativa. Per saltare questo passaggio, scegli Next (Successivo).
12. Viene visualizzato un riepilogo delle proprietà della tabella. Se tutto appare come previsto, scegliere Crea. In caso contrario, scegli Back (Indietro) e modifica in base alle necessità.

Congratulazioni, hai creato manualmente una tabella in modo corretto e l'hai associata a un database. La tabella appena creata apparirà nel pannello di controllo Tables (Tabelle). Dal pannello di controllo, puoi modificare e gestire le tabelle.

Per ulteriori informazioni, consulta Utilizzo di tabelle nella console AWS Glue.

## Fasi successive

Fasi successive

Ora che il catalogo dati è popolato, è possibile iniziare a creare i processi in AWS Glue. Consultare Creazione di processi ETL visivi con AWS Glue Studio.

Oltre a utilizzare la console, esistono altri modi per definire le tabelle nel catalogo dati, tra cui:

- Creare ed eseguire un crawler
- Aggiungere classificatori a un crawler in AWS Glue
- Usare la Tabella API AWS Glue
- Usare il modello AWS Glue Data Catalog
- Eseguire la migrazione di un metastore Apache Hive
- Utilizzando Boto3 o AWS CLI il linguaggio di definizione dei dati (DDL)

  Di seguito sono riportati esempi di come è possibile utilizzare la CLI, Boto3 o DDL per definire una tabella basata sullo stesso file flights_data.csv dal bucket S3 utilizzato nel tutorial.

  Consulta la documentazione su come strutturare un comando. AWS CLI L'esempio della CLI contiene la sintassi JSON per il valore "aws glue create-table --table-input".

  CLI

```
{
        "Name": "flights_data_cli",
        "StorageDescriptor": {
            "Columns": [
```

```
                {
                    "Name": "year",
                    "Type": "bigint"
                },
                {
                    "Name": "quarter",
                    "Type": "bigint"
                }
            ],
            "Location": "s3://crawler-public-us-west-2/flight/2016/csv",
            "InputFormat": "org.apache.hadoop.mapred.TextInputFormat",
            "OutputFormat":
"org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat",
            "Compressed": false,
            "NumberOfBuckets": -1,
            "SerdeInfo": {
                "SerializationLibrary":
"org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe",
                "Parameters": {
                    "field.delim": ",",
                    "serialization.format": ","
                }
            }
        },
        "PartitionKeys": [
            {
                "Name": "mon",
                "Type": "string"
            }
        ],
        "TableType": "EXTERNAL_TABLE",
        "Parameters": {
            "EXTERNAL": "TRUE",
            "classification": "csv",
            "columnsOrdered": "true",
            "compressionType": "none",
            "delimiter": ",",
            "skip.header.line.count": "1",
            "typeOfData": "file"
        }
    }
```

Boto3

```
import boto3

glue_client = boto3.client("glue")

response = glue_client.create_table(
    DatabaseName='sampledb',
    TableInput={
        'Name': 'flights_data_manual',
    'StorageDescriptor': {
      'Columns': [{
        'Name': 'year',
        'Type': 'bigint'
      }, {
        'Name': 'quarter',
        'Type': 'bigint'
      }],
      'Location': 's3://crawler-public-us-west-2/flight/2016/csv',
      'InputFormat': 'org.apache.hadoop.mapred.TextInputFormat',
      'OutputFormat':
 'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat',
      'Compressed': False,
      'NumberOfBuckets': -1,
      'SerdeInfo': {
        'SerializationLibrary':
 'org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe',
        'Parameters': {
          'field.delim': ',',
          'serialization.format': ','
        }
      },
    },
    'PartitionKeys': [{
      'Name': 'mon',
      'Type': 'string'
    }],
    'TableType': 'EXTERNAL_TABLE',
    'Parameters': {
      'EXTERNAL': 'TRUE',
      'classification': 'csv',
      'columnsOrdered': 'true',
      'compressionType': 'none',
```

```
      'delimiter': ',',
      'skip.header.line.count': '1',
      'typeOfData': 'file'
    }
    }
)
```

DDL

```
CREATE EXTERNAL TABLE `sampledb`.`flights_data` (
  `year` bigint,
  `quarter` bigint)
PARTITIONED BY (
  `mon` string)
ROW FORMAT DELIMITED
  FIELDS TERMINATED BY ','
STORED AS INPUTFORMAT
  'org.apache.hadoop.mapred.TextInputFormat'
OUTPUTFORMAT
  'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
LOCATION
  's3://crawler-public-us-west-2/flight/2016/csv/'
TBLPROPERTIES (
  'classification'='csv',
  'columnsOrdered'='true',
  'compressionType'='none',
  'delimiter'=',',
  'skip.header.line.count'='1',
  'typeOfData'='file')
```

# Impostazione dell'accesso di rete agli archivi di dati

Per eseguire i processi ETL (Extract, Transform and Load, estrazione, trasformazione e caricamento), AWS Glue deve poter accedere ai datastore. Se un processo non deve essere necessariamente eseguito nella tua sottorete Virtual Private Cloud (VPC) (es. trasformazione di dati da Amazon S3 ad Amazon S3) non servono ulteriori configurazioni.

Se un processo deve essere eseguito nella tua sottorete VPC, (es. trasformazione di dati da un datastore JDBC a una sottorete privata), AWS Glue imposta le interfacce di rete elastiche che

consentono di connettere i processi dell'utente ad altre risorse all'interno del VPC in modo sicuro. A ogni interfaccia di rete elastica è assegnato un indirizzo IP privato preso dall'intervallo di indirizzi IP nella sottorete che hai specificato. Nessun indirizzo IP pubblico assegnato. I gruppi di sicurezza specificati nella connessione AWS Glue vengono applicati a ciascuna delle interfacce di rete elastiche. Per ulteriori informazioni, consulta Impostazione di Amazon VPC per le connessioni JDBC ai data store Amazon RDS su JDBC da AWS Glue.

Tutti i datastore JDBC ai quali il processo accede devono essere disponibili dalla sottorete VPC. Per accedere ad Amazon S3 dal VPC, serve un endpoint VPC. Se il processo deve accedere sia alle risorse VPC che alla rete Internet pubblica, il VPC deve disporre di un gateway NAT (Network Address Translation) al suo interno.

Un processo o endpoint di sviluppo può accedere a un solo VPC (e sottorete) alla volta. Se è necessario accedere a archivi dati in diversi VPCs, sono disponibili le seguenti opzioni:

- Utilizza VPC in peering per accedere ai datastore. Per ulteriori informazioni su VPC in peering, consulta Nozioni di base sul VPC in peering
- Usa un bucket Amazon S3 come posizione di storage intermedia. Dividi il lavoro in due processi, con l'output Amazon S3 del processo 1 come input per il processo 2.

Per dettagli su come connettersi a un datastore Amazon Redshift utilizzando Amazon VPC, consulta la pagina the section called “Configurazione di Redshift”.

Per dettagli su come connettersi a un datastore Amazon RDS utilizzando Amazon VPC, consulta la pagina the section called “Impostazione di Amazon VPC per connettersi ai data store Amazon RDS”.

Una volta impostate le regole necessarie in Amazon VPC, puoi creare una connessione in AWS Glue con le proprietà necessarie per connetterti ai datastore. Per ulteriori informazioni sulla connessione, consulta Connessione ai dati.

> **Note**
>
> Assicurati di configurare l'ambiente DNS per AWS Glue. Per ulteriori informazioni, consulta Configurazione di DNS nel VPC.

Argomenti

- Configurazione di un VPC per la connessione a PyPI per AWS Glue

- Configurazione di DNS nel VPC

# Configurazione di un VPC per la connessione a PyPI per AWS Glue

Il Python Package Index (PyPI) è un repository di software per il linguaggio di programmazione Python. Questo argomento affronta i dettagli necessari per supportare l'utilizzo dei pacchetti pip installati (come specificato dal creatore della sessione utilizzando il flag `--additional-python-modules`).

L'utilizzo di sessioni AWS Glue interattive con un connettore comporta l'uso della rete VPC tramite la sottorete specificata per il connettore. Di conseguenza, AWS i servizi e le altre destinazioni di rete non sono disponibili a meno che non si configuri una configurazione speciale.

Le soluzioni a questo problema includono:

- Utilizzo di un gateway Internet raggiungibile dalla sessione.
- Configurazione e utilizzo di un bucket S3 con un PyPI/simple repository contenente la chiusura transitiva delle dipendenze di un set di pacchetti.
- Utilizzo di un CodeArtifact repository che rispecchia PyPI e che è collegato al VPC.

## Impostazione di un gateway Internet

Gli aspetti tecnici sono descritti in dettaglio nei casi d'uso del gateway NAT, ma tieni presente questi requisiti per l'utilizzo di `--additional-python-modules`. In particolare, `--additional-python-modules` richiede l'accesso a pypi.org, che è determinato dalla configurazione del tuo VPC. Si notino i requisiti seguenti:

1. Il requisito di installare moduli python aggiuntivi tramite pip install per la sessione di un utente. Se la sessione utilizza un connettore, la configurazione potrebbe risentirne.
2. Quando viene utilizzato un connettore con `--additional-python-modules`, all'avvio della sessione la sottorete associata al connettore `PhysicalConnectionRequirements` deve fornire un percorso di rete per raggiungere pypi.org.
3. È necessario determinare se la configurazione è corretta o meno.

## Configurazione di un bucket Amazon S3 per ospitare un repository mirato PyPI/simple

Questo esempio configura un mirror PyPI in Amazon S3 per un set di pacchetti e le relative dipendenze.

Per configurare il mirror PyPI per un set di pacchetti:

```
# pip download all the dependencies
pip download -d s3pypi --only-binary :all: plotly gglplot
pip download -d s3pypi --platform manylinux_2_17_x86_64 --only-binary :all: psycopg2-
binary
# create and upload the pypi/simple index and wheel files to the s3 bucket
s3pypi -b test-domain-name --put-root-index -v s3pypi/*
```

Se disponi già di un repository di artefatti esistente, esso avrà un URL di indice per l'utilizzo di pip che puoi fornire al posto dell'URL di esempio per il bucket Amazon S3 di cui sopra.

Per utilizzare l'index-url personalizzato, con alcuni pacchetti di esempio:

```
%%configure
{
    "--additional-python-modules": "psycopg2_binary==2.9.5",
    "python-modules-installer-option": "--no-cache-dir --verbose --index-url https://
test-domain-name.s3.amazonaws.com/ --trusted-host test-domain-name.s3.amazonaws.com"
}
```

## Configurazione di un CodeArtifact mirror di pypi collegato al tuo VPC

Per configurare un mirror:

1. Crea un repository nella stessa regione della sottorete usata dal connettore.

   Seleziona `Public upstream repositories` e scegli `pypi-store`.
2. Fornisci l'accesso al repository dal VPC per la sottorete.
3. Specifica il valore `--index-url` corretto utilizzando l'`python-modules-installer-option`.

   ```
   %%configure
   {
       "--additional-python-modules": "psycopg2_binary==2.9.5",
       "python-modules-installer-option": "--no-cache-dir --verbose --index-url https://
   test-domain-name.s3.amazonaws.com/ --trusted-host test-domain-name.s3.amazonaws.com"
   ```

```
}
```

Per ulteriori informazioni, consulta Utilizzo CodeArtifact da un VPC.

### Configurazione di DNS nel VPC

Domain Name System (DNS) è uno standard che consente di risolvere i nomi utilizzati su Internet nei corrispondenti indirizzi IP. Un nome host DNS assegna un nome a un computer in modo univoco ed è costituito da un nome host e un nome di dominio. I server DNS risolvono i nomi host DNS nei corrispondenti indirizzi IP.

Per configurare il DNS nel VPC, accertarsi che i nomi host DNS e la risoluzione DNS siano abilitati nel VPC. Gli attributi di rete VPC `enableDnsHostnames` e `enableDnsSupport` devono essere impostati su `true`. Per visualizzare e modificare questi attributi, vai alla console VPC all'indirizzo https://console.aws.amazon.com/vpc/.

Per ulteriori informazioni, consulta Utilizzo del DNS con il tuo VPC. Inoltre, è possibile utilizzare AWS CLI e chiamare il modify-vpc-attributecomando per configurare gli attributi di rete VPC.

> **Note**
>
> Se usi Route 53, verifica che la configurazione non sostituisca gli attributi di rete DNS.

## Configurazione della crittografia in AWS Glue

L'esempio di flusso di lavoro seguente evidenzia le opzioni da configurare quando si usa la crittografia con AWS Glue. L'esempio dimostra l'uso di tasti specifici AWS Key Management Service (AWS KMS), ma è possibile scegliere altre impostazioni in base alle proprie esigenze particolari. Questo flusso di lavoro mette in evidenza solo le opzioni della configurazione di AWS Glue relative alla crittografia.

1. Se l'utente della console AWS Glue non usa una policy di autorizzazioni che permette tutte le operazioni API AWS Glue (ad esempio, `"glue:*"`), verificaì che siano permesse le operazioni seguenti:
   - `"glue:GetDataCatalogEncryptionSettings"`
   - `"glue:PutDataCatalogEncryptionSettings"`

- "glue:CreateSecurityConfiguration"
- "glue:GetSecurityConfiguration"
- "glue:GetSecurityConfigurations"
- "glue:DeleteSecurityConfiguration"

2. Qualsiasi client che accede o scrive in un catalogo crittografato, ovvero qualsiasi utente della console, crawler, processo o endpoint di sviluppo, necessita delle autorizzazioni seguenti.
3. Qualsiasi utente o ruolo che accede a una password di connessione crittografata necessita delle autorizzazioni seguenti.
4. Il ruolo di qualsiasi processo di estrazione, trasformazione e caricamento (ETL) che scrive dati crittografati in Amazon S3 necessita delle autorizzazioni seguenti.

   JSON

   ```
   {
     "Version":"2012-10-17",
     "Statement": {
       "Effect": "Allow",
       "Action": [
         "kms:Decrypt",
         "kms:Encrypt",
         "kms:GenerateDataKey"
       ],
       "Resource": "arn:aws:kms:us-east-1:111122223333:key/key-id"
     }
   }
   ```

5. Qualsiasi job o crawler ETL che scrive Amazon CloudWatch Logs crittografati richiede le seguenti autorizzazioni nella chiave e nelle policy IAM.

   Nella policy della chiave (non nella policy IAM):

   ```
   {
     "Effect": "Allow",
     "Principal": {
      "Service": "logs.region.amazonaws.com"
     },
     "Action": [
      "kms:Encrypt*",
      "kms:Decrypt*",
      "kms:ReEncrypt*",
   ```

```
   "kms:GenerateDataKey*",
   "kms:Describe*"
  ],
  "Resource": "<arn of key used for ETL/crawler cloudwatch encryption>"
 }
```

Per ulteriori informazioni sulle policy delle chiavi , consulta Utilizzo delle policy delle chiavi in AWS KMS nella Guida per gli sviluppatori di AWS Key Management Service .

Nella policy IAM collega l'autorizzazione `logs:AssociateKmsKey`:

```
{
  "Effect": "Allow",
  "Principal": {
   "Service": "logs.region.amazonaws.com"
  },
  "Action": [
   "logs:AssociateKmsKey"
  ],
  "Resource": "<arn of key used for ETL/crawler cloudwatch encryption>"
 }
```

6. Un processo ETL che usa un segnalibro di processo crittografato necessita delle autorizzazioni seguenti.

   JSON

   ```
   {
     "Version":"2012-10-17",
     "Statement": {
       "Effect": "Allow",
       "Action": [
         "kms:Decrypt",
         "kms:Encrypt"
       ],
       "Resource": "arn:aws:kms:us-east-1:111122223333:key/*"
     }
   }
   ```

7. Nella console AWS Glue scegliere Settings (Impostazioni) nel riquadro di navigazione.

a. Nella pagina Data catalog settings (Impostazioni del catalogo dati) crittografare il catalogo dati selezionando la casella di controllo Metadata encryption (Crittografia dei metadati). Questa opzione crittografa tutti gli oggetti nel Data Catalog con la chiave che scegli. AWS KMS

b. Per la chiave AWS KMS , scegliere aws/glue. Puoi anche scegliere una AWS KMS chiave che hai creato.

> **Important**
>
> AWS Gluesupporta solo chiavi master simmetriche del cliente ()CMKs. L'elenco di chiavi AWS KMS  mostra solo le chiavi simmetriche. Tuttavia, se si seleziona Scegli un ARN AWS KMS per una chiave, la console consente di inserire un ARN per qualsiasi tipo di chiave. Assicurati di inserire solo chiavi ARNs simmetriche.

Quando la crittografia è abilitata, il client che accede al catalogo dati deve disporre delle autorizzazioni per AWS KMS .

8. Nel riquadro di navigazione scegliere Security configurations (Configurazioni di sicurezza). Una configurazione della sicurezza è un set di proprietà di sicurezza che è possibile usare per configurare i processi di AWS Glue. Scegliere quindi Add security configuration (Aggiungi configurazione di sicurezza). Nella configurazione scegliere tra le opzioni seguenti:

   a. Selezionare la crittografia S3. Per Encryption mode (Modalità crittografia), scegliere SSE-KMS. Per AWS KMS key (Chiave AWS KMS), scegliere aws/s3 (verificare che l'utente disponga dell'autorizzazione per usare questa chiave). Ciò consente ai dati scritti dal processo su Amazon S3 di utilizzare la chiave AWS gestita AWS Glue AWS KMS .

   b. Seleziona la crittografia CloudWatch dei log e scegli un CMK. (Assicurarsi che l'utente disponga dell'autorizzazione per utilizzare questa chiave). Per ulteriori informazioni, consulta Encrypt Log Data in CloudWatch Logs Using nella Developer Guide AWS KMS.AWS Key Management Service

   > **Important**
   >
   > AWS Gluesupporta solo le chiavi master simmetriche del cliente (). CMKs L'elenco di chiavi AWS KMS  mostra solo le chiavi simmetriche. Tuttavia, se si seleziona Scegli un ARN AWS KMS per una chiave, la console consente di inserire un ARN per qualsiasi tipo di chiave. Assicurati di inserire solo chiavi ARNs simmetriche.

c. Scegliere Advanced properties (Proprietà avanzate) e selezionare la casella di controllo Job bookmark encryption (Crittografia segnalibro del processo). Per AWS KMS key (Chiave AWS KMS), scegliere aws/glue (verificare che l'utente disponga dell'autorizzazione per usare questa chiave). Ciò consente la crittografia dei segnalibri di lavoro scritti su Amazon S3 con AWS Glue AWS KMS la chiave.

9. Nel riquadro di navigazione, scegli Connections (Connessioni).
   a. Scegliere Add connection (Aggiungi connessione) per creare una connessione al datastore JDBC (Java Database Connectivity) di destinazione del processo ETL.
   b. Per applicare la crittografia SSL (Secure Sockets Layer), selezionare la casella di controllo Require SSL connection (Richiedi connessione SSL) e testare la connessione.
10. Nel riquadro di navigazione scegliere Jobs (Processi).
   a. Scegliere Add job (Aggiungi processo) per creare un processo che trasforma i dati.
   b. Nella definizione del processo scegliere la configurazione della sicurezza creata.
11. Nella console AWS Glue eseguire il processo on demand. Verifica che tutti i dati di Amazon S3 scritti dal processo, CloudWatch i log scritti dal processo e i segnalibri del lavoro siano tutti crittografati.

# Creazione di reti per lo sviluppo per AWS Glue

Per eseguire gli script di estrazione, trasformazione e caricamento (ETL) con AWS Glue, puoi sviluppare e testare gli script usando un endpoint di sviluppo. Gli endpoint di sviluppo non sono supportati per l'utilizzo con i processi AWS Glue versione 2.0. Per le versioni 2.0 e successive, il metodo di sviluppo preferito è l'utilizzo di Jupyter Notebook con uno dei kernel AWS Glue. Per ulteriori informazioni, consulta the section called “Nozioni di base sulle sessioni interattive AWS Glue”.

## Impostazione della rete per un endpoint di sviluppo

Quando imposti un endpoint di sviluppo, specifichi un Virtual Private Cloud (VPC), una sottorete e i gruppi di sicurezza.

> **Note**
>
> Assicurati di configurare l'ambiente DNS per AWS Glue. Per ulteriori informazioni, consulta Configurazione di DNS nel VPC.

Per permettere a AWS Glue di accedere alle risorse necessarie, aggiungi una riga nella tabella di routing della sottorete per associare un elenco di prefissi per Amazon S3 all'endpoint VPC. È necessario un ID elenco prefisso per la creazione di una regola del gruppo di sicurezza in uscita che consenta al traffico da un VPC di accedere a un servizio AWS tramite un endpoint VPC. Per semplificare la connessione a un server notebook associato a questo endpoint di sviluppo, dal computer locale, aggiungi una riga alla tabella di routing per aggiungere un ID Internet Gateway. Per ulteriori informazioni, consulta Endpoint VPC. Aggiorna la tabella di routing della sottorete in modo simile alla tabella seguente:

| Destinazione | Target | | |
|---|---|---|---|
| 10.0.0.0/16 | locale | | |
| pl-id per Amazon S3 | vpce-id | | |
| 0.0.0.0/0 | igw-xxxx | | |

Per permettere la comunicazione dei componenti di AWS Glue, specifica un gruppo di sicurezza con una regola per il traffico in entrata autoreferenziale per tutte le porte TCP. Creando una regola autoreferenziale, puoi limitare l'origine allo stesso gruppo di sicurezza del VPC senza essere aperta a tutte le reti. Il gruppo di sicurezza predefinito per il tuo VPC potrebbe già avere una regola autoreferenziata in entrata per ALL Traffic.

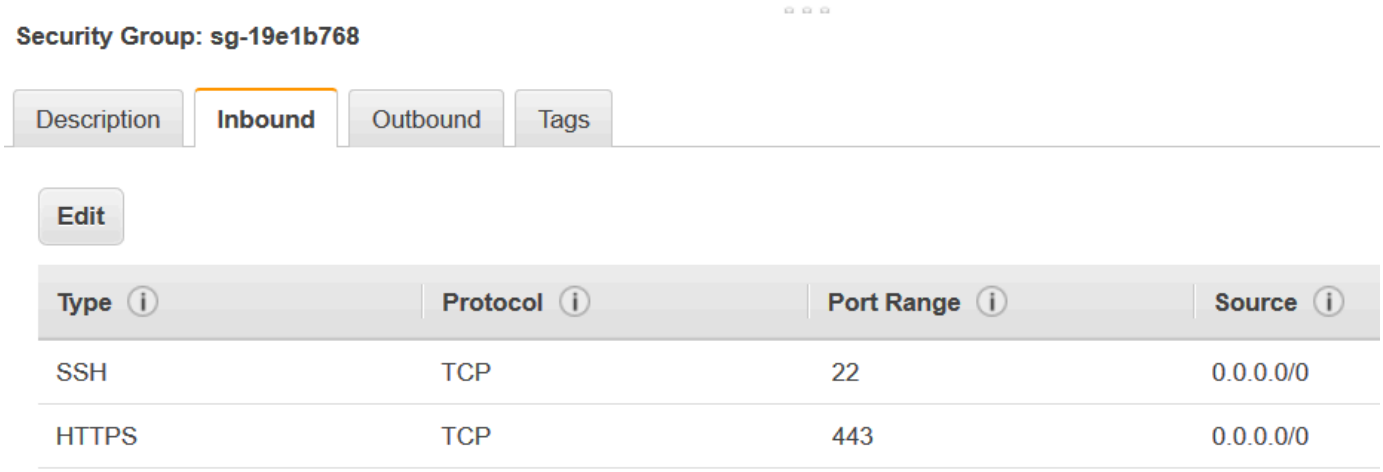
Per configurare un gruppo di sicurezza

1. Accedi Console di gestione AWS e apri la console Amazon EC2 all'indirizzo. https://console.aws.amazon.com/ec2/
2. Nel riquadro di navigazione a sinistra, scegli Security Groups (Gruppi di sicurezza).
3. Scegli un gruppo di sicurezza esistente dall'elenco o Create Security Group (Crea gruppo di sicurezza) da usare con l'endpoint di sviluppo.
4. Nel riquadro del gruppo di sicurezza, passa alla scheda Inbound (In entrata).
5. Aggiungi una regola autoreferenziale per permettere ai componenti di AWS Glue di comunicare tra loro. In particolare, aggiungi o verifica che sia presente una regola con Tipo `All TCP`, Protocollo `TCP`, Intervallo porte che include tutte le porte e Origine corrispondente al nome del gruppo di sicurezza indicato da ID gruppo.

   La regola in entrata è simile alla seguente:

| Tipo | Protocollo | Intervallo porte | Origine |
| --- | --- | --- | --- |
| Tutte le regole TCP | TCP | 0–65535 | *security-group* |

Il seguente è un esempio di regola in entrata autoreferenziale:




6. Aggiungi una regola anche per il traffico in uscita. Apri il traffico in uscita a tutte le porte o crea una regola autoreferenziale di Type (Tipo) `All TCP`, con Protocol (Protocollo) `TCP` e Port Range (Intervallo porte) che includa tutte le porte, la cui Source (Origine) abbia lo stesso nome del gruppo di sicurezza di Group ID (ID gruppo).

   La regola in uscita è simile a una delle seguenti regole:

| Tipo | Protocollo | Intervallo porte | Destinazione |
| --- | --- | --- | --- |
| Tutte le regole TCP | TCP | 0–65535 | *security-group* |
| All Traffic | ALL | ALL | 0.0.0.0/0 |

## Impostazione di Amazon EC2 per un server notebook

Con un endpoint di sviluppo, puoi creare un server notebook per testare gli script ETL con i notebook Jupyter. Per abilitare la comunicazione con il notebook, specifica un gruppo di sicurezza con regole in entrata per HTTPS (porta 443) e SSH (porta 22). Verifica che l'origine della regola sia 0.0.0.0/0 o l'indirizzo IP del computer che si collega al notebook.

Per configurare un gruppo di sicurezza

1. Accedi Console di gestione AWS e apri la console Amazon EC2 all'indirizzo. https://console.aws.amazon.com/ec2/
2. Nel riquadro di navigazione a sinistra, scegli Security Groups (Gruppi di sicurezza).
3. Scegli un gruppo di sicurezza esistente dall'elenco o Create Security Group (Crea gruppo di sicurezza) da usare con il server notebook. Il gruppo di sicurezza associato al tuo endpoint di sviluppo viene utilizzato anche per creare il server notebook.
4. Nel riquadro del gruppo di sicurezza, passa alla scheda Inbound (In entrata).
5. Aggiungi le regole in entrata simili alla seguente:

| Tipo | Protocollo | Intervallo porte | Origine |
|---|---|---|---|
| SSH | TCP | 22 | 0.0.0.0/0 |
| HTTPS | TCP | 443 | 0.0.0.0/0 |

Di seguito è riportato un esempio di regole in entrata per il gruppo di sicurezza:

# Scoperta e catalogazione dei dati in AWS Glue

AWS Glue Data Catalog Si tratta di un repository centralizzato che archivia i metadati relativi ai set di dati dell'organizzazione. Funge da indice per la posizione, lo schema e le metriche di runtime delle origini dati. I metadati vengono memorizzati come tabelle di metadati in cui ogni tabella rappresenta un singolo datastore.

È possibile popolare il Catalogo dati utilizzando un crawler, che analizza automaticamente le origini dati ed estrae i metadati. Un crawler può connettersi a origini dati interne (basate su AWS) ed esterne ad AWS.

Per ulteriori informazioni sulle origini dati supportate, consultare Origini dati supportate per il crawling.

È inoltre possibile creare manualmente tabelle nel Catalogo dati definendo la struttura della tabella, lo schema e la struttura di partizionamento in base ai requisiti specifici.

Per ulteriori informazioni sulla creazione manuale di una tabella di metadati, consultare Definizione manuale dei metadati.

È possibile utilizzare le informazioni presenti nel Catalogo dati per creare e monitorare i processi ETL. Il Data Catalog si integra con altri servizi di AWS analisi, fornendo una visione unificata delle fonti di dati che semplifica la gestione e l'analisi dei dati.

- Amazon Athena: archivia ed esegue le query sui metadati delle tabelle nel Catalogo dati per i dati di Amazon S3 tramite SQL.
- AWS Lake Formation — Definizione e gestione centralizzata di politiche di accesso ai dati dettagliate e verifica l'accesso ai dati.
- Amazon EMR: accede alle origini dati definite nel Catalogo dati per l'elaborazione di big data.
- Amazon SageMaker AI — Crea, addestra e implementa modelli di machine learning in modo rapido e sicuro.

Caratteristiche principali del Catalogo dati

Di seguito sono riportati gli aspetti chiave del Catalogo dati.

Repository di metadati

Catalogo dati funge da repository centrale di metadati e memorizza informazioni sulla posizione, lo schema e le proprietà delle origini dati. Questi metadati sono organizzati in database e tabelle, in modo simile a un tradizionale catalogo di database relazionali.

Rivelabilità dei dati automatica

Crawler di AWS Glue s è in grado di scoprire e catalogare automaticamente fonti di dati nuove o aggiornate, riducendo il sovraccarico legato alla gestione manuale dei metadati e garantendo la continuità del Data Catalog. up-to-date Catalogando le origini dati, il Catalogo dati consente agli utenti e alle applicazioni di scoprire e comprendere più facilmente le risorse di dati disponibili all'interno dell'organizzazione, promuovendo il riutilizzo dei dati e la collaborazione.

Il Catalogo dati supporta un'ampia gamma di origini dati, tra cui Amazon S3, Amazon RDS, Amazon Redshift, Apache Hive e altre ancora. Può dedurre e archiviare automaticamente i metadati da queste fonti utilizzando s. Crawler di AWS Glue

Per ulteriori informazioni, consultare Utilizzo dei crawler per compilare il Catalogo dati.

Gestione dello schema

Catalogo dati acquisisce e gestisce automaticamente lo schema delle origini dati, inclusi l'inferenza dello schema, l'evoluzione e il controllo delle versioni. È possibile aggiornare lo schema e le partizioni nel Data Catalog utilizzando AWS Glue i job ETL.

Ottimizzazione tabelle

Per migliorare le prestazioni di lettura da parte di servizi di AWS analisi come Amazon Athena e Amazon EMR e i processi AWS Glue ETL, il Data Catalog offre la compattazione gestita (un processo che compatta piccoli oggetti Amazon S3 in oggetti più grandi) per le tabelle Iceberg nel Data Catalog. Puoi utilizzare AWS Glue console AWS CLI, AWS Lake Formation console o AWS API per abilitare o disabilitare la compattazione per le singole tabelle Iceberg presenti nel Data Catalog.

Per ulteriori informazioni, consulta Ottimizzazione delle tabelle Iceberg.

Statistiche delle colonne

È possibile calcolare statistiche a livello di colonna per tabelle del Catalogo dati in formati di dati come Parquet, ORC, JSON, ION, CSV e XML senza configurare pipeline di dati aggiuntive. Le statistiche delle colonne consentono di comprendere i profili di dati ottenendo informazioni dettagliate sui valori all'interno di una colonna. Catalogo dati supporta la generazione di statistiche

per valori di colonna come valore minimo, valore massimo, valori nulli totali, valori distinti totali, lunghezza media dei valori e occorrenze totali di valori reali.

Per ulteriori informazioni, consulta Ottimizzazione delle prestazioni delle query con le statistiche delle colonne.

Data lineage

Catalogo dati registra le trasformazioni e le operazioni eseguite sui dati, fornendo informazioni sulla data lineage. Queste informazioni sono utili per il controllo, la conformità e la comprensione della provenienza dei dati.

Integrazione con altri servizi AWS

Il Data Catalog si integra perfettamente con altri AWS servizi, come Amazon Athena AWS Lake Formation, Amazon Redshift Spectrum e Amazon EMR. Questa integrazione consente di eseguire query e analizzare i dati su vari datastore utilizzando un unico livello di metadati coerente.

Sicurezza e controllo degli accessi

AWS Glue si integra AWS Lake Formation per supportare un controllo granulare degli accessi alle risorse di Data Catalog, consentendoti di gestire le autorizzazioni e l'accesso sicuro alle tue risorse di dati in base alle politiche e ai requisiti dell'organizzazione. AWS Glue si integra con AWS Key Management Service (AWS KMS) per crittografare i metadati archiviati nel Data Catalog.

Viste materializzate

Il Data Catalog supporta le viste materializzate di Apache Iceberg, che sono tabelle gestite che memorizzano i risultati precalcolati delle query SQL e si aggiornano automaticamente in base alle modifiche dei dati di origine sottostanti. Le viste materializzate semplificano le pipeline di trasformazione dei dati e accelerano le prestazioni delle query eliminando i calcoli ridondanti.

Puoi creare viste materializzate utilizzando Apache Spark SQL in AWS Glue versione 5.1 e successive, Amazon EMR versione 7.12.0 e successive e Amazon Athena. Il Data Catalog monitora automaticamente le tabelle di origine di Apache Iceberg e aggiorna le viste materializzate utilizzando un'infrastruttura di elaborazione gestita. I motori Spark di AWS Glue, Amazon EMR e Amazon Athena possono riscrivere automaticamente le query per utilizzare viste materializzate quando offrono prestazioni migliori.

Le viste materializzate vengono archiviate come tabelle Apache Iceberg in bucket Amazon S3 Tables o bucket Amazon S3 per uso generico all'interno del tuo account, rendendole accessibili da più motori di query. Il Data Catalog gestisce tutti gli aspetti del ciclo di vita delle

viste materializzate, tra cui la pianificazione automatica degli aggiornamenti, gli aggiornamenti incrementali e la gestione dei metadati.

Per ulteriori informazioni, consulta Utilizzo delle viste materializzate con AWS Glue e Utilizzo delle viste materializzate con Amazon EMR.

Argomenti


- Compilazione del catalogo AWS Glue dati
- Popolamento e gestione delle tabelle transazionali
- Gestione del Catalogo dati
- Accesso al Catalogo dati
- AWS Glue Procedure ottimali per Data Catalog
- Monitoraggio delle metriche di utilizzo del Data Catalog in Amazon CloudWatch
- Registro dello schema di AWS Glue


# Compilazione del catalogo AWS Glue dati

È possibile compilare il file AWS Glue Data Catalog utilizzando i seguenti metodi:

- Crawler di AWS Glue — An Crawler di AWS Glue può scoprire e catalogare automaticamente fonti di dati come database, data lake e dati in streaming. I crawler sono il metodo più comune e consigliato per popolare il catalogo dati in quanto possono scoprire e dedurre automaticamente i metadati per un'ampia varietà di origini dati.
- Aggiungere manualmente i metadati: puoi definire manualmente database, tabelle e dettagli di connessione e aggiungerli al Data Catalog utilizzando la AWS Glue console, la console Lake Formation o AWS Glue APIs. AWS CLI L'inserimento manuale è utile quando si desidera catalogare origini dati che non possono essere sottoposte a crawling.
- Integrazione con altri AWS servizi: puoi popolare il Data Catalog con metadati di servizi come Amazon AWS Lake Formation Athena. Questi servizi possono scoprire e registrare origini dati nel catalogo dati.
- Compilazione da un repository di metadati esistente: se disponi di un archivio di metadati esistente come Apache Hive Metastore, puoi utilizzarlo per importare tali metadati nel Data Catalog. AWS Glue Per ulteriori informazioni, consulta Migrazione tra Hive Metastore e on. AWS Glue Data Catalog GitHub

Argomenti

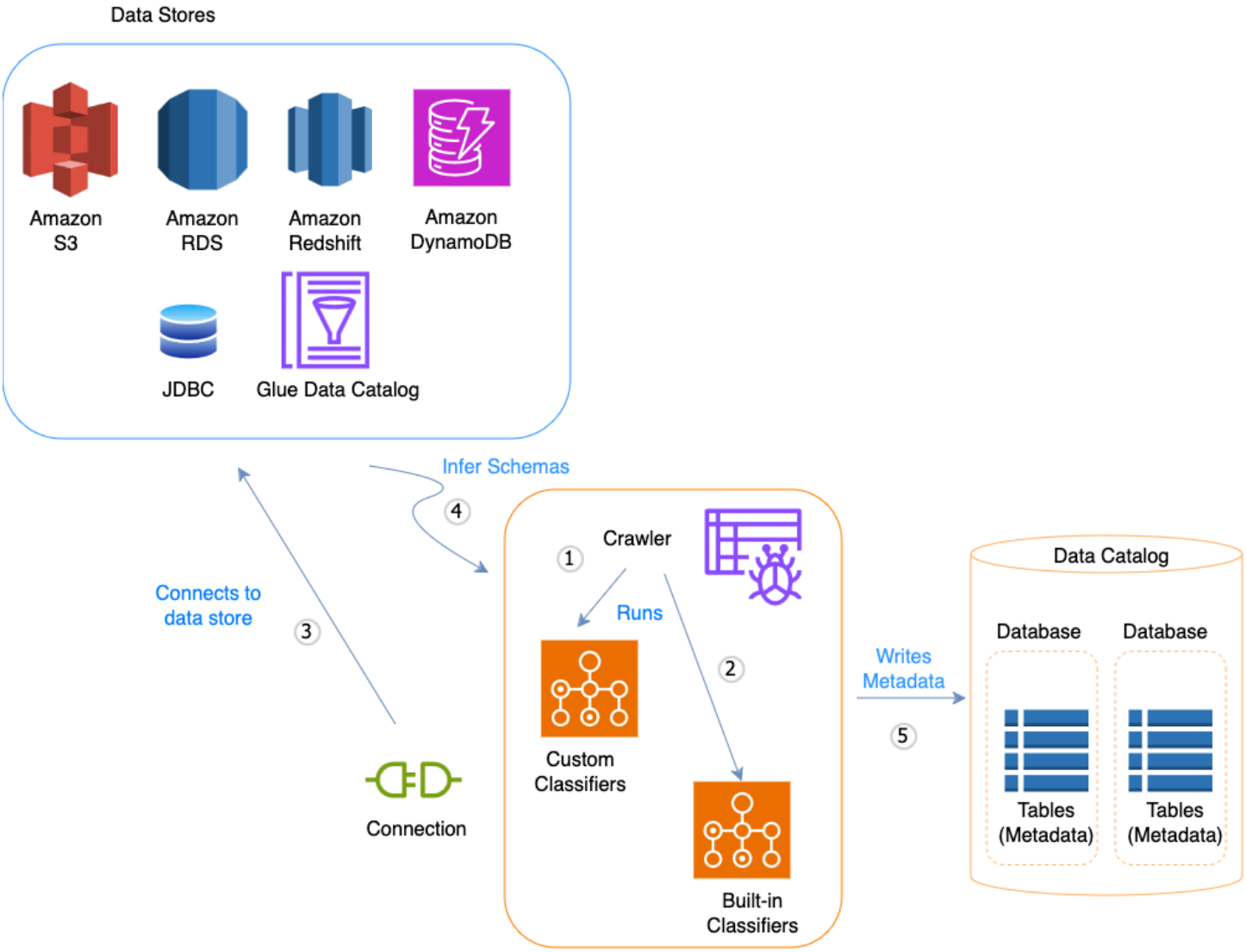

- Utilizzo dei crawler per compilare il Catalogo dati
- Definizione manuale dei metadati
- Integrazione con le tabelle Amazon S3
- Integrazione con altri servizi AWS
- Impostazioni del Catalogo dati


# Utilizzo dei crawler per compilare il Catalogo dati

È possibile utilizzare an Crawler di AWS Glue per popolarli AWS Glue Data Catalog con database e tabelle. Questo è il metodo principale utilizzato dalla maggior parte AWS Glue degli utenti. Un crawler è in grado di eseguire il crawling di archivi dati con una sola esecuzione. Al termine dell'operazione, il crawler crea o aggiorna una o più tabelle nel catalogo dati. I processi di estrazione, trasformazione e caricamento (ETL) definiti in AWS Glue usano queste tabelle del catalogo dati come origini e destinazioni. Il processo ETL legge e scrive negli archivi dati specificati nelle tabelle del catalogo dati di origine e di destinazione.

## Flusso di lavoro

Il diagramma del flusso di lavoro seguente mostra in che modo i crawler AWS Glue interagiscono con i datastore e altri elementi per popolare il catalogo dati.




Di seguito è riportato il flusso di lavoro generale che descrive il modo in cui il crawler popola il AWS Glue Data Catalog:

1. Un crawler esegue qualsiasi classificatore personalizzato da te selezionato per dedurre il formato e lo schema dei dati. É possibile fornire il codice per i classificatori personalizzati, i quali vengono eseguiti nell'ordine specificato.

   Il primo classificatore personalizzato che riconosce in modo corretto la struttura dei dati viene utilizzato per creare uno schema. I classificatori personalizzati in basso nell'elenco vengono ignorati.

2. Se nessun classificatore personalizzato è in grado di abbinar lo schema dei dati, i classificatori integrati ne tenteranno il riconoscimento. Un esempio di un classificatore incorporato è quello che riconosce JSON.

3. Il crawler si collega al datastore. Alcuni datastore richiedono proprietà di connessione per l'accesso al crawler.
4. Lo schema dedotto dei dati viene creato.
5. Il crawler scrive metadati nel catalogo dati. Una definizione di tabella contiene i metadati sui dati presenti nel datastore. La tabella viene scritta in un database, che è un container di tabelle nel catalogo dati. Gli attributi di una tabella includono la classificazione, ovvero un'etichetta creata dal classificatore che ha dedotto lo schema della tabella.

Argomenti


- Come funzionano i crawler
- In che modo un crawler determina quando creare le partizioni?
- Origini dati supportate per il crawling
- Prerequisiti del crawler
- Definizione e gestione dei classificatori
- Configurazione di un crawler
- Pianificazione di un crawler
- Visualizzazione dei risultati e dei dettagli del crawler
- Personalizzare il comportamento del crawler
- Tutorial: aggiunta di un crawler AWS Glue


## Come funzionano i crawler

Quando un crawler viene eseguito, è necessario effettuare le seguenti operazioni per interrogare un archivio dati:

- Classifica dei dati per determinare il formato, lo schema e le relative proprietà dei dati grezzi : è possibile configurare i risultati di classificazione creando un classificatore personalizzato.
- Raggruppamento di dati in tabelle o partizioni : i dati vengono raggruppati in base a processi euristici del crawler.
- Scrittura dei metadati nel catalogo dati: è possibile configurare il modo in cui il crawler aggiunge, aggiorna ed elimina tabelle e partizioni.

Quando si definisce un crawler, si scelgono uno o più classificatori che valutano il formato dei dati per dedurre uno schema. Quando il crawler viene eseguito, il primo classificatore nella lista per riconoscere con successo gli archivi dati è utilizzato per creare uno schema per la tabella. È possibile utilizzare classificatori incorporati o definirne uno proprio. Definisci i classificatori predefiniti in un'operazione separata prima di definire i crawler. AWS Glue fornisce classificatori predefiniti per dedurre gli schemi da file comuni con formati che includono JSON, CSV e Apache Avro. Per l'elenco aggiornato dei classificatori integrati nel AWS Glue, consulta Classificatori integrati.

Le tabelle di metadati create dal crawler sono contenute in un database nel momento in cui si definisce il crawler. Se il crawler non specifica un database, le tabelle sono posizionate in un database di default. Inoltre, ogni tabella ha una colonna di classificazione che viene compilata dal classificatore che per primo ha riconosciuto con successo l'archivio dati.

Se il file sottoposto a crawling è compresso, il crawler deve scaricarlo per elaborarlo. Quando viene eseguito, un crawler interroga i file per determinarne il formato e il tipo di compressione e scrive queste proprietà nel catalogo dati. Alcuni formati di file, ad esempio Apache Parquet, permettono di comprimere parti del file non appena vengono scritte. Per questi file, i dati compressi sono un componente interno del file e AWS Glue non popola la proprietà `compressionType` quando scrive tabelle nel catalogo dati. Al contrario, se un intero file viene compresso tramite un algoritmo di compressione, ad esempio gzip, la proprietà `compressionType` viene popolata quando vengono scritte tabelle nel catalogo dati.

Il crawler genera i nomi per le tabelle che crea. I nomi delle tabelle archiviate in base alle seguenti AWS Glue Data Catalog regole:

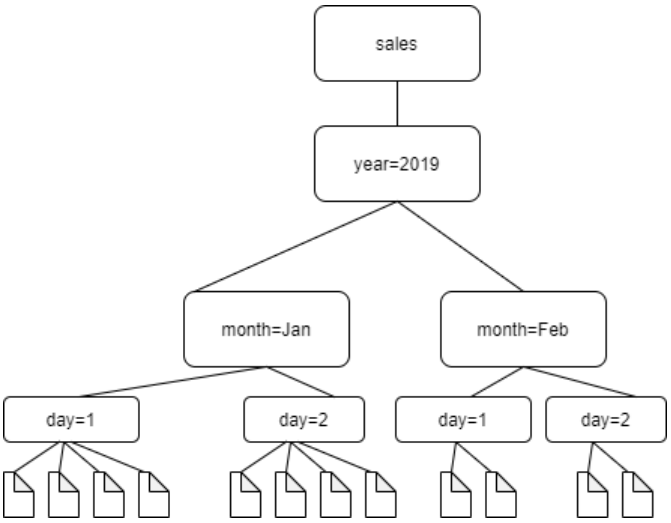
- Sono ammessi solo caratteri alfanumerici e caratteri di sottolineatura (_).
- Qualsiasi prefisso personalizzato non può essere più lungo di 64 caratteri.
- La lunghezza massima del nome non può essere superiore a 128 caratteri. Il crawler tronca i nomi generati per rientrare nel limite.
- Se vengono rilevati nomi di tabelle duplicati, il crawler aggiunge al nome un suffisso stringa hash.

Se il crawler viene eseguito più di una volta, su pianificazione, cerca file nuovi o modificati o tabelle nel tuo archivio dati. L'output del crawler include nuove tabelle e partizioni trovate nel corso di un'esecuzione precedente.

## In che modo un crawler determina quando creare le partizioni?

Quando un AWS Glue crawler esegue la scansione del data store Amazon S3 e rileva più cartelle in un bucket, determina la radice di una tabella nella struttura delle cartelle e quali cartelle sono partizioni di una tabella. Il nome della tabella si basa sul prefisso Amazon S3 o sul nome della cartella. Si deve specificare un Include path (Percorso di inclusione) che indichi il livello della cartella su cui eseguire il crawling. Quando la maggior parte degli schemi a un livello della cartella sono simili, il crawler crea partizioni di una tabella invece di tabelle separate. Per fare in modo che il crawler crei tabelle separate, aggiungere ogni cartella radice della tabella come archivio dati separato quando si definisce il crawler.

Considera, ad esempio, la seguente struttura di cartelle Amazon S3.




I percorsi alle quattro cartelle più in basso sono i seguenti:

```
S3://sales/year=2019/month=Jan/day=1
S3://sales/year=2019/month=Jan/day=2
S3://sales/year=2019/month=Feb/day=1
S3://sales/year=2019/month=Feb/day=2
```

Supponiamo che la destinazione del crawler sia impostata su `Sales` e che tutti i file nelle cartelle `day=`*n* siano nello stesso formato (ad esempio, JSON, non crittografato) e abbiano schemi uguali o molto simili. Il crawler creerà una singola tabella con quattro partizioni, con chiavi di partizione `year`, `month`, e `day`.

Nel prossimo esempio consideriamo la seguente struttura di cartelle Amazon S3:

```
s3://bucket01/folder1/table1/partition1/file.txt
s3://bucket01/folder1/table1/partition2/file.txt
s3://bucket01/folder1/table1/partition3/file.txt
s3://bucket01/folder1/table2/partition4/file.txt
s3://bucket01/folder1/table2/partition5/file.txt
```

Se gli schemi per i file sotto `table1` e `table2` sono simili e nel crawler viene definito un unico archivio dati con Include path (Percorso di inclusione) `s3://bucket01/folder1/`, il crawler crea un'unica tabella con due colonne delle chiavi di partizione. La prima colonna delle chiavi di partizione contiene `table1` e `table2` e la seconda colonna delle chiavi di partizione contiene da `partition1` a `partition3` per la partizione `table1` e `partition4` e `partition5` per la partizione `table2`. Per creare due tabelle separate, definire il crawler con due archivi dati. In questo esempio, definire il primo Include path (Percorso di inclusione) come `s3://bucket01/folder1/table1/` e il secondo come `s3://bucket01/folder1/table2`.

> **Note**
>
> In Amazon Athena, ogni tabella corrisponde a un prefisso Amazon S3 con tutti gli oggetti contenuti. Se gli oggetti hanno schemi diversi, Athena non riconosce gli oggetti diversi all'interno dello stesso prefisso come tabelle separate. Questo può verificarsi se un crawler crea più tabelle dallo stesso prefisso Amazon S3. In questo caso, alcune query in Athena possono restituire zero risultati. Perché Athena riconosca correttamente le tabelle ed esegua query sulle tabelle, crea il crawler con un Include path (Percorso di inclusione) separato per ogni schema di tabella diverso nella struttura di cartelle Amazon S3. Per ulteriori informazioni, consulta Best practice durante l'utilizzo di Athena con AWS Glue e questo articolo del Portale del sapere AWS.

## Origini dati supportate per il crawling

Un crawler può eseguire il crawling dei seguentgli archivi dati basati su file e su tabelle.

| Tipo di accesso utilizzato dal crawler | Archivi dati |
|---|---|
| Client nativo | • Amazon Simple Storage Service (Amazon S3)<br>• Amazon DynamoDB<br>• Delta Lake 2.0.x |

| Tipo di accesso utilizzato dal crawler | Archivi dati |
| --- | --- |
| | • Apache Iceberg 1.5<br>• Apache Hudi 0.14 |
| JDBC | Amazon Redshift<br><br>Snowflake<br><br>All'interno di Amazon Relational Database Service (Amazon RDS) o esterno ad Amazon RDS:<br><br>• Amazon Aurora<br>• MariaDB<br>• Microsoft SQL Server<br>• MySQL<br>• Oracle<br>• PostgreSQL |
| Client MongoDB | • MongoDB<br>• MongoDB Atlas<br>• Amazon DocumentDB (compatibile con MongoDB) |

> **Note**
>
> Attualmente AWS Glue non supporta i crawler per i flussi di dati.

Per gli archivi dati JDBC, MongoDB e Amazon DocumentDB (compatibile con MongoDB), è necessario specificare una connessione AWS Glue che il crawler può utilizzare per connettersi all'archivio dati. Per Amazon S3, puoi facoltativamente specificare una connessione di tipo Rete. Una connessione è un oggetto del catalogo dati che memorizza informazioni di connessione, ad esempio credenziali, URL, informazioni su Amazon Virtual Private Cloud e altro ancora. Per ulteriori informazioni, consulta Connessione ai dati.

Di seguito sono indicate le versioni dei driver supportate dal crawler:

| Prodotto | Driver supportato dal crawler |
| --- | --- |
| PostgreSQL | 42,21 |
| Amazon Aurora | Uguali ai driver dei crawler nativi |
| MariaDB | 8.0.13 |
| Microsoft SQL Server | 6.1,0 |
| MySQL | 8.0.13 |
| Oracle | 11.2.2 |
| Amazon Redshift | 4.1 |
| Snowflake | 3,13,20 |
| MongoDB | 4,7,2 |
| Atlante MongoDB | 4.7.2 |

Di seguito sono elencate le note sui vargli archivi dati.

Simple Storage Service (Amazon S3)

Puoi scegliere di eseguire il crawling di un percorso nel tuo account o in un altro account. Se tutti i file Amazon S3 in una cartella hanno lo stesso schema, il crawler crea una tabella. Inoltre, se l'oggetto Amazon S3 è partizionato, viene creata solo una tabella di metadati e le informazioni di partizionamento vengono aggiunte al catalogo dati per tale tabella.

Amazon S3 e Amazon DynamoDB

I crawler utilizzano un ruolo AWS Identity and Access Management (IAM) per l'autorizzazione ad accedere ai tuoi archivi di dati. Il ruolo passato al crawler deve avere l'autorizzazione per accedere ai percorsi Amazon S3 e alle tabelle Amazon DynamoDB di cui viene eseguito il crawling.

Amazon DynamoDB

Quando definisci un crawler usando la console AWS Glue, devi specificare una tabella DynamoDB. Se usi l'API di AWS Glue, puoi specificare un elenco di tabelle. È possibile scegliere

di eseguire il crawling solo di un piccolo campione di dati per ridurre i tempi di esecuzione del crawler.

Delta Lake

Per ogni archivio dati Delta Lake, specifichi la modalità di creazione delle tabelle Delta:

- Creazione di tabelle native: consente l'integrazione con i motori di query che supportano l'interrogazione diretta del log delle transazioni Delta. Per ulteriori informazioni, consulta Interrogare le tabelle Delta Lake.
- Creazione di tabelle Symlink: crea una cartella `_symlink_manifest` con file manifesti partizionati con chiavi di partizioni in base ai parametri di configurazione specificati.

Iceberg

Per ogni datastore Iceberg, specifichi un percorso Amazon S3 che contiene i metadati per le tabelle Iceberg. Se il crawler rileva i metadati della tabella Iceberg, li registra in Catalogo dati. È possibile impostare una pianificazione per il crawler per mantenere aggiornate le tabelle.

È possibile definire questi parametri per il datastore:

- Esclusioni: consente di ignorare determinate cartelle.
- Profondità di attraversamento massima: imposta il limite di profondità che il crawler può esplorare nel bucket Amazon S3. La profondità di attraversamento massima predefinita è 10, mentre la profondità massima che puoi impostare è 20.

Hudi

Per ogni datastore Iceberg, specifichi un percorso Amazon S3 che contiene i metadati per le tabelle Hudi. Se il crawler rileva i metadati della tabella Hudi, li registra in Catalogo dati. È possibile impostare una pianificazione per il crawler per mantenere aggiornate le tabelle.

È possibile definire questi parametri per il datastore:

- Esclusioni: consente di ignorare determinate cartelle.
- Profondità di attraversamento massima: imposta il limite di profondità che il crawler può esplorare nel bucket Amazon S3. La profondità di attraversamento massima predefinita è 10, mentre la profondità massima che puoi impostare è 20.

> **Note**
>
> Le colonne di timestamp con tipi logici `millis` verranno interpretate come `bigint` a causa di un'incompatibilità con Hudi 0.13.1 e i tipi di timestamp. Una risoluzione potrebbe essere fornita nella prossima versione di Hudi.

Le tabelle Hudi sono classificate come segue, con implicazioni specifiche per ciascuna di esse:

- Copia in scrittura (CoW): i dati vengono archiviati in un formato colonnare (Parquet) e ogni aggiornamento crea una nuova versione dei file durante una scrittura.
- Unisci in lettura (MoW): i dati vengono archiviati utilizzando una combinazione di formati colonnare (Parquet) e basati su righe (Avro). Gli aggiornamenti vengono registrati nei file delta basati su righe e vengono compattati come necessario per creare nuove versioni dei file colonnari.

Con i set di dati CoW, ogni volta che c'è un aggiornamento a un record, il file che contiene il record viene riscritto con i valori aggiornati. Quando si lavora con un set di dati MoR, ogniqualvolta è disponibile un aggiornamento Hudi scrive solo la riga per il registro modificato. MoR è più adatto per carichi di lavoro pesanti in scrittura o modifiche con meno letture. CoW è più adatto per carichi di lavoro pesanti di lettura su dati che cambiano meno frequentemente.

Hudi fornisce tre tipi di query per accedere ai dati:

- Query snapshot: query che visualizzano l'ultimo snapshot della tabella a partire da una determinata operazione di commit o compattazione. Per le tabelle MoR, le query snapshot espongono lo stato più recente della tabella unendo i file di base e delta della parte di file più recente al momento della query.
- Query incrementali: le query vedono solo i nuovi dati scritti nella tabella dal momento di un determinato commit/compattazione. Questo fornisce in modo efficace flussi di modifica per abilitare pipeline di dati incrementali.
- Query ottimizzate per la lettura (ReadOptimized): per le tabelle MoR, le query vedono i dati compattati più recenti. Per le tabelle CoW, le query vedono i dati più recenti impegnati.

Per le Copy-On-Write tabelle, i crawler creano una singola tabella nel Data Catalog con il serde. ReadOptimized `org.apache.hudi.hadoop.HoodieParquetInputFormat`

Per le Merge-On-Read tabelle, il crawler crea due tabelle nel Data Catalog per la stessa posizione della tabella:

- Una tabella con suffisso `_ro` che utilizza il serde. ReadOptimized `org.apache.hudi.hadoop.HoodieParquetInputFormat`
- Una tabella con suffisso `_rt` che utilizza RealTime Serde che consente di eseguire query Snapshot:. `org.apache.hudi.hadoop.realtime.HoodieParquetRealtimeInputFormat`

MongoDB e Amazon DocumentDB (compatibile con MongoDB)

Sono supportate le versioni 3.2 e successive di MongoDB. È possibile scegliere di eseguire il crawling solo di un piccolo campione di dati per ridurre i tempi di esecuzione del crawler.

Database relazionale

L'autenticazione avviene con un nome utente e una password del database. A seconda del tipo di motore di database, è possibile scegliere quali oggetti sono sottoposti a crawling, ad esempio database, schemi e tabelle.

Snowflake

Il crawler JDBC Snowflake supporta il crawling della tabella, della tabella esterna, della vista e della vista materializzata. La definizione della vista materializzata non verrà compilata.

Per le tabelle esterne Snowflake, il crawler eseguirà il crawling se punta a una posizione Amazon S3. Oltre allo schema della tabella, il crawler eseguirà il crawling anche della posizione di Amazon S3, del formato del file e dell'output come parametri di tabella nella tabella del catalogo di dati. Tenere presente che le informazioni sulla partizione della tabella esterna con partizioni non sono compilate.

ETL non è attualmente supportato per le tabelle del catalogo dati create utilizzando il crawler Snowflake.

## Prerequisiti del crawler

Il crawler presuppone le autorizzazioni del ruolo AWS Identity and Access Management (IAM) specificato al momento della definizione. Questo ruolo IAM deve avere le autorizzazioni necessarie per estrarre dati dall'archivio dati e scriverli nel catalogo dati. La console AWS Glue elenca solo i ruoli IAM cui è collegata una policy di trust per il servizio dell'entità principale AWS Glue. Dalla console puoi anche creare un ruolo IAM con una policy IAM per accedere ad archivi dati Amazon S3 cui accede il crawler. Per ulteriori informazioni su come specificare ruoli per AWS Glue, consulta Politiche basate sull'identità per Glue AWS.

> **Note**
>
> Quando esegui la scansione di un data store Delta Lake, devi disporre Read/Write delle autorizzazioni per la posizione Amazon S3.

Per il crawler, è possibile creare un ruolo e allegare le seguenti policy:

- La policy `AWSGlueServiceRole` AWS gestita, che concede le autorizzazioni necessarie sul Data Catalog
- Una policy inline che concede le autorizzazioni per l'origine dati.
- Una policy inline che concede l'autorizzazione `iam:PassRole` sul ruolo.

Un approccio più rapido consiste nel lasciare che la procedura guidata del crawler della console AWS Glue crei un ruolo per te. Il ruolo che crea è specifico per il crawler e include la policy `AWSGlueServiceRole` AWS gestita più la policy in linea richiesta per l'origine dati specificata.

Se si specifica un ruolo esistente per un crawler, bisogna assicurarsi che includa la policy `AWSGlueServiceRole` o equivalente (o una versione ridotta di questa policy), oltre alle policy inline richieste. Ad esempio, per un archivio dati Amazon S3, la policy inline sarebbe almeno la seguente:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::bucket/object*"
      ]
    }
  ]
}
```

Per un archivio dati Amazon DynamoDB, la policy sarebbe almeno la seguente:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "dynamodb:DescribeTable",
        "dynamodb:Scan"
      ],
      "Resource": [
        "arn:aws:dynamodb:us-east-1:111122223333:table/table-name*"
      ]
    }
  ]
}
```

Inoltre, se il crawler legge AWS Key Management Service (AWS KMS) dati Amazon S3 crittografati, il ruolo IAM deve disporre dell'autorizzazione di decrittografia sulla chiave. AWS KMS Per ulteriori informazioni, consulta Fase 2: creare un ruolo IAM per AWS Glue.

## Definizione e gestione dei classificatori

Un classificatore legge i dati in un archivio dati. Se riconosce il formato dei dati, genera uno schema. Il classificatore inoltre restituisce un numero di certezza per indicare quanto è stato certo il riconoscimento del formato.

AWS Glue fornisce un set di classificatori predefiniti, ma puoi anche crearne di personalizzati. AWS Glue richiama per primi i classificatori personalizzati, in base all'ordine che hai specificato nella definizione del crawler. A seconda dei risultati restituiti dai classificatori personalizzati, AWS Glue può anche richiamare i classificatori predefiniti. Se un classificatore restituisce `certainty=1.0` durante l'elaborazione, significa che la certezza di creare uno schema corretto è del 100%. AWS Glue usa quindi l'oputput di questo classificatore.

Se nessun classificatore restituisce `certainty=1.0`, AWS Glue usa l'output del classificatore che indica la certezza maggiore. Se nessun classificatore restituisce una certezza superiore a `0.0`, AWS Glue restituisce la stringa di classificazione predefinita `UNKNOWN`.

### Quando si utilizza un classificatore?

Puoi usare classificatori quando esegui il crawling di un datastore per definire tabelle di metadati nel AWS Glue Data Catalog. È possibile configurare il crawler con un set ordinato di classificatori. Quando il crawler richiama un classificatore, il classificatore determina se i dati vengono riconosciuti. Se il classificatore non è in grado di riconoscere i dati o non è certo al 100 percento, il crawler richiama il prossimo classificatore nell'elenco per determinare se è in grado di riconoscere i dati.

Per ulteriori informazioni sulla creazione di un classificatore tramite la console AWS Glue, consulta Creazione di classificatori usando la console AWS Glue.

### Classificatori personalizzati

L'output di un classificatore include una stringa che indica la classificazione del file o il formato (ad esempio, `json`) e lo schema del file. Per classificatori personalizzati, è possibile definire la logica per la creazione di uno schema in base al tipo di classificatore. I tipi di classificatore includono definizioni di schemi in base a pattern grok, tag XML e percorsi JSON.

Se si modifica una definizione di classificatore, qualsiasi tipo di dati precedentemente sottoposto a crawling utilizzando il classificatore non è riclassificato. Un crawler tiene traccia dei dati precedentemente sottoposti a crawling. I nuovi dati sono classificati con il classificatore aggiornato, che potrebbe dare origine a uno schema aggiornato. Se lo schema dei dati si è evoluto, aggiornare il classificatore per verificare qualsiasi modifica dello schema quando viene eseguito il crawler. Per riclassificare i dati al fine di correggere un classificatore errato, creare un nuovo crawler con il classificatore aggiornato.

Per ulteriori informazioni sulla creazione di classificatori personalizzati in AWS Glue, consulta Scrittura di classificatori personalizzati per diversi formati di dati.

> **Note**
>
> Se il formato di dati è riconosciuto da uno dei classificatori integrati, non è necessario creare un classificatore personalizzato.

### Classificatori integrati

AWS Glue fornisce classificatori predefiniti per diversi formati, tra cui JSON, CSV, log Web e molti sistemi di database.

Se AWS Glue non trova un classificatore personalizzato appropriato al formato di dati di input con una certezza del 100%, richiama i classificatori predefiniti in base all'ordine mostrato nella tabella seguente. Il classificatore integrato restituisce un risultato per indicare se il formato corrisponde (`certainty=1.0`) o non, corrisponde (`certainty=0.0`). Il primo classificatore con `certainty=1.0` fornisce la stringa di classificazione e lo schema per una tabella di metadati nel catalogo dati.

| Tipo di classificatore | Stringa di classificazione | Note |
|---|---|---|
| Apache Avro | `avro` | Legge lo schema nella parte iniziale del file per determinare il formato. |
| Apache ORC | `orc` | Legge i metadati dei file per determinare il formato. |
| Apache Parquet | `parquet` | Legge lo schema nella parte finale del file per determinare il formato. |
| JSON | `json` | Legge la parte iniziale del file per determinare il formato. |
| Binary JSON | `bson` | Legge la parte iniziale del file per determinare il formato. |
| XML | `xml` | Legge la parte iniziale del file per determinare il formato. AWS Glue determina lo schema della tabella in base ai tag XML nel documento.<br><br>Per ulteriori informazioni su come creare un classificatore XML personalizzato per specificare le righe nel documento, consulta Scrittura di classificatori personalizzati XML. |
| Amazon Ion | `ion` | Legge la parte iniziale del file per determinare il formato. |
| Combined Apache log | `combined_apache` | Determina i formati dei log attraverso un pattern grok. |

| Tipo di classificatore | Stringa di classificazione | Note |
|---|---|---|
| Apache log | `apache` | Determina i formati dei log attraverso un pattern grok. |
| Linux kernel log | `linux_kernel` | Determina i formati dei log attraverso un pattern grok. |
| Microsoft log | `microsoft_log` | Determina i formati dei log attraverso un pattern grok. |
| Ruby log | `ruby_logger` | Legge la parte iniziale del file per determinare il formato. |
| Squid 3.x log | `squid` | Legge la parte iniziale del file per determinare il formato. |
| Redis monitor log | `redismonlog` | Legge la parte iniziale del file per determinare il formato. |
| Redis log | `redislog` | Legge la parte iniziale del file per determinare il formato. |
| CSV | `csv` | Verifica dei seguenti delimitatori: virgola (,), barra verticale (\|), tabulazione (\ t), punto e virgola (;) e Ctrl-A (\u0001). Ctrl-A è il carattere di controllo Unicode per `Start Of Heading`. |
| Amazon Redshift | `redshift` | Utilizza la connessione JDBC per importare i metadati. |
| MySQL | `mysql` | Utilizza la connessione JDBC per importare i metadati. |
| PostgreSQL | `postgresql` | Utilizza la connessione JDBC per importare i metadati. |

| Tipo di classificatore | Stringa di classificazione | Note |
|---|---|---|
| Oracle database | `oracle` | Utilizza la connessione JDBC per importare i metadati. |
| Microsoft SQL Server | `sqlserver` | Utilizza la connessione JDBC per importare i metadati. |
| Amazon DynamoDB | `dynamodb` | Legge i dati dalla tabella DynamoDB. |

I file nei seguenti formati compressi possono essere classificati:

- ZIP (supportato per gli archivi che contengono un solo file). Il formato ZIP non è supportato in modo adeguato in altri servizi (a causa dell'archivio).
- BZIP
- GZIP
- LZ4
- Snappy (supportato sia per i formati Snappy standard che nativi Hadoop)

Classificatore CSV integrato

Il classificatore CSV predefinito analizza il contenuto dei file CSV per determinare lo schema per una tabella AWS Glue. Questo classificatore controlla i seguenti delimitatori:

- Virgola (,)
- Barra verticale (|)
- Tabulazione (\t)
- Punto e virgola (;)
- Ctrl-A (\u0001)

  Ctrl-A è il carattere di controllo Unicode per `Start Of Heading`.

Per essere classificato come CSV, lo schema della tabella deve avere almeno due colonne e due righe di dati. Il classificatore CSV utilizza una serie di procedimenti euristici per determinare se

un'intestazione è presente in un determinato file. Se il classificatore non è in grado di determinare l'intestazione della prima riga di dati, le intestazioni delle colonne vengono visualizzate come `col1`, `col2`, `col3` e così via. Il classificatore CSV integrato stabilisce se dedurre un'intestazione valutando le seguenti caratteristiche del file:

- Ogni colonna in una potenziale intestazione compare come tipo di dati STRING.
- Ad eccezione dell'ultima colonna, ogni colonna in una potenziale intestazione contiene meno di 150 caratteri. Per consentire un delimitatore finale, l'ultima colonna può essere vuota in tutto il file.
- Ogni colonna in una potenziale intestazione deve soddisfare i requisiti di AWS Glue in `regex`.
- La riga di intestazione deve essere sufficientemente diversa dalle righe di dati. Per determinarla, una o più righe devono comparire come diverse dal tipo STRING. Se tutte le colonne sono di tipo STRING, la prima riga di dati non è abbastanza diversa dalle righe successive per essere utilizzata come intestazione.

> **Note**
>
> Se il classificatore CSV predefinito non crea la tabella AWS Glue come desideri, puoi usare una delle alternative seguenti:
>
> - Modifica i nomi delle colonne nel catalogo dati, imposta `SchemaChangePolicy` su LOG e imposta la configurazione di output della partizione su `InheritFromTable` per le future esecuzioni del crawler.
> - Creare un classificatore grok personalizzato per analizzare i dati e assegnare le colonne desiderate.
> - Il classificatore CSV integrato crea tabelle referenziando `LazySimpleSerDe` come libreria di serializzazione, che è una valida scelta per l'inferenza del tipo. Tuttavia, se i dati CSV contengono stringhe tra virgolette, modificate la definizione della tabella e cambiate la libreria in. SerDe `OpenCSVSerDe` Modificare tutti i tipi dedotti in STRING, impostare `SchemaChangePolicy` su LOG e impostare la configurazione di output delle partizioni su `InheritFromTable` per le future esecuzioni del crawler. Per ulteriori informazioni sulle SerDe librerie, consulta SerDe Reference nella Amazon Athena User Guide.

## Scrittura di classificatori personalizzati per diversi formati di dati

Puoi fornire un classificatore personalizzato per classificare i dati in AWS Glue. Puoi creare un classificatore personalizzato utilizzando un pattern grok, un tag XML, JavaScript Object Notation (JSON) o valori separati da virgole (CSV). Un crawler AWS Glue richiama un classificatore personalizzato. Se il classificatore riconosce i dati, esso restituisce la classificazione e lo schema dei dati al crawler. Potrebbe essere necessario definire un classificatore personalizzato nel caso in cui i dati non corrispondano ad alcun classificatore integrato, oppure se si intende personalizzare le tabelle create dal crawler.

Per ulteriori informazioni sulla creazione di un classificatore tramite la console AWS Glue, consulta Creazione di classificatori usando la console AWS Glue.

AWS Glue esegue i classificatori personalizzati prima dei classificatori predefiniti, in base all'ordine da te specificato. Quando un crawler individua un classificatore corrispondente ai dati, lo schema e la stringa di classificazione vengono usati nella definizione delle tabelle che vengono scritte nel AWS Glue Data Catalog.

Argomenti

- Scrittura di classificatori personalizzati grok
- Scrittura di classificatori personalizzati XML
- Scrittura di classificatori personalizzati JSON
- Scrittura di classificatori personalizzati CSV

### Scrittura di classificatori personalizzati grok

Grok è uno strumento che consente di analizzare dati testuali dato un pattern corrispondente. Un pattern grok è un set denominato di espressioni regolari (regex) che vengono usate per confrontare i dati di una riga per volta. AWS Glue usa pattern grok dedurre lo schema dei dati. Quando un pattern grok individua la corrispondenza dei dati, AWS Glue usa il pattern per determinare la struttura dei dati e mapparla in campi.

AWS Glue offre numerosi pattern predefiniti, ma puoi definirne di personalizzati. È possibile creare un pattern grok tramite pattern integrati o pattern personalizzati nella definizione di classificatori personalizzati. È possibile adattare un pattern grok per classificare i formati di file di testo personalizzati.

> **Note**
>
> I classificatori personalizzati grok AWS Glue usano la libreria di serializzazione `GrokSerDe` per le tabelle create nel AWS Glue Data Catalog. Se utilizzi Amazon Athena, Amazon EMR o Redshift Spectrum, consulta la documentazione relativa a tali servizi per informazioni sul supporto di. AWS Glue Data Catalog `GrokSerDe` Al momento, possono verificarsi problemi di esecuzione di query su tabelle create con `GrokSerDe` da Amazon EMR e Redshift Spectrum.

Di seguito è riportata la sintassi di base per i componenti di un pattern grok:

```
%{PATTERN:field-name}
```

I dati corrispondenti al `PATTERN` denominato vengono mappati alla colonna `field-name` nello schema, con un tipo di `string` di dati predefinito. Facoltativamente, è possibile eseguire il cast del tipo di dati per il campo in `byte`, `boolean`, `double`, `short`, `int`, `long` o `float` nello schema risultante.

```
%{PATTERN:field-name:data-type}
```

Ad esempio, per trasmettere un campo `num` a un tipo di dati `int`, si può utilizzare questo pattern:

```
%{NUMBER:num:int}
```

I pattern possono essere costituiti da altri pattern. Ad esempio, può esserci un pattern per un timestamp `SYSLOG` definito da pattern per mese, giorno del mese e ora (ad esempio, `Feb 1 06:25:43`). Per questi dati, è possibile definire il pattern seguente:

```
SYSLOGTIMESTAMP %{MONTH} +%{MONTHDAY} %{TIME}
```

> **Note**
>
> I pattern grok sono in grado di elaborare una sola riga alla volta. I pattern a righe multiple non sono supportati. In aggiunta, le interruzioni di riga all'interno dei pattern non sono supportate.

Valori personalizzati per il classificatore Grok

Quando definisci un classificatore Grok, devi specificare i valori seguenti per creare il classificatore personalizzato.

Name

Nome del classificatore.

Classificazione

La stringa di testo scritta per descrivere il formato dei dati da classificare, ad esempio `special-logs`.

Pattern grok

Il set di pattern applicati al datastore per determinare l'esistenza di corrispondenze. Questi pattern provengono dai AWS Gluepattern predefiniti di e da eventuali pattern personalizzati da te definiti.

Di seguito è riportato un esempio di pattern grok:

```
%{TIMESTAMP_ISO8601:timestamp} \[%{MESSAGEPREFIX:message_prefix}\]
 %{CRAWLERLOGLEVEL:loglevel} : %{GREEDYDATA:message}
```

Quando i dati corrispondono a `TIMESTAMP_ISO8601`, viene creata una colonna di schema `timestamp`. Il funzionamento è analogo per i restanti pattern denominati nell'esempio.

Pattern personalizzati

Pattern personalizzati facoltativi da te definiti. Il pattern grok che classifica i dati fa riferimento a questi pattern. È possibile fare riferimento a questi pattern personalizzati nel pattern grok applicato ai dati. Ciascun pattern personalizzato che compone il pattern grok deve trovarsi su righe separate. Per definire il pattern, si utilizza la sintassi Espressione regolare (regex).

Di seguito è riportato un esempio di utilizzo di pattern personalizzati:

```
CRAWLERLOGLEVEL (BENCHMARK|ERROR|WARN|INFO|TRACE)
MESSAGEPREFIX .*-.*-.*-.*-.*
```

Il primo pattern personalizzato denominato, `CRAWLERLOGLEVEL`, costituisce una corrispondenza quando i dati corrispondono a una delle stringhe enumerate. Il secondo pattern personalizzato, `MESSAGEPREFIX`, tenta di abbinare una stringa di messaggio di prefisso.

AWS Glue tiene traccia della data e dell'ora di creazione, della data e dell'ora dell'ultimo aggiornamento e della versione del classificatore.

Pattern predefiniti

AWS Glue offre molti pattern comuni che puoi usare per creare un classificatore personalizzato. É possibile aggiungere un pattern denominato al `grok pattern` all'interno di una definizione di classificatore.

L'elenco seguente comprende una riga per ciascun pattern. In ciascuna riga, il nome del pattern è seguito dalla sua definizione. Per definire il pattern, viene usata la sintassi delle espressioni regolari (regex).

```
#<noloc>&GLU;</noloc> Built-in patterns
 USERNAME [a-zA-Z0-9._-]+
 USER %{USERNAME:UNWANTED}
 INT (?:[+-]?(?:[0-9]+))
 BASE10NUM (?<![0-9.+-])(?>[+-]?(?:(?:[0-9]+(?:\.[0-9]+)?)|(?:\.[0-9]+)))
 NUMBER (?:%{BASE10NUM:UNWANTED})
 BASE16NUM (?<![0-9A-Fa-f])(?:[+-]?(?:0x)?(?:[0-9A-Fa-f]+))
 BASE16FLOAT \b(?<![0-9A-Fa-f.])(?:[+-]?(?:0x)?(?:(?:[0-9A-Fa-f]+(?:\.[0-9A-Fa-f]*)?)|
(?:\.[0-9A-Fa-f]+)))\b
 BOOLEAN (?i)(true|false)

 POSINT \b(?:[1-9][0-9]*)\b
 NONNEGINT \b(?:[0-9]+)\b
 WORD \b\w+\b
 NOTSPACE \S+
 SPACE \s*
 DATA .*?
 GREEDYDATA .*
 #QUOTEDSTRING (?:(?<!\\)(?:"(?:\\.|[^\\"])*"|(?:'(?:\\.|[^\\'])*')|(?:`(?:\\.|[^\
\`])*`)))
 QUOTEDSTRING (?>(?<!\\)(?>"(?>\\.|[^\\"]+)+"|""|(?>'(?>\\.|[^\\']+)+')|''|(?>`(?>\\.|
[^\\`]+)+`)|``))
 UUID [A-Fa-f0-9]{8}-(?:[A-Fa-f0-9]{4}-){3}[A-Fa-f0-9]{12}

 # Networking
 MAC (?:%{CISCOMAC:UNWANTED}|%{WINDOWSMAC:UNWANTED}|%{COMMONMAC:UNWANTED})
 CISCOMAC (?:(?:[A-Fa-f0-9]{4}\.){2}[A-Fa-f0-9]{4})
 WINDOWSMAC (?:(?:[A-Fa-f0-9]{2}-){5}[A-Fa-f0-9]{2})
 COMMONMAC (?:(?:[A-Fa-f0-9]{2}:){5}[A-Fa-f0-9]{2})
```

```
 IPV6 ((([0-9A-Fa-f]{1,4}:){7}([0-9A-Fa-f]{1,4}|:))|(([0-9A-Fa-f]{1,4}:){6}(:[0-9A-
Fa-f]{1,4}|((25[0-5]|2[0-4]\d|1\d\d|[1-9]?\d)(\.(25[0-5]|2[0-4]\d|1\d\d|[1-9]?\d))
{3})|:))|(([0-9A-Fa-f]{1,4}:){5}(((:[0-9A-Fa-f]{1,4}){1,2})|:((25[0-5]|2[0-4]\d|1\d
\d|[1-9]?\d)(\.(25[0-5]|2[0-4]\d|1\d\d|[1-9]?\d)){3})|:))|(([0-9A-Fa-f]{1,4}:){4}(((:
[0-9A-Fa-f]{1,4}){1,3})|((:[0-9A-Fa-f]{1,4})?:((25[0-5]|2[0-4]\d|1\d\d|[1-9]?\d)(\.
(25[0-5]|2[0-4]\d|1\d\d|[1-9]?\d)){3}))|:))|(([0-9A-Fa-f]{1,4}:){3}(((:[0-9A-Fa-f]
{1,4}){1,4})|((:[0-9A-Fa-f]{1,4}){0,2}:((25[0-5]|2[0-4]\d|1\d\d|[1-9]?\d)(\.(25[0-5]|
2[0-4]\d|1\d\d|[1-9]?\d)){3}))|:))|(([0-9A-Fa-f]{1,4}:){2}(((:[0-9A-Fa-f]{1,4}){1,5})|
((:[0-9A-Fa-f]{1,4}){0,3}:((25[0-5]|2[0-4]\d|1\d\d|[1-9]?\d)(\.(25[0-5]|2[0-4]\d|1\d
\d|[1-9]?\d)){3}))|:))|(([0-9A-Fa-f]{1,4}:){1}(((:[0-9A-Fa-f]{1,4}){1,6})|((:[0-9A-Fa-
f]{1,4}){0,4}:((25[0-5]|2[0-4]\d|1\d\d|[1-9]?\d)(\.(25[0-5]|2[0-4]\d|1\d\d|[1-9]?\d))
{3}))|:))|(:(((:[0-9A-Fa-f]{1,4}){1,7})|((:[0-9A-Fa-f]{1,4}){0,5}:((25[0-5]|2[0-4]\d|
1\d\d|[1-9]?\d)(\.(25[0-5]|2[0-4]\d|1\d\d|[1-9]?\d)){3}))|:)))(%.+)?
 IPV4 (?<![0-9])(?:(?:25[0-5]|2[0-4][0-9]|[0-1]?[0-9]{1,2})[.](?:25[0-5]|2[0-4][0-9]|
[0-1]?[0-9]{1,2})[.](?:25[0-5]|2[0-4][0-9]|[0-1]?[0-9]{1,2})[.](?:25[0-5]|2[0-4][0-9]|
[0-1]?[0-9]{1,2}))(?![0-9])
 IP (?:%{IPV6:UNWANTED}|%{IPV4:UNWANTED})
 HOSTNAME \b(?:[0-9A-Za-z][0-9A-Za-z-_]{0,62})(?:\.(?:[0-9A-Za-z][0-9A-Za-z-_]
{0,62}))*(\.?|\b)
 HOST %{HOSTNAME:UNWANTED}
 IPORHOST (?:%{HOSTNAME:UNWANTED}|%{IP:UNWANTED})
 HOSTPORT (?:%{IPORHOST}:%{POSINT:PORT})

 # paths
 PATH (?:%{UNIXPATH}|%{WINPATH})
 UNIXPATH (?>/(?>[\w_%!$@:.,~-]+|\\.)*)+
 #UNIXPATH (?<![\w\/])(?:/[^\/\s?*]*)+
 TTY (?:/dev/(pts|tty([pq])?)(\w+)?/?(?:[0-9]+))
 WINPATH (?>[A-Za-z]+:|\\)(?:\\[^\\?*]*)+
 URIPROTO [A-Za-z]+(\+[A-Za-z+]+)?
 URIHOST %{IPORHOST}(?::%{POSINT:port})?
 # uripath comes loosely from RFC1738, but mostly from what Firefox
 # doesn't turn into %XX
 URIPATH (?:/[A-Za-z0-9$.+!*'(){},~:;=@#%_\-]*)+
 #URIPARAM \?(?:[A-Za-z0-9]+(?:=(?:[^&]*))?(?:&(?:[A-Za-z0-9]+(?:=(?:[^&]*))?)?)*)?
 URIPARAM \?[A-Za-z0-9$.+!*'|(){},~@#%&/=:;_?\-\[\]]*
 URIPATHPARAM %{URIPATH}(?:%{URIPARAM})?
 URI %{URIPROTO}://(?:%{USER}(?::[^@]*)?@)?(?:%{URIHOST})?(?:%{URIPATHPARAM})?

 # Months: January, Feb, 3, 03, 12, December
 MONTH \b(?:Jan(?:uary)?|Feb(?:ruary)?|Mar(?:ch)?|Apr(?:il)?|May|Jun(?:e)?|Jul(?:y)?|
Aug(?:ust)?|Sep(?:tember)?|Oct(?:ober)?|Nov(?:ember)?|Dec(?:ember)?)\b
 MONTHNUM (?:0?[1-9]|1[0-2])
 MONTHNUM2 (?:0[1-9]|1[0-2])
```

```
 MONTHDAY (?:(?:0[1-9])|(?:[12][0-9])|(?:3[01])|[1-9])

 # Days: Monday, Tue, Thu, etc...
 DAY (?:Mon(?:day)?|Tue(?:sday)?|Wed(?:nesday)?|Thu(?:rsday)?|Fri(?:day)?|
Sat(?:urday)?|Sun(?:day)?)

 # Years?
 YEAR (?>\d\d){1,2}
 # Time: HH:MM:SS
 #TIME \d{2}:\d{2}(?::\d{2}(?:\.\d+)?)?
 # TIME %{POSINT<24}:%{POSINT<60}(?::%{POSINT<60}(?:\.%{POSINT})?)?
 HOUR (?:2[0123]|[01]?[0-9])
 MINUTE (?:[0-5][0-9])
 # '60' is a leap second in most time standards and thus is valid.
 SECOND (?:(?:[0-5]?[0-9]|60)(?:[:.,][0-9]+)?)
 TIME (?!<[0-9])%{HOUR}:%{MINUTE}(?::%{SECOND})(?![0-9])
 # datestamp is YYYY/MM/DD-HH:MM:SS.UUUU (or something like it)
 DATE_US %{MONTHNUM}[/-]%{MONTHDAY}[/-]%{YEAR}
 DATE_EU %{MONTHDAY}[./-]%{MONTHNUM}[./-]%{YEAR}
 DATESTAMP_US %{DATE_US}[- ]%{TIME}
 DATESTAMP_EU %{DATE_EU}[- ]%{TIME}
 ISO8601_TIMEZONE (?:Z|[+-]%{HOUR}(?::?%{MINUTE}))
 ISO8601_SECOND (?:%{SECOND}|60)
 TIMESTAMP_ISO8601 %{YEAR}-%{MONTHNUM}-%{MONTHDAY}[T ]%{HOUR}:?%{MINUTE}(?::?
%{SECOND})?%{ISO8601_TIMEZONE}?
 TZ (?:[PMCE][SD]T|UTC)
 DATESTAMP_RFC822 %{DAY} %{MONTH} %{MONTHDAY} %{YEAR} %{TIME} %{TZ}
 DATESTAMP_RFC2822 %{DAY}, %{MONTHDAY} %{MONTH} %{YEAR} %{TIME} %{ISO8601_TIMEZONE}
 DATESTAMP_OTHER %{DAY} %{MONTH} %{MONTHDAY} %{TIME} %{TZ} %{YEAR}
 DATESTAMP_EVENTLOG %{YEAR}%{MONTHNUM2}%{MONTHDAY}%{HOUR}%{MINUTE}%{SECOND}
 CISCOTIMESTAMP %{MONTH} %{MONTHDAY} %{TIME}

 # Syslog Dates: Month Day HH:MM:SS
 SYSLOGTIMESTAMP %{MONTH} +%{MONTHDAY} %{TIME}
 PROG (?:[\w._/%-]+)
 SYSLOGPROG %{PROG:program}(?:\[%{POSINT:pid}\])?
 SYSLOGHOST %{IPORHOST}
 SYSLOGFACILITY <%{NONNEGINT:facility}.%{NONNEGINT:priority}>
 HTTPDATE %{MONTHDAY}/%{MONTH}/%{YEAR}:%{TIME} %{INT}

 # Shortcuts
 QS %{QUOTEDSTRING:UNWANTED}

 # Log formats
```

```
 SYSLOGBASE %{SYSLOGTIMESTAMP:timestamp} (?:%{SYSLOGFACILITY} )?%{SYSLOGHOST:logsource}
 %{SYSLOGPROG}:

 MESSAGESLOG %{SYSLOGBASE} %{DATA}

 COMMONAPACHELOG %{IPORHOST:clientip} %{USER:ident} %{USER:auth}
 \[%{HTTPDATE:timestamp}\] "(?:%{WORD:verb} %{NOTSPACE:request}(?: HTTP/
%{NUMBER:httpversion})?|%{DATA:rawrequest})" %{NUMBER:response} (?:%{Bytes:bytes=
%{NUMBER}|-})
 COMBINEDAPACHELOG %{COMMONAPACHELOG} %{QS:referrer} %{QS:agent}
 COMMONAPACHELOG_DATATYPED %{IPORHOST:clientip} %{USER:ident;boolean} %{USER:auth}
 \[%{HTTPDATE:timestamp;date;dd/MMM/yyyy:HH:mm:ss Z}\] "(?:%{WORD:verb;string}
 %{NOTSPACE:request}(?: HTTP/%{NUMBER:httpversion;float})?|%{DATA:rawrequest})"
 %{NUMBER:response;int} (?:%{NUMBER:bytes;long}|-)


 # Log Levels
 LOGLEVEL ([A|a]lert|ALERT|[T|t]race|TRACE|[D|d]ebug|DEBUG|[N|n]otice|NOTICE|[I|i]nfo|
INFO|[W|w]arn?(?:ing)?|WARN?(?:ING)?|[E|e]rr?(?:or)?|ERR?(?:OR)?|[C|c]rit?(?:ical)?|
CRIT?(?:ICAL)?|[F|f]atal|FATAL|[S|s]evere|SEVERE|EMERG(?:ENCY)?|[Ee]merg(?:ency)?)
```

Scrittura di classificatori personalizzati XML

XML definisce la struttura di un documento tramite l'uso di tag nel file. Con un classificatore personalizzato XML, è possibile specificare il nome tag utilizzato per definire una riga.

Valori personalizzati per un classificatore XML

Quando definisci un classificatore XML, devi specificare i valori seguenti in AWS Glue per creare il classificatore. Il campo classificazione di questo classificatore è impostato su `xml`.

Name

Nome del classificatore.

Tag di riga

Il nome tag XML che definisce una riga di tabella nel documento XML, senza parentesi angolate < >. Il nome deve rispettare le regole XML relative ai tag.

> **Note**
>
> L'elemento contenente i dati di riga non può essere un elemento vuoto che si auto-chiude. Questo elemento vuoto, ad esempio, non viene analizzato da AWS Glue:

```
<row att1=”xx” att2=”yy” />
```

È possibile scrivere gli elementi vuoti come segue:

```
<row att1=”xx” att2=”yy”> </row>
```

AWS Glue tiene traccia della data e dell'ora di creazione, della data e dell'ora dell'ultimo aggiornamento e della versione del classificatore.

Supponi, ad esempio, di avere il file XML seguente. Per creare una tabella AWS Glue che contiene solo le colonne per autore e titolo, crea un classificatore nella console AWS Glue con Row tag (Tag di riga) impostato su `AnyCompany`. Aggiungi ed esegui quindi un crawler che usa questo classificatore personalizzato.

```
<?xml version="1.0"?>
<catalog>
   <book id="bk101">
     <AnyCompany>
       <author>Rivera, Martha</author>
       <title>AnyCompany Developer Guide</title>
     </AnyCompany>
   </book>
   <book id="bk102">
     <AnyCompany>
       <author>Stiles, John</author>
       <title>Style Guide for AnyCompany</title>
     </AnyCompany>
   </book>
</catalog>
```

### Scrittura di classificatori personalizzati JSON

JSON è un formato per lo scambio di dati. Definisce le strutture di dati con coppie nome-valore o con un elenco ordinato di valori. Con un classificatore personalizzato JSON, puoi specificare il percorso JSON di una struttura di dati usata per definire lo schema per la tabella.

#### Valori dei classificatori personalizzati in AWS Glue

Quando definisci un classificatore JSON, devi specificare i valori seguenti in AWS Glue per creare il classificatore. Il campo classificazione di questo classificatore è impostato su `json`.

Name

Nome del classificatore.

Percorso JSON

Percorso JSON che fa riferimento a un oggetto usato per definire uno schema di tabella. È possibile scrivere il percorso JSON in notazione punto o in notazione parentesi. Sono supportati i seguenti operatori:

| Description |
| --- |
| Elemento radice di un oggetto JSON. Avvia tutte le espressioni del percorso |
| Carattere jolly. Disponibile ovunque siano necessari un nome o un elemento numerico nel percorso JSON. |
| Figlio con notazione dot. Specifica un campo figlio in un oggetto JSON. |
| Figlio con notazione parentesi. Specifica campi figli in un oggetto JSON. Solo un singolo campo figlio può essere specificato. |
| Indice di matrice. Specifica il valore di una matrice in base all'indice. |

AWS Glue tiene traccia della data e dell'ora di creazione, della data e dell'ora dell'ultimo aggiornamento e della versione del classificatore.

Example Utilizzo di un classificatore JSON per estrarre record da una matrice

Supponi che i dati JSON siano costituiti da una matrice di record. Ad esempio, le prime righe del file potrebbero apparire come segue:

```
[
  {
    "type": "constituency",
    "id": "ocd-division\/country:us\/state:ak",
    "name": "Alaska"
  },
  {
    "type": "constituency",
    "id": "ocd-division\/country:us\/state:al\/cd:1",
    "name": "Alabama's 1st congressional district"
  },
  {
    "type": "constituency",
    "id": "ocd-division\/country:us\/state:al\/cd:2",
    "name": "Alabama's 2nd congressional district"
  },
  {
    "type": "constituency",
    "id": "ocd-division\/country:us\/state:al\/cd:3",
    "name": "Alabama's 3rd congressional district"
  },
  {
    "type": "constituency",
    "id": "ocd-division\/country:us\/state:al\/cd:4",
    "name": "Alabama's 4th congressional district"
  },
  {
    "type": "constituency",
    "id": "ocd-division\/country:us\/state:al\/cd:5",
    "name": "Alabama's 5th congressional district"
  },
  {
    "type": "constituency",
    "id": "ocd-division\/country:us\/state:al\/cd:6",
    "name": "Alabama's 6th congressional district"
  },
  {
    "type": "constituency",
```

```
    "id": "ocd-division\/country:us\/state:al\/cd:7",
    "name": "Alabama's 7th congressional district"
  },
  {
    "type": "constituency",
    "id": "ocd-division\/country:us\/state:ar\/cd:1",
    "name": "Arkansas's 1st congressional district"
  },
  {
    "type": "constituency",
    "id": "ocd-division\/country:us\/state:ar\/cd:2",
    "name": "Arkansas's 2nd congressional district"
  },
  {
    "type": "constituency",
    "id": "ocd-division\/country:us\/state:ar\/cd:3",
    "name": "Arkansas's 3rd congressional district"
  },
  {
    "type": "constituency",
    "id": "ocd-division\/country:us\/state:ar\/cd:4",
    "name": "Arkansas's 4th congressional district"
  }
]
```

Quando esegui un crawler usando il classificatore JSON predefinito, l'intero file viene utilizzato per definire lo schema. Poiché non specifichi un percorso JSON, il crawler tratta i dati come un unico oggetto, ovvero, come una matrice. Ad esempio, lo schema potrebbe apparire come segue:

```
root
|-- record: array
```

Tuttavia, per creare uno schema basato su ciascun record presente nella matrice JSON, crea un classificatore JSON personalizzato e specifica il percorso JSON come `$[*]`. Quando si specifica questo percorso JSON, il classificatore interroga tutti i 12 record presenti nella matrice per determinare lo schema. Lo schema risultante contiene campi separati per ciascun oggetto, analogamente all'esempio seguente:

```
root
|-- type: string
```

```
|-- id: string
|-- name: string
```

Example Utilizzo di un classificatore JSON per esaminare solo alcune parti di un file

Supponi che i dati JSON seguano il pattern del file di esempio JSON `s3://awsglue-datasets/examples/us-legislators/all/areas.json` estratto dal sito http://everypolitician.org/. Gli oggetti di esempio nel file JSON hanno l'aspetto seguente:

```
{
  "type": "constituency",
  "id": "ocd-division\/country:us\/state:ak",
  "name": "Alaska"
}
{
  "type": "constituency",
  "identifiers": [
    {
      "scheme": "dmoz",
      "identifier": "Regional\/North_America\/United_States\/Alaska\/"
    },
    {
      "scheme": "freebase",
      "identifier": "\/m\/0hjy"
    },
    {
      "scheme": "fips",
      "identifier": "US02"
    },
    {
      "scheme": "quora",
      "identifier": "Alaska-state"
    },
    {
      "scheme": "britannica",
      "identifier": "place\/Alaska"
    },
    {
      "scheme": "wikidata",
      "identifier": "Q797"
    }
  ],
```

```
  "other_names": [
    {
      "lang": "en",
      "note": "multilingual",
      "name": "Alaska"
    },
    {
      "lang": "fr",
      "note": "multilingual",
      "name": "Alaska"
    },
    {
      "lang": "nov",
      "note": "multilingual",
      "name": "Alaska"
    }
  ],
  "id": "ocd-division\/country:us\/state:ak",
  "name": "Alaska"
}
```

Quando esegui un crawler usando il classificatore JSON predefinito, l'intero file viene utilizzato per creare lo schema. È possibile ritrovarsi con uno schema di questo tipo:

```
root
|-- type: string
|-- id: string
|-- name: string
|-- identifiers: array
|    |-- element: struct
|    |    |-- scheme: string
|    |    |-- identifier: string
|-- other_names: array
|    |-- element: struct
|    |    |-- lang: string
|    |    |-- note: string
|    |    |-- name: string
```

Tuttavia, per creare uno schema usando solo l'oggetto "`id`", crea un classificatore JSON personalizzato e specifica il percorso JSON come `$.id`. Lo schema sarà così basato solo sul campo "`id`":

```
root
|-- record: string
```

Le prime righe di dati estratte con questo schema hanno l'aspetto seguente:

```
{"record": "ocd-division/country:us/state:ak"}
{"record": "ocd-division/country:us/state:al/cd:1"}
{"record": "ocd-division/country:us/state:al/cd:2"}
{"record": "ocd-division/country:us/state:al/cd:3"}
{"record": "ocd-division/country:us/state:al/cd:4"}
{"record": "ocd-division/country:us/state:al/cd:5"}
{"record": "ocd-division/country:us/state:al/cd:6"}
{"record": "ocd-division/country:us/state:al/cd:7"}
{"record": "ocd-division/country:us/state:ar/cd:1"}
{"record": "ocd-division/country:us/state:ar/cd:2"}
{"record": "ocd-division/country:us/state:ar/cd:3"}
{"record": "ocd-division/country:us/state:ar/cd:4"}
{"record": "ocd-division/country:us/state:as"}
{"record": "ocd-division/country:us/state:az/cd:1"}
{"record": "ocd-division/country:us/state:az/cd:2"}
{"record": "ocd-division/country:us/state:az/cd:3"}
{"record": "ocd-division/country:us/state:az/cd:4"}
{"record": "ocd-division/country:us/state:az/cd:5"}
{"record": "ocd-division/country:us/state:az/cd:6"}
{"record": "ocd-division/country:us/state:az/cd:7"}
```

Per creare uno schema basato su un oggetto con nidificazione profonda, come "`identifier`", nel file JSON puoi creare un classificatore JSON personalizzato e specificare il percorso JSON come `$.identifiers[*].identifier`. Sebbene lo schema sia simile a quello dell'esempio precedente, si basa su un oggetto diverso nel file JSON.

Lo schema ha il seguente aspetto:

```
root
```

```
|-- record: string
```

Elencando le prime righe di dati della tabella è possibile vedere che lo schema si basa sui dati nell'oggetto "`identifier`":

```
{"record": "Regional/North_America/United_States/Alaska/"}
{"record": "/m/0hjy"}
{"record": "US02"}
{"record": "5879092"}
{"record": "4001016-8"}
{"record": "destination/alaska"}
{"record": "1116270"}
{"record": "139487266"}
{"record": "n79018447"}
{"record": "01490999-8dec-4129-8254-eef6e80fadc3"}
{"record": "Alaska-state"}
{"record": "place/Alaska"}
{"record": "Q797"}
{"record": "Regional/North_America/United_States/Alabama/"}
{"record": "/m/0gyh"}
{"record": "US01"}
{"record": "4829764"}
{"record": "4084839-5"}
{"record": "161950"}
{"record": "131885589"}
```

Per creare una tabella basata su un altro oggetto con nidificazione profonda, come il campo "`name`" nella matrice "`other_names`" nel file JSON, puoi creare un classificatore JSON personalizzato e specificare il percorso JSON come `$.other_names[*].name`. Sebbene lo schema sia simile a quello dell'esempio precedente, si basa su un oggetto diverso nel file JSON. Lo schema ha il seguente aspetto:

```
root
|-- record: string
```

Elencando le prime righe di dati della tabella è possibile vedere che lo schema si basa sui dati nell'oggetto "`name`" nella matrice "`other_names`":

```
{"record": "Alaska"}
{"record": "Alaska"}
{"record": "Аляска"}
{"record": "Alaska"}
{"record": "Alaska"}
{"record": "Alaska"}
{"record": "Alaska"}
{"record": "Alaska"}
{"record": "Alaska"}
{"record": "######"}
{"record": "######"}
{"record": "######"}
{"record": "Alaska"}
{"record": "Alyaska"}
{"record": "Alaska"}
{"record": "Alaska"}
{"record": "Штат Аляска"}
{"record": "Аляска"}
{"record": "Alaska"}
{"record": "#######"}
```

Scrittura di classificatori personalizzati CSV

I classificatori CSV personalizzati consentono di specificare i tipi di dati per ogni colonna nel campo del classificatore CSV personalizzato. È possibile specificare il tipo di dati di ogni colonna separato da una virgola. Specificando i tipi di dati, è possibile sovrascrivere i tipi di dati dedotti dai crawler e garantire che i dati vengano classificati in modo appropriato.

Puoi impostare il file CSV SerDe per l'elaborazione nel classificatore, che verrà applicato nel Data Catalog.

Quando crei un classificatore personalizzato, puoi anche riutilizzare il classificatore per diversi crawler.

- Per i file csv con solo intestazioni (nessun dato), questi file verranno classificati come SCONOSCIUTI poiché non vengono fornite informazioni sufficienti. Se specifichi che il CSV "contiene titoli" nell'opzione Column headings (Intestazioni delle colonne) e fornisci i tipi di dati, possiamo classificare correttamente questi file.

Puoi usare un classificatore CSV personalizzato per dedurre lo schema di vari tipi di dati CSV. Gli attributi personalizzati che puoi fornire per il classificatore includono delimitatori, un' SerDe opzione CSV, opzioni relative all'intestazione e se eseguire determinate convalide sui dati.

Valori dei classificatori personalizzati in AWS Glue

Quando definisci un classificatore CSV, devi fornire i valori seguenti in AWS Glue per creare il classificatore. Il campo classificazione di questo classificatore è impostato su `csv`.

Nome del classificatore

Nome del classificatore.

SerDe CSV

Imposta il file CSV SerDe per l'elaborazione del classificatore, che verrà applicato nel Data Catalog. Le opzioni sono Open CSV SerDe, Lazy SerDe Simple e None. È possibile specificare il valore None quando si desidera che il crawler esegua il rilevamento.

Delimitatore di colonna

Un simbolo personalizzato per indicare il separatore di ogni voce di colonna nella riga. Fornisci un carattere unicode. Se non riesci a digitare il delimitatore, puoi copiarlo e incollarlo. Questo vale per i caratteri stampabili, compresi quelli non supportati dal sistema (in genere indicati come □).

Simbolo di virgolette

Un simbolo personalizzato per indicare la combinazione dei contenuti in un singolo valore di colonna. Deve essere diverso dal delimitatore di colonna. Fornisci un carattere unicode. Se non riesci a digitare il delimitatore, puoi copiarlo e incollarlo. Questo vale per i caratteri stampabili, compresi quelli non supportati dal sistema (in genere indicati come □).

Intestazioni di colonna

Indica il comportamento per il modo in cui le intestazioni di colonna devono essere rilevate nel file CSV. Se il file CSV personalizzato include le intestazioni di colonna, inserisci un elenco di intestazioni di colonna delimitate da virgole.

Opzioni di elaborazione: consenti i file con una singola colonna

Abilita l'elaborazione dei file che contengono una sola colonna.

Opzioni di elaborazione: taglia lo spazio vuoto prima dell'identificazione dei valori di colonna

Specifica se tagliare i valori prima di individuare il tipo dei valori di colonna.

Tipi di dati personalizzati: facoltativo

Inserisci il tipo di dati personalizzato separato da una virgola. Specifica i tipi di dati personalizzati nel file CSV. Il tipo di dati personalizzato deve essere un tipo di dati supportato. I tipi di dati supportati sono: "BINARY", "BOOLEAN", "DATE", "DECIMAL", "DOUBLE", "FLOAT", "INT", "LONG", "SHORT", "STRING", "TIMESTAMP". I tipi di dati non supportati mostreranno un errore.

## Creazione di classificatori usando la console AWS Glue

Un classificatore determina lo schema dei dati. Puoi scrivere un classificatore personalizzato e puntarvi da AWS Glue.

### Creazione dei classificatori

Per aggiungere un classificatore nella console AWS Glue, scegli Add classifier (Aggiungi classificatore). Quando definisci un classificatore, specifichi i valori per le seguenti opzioni:

- Classifier name (Nome del classificatore) – Fornisci un nome univoco per il tuo classificatore.
- Classifier type (Tipo di classificazione) – Il tipo di classificazione delle tabelle dedotte dal classificatore.
- Last updated (Ultimo aggiornamento) – L'ultima volta in cui è stato aggiornato il classificatore.

Nome del classificatore

Fornisci un nome univoco per il tuo classificatore.

Tipo di classificatore

Scegli il tipo di classificatore da creare.

A seconda del tipo di classificatore scelto, configurare le seguenti proprietà per il classificatore:

Grok

- Classificazione

  Descrivi il formato o il tipo di dati classificati o fornisci un'etichetta personalizzata.
- Pattern grok

Viene utilizzato per analizzare i dati in uno schema strutturato. Il pattern grok è composto da modelli denominati che descrivono il formato dell'archivio dati. Puoi scrivere questo pattern grok usando il modello predefinito denominato fornito da AWS Glue e i modelli personalizzati che scrivi e includi nel campo Custom patterns (Modelli personalizzati). Anche se i risultati dei debugger grok potrebbero non corrispondere esattamente ai risultati di AWS Glue, ti consigliamo di provare il modello usando alcuni dati di esempio con un debugger grok. Puoi trovare i debugger grok sul Web. I modelli predefiniti denominati forniti da AWS Glue sono generalmente compatibili con i modelli grok disponibili nel Web.

Crea il tuo pattern grok aggiungendo iterativamente i modelli denominati e controlla i risultati in un debugger. Questa attività garantisce che i dati vengano analizzati quando il crawler AWS Glue esegue il pattern grok.

- Pattern personalizzati

  Per i classificatori grok, questi sono elementi costitutivi facoltativi per il Grok pattern (Pattern grok) che scrivi. Quando i modelli integrati non sono in grado di analizzare i dati, potrebbe essere necessario scrivere un modello personalizzato. Questi modelli personalizzati sono definiti in questo campo e referenziati nel campo Grok pattern (Pattern grok). Ciascun modello personalizzato è definito su una riga separata. Proprio come i modelli integrati, è costituito da una definizione di modello denominato che utilizza la sintassi di espressione regolare (regex).

  Ad esempio, di seguito è riportato il nome `MESSAGEPREFIX` seguito da una definizione di espressione regolare da applicare ai dati per determinare se segue il modello.

  ```
  MESSAGEPREFIX .*-.*-.*-.*-.*
  ```

XML

- Tag di riga

  Per i classificatori XML, questo è il nome del tag XML che definisce una riga di tabella nel documento XML. Digita il nome senza parentesi angolari < >. Il nome deve rispettare le regole XML relative ai tag.

  Per ulteriori informazioni, consulta Scrittura di classificatori personalizzati XML.

JSON

- Percorso JSON

  Per i classificatori JSON, questo è il percorso JSON dell'oggetto, della matrice o del valore che definisce una riga della tabella creata. Digita il nome nella sintassi JSON con punti o parentesi usando gli operatori supportati in AWS Glue.

  Per ulteriori informazioni, vedi l'elenco degli operatori in Scrittura di classificatori personalizzati JSON.

CSV

- Delimitatore di colonna

  Un singolo carattere o simbolo personalizzato per indicare il separatore di ogni voce di colonna nella riga. Scegli il delimitatore dall'elenco o scegli `Other` per immettere un delimitatore personalizzato.

- Simbolo di virgolette

  Un singolo carattere o simbolo personalizzato per indicare la combinazione dei contenuti in un singolo valore di colonna. Deve essere diverso dal delimitatore di colonna. Scegli il simbolo di virgolette dall'elenco o scegli `Other` per immettere delle virgolette personalizzate.

- Intestazioni di colonna

  Indica il comportamento per il modo in cui le intestazioni di colonna devono essere rilevate nel file CSV. È possibile scegliere `Has headings`, `No headings`, oppure `Detect headings`. Se il file CSV personalizzato include le intestazioni di colonna, inserisci un elenco di intestazioni di colonna delimitate da virgole.

- Consenti i file con una singola colonna

  Per essere classificato come CSV, i dati devono avere almeno due colonne e due righe di dati. Utilizza questa opzione per consentire l'elaborazione dei file che contengono una sola colonna.

- Taglia lo spazio vuoto prima dell'identificazione dei valori di colonna

  Questa opzione specifica se tagliare i valori prima di individuare il tipo dei valori di colonna.

- Tipo di dati personalizzato

(Facoltativo) - Inserisci tipi di dati personalizzati in un elenco delimitato da virgole. I tipi di dati supportati sono: "BINARY", "BOOLEAN", "DATE", "DECIMAL", "DOUBLE", "FLOAT", "INT", "LONG", "SHORT", "STRING", "TIMESTAMP".

- SerDe CSV

  (Facoltativo): A SerDe per l'elaborazione del file CSV nel classificatore, che verrà applicato nel Data Catalog. Scegli tra `Open CSV SerDe`, `Lazy Simple SerDe` o `None`. È possibile specificare il valore `None` quando si desidera che il crawler esegua il rilevamento.

Per ulteriori informazioni, consulta Scrittura di classificatori personalizzati per diversi formati di dati.

Visualizzazione dei classificatori

Per visualizzare un elenco di tutti i classificatori creati, apri la console AWS Glue all'indirizzo https://console.aws.amazon.com/glue/ e scegli la scheda Classifiers (Classificatori).

Nell'elenco sono riportate le seguenti proprietà per ogni classificatore:

- Classifier (Classificatore) – Il nome del classificatore. Quando crei un classificatore, devi specificarne il nome.
- Classification (Classificazione) – Il tipo di classificazione delle tabelle dedotte dal classificatore.
- Last updated (Ultimo aggiornamento) – L'ultima volta in cui è stato aggiornato il classificatore.

Gestione dei classificatori

Nell'elenco Classifiers (Classificatori) nella console AWS Glue puoi aggiungere, modificare ed eliminare classificatori. Per visualizzare ulteriori dettagli per un classificatore, scegli il nome nell'elenco. I dettagli sono le informazioni cha hai definito al momento della creazione del classificatore.

## Configurazione di un crawler

Un crawler accede al datastore, identifica i metadati e crea definizioni di tabella nel AWS Glue Data Catalog. Il riquadro Crawler della AWS Glue console elenca tutti i crawler creati. L'elenco mostra stato e parametri dall'ultima esecuzione del crawler.

Questo argomento contiene il step-by-step processo di configurazione di un crawler, trattando aspetti essenziali come l'impostazione dei parametri del crawler, la definizione delle fonti di dati da sottoporre a indicizzazione, l'impostazione della sicurezza e la gestione dei dati sottoposti a indicizzazione.

Argomenti


- Fase 1: Configurazione delle proprietà del crawler
- Fase 2: Scelta delle origini dei dati e dei classificatori
- Fase 3: configurare le impostazioni di sicurezza
- Fase 4: Configurazione dell'output e della pianificazione
- Passaggio 5: revisione e creazione


## Fase 1: Configurazione delle proprietà del crawler

Per configurare un crawler

1. Accedi a Console di gestione AWS e apri la AWS Glue console all'indirizzo https://console.aws.amazon.com/glue/. Nel riquadro di navigazione scegliere Crawler.
2. Scegli Crea crawler e segui le istruzioni della procedura guidata Aggiungi crawler. La procedura guidata ti indicherà i passaggi necessari per creare un crawler. Se vuoi aggiungere classificatori personalizzati per definire lo schema, consulta Definizione e gestione dei classificatori.
3. Inserisci un nome per il crawler e una descrizione (facoltativa). Facoltativamente puoi applicare il tag al crawler tramite una chiave tag e un valore tag opzionale. Una volta create, le chiavi tag sono di sola lettura. Usa i tag su alcune risorse per facilitarne l'organizzazione e l'individuazione. Per ulteriori informazioni, vedi AWS tags in AWS Glue.

   Name

   Il nome può contenere un massimo di 255 caratteri, tra cui lettere (A-Z), numeri (0-9), trattini (-) o caratteri di sottolineatura (_).

   Description

   La descrizione può contenere un massimo di 2048 caratteri.

   Tag

   Utilizza i tag per organizzare e identificare le risorse. Per ulteriori informazioni, consulta gli argomenti seguenti:

- AWS tag in AWS Glue

Fase 2: Scelta delle origini dei dati e dei classificatori

Quindi, configura le origini dati e i classificatori per il crawler.

Per ulteriori informazioni sulle origini dati supportate, consulta Origini dati supportate per il crawling.

Configurazione dell'origine dati

Seleziona l'opzione appropriata per I tuoi dati sono già mappati alle tabelle AWS Glue? scegli 'Non ancora' or 'Sì'. Per impostazione predefinita, è selezionata la risposta "Non ancora".

Il crawler è in grado di accedere agli archivi dati direttamente come origine del crawling oppure utilizza le tabelle di catalogo esistenti come origine. Se il crawler utilizza le tabelle di catalogo esistenti, esegue il crawling degli archivi dati specificati dalle tabelle di catalogo.

- Non ancora: seleziona una o più origini dati da sottoporre a crawling. Un crawler è in grado di eseguire il crawling di più archivi dati di diversi tipi (Amazon S3, JDBC e così via).

  È possibile configurare un solo archivio dati alla volta. Dopo aver fornito le informazioni di connessione e incluso percorsi ed escluso i modelli, è possibile aggiungere un altro archivio dati.
- Sì: seleziona le tabelle esistenti dal tuo catalogo dati AWS Glue. Le tabelle del catalogo specificano gli archivi dati da sottoporre a crawling. In una singola esecuzione, il crawler è in grado di eseguire il crawling solo di tabelle di catalogo; non può combinare altri tipi di origine.

  Un motivo comune per specificare una tabella di catalogo come origine è quando la tabella viene creata manualmente (poiché si conosce già la struttura dell'archivio dati) e desideri che un crawler mantenga aggiornata la tabella, inclusa l'aggiunta di nuove partizioni. Per una discussione degli altri motivi, consulta Aggiornamento delle tabelle del catalogo dati create manualmente usando i crawler.

  Quando specifichi le tabelle esistenti come tipo di origine crawler, si applicano le seguenti condizioni:

  - Il nome del database è facoltativo.
  - Solo le tabelle di catalogo che specificano archivi dati Amazon S3, Amazon DynamoDB o Delta Lake sono consentite.

- Nessuna nuova tabella di catalogo viene creata quando viene eseguito il crawler. Le tabelle esistenti vengono aggiornate in base alle esigenze, inclusa l'aggiunta di nuove partizioni.
- Gli oggetti eliminati trovati negli archivi dati vengono ignorati e non vengono eliminate le tabelle di catalogo. Al contrario, il crawler scrive un messaggio di log. (`SchemaChangePolicy.DeleteBehavior=LOG`)
- L'opzione di configurazione del crawler per creare un singolo schema per ogni percorso Amazon S3 è abilitata per impostazione predefinita e non può essere disattivata. (`TableGroupingPolicy=CombineCompatibleSchemas`) Per ulteriori informazioni, consulta Creare un singolo schema per ogni percorso di inclusione Amazon S3
- Non è possibile combinare le tabelle di catalogo come origine con altri tipi di origine (ad esempio Amazon S3 o Amazon DynamoDB).

Per utilizzare le tabelle Delta, crea prima una tabella Delta utilizzando Athena DDL o l'API. AWS Glue

Usando Athena, imposta la posizione sulla cartella Amazon S3 e il tipo di tabella su "DELTA".

```
CREATE EXTERNAL TABLE database_name.table_name
LOCATION 's3://bucket/folder/'
TBLPROPERTIES ('table_type' = 'DELTA')
```

Utilizzando l' AWS Glue API, specifica il tipo di tabella all'interno della mappa dei parametri della tabella. I parametri della tabella devono includere la key/value coppia seguente. Per maggiori informazioni su come creare una tabella, consulta la documentazione Boto3 per create_table.

```
{
    "table_type":"delta"
}
```

Origini dati

Seleziona o aggiungi l'elenco di origini dati che devono essere scansionate dal crawler.

(Facoltativo) Se scegli JDBC come origine dati, puoi utilizzare i tuoi driver JDBC per specificare l'accesso alla connessione in cui sono archiviate le informazioni sul driver.

Percorso di inclusione

Per stabilire cosa includere o escludere dal crawling, il crawler inizia la valutazione del percorso di inclusione richiesto. Per gli archivi dati Amazon S3, MongoDB, MongoDB Atlas, Amazon

DocumentDB (compatibile con MongoDB) e relazionali, è necessario specificare un percorso di inclusione.

Per un archivio dati Amazon S3

Scegli se specificare un percorso in questo account o in uno differente, quindi seleziona un percorso Amazon S3.

Per gli archivi dati Amazon S3, la sintassi del percorso di inclusione è `bucket-name/folder-name/file-name.ext`. Per eseguire il crawling di tutti gli oggetti in un bucket, devi specificare solo il nome del bucket nel percorso di inclusione. Il modello di esclusione è relativo rispetto al percorso di inclusione

Per un archivio dati Delta Lake

Specificare uno o più percorsi Amazon S3 alle tabelle Delta come s3:////*bucket*. *prefix object*

Per un datastore Iceberg o Hudi

Specificare uno o più percorsi Amazon S3 che contengono cartelle con metadati delle tabelle Iceberg o Hudi come s3:///. *bucket prefix*

Per gli archivi dati Iceberg e Hudi, la Iceberg/Hudi cartella può trovarsi in una cartella secondaria della cartella principale. Il crawler scansionerà tutte le cartelle al di sotto del percorso di una cartella Hudi.

Per un archivio dati JDBC

Immettere*<database>*/*<schema>*/*<table>*o*<database>*/*<table>*, a seconda del prodotto del database. Oracle Database e MySQL non supportano lo schema nel percorso. È possibile sostituire *<schema>* o *<table>* con il carattere percentuale (%). Ad esempio, per un database Oracle con un identificatore di sistema (SID) di `orcl`, immettere `orcl/%` per importare tutte le tabelle a cui può accedere l'utente denominato nella connessione.

> **Important**
>
> Questo campo fa distinzione tra minuscole e maiuscole.

> **Note**
>
> Se scegli di importare le tue versioni dei driver JDBC, i crawler AWS Glue consumeranno risorse nei processi AWS Glue e nei bucket Amazon S3 per garantire che i driver forniti vengano eseguiti nel tuo ambiente. L'utilizzo aggiuntivo delle risorse si rifletterà nel tuo account. I driver sono limitati alle proprietà descritte nella sezione Adding an AWS Glue connection.

Per un archivio dati MongoDB, MongoDB Atlas o Amazon DocumentDB

Per MongoDB, MongoDB Atlas e Amazon DocumentDB (compatibile con MongoDB), la sintassi è `database/collection`.

Per gli archivi dati JDBC, la sintassi è `database-name/schema-name/table-name` oppure `database-name/table-name`. La sintassi dipende dal fatto che il motore di database supporti schemi all'interno di un database. Ad esempio, per i motori di database come MySQL o Oracle, non è necessario specificare un `schema-name` nel percorso di inclusione. È possibile sostituire il segno di percentuale (`%`) per uno schema o una tabella nel percorso di inclusione per rappresentare tutti gli schemi o tutte le tabelle all'interno di un database. Non è possibile sostituire il segno di percentuale (`%`) per il database nel percorso di inclusione.

Profondità trasversale massima (solo per i datastore Iceberg o Hudi)

Definisce la profondità massima del percorso Amazon S3 che il crawler può percorrere per scoprire la cartella di metadati Iceberg o Hudi nel percorso Amazon S3. Lo scopo di questo parametro è limitare il tempo di esecuzione del crawler. Il valore predefinito è 10, il valore massimo è 20.

Modelli di esclusione

Consentono di escludere alcuni file o tabelle dal crawling. Il percorso di esclusione è relativo rispetto al percorso di inclusione. Ad esempio, per escludere una tabella nel tuo archivio dati JDBC, digita il nome della tabella nel percorso di esclusione.

Un crawler si connette a un archivio dati JDBC usando una connessione AWS Glue che contiene una stringa di connessione URI JDBC. Il crawler ha accesso a oggetti nel motore di database solo usando il nome utente e la password JDBC nella connessione AWS Glue. Il crawler può creare solo le tabelle cui può accedere tramite la connessione JDBC. Dopo che il crawler accede al motore del database con l'URI JDBC, viene usato il percorso di inclusione per determinare quali

tabelle del motore del database vengono create nel catalogo dati. Ad esempio, se con MySQL specifichi il percorso di inclusione `MyDatabase/%`, nel catalogo dati verranno create tutte le tabelle presenti in `MyDatabase`. Se quando accedi ad Amazon Redshift specifichi il percorso di inclusione di `MyDatabase/%`, nel catalogo dati vengono create tutte le tabelle in tutti gli schemi per il database `MyDatabase`. Se si specifica un percorso di inclusione di `MyDatabase/MySchema/%`, vengono create tutte le tabelle nel database `MyDatabase` e lo schema `MySchema`.

Dopo aver specificato un percorso di inclusione, puoi quindi escludere dal crawling oggetti che il percorso di inclusione altrimenti includerebbe, specificando uno o più modelli di esclusione `glob` in stile Unix. Questi modelli vengono applicati al tuo percorso di inclusione per determinare quali oggetti sono esclusi. Questi modelli vengono inoltre memorizzati come proprietà delle tabelle create dal crawler. AWS Glue PySpark le estensioni, ad esempio`create_dynamic_frame.from_catalog`, leggono le proprietà della tabella ed escludono gli oggetti definiti dal modello di esclusione.

AWS Glue supporta i seguenti modelli `glob` nel modello di esclusione.

| Modello di esclusione | Description |
| --- | --- |
| `*.csv` | Individua un percorso Amazon S3 che rappresenta un nome di oggetto nella cartella corrente che termina con `.csv` |
| `*.*` | Individua tutti i nomi degli oggetti che contengono un punto |
| `*.{csv,avro}` | Individua i nomi di oggetti che terminano con `.csv` o `.avro` |
| `foo.?` | Individua i nomi degli oggetti che iniziano con `foo.` che sono seguiti da un'estensione di un singolo carattere. |
| `myfolder/*` | Individua gli oggetti in un livello di sottocartella di `myfolder`, ad esempio `/myfolder/mysource` |

| Modello di esclusione | Description |
| --- | --- |
| `myfolder/*/*` | Individua gli oggetti in due livelli di sottocartella di `myfolder`, ad esempio `/myfolder/mysource/data` |
| `myfolder/**` | Individua gli oggetti in tutte le sottocartelle di `myfolder`, ad esempio `/myfolder/mysource/mydata` e `/myfolder/mysource/data` |
| `myfolder**` | Individua la sottocartella `myfolder`, insieme ai file all'interno di `myfolder`, ad esempio `/myfolder` e `/myfolder/mydata.txt` |
| `Market*` | Individua le tabelle in un database JDBC con i nomi che iniziano con `Market`, quali `Market_us` e `Market_fr` |

AWS Glue interpreta i modelli di esclusione `glob` in questo modo:

- La barra (`/`) è il delimitatore per separare chiavi Amazon S3 in una gerarchia di cartelle.
- L'asterisco (`*`) individua zero o più caratteri di un componente del nome entro i limiti della cartella.
- Un doppio asterisco (`**`) individua zero o più caratteri al di fuori della cartella o dello schema.
- Il punto interrogativo (`?`) individua esattamente un carattere di un componente del nome.
- La barra rovesciata (`\`) è utilizzata per ignorare i caratteri che potrebbero essere interpretati come caratteri speciali. L'espressione `\\` individua una singola barra rovesciata, mentre `\{` a una parentesi aperta.
- Le parentesi `[ ]` creano un'espressione tra parentesi che individua un singolo carattere di un componente nome fuori da un set di caratteri. Ad esempio, `[abc]` individua `a`, `b` o `c`. Il trattino (`-`) può essere utilizzato per specificare un intervallo, quindi `[a-z]` specifica un intervallo `a` – `z` (estremi inclusi). Questi moduli possono essere mescolati, perciò `[abce-g]` individua `a`, `b`, `c`, `e`, `f` o `g`. Se il carattere dopo la parentesi (`[`) è un punto esclamativo (`!`), l'espressione all'interno della parentesi viene ignorata. Ad esempio, `[!a-c]` corrisponde a tutti i caratteri, tranne `a`, `b` o `c`.

All'interno di un'espressione tra parentesi, i caratteri `*`, `?` e `\` corrispondono. Il trattino (`-`) corrisponde a se stesso se è il primo carattere all'interno della parentesi oppure se è il primo carattere dopo il `!` quando l'espressione viene ignorata.

- Le parentesi graffe (`{ }`) racchiudono un gruppo di sotto-modelli, dove il gruppo individua un eventuale sotto-modello corrispondente all'interno del gruppo. Una virgola (`,`) è utilizzata per separare i sotto-modelli. I gruppi non possono essere annidati.
- Il punto iniziale o i punti nei nomi di file vengono trattati come caratteri normali nelle operazioni di confronto. Ad esempio, il modello di esclusione `*` corrisponde al nome di file `.hidden`.

Example Modelli di esclusione Amazon S3

Ogni modello di esclusione è valutato rispetto al percorso di inclusione. Ad esempio, supponiamo di avere la seguente struttura di directory Amazon S3:

```
/mybucket/myfolder/
   departments/
      finance.json
      market-us.json
      market-emea.json
      market-ap.json
   employees/
      hr.json
      john.csv
      jane.csv
      juan.txt
```

Dato il percorso di inclusione `s3://mybucket/myfolder/`, i seguenti sono esempi di risultati per i modelli di esclusione:

| Modello di esclusione | Risultati |
|---|---|
| `departments/**` | Esclude tutti i file e le cartelle sotto `departments` e include la cartella `employees` e i relativi file |
| `departments/market*` | Esclude `market-us.json` , `market-emea.json` e `market-ap.json` |

| Modello di esclusione | Risultati |
| --- | --- |
| `**.csv` | Esclude tutti gli oggetti sotto `myfolder` che hanno un nome che termina con `.csv` |
| `employees/*.csv` | Esclude tutti i file `.csv` nella cartella `employees` |

Example Esclusione di un sottoinsieme di partizioni Amazon S3

Supponiamo che i dati vengano partizionati in base al giorno, in modo che ogni giorno di un anno si trovi in una partizione Amazon S3 separata. Per gennaio 2015, ci sono 31 partizioni. Ora, per eseguire il crawling solo per i dati della prima settimana di gennaio, devi escludere tutte le partizioni eccetto i giorni da 1 a 7:

```
2015/01/{[!0],0[8-9]}**, 2015/0[2-9]/**, 2015/1[0-2]/**
```

Diamo uno sguardo alle parti di questo modello glob. La prima parte, `2015/01/{[!0],0[8-9]}**`, esclude tutti i giorni che non iniziano con "0" oltre ai giorni 08 e 09 dal mese 01 nel 2015. Tieni presente che "**" viene utilizzato come suffisso per il modello del numero del giorno e si estende alle cartelle di livello inferiore. Se viene utilizzato "*", i livelli di cartella inferiori non sono esclusi.

La seconda parte, `2015/0[2-9]/**`, esclude i giorni nei mesi da 02 a 09, nel 2015.

La terza parte, `2015/1[0-2]/**`, esclude i giorni nei mesi 10, 11 e 12, nel 2015.

Example Modelli di esclusione JDBC

Supponiamo di eseguire il crawling di un database JDBC con la seguente struttura di schema:

```
MyDatabase/MySchema/
   HR_us
   HR_fr
   Employees_Table
   Finance
   Market_US_Table
   Market_EMEA_Table
```

```
Market_AP_Table
```

Dato il percorso di inclusione `MyDatabase/MySchema/%`, i seguenti sono esempi di risultati per i modelli di esclusione:

| Modello di esclusione | Risultati |
| --- | --- |
| `HR*` | Esclude le tabelle con nomi che iniziano con `HR` |
| `Market_*` | Esclude le tabelle con nomi che iniziano con `Market_` |
| `**_Table` | Esclude tutte le tabelle con nomi che terminano con `_Table` |

Ulteriori parametri per l'origine del crawler

Ogni tipo di origine richiede un diverso set di parametri aggiuntivi.

Connessione

Selezionare o aggiungere una connessione AWS Glue. Per informazioni sulle connessioni, consulta Connessione ai dati.

Metadati aggiuntivi: facoltativi (per gli archivi di dati JDBC)

Seleziona proprietà di metadati aggiuntive per il crawler da sottoporre a crawling.

- Commenti: esegui il crawling dei commenti associati a livello di tabella e colonna.
- Tipi non elaborati: mantengono i tipi di dati non elaborati delle colonne della tabella in metadati aggiuntivi. Come comportamento predefinito, il crawler traduce i tipi di dati non elaborati in tipi compatibili con Hive.

Nome della classe del driver JDBC: facoltativo (per i datastore JDBC)

Digita un nome di classe del driver JDBC personalizzato per consentire al crawler di connettersi all'origine dati:

- Postgres: org.postgresql.Driver
- MySQL: com.mysql.jdbc.Driver, com.mysql.cj.jdbc.Driver

- Redshift: com.amazon.redshift.jdbc.Driver, com.amazon.redshift.jdbc42.Driver
- Oracle: oracle.jdbc.driver. OracleDriver
- SQL Server: com.microsoft.sqlserver.jdbc. SQLServerAutista

Percorso S3 del driver JDBC: facoltativo (per i datastore JDBC)

Scegli un percorso Amazon S3 esistente verso un file `.jar`. Questa è la posizione in cui verrà archiviato il file `.jar` quando si utilizza un driver JDBC personalizzato per la connessione del crawler all'origine dati.

Abilitazione del campionamento dei dati (solo per gli archivi dati Amazon DynamoDB, MongoDB, MongoDB Atlas e Amazon DocumentDB)

Selezionare se eseguire il crawling solo di un campione di dati. Se non è selezionata, l'intera tabella viene sottoposta a crawling. La scansione di tutti i registri può richiedere molto tempo quando la tabella non è una tabella di throughput elevato.

Creazione di tabelle per l'esecuzione di query (solo per gli archivi dati Delta Lake)

Seleziona come desideri creare le tabelle Delta Lake:

- Creazione di tabelle native: consente l'integrazione con i motori di query che supportano l'interrogazione diretta del log delle transazioni Delta.
- Creazione di tabelle Symlink: crea una cartella dei manifesti Symlink con file manifesti partizionati con chiavi di partizioni in base ai parametri di configurazione specificati.

Velocità di scansione (facoltativo, solo per i datastore DynamoDB)

Specifica la percentuale delle unità di capacità di lettura della tabella DynamoDB che deve essere utilizzata dal crawler. L'unità di capacità di lettura è un termine definito da DynamoDB ed è un valore numerico che funge da limitatore di velocità per il numero di letture che possono essere eseguite su tale tabella al secondo. Inserire un valore tra 0.1 e 1.5. Se non specificato, il valore predefinito è 0,5 per le tabelle con provisioning e 1/4 della capacità massima configurata per le tabelle on demand. Tieni presente che con i AWS Glue crawler deve essere utilizzata solo la modalità di capacità assegnata.

> **Note**
>
> Per gli archivi dati DynamoDB, impostare la modalità di capacità assegnata per l'elaborazione delle letture e delle scritture sulle tabelle. Il AWS Glue crawler non deve essere utilizzato con la modalità di capacità su richiesta.

Connessione di rete: facoltativa (per i datastore di destinazione Amazon S3, Delta, Iceberg, Hudi e Catalog)

Facoltativamente, includi una connessione di rete da utilizzare con questa destinazione Amazon S3. Tieni presente che ogni crawler è limitato a una connessione di rete, quindi anche qualsiasi altra destinazione Amazon S3 utilizzerà la stessa connessione (o nessuna, se lasciata vuota).

Per informazioni sulle connessioni, consulta Connessione ai dati.

Campionamento di un solo sottoinsieme di file e dimensioni del campione (solo per gli archivi dati Amazon S3)

Specificare il numero di file in ogni cartella foglia da sottoporre al crawling durante il crawling di file di esempio in un set di dati. Quando questa caratteristica è attivata, invece di eseguire il crawling di tutti i file in questo set di dati, il crawler seleziona in modo casuale alcuni file in ogni cartella foglia da sottoporre a crawling.

Il crawler di campionamento è adatto ai clienti che hanno conoscono i loro formati di dati e sanno che gli schemi nelle loro cartelle non cambiano. L'attivazione di questa funzione ridurrà significativamente il tempo di esecuzione del crawler.

Un valore valido è un numero intero compreso tra 1 e 249. Se non è specificato, tutti i file vengono sottoposti al crawling.

Esecuzioni successive del crawler

Questo campo è un campo globale che riguarda tutte le origini dati Amazon S3.

- Esecuzione del crawling di tutte le sottocartelle: esegui nuovamente il crawling di tutte le cartelle ad ogni crawling successivo.
- Esecuzione del crawling solo delle nuove sottocartelle: verrà eseguito il crawling solo delle cartelle Amazon S3 che sono state aggiunte dall'ultimo crawling. Se gli schemi sono compatibili, verranno aggiunte nuove partizioni alle tabelle esistenti. Per ulteriori informazioni, consulta the section called "Pianificazione di crawling incrementale".
- Crawling in base agli eventi: basati sugli eventi di Amazon S3 per controllare quali cartelle sottoporre a crawling. Per ulteriori informazioni, consulta Accelerazione del crawling con le notifiche eventi Amazon S3.

Classificatori personalizzati: facoltativi

Definisci i classificatori personalizzati prima di definire i crawler. Un classificatore verifica se un determinato file è in un formato che può essere gestito dal crawler. In questo caso il classificatore crea uno schema nel formato di un oggetto `StructType` che corrisponde a quel formato di dati.

Per ulteriori informazioni, consulta Definizione e gestione dei classificatori.

Fase 3: configurare le impostazioni di sicurezza

Ruolo IAM

Il crawler assume questo ruolo. Deve avere autorizzazioni simili alla policy AWS `AWSGlueServiceRole` gestita. Per le origini Amazon S3 e DynamoDB, deve disporre anche delle autorizzazioni per accedere all'archivio dati. Se il crawler legge i dati di Amazon S3 crittografati con AWS Key Management Service (AWS KMS), il ruolo deve disporre delle autorizzazioni di decrittografia sulla chiave. AWS KMS

Per un archivio dati Amazon S3, autorizzazioni aggiuntive collegate al ruolo saranno simili alle seguenti:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::bucket/object*"
      ]
    }
  ]
}
```

Per un archivio dati Amazon DynamoDB, autorizzazioni aggiuntive collegate al ruolo saranno simili alle seguenti:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
```

```
      "Effect": "Allow",
      "Action": [
        "dynamodb:DescribeTable",
        "dynamodb:Scan"
      ],
      "Resource": [
        "arn:aws:dynamodb:*:111122223333:table/table-name*"
      ]
    }
  ]
}
```

Per aggiungere il proprio driver JDBC, è necessario aggiungere ulteriori autorizzazioni.

- Concedi le autorizzazioni per le seguenti operazioni di processo: `CreateJob`, `DeleteJob`, `GetJob`, `GetJobRun`, `StartJobRun`.
- Concedi le autorizzazioni per le operazioni di Amazon S3: `s3:DeleteObjects`, `s3:GetObject`, `s3:ListBucket`, `s3:PutObject`.

> **Note**
>
> Il valore `s3:ListBucket` non è necessario se la policy del bucket Amazon S3 è disabilitata.

- Concedi l'accesso principale al servizio bucket/folder nella policy di Amazon S3.

Esempio di policy Amazon S3:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObject",
                "s3:ListBucket",
                "s3:DeleteObject"
            ],
```

```
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket/driver-parent-folder/
driver.jar",
                "arn:aws:s3:::amzn-s3-demo-bucket"
            ]
        }
    ]
}
```

AWS Glue crea le seguenti cartelle (`_crawler` e `_glue_job_crawler` allo stesso livello del driver JDBC) nel tuo bucket Amazon S3. Ad esempio, se il percorso del driver è `<s3-path/driver_folder/driver.jar>`, verranno create le seguenti cartelle, se non esistono ancora:

- <s3-path/driver_folder/_crawler>
- <s3-path/driver_folder/_glue_job_crawler>

Puoi facoltativamente aggiungere una configurazione di sicurezza a un crawler per specificare opzioni di crittografia dei dati inattivi.

Per ulteriori informazioni, consultare Fase 2: creare un ruolo IAM per AWS Glue e Gestione delle identità e degli accessi per AWS Glue.

Configurazione Lake Formation: facoltativa

Consenti al crawler di utilizzare le credenziali di Lake Formation per il crawling dell'origine dati.

Selezionando Use Lake Formation credentials for crawling S3 data source (Usa le credenziali di Lake Formation per il crawling dell'origine dati S3), il crawler potrà utilizzare le credenziali di Lake Formation per il crawling dell'origine dati. Se l'origine appartiene a un altro account, è necessario fornire l'ID dell'account registrato. Altrimenti, il crawler eseguirà il crawling solo delle origini dati associate all'account. Applicabile solo a origini dati Amazon S3 e del catalogo dati.

Configurazione di sicurezza: facoltativa

Le impostazioni includono le configurazioni di sicurezza. Per ulteriori informazioni, consulta gli argomenti seguenti:

- Crittografia dei dati scritti da AWS Glue

> **Note**
>
> Una volta impostata una configurazione di sicurezza su un crawler, puoi modificarla, ma non rimuoverla. Per abbassare il livello di sicurezza su un crawler, imposta esplicitamente

la funzionalità di sicurezza su `DISABLED` nella configurazione oppure crea un nuovo crawler.

Fase 4: Configurazione dell'output e della pianificazione

Configurazione dell'uscita

Le opzioni includono in che modo il crawler deve gestire le modifiche dello schema rilevate, gli oggetti eliminati nell'archivio dati e altro. Per ulteriori informazioni, consulta Personalizzare il comportamento del crawler

Pianificazione del crawler

Puoi definire un crawler on demand oppure definire una pianificazione temporale per i crawler e i processi in AWS Glue. La definizione di queste pianificazioni usa la sintassi cron di tipo Unix. Per ulteriori informazioni, consulta Pianificazione di un crawler.

Passaggio 5: revisione e creazione

Controlla le impostazioni del crawler configurate e crea il crawler.

## Pianificazione di un crawler

Puoi eseguire un crawler AWS Glue on demand o in base a una pianificazione regolare. Quando si imposta un crawler basato su una pianificazione, è possibile specificare alcuni vincoli, come la frequenza, i giorni della settimana e l'orario di esecuzione del crawler. Si possono creare queste pianificazioni personalizzate in formato cron. Per ulteriori informazioni, consulta cron in Wikipedia.

Quando si configura la pianificazione di un crawler, devi tener conto delle caratteristiche e delle limitazioni del cron. Ad esempio, se vuoi eseguire il crawler il giorno 31 di ogni mese, devi ricordare che alcuni mesi non sono di 31 giorni.

Argomenti

- Creazione di una pianificazione del crawler
- Creare una pianificazione per un crawler esistente

Creazione di una pianificazione del crawler

Puoi creare una pianificazione per il crawler utilizzando la AWS Glue console o. AWS CLI

Console di gestione AWS

1. Accedi a Console di gestione AWS, e apri la AWS Glue console all'indirizzo. https://console.aws.amazon.com/glue/
2. Nel riquadro di navigazione scegliere Crawler.
3. Segui i passaggi da 1 a 3 nella sezione Configurazione di un crawler.
4. In Fase 4: Configurazione dell'output e della pianificazione, scegli una pianificazione del Crawler per impostare la frequenza di esecuzione. Puoi scegliere di eseguire il crawler ogni ora, ogni giorno, ogni settimana, ogni mese o definire una pianificazione personalizzata utilizzando espressioni Cron.

   Un'espressione Cron è una stringa che rappresenta un modello di pianificazione, ed è composta da 6 campi separati da spazi: * * * * * <minuto> <ora> <giorno del mese> <mese> <giorno della settimana> <anno>

   Ad esempio, per eseguire un'attività ogni giorno a mezzanotte, l'espressione Cron è: 0 0 * * ? *

   Per ulteriori informazioni consulta Espressioni Cron.
5. Rivedi le impostazioni del crawler che hai configurato e crea il crawler da eseguire secondo una pianificazione.

AWS CLI

```
aws glue create-crawler
 --name myCrawler \
 --role AWSGlueServiceRole-myCrawler  \
 --targets '{"S3Targets":[{Path="s3://amzn-s3-demo-bucket/"}]}' \
 --schedule cron(15 12 * * ? *)
```

Per ulteriori informazioni sull'utilizzo di cron per pianificare processi e crawler, consulta Pianificazioni basate sul tempo per processi e crawler.

Creare una pianificazione per un crawler esistente

Completare la procedura riportata di seguito per impostare una pianificazione ricorrente per un crawler esistente.

Console di gestione AWS

1. Accedi a Console di gestione AWS e apri la AWS Glue console all'indirizzo https://console.aws.amazon.com/glue/.
2. Nel riquadro di navigazione scegliere Crawler.
3. Scegliere un crawler da pianificare dall'elenco disponibile.
4. Scegli Modifica dal menu Azioni.
5. Scorrere in basso fino alla Fase 4: Configurare l'output e la pianificazione, e scegliere Modifica.
6. Aggiornare la pianificazione del crawler nella sezione Pianificazione del crawler.
7. Scegliere Aggiorna.

AWS CLI

Utilizzare il seguente comando dell'interfaccia a riga di comando (CLI) per aggiornare la configurazione del crawler esistente:

```
aws glue update-crawler-schedule
   --crawler-name myCrawler
   --schedule cron(15 12 * * ? *)
```

## Visualizzazione dei risultati e dei dettagli del crawler

Dopo l'esecuzione corretta del crawler, questo crea le definizioni di tabella nel catalogo dati. Scegli Tables (Tabelle) nel pannello di navigazione per visualizzare le tabelle create dal crawler nel database specificato.

È possibile visualizzare le informazioni relative al crawler stesso nel modo seguente:

- La pagina Crawler sulla AWS Glue console mostra le seguenti proprietà per un crawler:

| Proprietà | Description |
| --- | --- |
| Nome | Quando crei un crawler, devi assegnargli un nome univoco. |

| Proprietà | Description |
| --- | --- |
| Stato | Un crawler può essere pronto, in fase avvio, in fase di arresto, pianificato o con pianifica zione in pausa. Un crawler in esecuzione avanza dall'avvio all'arresto. Puoi riprendere o sospendere una pianificazione collegata a un crawler. |
| Pianificazione | Puoi scegliere di eseguire il tuo crawler on demand oppure scegliere una frequenza mediante una pianificazione. Per ulteriori informazioni sulla pianificazione di un crawler, consulta Pianificazione di un crawler. |
| Ultima esecuzione | Data e ora dell'ultima esecuzione del crawler. |
| Log | Link ai log disponibili dall'ultima esecuzione del crawler. |
| Modifiche alle tabelle rispetto all'ultima esecuzione | Il numero di tabelle del crawler AWS Glue Data Catalog che sono state aggiornate dall'ultima esecuzione del crawler. |

- Per visualizzare la cronologia di un crawler, scegli Crawlers (Crawler) nel pannello di navigazione per visualizzare i crawler creati. Scegli un crawler dall'elenco dei crawler disponibili. Puoi visualizzare le proprietà e la cronologia del crawler nella scheda Crawler runs (Esecuzioni del crawler).

  La scheda Crawler runs (Esecuzioni del crawler) mostra le informazioni relative a ogni esecuzione del crawler, tra cui Start time (UTC) (Ora di inizio [UTC]), End time (UTC) (Ora di fine [UTC]), Duration (Durata), Status (Stato), DPU hours (Ore DPU) e Table changes (Modifiche alla tabella).

  La scheda Esecuzioni del crawler riporta solo i crawling che si sono verificati dalla data di avvio della funzionalità di cronologia del crawler e conserva solo fino a 12 mesi di crawling. I crawling più vecchi non verranno restituiti.
- Per visualizzare le informazioni aggiuntive, scegli una scheda nella pagina dei dettagli del crawler. Ogni scheda mostrerà le informazioni relative al crawler.

- Schedule (Pianificazione): tutte le pianificazioni create per il crawler saranno visibili qui.
- Data sources (Origini dei dati): tutte le origini dei dati scansionate dal crawler saranno visibili qui.
- Classifiers (Classificatori): tutti i classificatori assegnati al crawler saranno visibili qui.
- Tag: tutti i tag creati e assegnati a una AWS risorsa saranno visibili qui.

Parametri impostati sulle tabelle del catalogo dati dal crawler

Queste proprietà della tabella sono impostate dai AWS Glue crawler. Ci aspettiamo che gli utenti utilizzino `compressionType` e le proprietà `classification`. Altre proprietà, comprese le stime delle dimensioni delle tabelle, vengono utilizzate per i calcoli interni e non garantiamo la loro accuratezza o applicabilità ai casi d'uso dei clienti. La modifica di questi parametri può alterare il comportamento del crawler, non supportiamo questo flusso di lavoro.

| Chiave di proprietà | Valore proprietà |
| --- | --- |
| `UPDATED_BY_CRAWLER` | Nome del crawler che esegue l'aggiornamento. |
| `connectionName` | Il nome della connessione nel catalogo di dati per il crawler utilizzato per connettersi all'archivio dati. |
| `recordCount` | Stima del numero di registri nella tabella, in base alle dimensioni e alle intestazioni dei file. |
| `skip.header.line.count` | Righe saltate per saltare l'intestazione. Impostato su tabelle classificate come CSV. |
| `CrawlerSchemaSerializerVersion` | Per uso interno |
| `classification` | Formato dei dati, dedotto dal crawler. Per ulteriori informazioni sui formati di dati supportati dai AWS Glue crawler, consulta. the section called "Classificatori integrati" |

| Chiave di proprietà | Valore proprietà |
| --- | --- |
| `CrawlerSchemaDeserializerVersion` | Per uso interno |
| `sizeKey` | Dimensione combinata dei file nella tabella sottoposti a scansione. |
| `averageRecordSize` | La dimensione media della riga nella tabella in byte. |
| `compressionType` | Tipo di compressione utilizzato sui dati della tabella. Per ulteriori informazioni sui tipi di compressione supportati dai AWS Glue crawler, consultate. the section called "Classificatori integrati" |
| `typeOfData` | `file`, `table`, oppure `view`. |
| `objectCount` | Numero di oggetti nel percorso Amazon S3 per la tabella. |

Queste proprietà aggiuntive delle tabelle vengono impostate dai AWS Glue crawler per gli archivi dati Snowflake.

| Chiave di proprietà | Valore proprietà |
| --- | --- |
| `aws:RawTableLastAltered` | Registra l'ultimo timestamp modificato della tabella Snowflake. |
| `ViewOriginalText` | Visualizza l'istruzione SQL. |
| `ViewExpandedText` | Visualizza l'istruzione SQL codificata nel formato Base64. |
| `ExternalTable:S3Location` | La posizione di Amazon S3 della tabella esterna Snowflake. |

| Chiave di proprietà | Valore proprietà |
| --- | --- |
| `ExternalTable:FileFormat` | Formato di file Amazon S3 della tabella esterna Snowflake. |

Queste proprietà di tabella aggiuntive vengono impostate dai AWS Glue crawler per archivi dati di tipo JDBC come Amazon Redshift, Microsoft SQL Server, MySQL, PostgreSQL e Oracle.

| Chiave di proprietà | Valore proprietà |
| --- | --- |
| `aws:RawType` | Quando un crawler archivia i dati nel catalogo dati, traduce i tipi di dati in tipi compatibili con Hive, il che spesso causa la perdita delle informazioni sul tipo di dati nativo. Il crawler emette il parametro `aws:RawType` per fornire il tipo di dati a livello nativo. |
| `aws:RawColumnComment` | Se un commento è associato a una colonna del database, il crawler emette il commento corrispondente nella tabella del catalogo. La stringa di commento viene troncata a 255 byte.<br><br>I commenti non sono supportati per Microsoft SQL Server. |
| `aws:RawTableComment` | Se un commento è associato a una tabella nel database, il crawler emette il commento corrispondente nella tabella del catalogo. La stringa di commento viene troncata a 255 byte.<br><br>I commenti non sono supportati per Microsoft SQL Server. |

## Personalizzare il comportamento del crawler

Quando configuri un Crawler di AWS Glue, hai diverse opzioni per definire il comportamento del tuo crawler.

- Crawling incrementali: puoi configurare un crawler per eseguire crawling incrementali al fine di aggiungere solo nuove partizioni allo schema della tabella.
- Indici di partizione: un crawler crea indici delle partizioni per le destinazioni Amazon S3 e Delta Lake per impostazione predefinita, al fine di fornire una ricerca efficiente per partizioni specifiche.

- Accelerare il tempo di crawling utilizzando gli eventi Amazon S3: puoi configurare un crawler in modo che utilizzi gli eventi Amazon S3 per identificare le modifiche tra due crawling elencando tutti i file della sottocartella che ha attivato l'evento invece di elencare l'intera destinazione Amazon S3 o catalogo dati.
- Gestione delle modifiche allo schema: puoi impedire a un crawler di apportare modifiche allo schema esistente. Puoi utilizzare l'API Console di gestione AWS o l'AWS GlueAPI per configurare il modo in cui il crawler elabora determinati tipi di modifiche.
- Uno schema unico per più percorsi Amazon S3: puoi configurare un crawler in modo da creare uno schema unico per ciascun percorso S3, se i dati sono compatibili.
- Posizione della tabella e livello di partizionamento: l'opzione crawler a livello di tabella offre la flessibilità necessaria per indicare al crawler dove si trovano le tabelle e come si desidera creare le partizioni.
- Soglia della tabella: puoi specificare il numero massimo di tabelle che il crawler è autorizzato a creare specificando una soglia per le tabelle.
- AWS Lake Formation credenziali: puoi configurare un crawler per utilizzare le credenziali di Lake Formation per accedere a un data store Amazon S3 o a una tabella Data Catalog con una posizione Amazon S3 sottostante all'interno della stessa o di un'altra. Account AWS Account AWS

Per ulteriori informazioni sull'uso della console AWS Glue per aggiungere un crawler, consulta Configurazione di un crawler.

Argomenti

- Pianificazione di crawling incrementale per l'aggiunta di nuove partizioni
- Generazione di indici di partizione
- Impedire a un crawler di modificare uno schema esistente
- Creare un singolo schema per ogni percorso di inclusione Amazon S3
- Specificare la posizione della tabella e il livello di partizionamento
- Specificare il numero massimo di tabelle che il crawler può creare
- Configurare un crawler per utilizzare le credenziali di Lake Formation
- Accelerazione del crawling con le notifiche eventi Amazon S3

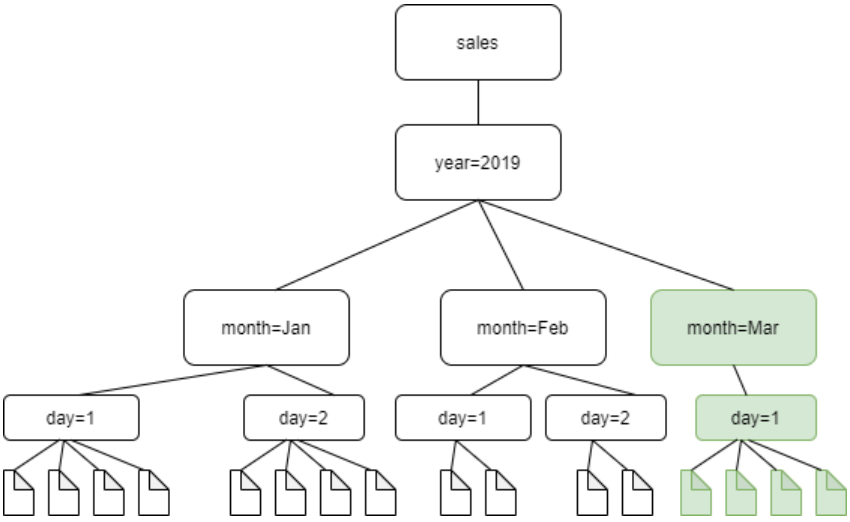
## Pianificazione di crawling incrementale per l'aggiunta di nuove partizioni

È possibile configurare Crawler di AWS Glue ed eseguire ricerche di indicizzazione incrementali per aggiungere solo nuove partizioni allo schema della tabella. Quando il crawler viene eseguito per la prima volta, esegue un crawling completo per elaborare l'intera origine dati per registrare lo schema completo e tutte le partizioni esistenti in AWS Glue Data Catalog.

I crawling successivi a quello iniziale saranno incrementali; in questi, il crawler identifica e aggiunge solo le nuove partizioni introdotte dopo il crawling precedente. Questo approccio consente tempi di crawling più rapidi, in quanto il crawler non deve più elaborare l'intera origine dati per ogni esecuzione, ma si concentra invece solo sulle nuove partizioni.

> **Note**
>
> I crawling incrementali non rilevano modifiche o eliminazioni di partizioni esistenti. Questa configurazione è più adatta per origini dati con uno schema stabile. Se si verifica una modifica importante dello schema una tantum, è consigliabile impostare temporaneamente il crawler in modo che esegua un crawling completo per acquisire il nuovo schema con precisione, e quindi tornare alla modalità di crawling incrementale.

Il diagramma seguente mostra che con l'impostazione di crawling abilitata, il crawler rileverà e aggiungerà solo la cartella appena aggiunta, month=March, al catalogo.




Segui questi passaggi per aggiornare il crawler per eseguire crawling incrementali:

Console di gestione AWS

1. Accedi a Console di gestione AWS e apri la console all'indirizzo. AWS Glue https://console.aws.amazon.com/glue/
2. Seleziona Crawler nel Catalogo dati.
3. Scegli un crawler che desideri configurare per il crawling incrementale.
4. Scegli Modifica.
5. Scegli Passaggio 2. Scegli fonti di dati e classificatori.
6. Seleziona l'origine dati per cui vuoi eseguire il crawling incrementale.
7. Scegli Modifica.
8. Scegli Esplora solo nuove sottocartelle in Esecuzioni successive del crawler.
9. Scegliere Aggiorna.

Per creare una pianificazione per un crawler, consulta the section called "Pianificazione di un crawler".

AWS CLI

```
aws glue update-crawler \
 --name myCrawler \
 --recrawl-policy RecrawlBehavior=CRAWL_NEW_FOLDERS_ONLY \
 --schema-change-policy UpdateBehavior=LOG,DeleteBehavior=LOG
```

Note e restrizioni

Quando questa opzione è attivata, non è possibile modificare gli archivi dati di destinazione Amazon S3 quando si modifica il crawler. Questa opzione influisce su alcune impostazioni di configurazione del crawler. Quando è attivata, impone il comportamento di aggiornamento e di eliminazione del crawler a `LOG`. Ciò significa che:

- Se rileva oggetti con schemi non compatibili, il crawler non aggiungerà gli oggetti nel Data Catalog e aggiungerà questi dettagli come log in Logs. CloudWatch
- Non aggiornerà gli oggetti eliminati nel catalogo dati.

### Generazione di indici di partizione

Il Catalogo dati supporta la creazione di indici di partizione per fornire una ricerca efficiente di partizioni specifiche. Per ulteriori informazioni, consulta la pagina relativa alla Creazione di indici di partizione. Per impostazione predefinita, il AWS Glue crawler crea indici di partizione per le destinazioni Amazon S3 e Delta Lake.

#### Console di gestione AWS

1. Accedi a e apri la console all'indirizzo. Console di gestione AWS AWS Glue https://console.aws.amazon.com/glue/
2. Seleziona Crawler nel Catalogo dati.
3. Quando si definisce un crawler, l'opzione Crea automaticamente indici di partizione è abilitata per impostazione predefinita in Opzioni avanzate nella pagina Imposta output e pianificazione.

   Per disabilitare questa opzione, puoi deselezionare la casella di controllo Crea automaticamente gli indici di partizione nella console.
4. Completa la configurazione del crawler e seleziona Crea crawler.

#### AWS CLI

Puoi anche disabilitare questa opzione usando la AWS CLI, impostando `CreatePartitionIndex` nel parametro `configuration`. Il valore di default è true.

```
aws glue update-crawler \
    --name myCrawler \
    --configuration '{"Version": 1.0, "CreatePartitionIndex": false }'
```

### Note di utilizzo sugli indici di partizione

- Le tabelle create dal crawler non hanno la variabile `partition_filtering.enabled` per impostazione predefinita. Per ulteriori informazioni, consulta la pagina AWS Glue partition indexing and filtering.
- La creazione di indici di partizione per partizioni crittografate non è supportata.

### Impedire a un crawler di modificare uno schema esistente

È possibile impedire a Crawler di AWS Glue s di apportare modifiche allo schema del Data Catalog durante l'esecuzione. Per impostazione predefinita, i crawler aggiornano lo schema nel Catalogo dati in modo che corrisponda all'origine dati sottoposta a scansione. Tuttavia, in alcuni casi, potresti voler impedire al crawler di modificare lo schema esistente, specialmente se hai trasformato o pulito i dati e non vuoi che lo schema originale sovrascriva le modifiche.

Segui questi passaggi per configurare il crawler affinché non sovrascriva lo schema esistente in una definizione di tabella.

#### Console di gestione AWS

1. Accedi a Console di gestione AWS e apri la AWS Glue console all'indirizzo https://console.aws.amazon.com/glue/.
2. Seleziona Crawler nel Catalogo dati.
3. Scegli un crawler dall'elenco e seleziona Modifica.
4. Scegli la fase 4: Configurazione dell'output e della pianificazione.
5. In Opzioni avanzate, seleziona Aggiungi solo nuove colonne o Ignora la modifica e non aggiornare la tabella nel Catalogo dati.
6. Puoi anche impostare un'opzione di configurazione per aggiornare tutte le partizioni nuove ed esistenti con metadati della tabella. In questo modo gli schemi delle partizioni vengono impostati in modo da ereditare quelli della tabella.
7. Scegliere Aggiorna.

#### AWS CLI

Il seguente esempio mostra come configurare un crawler in modo che non modifichi lo schema esistente, ma si limiti ad aggiungere nuove colonne:

```
aws glue update-crawler \
  --name myCrawler \
  --configuration '{"Version": 1.0, "CrawlerOutput": {"Tables":
 {"AddOrUpdateBehavior": "MergeNewColumns"}}}'
```

Il seguente esempio mostra come configurare un crawler in modo che non modifichi lo schema esistente e non aggiunga nuove colonne:

```
aws glue update-crawler \
  --name myCrawler \
  --schema-change-policy UpdateBehavior=LOG \
  --configuration '{"Version": 1.0, "CrawlerOutput": {"Partitions":
 { "AddOrUpdateBehavior": "InheritFromTable" }}}'
```

API

Se non vuoi che uno schema di tabella venga modificato in alcun modo durante l'esecuzione di un crawler, imposta la policy di modifica dello schema su `LOG`.

Quando configuri il crawler tramite l'API, imposta i parametri seguenti:

- Imposta il campo `UpdateBehavior` nella struttura `SchemaChangePolicy` su `LOG`.
- Imposta il campo `Configuration` con una rappresentazione di stringa dell'oggetto JSON seguente nell'API del crawler, ad esempio:

```
{
   "Version": 1.0,
   "CrawlerOutput": {
      "Partitions": { "AddOrUpdateBehavior": "InheritFromTable" }
   }
}
```

Creare un singolo schema per ogni percorso di inclusione Amazon S3

Per impostazione predefinita, quando un crawler definisce tabelle per i dati archiviati in Amazon S3, considera sia la compatibilità dei dati sia la somiglianza dello schema. I fattori di compatibilità dei dati presi in considerazione includono il fatto che i dati abbiano o meno lo stesso formato (ad esempio JSON), lo stesso tipo di compressione (ad esempio GZIP), la stessa struttura del percorso Amazon S3 e altri attributi di dati. La somiglianza dello schema misura il livello di somiglianza degli schemi di oggetti Amazon S3 separati.

Per descrivere meglio questa opzione, supponiamo di definire un crawler con un percorso di inclusione `s3://amzn-s3-demo-bucket/table1/`. Quando il crawler viene eseguito, individua due file JSON con le caratteristiche seguenti:

- File 1: `S3://amzn-s3-demo-bucket/table1/year=2017/data1.json`

- Contenuto del file: `{"A": 1, "B": 2}`
- Schema: `A:int, B:int`

- File 2: `S3://amzn-s3-demo-bucket/table1/year=2018/data2.json`
- Contenuto del file – `{"C": 3, "D": 4}`
- Schema – `C: int, D: int`

Per impostazione predefinita, il crawler crea due tabelle, denominate `year_2017` e `year_2018`, perché gli schemi non sono abbastanza simili. Tuttavia, se l'opzione Create a single schema for each S3 path (Crea un singolo schema per ogni percorso S3) è selezionata e se i dati sono compatibili, il crawler crea una tabella. La tabella contiene lo schema `A:int,B:int,C:int,D:int` e `partitionKey year:string`.

Console di gestione AWS

1. Accedi a Console di gestione AWS e apri la AWS Glue console all'indirizzo https://console.aws.amazon.com/glue/.
2. Seleziona Crawler nel Catalogo dati.
3. Quando configuri un nuovo crawler, in Output e pianificazione, vai su Opzioni avanzate e seleziona l'opzione Crea uno schema singolo per ogni percorso S3.

AWS CLI

È possibile configurare un crawler in modo che esegua l'operazione `CombineCompatibleSchemas` per combinare schemi compatibili in una definizione di tabella comune quando possibile. Con questa opzione, il crawler continua a considerare la compatibilità dei dati, ma ignora la somiglianza degli schemi specifici durante la valutazione di oggetti Amazon S3 nel percorso di inclusione specificato.

Quando configurate il crawler utilizzando il AWS CLI, impostate la seguente opzione di configurazione:

```
aws glue update-crawler \
   --name myCrawler \
   --configuration '{"Version": 1.0, "Grouping": {"TableGroupingPolicy":
 "CombineCompatibleSchemas" }}'
```

API

Quando configuri il crawler tramite l'API, imposta l'opzione di configurazione seguente:

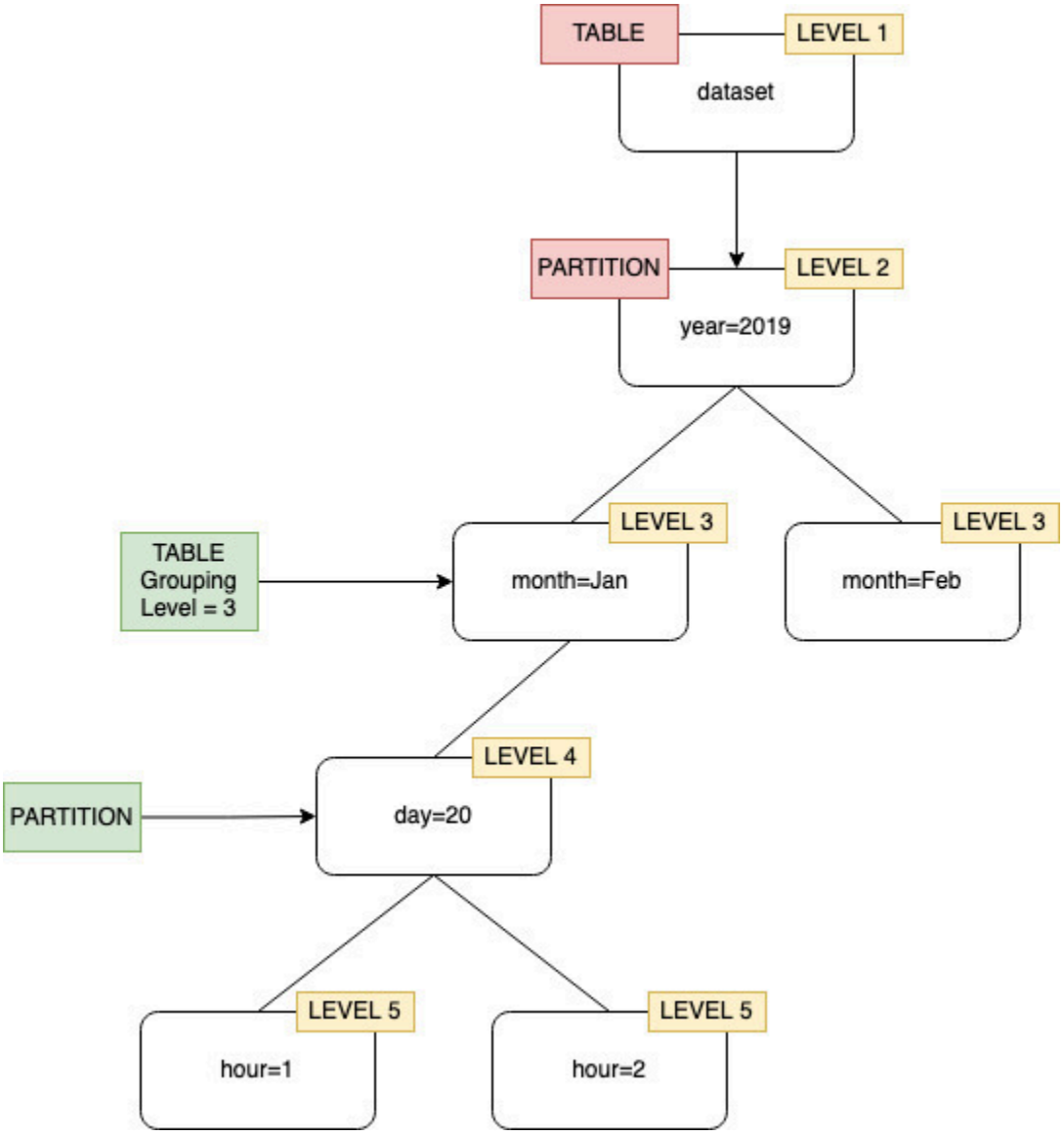
Imposta il campo `Configuration` con una rappresentazione di stringa dell'oggetto JSON seguente nell'API del crawler, ad esempio:

```
{
   "Version": 1.0,
   "Grouping": {
      "TableGroupingPolicy": "CombineCompatibleSchemas" }
}
```

Specificare la posizione della tabella e il livello di partizionamento

Per impostazione predefinita, quando un crawler definisce tabelle per i dati archiviati in Amazon S3, il crawler tenta di unire gli schemi e creare tabelle di primo livello (`year=2019`). In alcuni casi, è possibile che, invece di creare una tabella per la cartella `month=Jan` come previsto, il crawler crei una partizione poiché una cartella di pari livello (`month=Mar`) è stata unita alla stessa tabella.

L'opzione crawler a livello di tabella offre la flessibilità necessaria per indicare al crawler dove si trovano le tabelle e come si desidera creare le partizioni. Quando si specifica un Table level (Livello della tabella), la tabella viene creata a quel livello assoluto dal bucket Amazon S3.

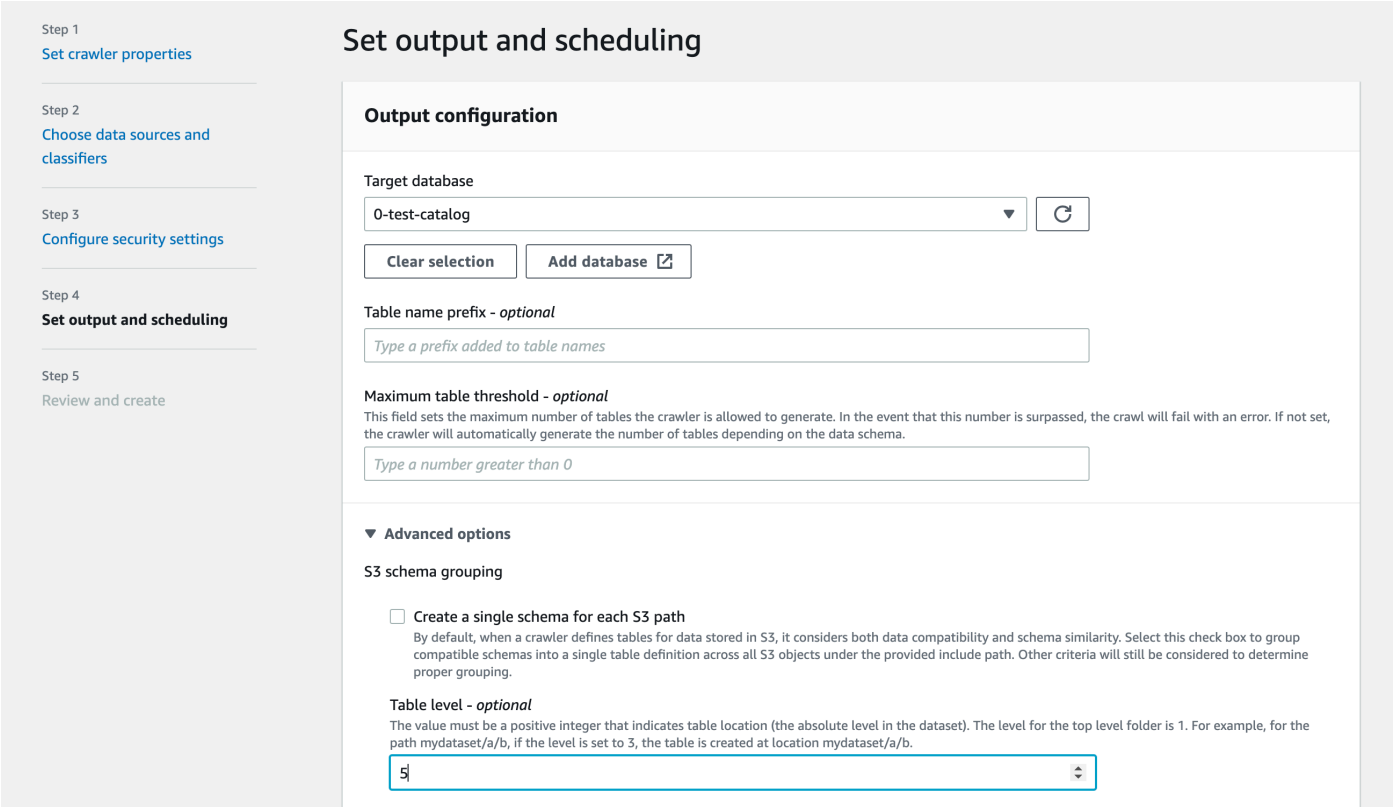



Quando si configura il crawler nella console, è possibile specificare un valore per l'opzione crawler Table level (Livello della tabella). Il valore deve essere un numero intero positivo che indica la posizione della tabella (il livello assoluto nel set di dati). Il livello per la cartella di livello superiore è 1. Ad esempio, per il percorso `mydataset/year/month/day/hour`, se il livello è impostato su 3, la tabella viene creata nella posizione `mydataset/year/month`.

Console di gestione AWS

1. Accedi a Console di gestione AWS e apri la AWS Glue console all'indirizzo https://console.aws.amazon.com/glue/.
2. Seleziona Crawler nel Catalogo dati.

3. Quando configuri un crawler, in Output e pianificazione, vai su Opzioni avanzate e seleziona Livello di tabella.




AWS CLI

Quando configurate il crawler utilizzando il AWS CLI, impostate il `configuration` parametro come mostrato nel codice di esempio:

```
aws glue update-crawler \
  --name myCrawler \
  --configuration '{"Version": 1.0, "Grouping": { "TableLevelConfiguration": 2 }}'
```

API

Quando configuri il crawler usando l'API, imposta il campo `Configuration` con una rappresentazione stringa del seguente oggetto JSON; per esempio:

```
configuration = jsonencode(
{
   "Version": 1.0,
   "Grouping": {
```

```
            TableLevelConfiguration = 2
        }
})
```

CloudFormation

In questo esempio, impostate l'opzione Table level disponibile nella console all'interno del modello: CloudFormation

```
"Configuration": "{
    \"Version\":1.0,
    \"Grouping\":{\"TableLevelConfiguration\":2}
}"
```

Specificare il numero massimo di tabelle che il crawler può creare

Facoltativamente, puoi specificare il numero massimo di tabelle che il crawler può creare specificando un `TableThreshold` tramite la console o. AWS Glue AWS CLI Se il numero di tabelle rilevate dal crawler durante il crawling è superiore a questo valore di input, il crawling ha esito negativo e non vengono scritti dati nel Catalogo dati.

Questo parametro è utile quando le tabelle che verrebbero rilevate e create dal crawler sono molto più grandi del previsto. I motivi possono essere molteplici, come ad esempio:

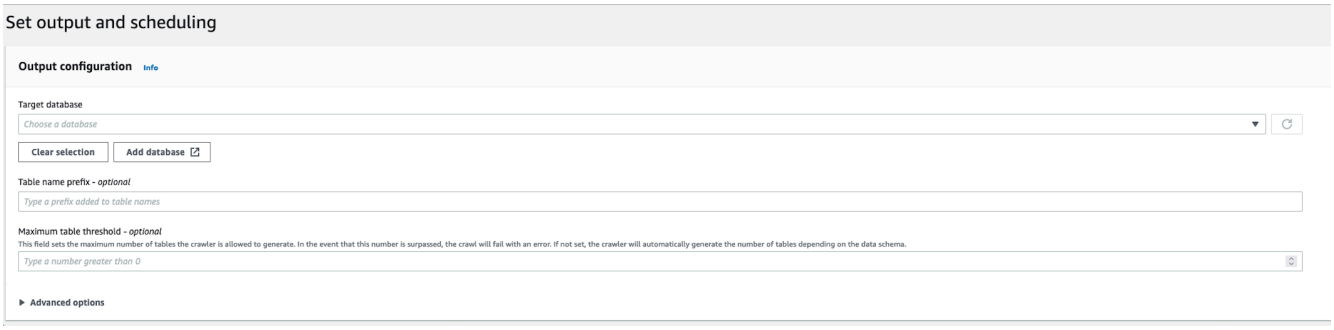
- Quando utilizzi un AWS Glue job per popolare le tue sedi Amazon S3, puoi ritrovarti con file vuoti allo stesso livello di una cartella. In questo caso, quando esegui un crawler su questa posizione Amazon S3, il crawler crea più tabelle a causa di file e cartelle presenti allo stesso livello.
- La mancata configurazione di `"TableGroupingPolicy": "CombineCompatibleSchemas"` potrebbe restituire un numero maggiore di tabelle rispetto al previsto.

Specifica il parametro `TableThreshold` come un valore intero maggiore di 0. Questo valore è configurato in base al crawler, quindi viene preso in considerazione per ogni crawling. Si supponga ad esempio di avere un crawler con il valore `TableThreshold` impostato su 5. In ogni ricerca per indicizzazione, AWS Glue confronta il numero di tabelle rilevate con questo valore di soglia della tabella (5) e, se il numero di tabelle rilevate è inferiore a 5, AWS Glue scrive le tabelle nel Data Catalog e, in caso contrario, la ricerca per indicizzazione fallisce senza scrivere nel Data Catalog.

Console di gestione AWS

Per impostare **`TableThreshold`** utilizzando: Console di gestione AWS

1. Accedi a Console di gestione AWS e apri la AWS Glue console all'indirizzo https://console.aws.amazon.com/glue/.
2. Quando configuri un crawler, in Output e pianificazione, imposta la soglia massima per le tabelle sul numero di tabelle che il crawler è autorizzato a generare.




AWS CLI

Per impostare `TableThreshold` utilizzando: AWS CLI

```
aws glue update-crawler \
    --name myCrawler \
    --configuration '{"Version": 1.0, "CrawlerOutput": {"Tables":
 { "TableThreshold": 5 }}}'
```

API

Per impostare `TableThreshold` tramite l'API:

```
"{"Version":1.0,
"CrawlerOutput":
{"Tables":{"AddOrUpdateBehavior":"MergeNewColumns",
"TableThreshold":5}}}";
```

I messaggi di errore vengono registrati per facilitare l'identificazione dei percorsi delle tabelle e la pulizia dei dati. Di seguito è riportato un esempio di accesso all'account in caso di esito negativo del crawler a causa del numero di tabelle superiore al valore di soglia:

```
Table Threshold value = 28, Tables detected - 29
```

In CloudWatch, registriamo tutte le posizioni delle tabelle rilevate come messaggio INFO. Le informazioni relative all'errore vengono registrate in un messaggio.

```
ERROR com.amazonaws.services.glue.customerLogs.CustomerLogService - CustomerLogService
 received CustomerFacingException with message
The number of tables detected by crawler: 29 is greater than the table threshold value
 provided: 28. Failing crawler without writing to Data Catalog.
com.amazonaws.services.glue.exceptions.CustomerFacingInternalException: The number of
 tables detected by crawler: 29 is greater than the table threshold value provided:
 28.
Failing crawler without writing to Data Catalog.
```

Configurare un crawler per utilizzare le credenziali di Lake Formation

Puoi configurare un crawler per utilizzare AWS Lake Formation le credenziali per accedere a un data store Amazon S3 o a una tabella Data Catalog con una posizione Amazon S3 sottostante all'interno della stessa o di un'altra. Account AWS Account AWSÈ possibile configurare una tabella del catalogo dati esistente come destinazione del crawler se entrambi si trovano nello stesso account. Attualmente, è consentita solo una singola destinazione del catalogo per una singola tabella quando si utilizza una tabella del catalogo dati come destinazione del crawler.

> **Note**
>
> Quando si definisce una tabella del catalogo dati come destinazione del crawler, assicurarsi che la posizione sottostante della tabella sia una posizione Amazon S3. I crawler che utilizzano le credenziali Lake Formation supportano solo le destinazioni del catalogo con le posizioni Amazon S3 sottostanti.

Configurazione richiesta quando il crawler e la posizione registrata di Amazon S3 o la tabella del catalogo dati si trovano nello stesso account (crawling all'interno dell'account)

Per consentire al crawler di accedere a un datastore o a una tabella del catalogo dati utilizzando le credenziali di Lake Formation, è necessario registrare la posizione dei dati con Lake Formation. Inoltre, il ruolo IAM del crawler deve disporre delle autorizzazioni necessarie per leggere i dati dalla destinazione in cui è registrato il bucket Amazon S3.

Puoi completare i seguenti passaggi di configurazione utilizzando o (). Console di gestione AWS AWS Command Line Interface AWS CLI

Console di gestione AWS

1. Prima di configurare un crawler per accedere alla sua origine, registra la posizione dei dati del datastore o del catalogo dati con Lake Formation. Nella console Lake Formation (https://console.aws.amazon.com/lakeformation/), registra una posizione Amazon S3 come posizione principale del tuo data lake nel punto in Account AWS cui è definito il crawler. Per ulteriori informazioni, consulta la pagina Registrazione di una posizione Amazon S3.
2. Concedi le autorizzazioni relative alla posizione dei dati al ruolo IAM utilizzato per l'esecuzione del crawler in modo che il crawler possa leggere i dati dalla destinazione in Lake Formation. Per ulteriori informazioni, consulta la pagina Concessione delle autorizzazioni per la posizione dei dati (stesso account).
3. Concessione al ruolo crawler delle autorizzazioni di accesso (`Create`) al database, che è specificato come database di output. Per ulteriori informazioni, consulta la pagina Concessione delle autorizzazioni al database tramite la console di Lake Formation e il metodo delle risorse denominate.
4. Nella console IAM (https://console.aws.amazon.com/iam/), crea un ruolo IAM per il crawler. Aggiungi la policy `lakeformation:GetDataAccess` al ruolo.
5. Nella AWS Glue console (https://console.aws.amazon.com/glue/), durante la configurazione del crawler, seleziona l'opzione Usa le credenziali di Lake Formation per la scansione dell'origine dati Amazon S3.

> **Note**
>
> Il campo accountId è facoltativo per il crawling all'interno dell'account.

AWS CLI

```
aws glue --profile demo create-crawler --debug --cli-input-json '{
    "Name": "prod-test-crawler",
    "Role": "arn:aws:iam::111122223333:role/service-role/AWSGlueServiceRole-prod-
test-run-role",
    "DatabaseName": "prod-run-db",
    "Description": "",
    "Targets": {
```

```
    "S3Targets":[
                {
                 "Path": "s3://amzn-s3-demo-bucket"
                }
                ]
                },
   "SchemaChangePolicy": {
      "UpdateBehavior": "LOG",
      "DeleteBehavior": "LOG"
  },
  "RecrawlPolicy": {
    "RecrawlBehavior": "CRAWL_EVERYTHING"
  },
  "LineageConfiguration": {
    "CrawlerLineageSettings": "DISABLE"
  },
  "LakeFormationConfiguration": {
    "UseLakeFormationCredentials": true,
    "AccountId": "111122223333"
  },
  "Configuration": {
           "Version": 1.0,
           "CrawlerOutput": {
             "Partitions": { "AddOrUpdateBehavior": "InheritFromTable" },
             "Tables": {"AddOrUpdateBehavior": "MergeNewColumns" }
           },
           "Grouping": { "TableGroupingPolicy": "CombineCompatibleSchemas" }
         },
  "CrawlerSecurityConfiguration": "",
  "Tags": {
    "KeyName": ""
  }
}'
```

Configurazione richiesta quando il crawler e la posizione registrata di Amazon S3 si trovano in account diversi (crawling tra più account)

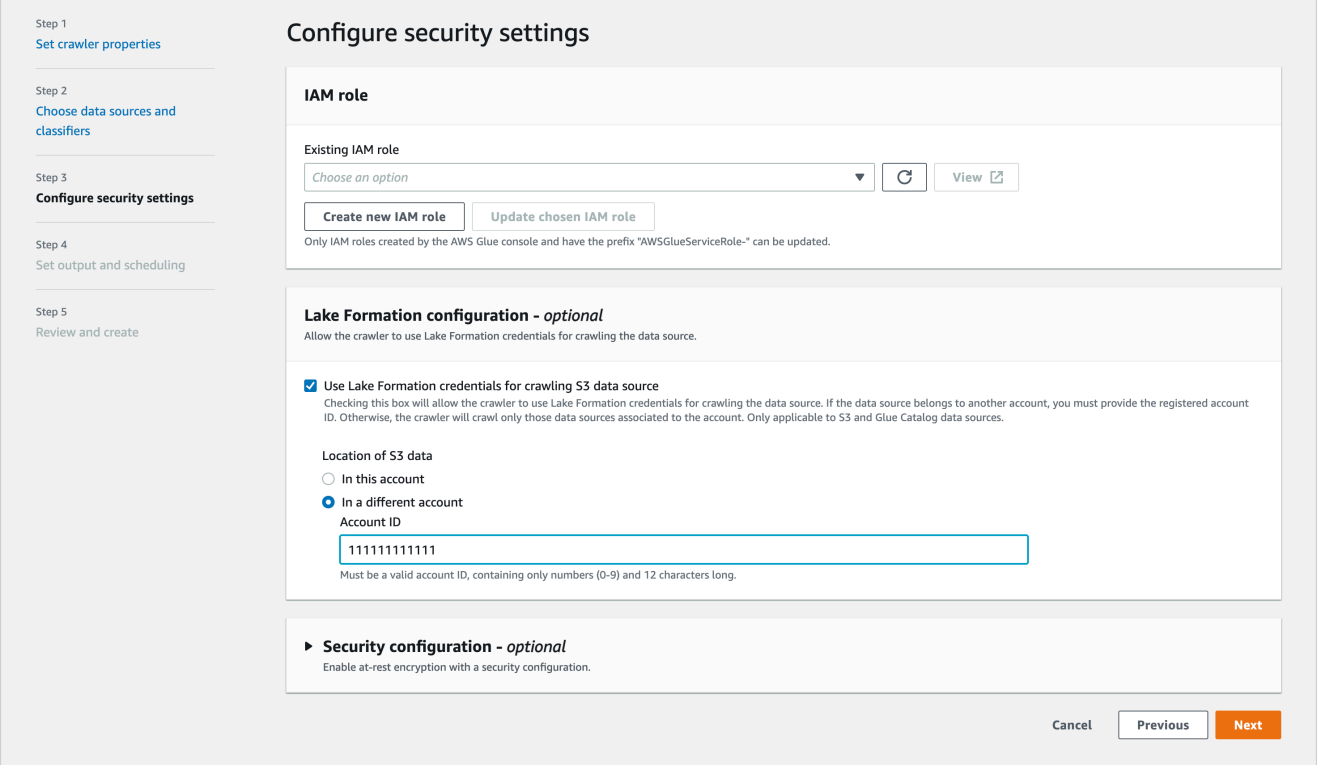
Per consentire al crawler di accedere a un datastore in un account diverso utilizzando le credenziali di Lake Formation, è necessario prima registrare la posizione dei dati di Amazon S3 con Lake Formation. Quindi, concedi le autorizzazioni per la posizione dei dati all'account del crawler eseguendo la procedura seguente.

È possibile completare i seguenti passaggi utilizzando Console di gestione AWS o AWS CLI.

Console di gestione AWS

1. Nell'account in cui è registrata la posizione Amazon S3 (account B):
   a. Registra un percorso Amazon S3 con Lake Formation. Per ulteriori informazioni, consulta la pagina Registrazione della posizione Amazon S3.
   b. Concedi le autorizzazioni per la posizione dei dati all'account (A) in cui verrà eseguito il crawler. Per ulteriori informazioni, consulta la pagina Concessione delle autorizzazioni per la posizione dei dati.
   c. Crea un database vuoto in Lake Formation con la posizione sottostante come posizione Amazon S3 di destinazione. Per ulteriori informazioni, consulta la pagina Creazione di un database.
   d. Concedi l'accesso al database all'account A (l'account in cui verrà eseguito il crawler) creato nel passaggio precedente. Per ulteriori informazioni, consulta la pagina Concessione delle autorizzazioni al database.
2. Nell'account in cui è stato creato e verrà eseguito il crawler (account A):
   a. Utilizzando la AWS RAM console, accetta il database condiviso dall'account esterno (account B). Per ulteriori informazioni, consulta Accettazione di un invito alla condivisione di risorse da AWS Resource Access Manager.
   b. Crea un ruolo IAM per il crawler. Aggiungi la policy `lakeformation:GetDataAccess` al ruolo.
   c. Nella console Lake Formation (https://console.aws.amazon.com/lakeformation/), concedi le autorizzazioni Data location sulla posizione Amazon S3 di destinazione al ruolo IAM utilizzato per l'esecuzione del crawler in modo che il crawler possa leggere i dati dalla destinazione in Lake Formation. Per ulteriori informazioni, consulta la pagina Concessione delle autorizzazioni per la posizione dei dati.
   d. Crea un collegamento alla risorsa nel database condiviso. Per ulteriori informazioni, consulta la pagina Creare un collegamento alla risorsa.
   e. Concessione al ruolo crawler delle autorizzazioni di accesso (`Create`) sul database condiviso e (`Describe`) sul collegamento alla risorsa. Il collegamento alla risorsa è specificato nell'output del crawler.
   f. Nella AWS Glue console (https://console.aws.amazon.com/glue/), durante la configurazione del crawler, seleziona l'opzione Usa le credenziali di Lake Formation per la scansione dell'origine dati Amazon S3.

Per la scansione tra più account, specifica l' Account AWS ID in cui è registrata la sede Amazon S3 di destinazione con Lake Formation. Per effettuare il crawling all'interno dell'account, il campo accountId è facoltativo.




## AWS CLI

```
aws glue --profile demo create-crawler --debug --cli-input-json '{
    "Name": "prod-test-crawler",
    "Role": "arn:aws:iam::111122223333:role/service-role/AWSGlueServiceRole-prod-
test-run-role",
    "DatabaseName": "prod-run-db",
    "Description": "",
    "Targets": {
    "S3Targets":[
                {
                 "Path": "s3://amzn-s3-demo-bucket"
                }
                ]
                },
   "SchemaChangePolicy": {
      "UpdateBehavior": "LOG",
      "DeleteBehavior": "LOG"
```

```
  },
  "RecrawlPolicy": {
    "RecrawlBehavior": "CRAWL_EVERYTHING"
  },
  "LineageConfiguration": {
    "CrawlerLineageSettings": "DISABLE"
  },
  "LakeFormationConfiguration": {
    "UseLakeFormationCredentials": true,
    "AccountId": "111111111111"
  },
  "Configuration": {
          "Version": 1.0,
          "CrawlerOutput": {
            "Partitions": { "AddOrUpdateBehavior": "InheritFromTable" },
            "Tables": {"AddOrUpdateBehavior": "MergeNewColumns" }
          },
          "Grouping": { "TableGroupingPolicy": "CombineCompatibleSchemas" }
        },
  "CrawlerSecurityConfiguration": "",
  "Tags": {
    "KeyName": ""
  }
}'
```

Note

- Un crawler che utilizza le credenziali Lake Formation è supportato solo per le destinazioni Amazon S3 e Catalogo dati.
- Per le destinazioni che utilizzano la distribuzione delle credenziali Lake Formation, le posizioni Amazon S3 sottostanti devono appartenere allo stesso bucket. Ad esempio, gli utenti possono utilizzare più destinazioni (s3://amzn-s3-demo-bucket1/folder1, s3://amzn-s3-demo-bucket1/folder2) purché tutte le posizioni di destinazione si trovino nello stesso bucket (amzn-s3-demo-bucket1). Non è consentito specificare bucket diversi (s3://amzn-s3-demo-bucket1/folder1, s3://amzn-s3-demo-bucket2/folder2).
- Per i crawler di destinazione del catalogo dati è attualmente consentita solo una singola destinazione del catalogo per una singola tabella.

Accelerazione del crawling con le notifiche eventi Amazon S3

Invece di elencare gli oggetti da una destinazione Amazon S3 o catalogo dati, puoi configurare il crawler in modo che utilizzi gli eventi Amazon S3 per trovare eventuali modifiche. Questa caratteristica migliora il tempo di recupero utilizzando gli eventi Amazon S3 per identificare le modifiche tra due ricerche per indicizzazione elencando tutti i file della sottocartella che ha attivato l'evento invece che elencare l'intera destinazione Amazon S3 o catalogo dati.

Il primo crawling elenca tutti gli oggetti Amazon S3 dalla destinazione. Dopo il primo crawling riuscito, è possibile scegliere di effettuare una ricerca manualmente o in base a una pianificazione prestabilita. Il crawler elencherà solo gli oggetti di tali eventi invece di elencare tutti gli oggetti.

Quando la destinazione è una tabella del Catalogo dati, il crawler aggiorna le tabelle esistenti al suo interno con le modifiche (ad esempio, partizioni aggiuntive in una tabella).

I vantaggi di passare a un crawler basato su eventi Amazon S3 sono:

- Non è necessario un nuovo crawling più rapido, poiché non è necessario l'elenco di tutti gli oggetti della destinazione, invece l'elenco di cartelle specifiche viene eseguito dove gli oggetti vengono aggiunti o eliminati.
- Si ha una riduzione del costo complessivo del crawling man mano che vengono elencate le cartelle specifiche nelle quali gli oggetti vengono aggiunti o eliminati.

Il crawling degli eventi Amazon S3 viene eseguito consumando gli eventi Amazon S3 dalla coda SQS in base alla pianificazione del crawler. Non ci saranno costi se non ci sono eventi nella coda. Gli eventi Amazon S3 possono essere configurati in modo che passino direttamente alla coda SQS o, nei casi in cui più utenti hanno bisogno dello stesso evento, verso una combinazione di SNS e SQS. Per ulteriori informazioni, consulta the section called “Impostazione dell'account per le notifiche eventi Amazon S3”.

Dopo aver creato e configurato il crawler in modalità evento, il primo crawling viene eseguito in modalità elenco eseguendo un elenco completo della destinazione Amazon S3 o catalogo dati. Il seguente log conferma il funzionamento del crawling consumando gli eventi Amazon S3 dopo la prima scansione riuscita: "il crawling è in esecuzione consumando eventi Amazon S3".

Dopo aver creato la ricerca per indicizzazione degli eventi Amazon S3 e aver aggiornato le proprietà del crawler che potrebbero influire sul crawling, quest'ultima funziona in modalità elenco e viene aggiunto il seguente log: "Il crawling non è in esecuzione in modalità evento S3".

> **Note**
>
> Il numero massimo di messaggi da consumare è 100.000 messaggi per ogni crawling.

### Considerazioni e limitazioni

Le seguenti considerazioni e limitazioni si applicano quando si configura un crawler in modo che usi le notifiche degli eventi Amazon S3 per individuare eventuali modifiche.

- Comportamento importante con partizioni eliminate

  Quando usi i crawler di eventi Amazon S3 con tabelle del Catalogo dati:

  - Se elimini una partizione usando la chiamata API `DeletePartition`, devi anche eliminare tutti gli oggetti S3 sotto quella partizione e selezionare Tutti gli eventi di rimozione oggetti quando configuri le notifiche degli eventi S3. Se gli eventi di eliminazione non sono configurati, il crawler ricrea la partizione eliminata durante l'esecuzione successiva.
- Il crawler di destinazione ne supporta una sola, sia per quanto riguarda le destinazioni Amazon S3 che per le destinazioni Amazon S3.
- L'SQS su VPC privato non è supportato.
- Il campionamento Amazon S3 non è supportato.
- La destinazione del crawler deve essere una cartella per una destinazione Amazon S3 o una o più tabelle di catalogo dati di AWS Glue per una destinazione catalogo dati.
- Il carattere jolly del percorso "tutto" non è supportato: s3: //%
- Per una destinazione catalogo dati, tutte le tabelle del catalogo devono puntare allo stesso bucket Amazon S3 per la modalità evento di Amazon S3.
- Per una destinazione catalogo dati, una tabella di catalogo non deve indicare una posizione Amazon S3 nel formato Delta Lake (contenente cartelle _symlink o controllando le tabelle del catalogo `InputFormat`).

### Argomenti

- Impostazione dell'account per le notifiche eventi Amazon S3
- Configurare un crawler per le notifiche eventi Amazon S3 per una destinazione Amazon S3
- Configurazione di un crawler per le notifiche degli eventi Amazon S3 per una tabella del Catalogo dati

Impostazione dell'account per le notifiche eventi Amazon S3

Completa i seguenti processi di configurazione. Nota che i valori tra parentesi fanno riferimento alle impostazioni configurabili dello script.

1. Occorre configurare le notifiche degli eventi per il bucket Amazon S3.

   Per ulteriori informazioni, consulta Notifiche degli eventi di Amazon S3.
2. Per utilizzare il crawler basato su eventi Amazon S3, è necessario abilitare la notifica degli eventi sul bucket Amazon S3 con eventi filtrati dal prefisso uguale alla destinazione S3 e archiviarlo in SQS. È possibile configurare SQS e la notifica degli eventi tramite la console seguendo la procedura descritta in Spiegazione passo per passo: come configurare un bucket per le notifiche.
3. Aggiungi la seguente policy SQS al ruolo utilizzato dal crawler.

   JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "sqs:DeleteMessage",
        "sqs:GetQueueUrl",
        "sqs:ListDeadLetterSourceQueues",
        "sqs:ReceiveMessage",
        "sqs:GetQueueAttributes",
        "sqs:ListQueueTags",
        "sqs:SetQueueAttributes",
        "sqs:PurgeQueue"
      ],
      "Resource": "arn:aws:sqs:us-east-1:111122223333:cfn-sqs-queue"
    }
  ]
}
```

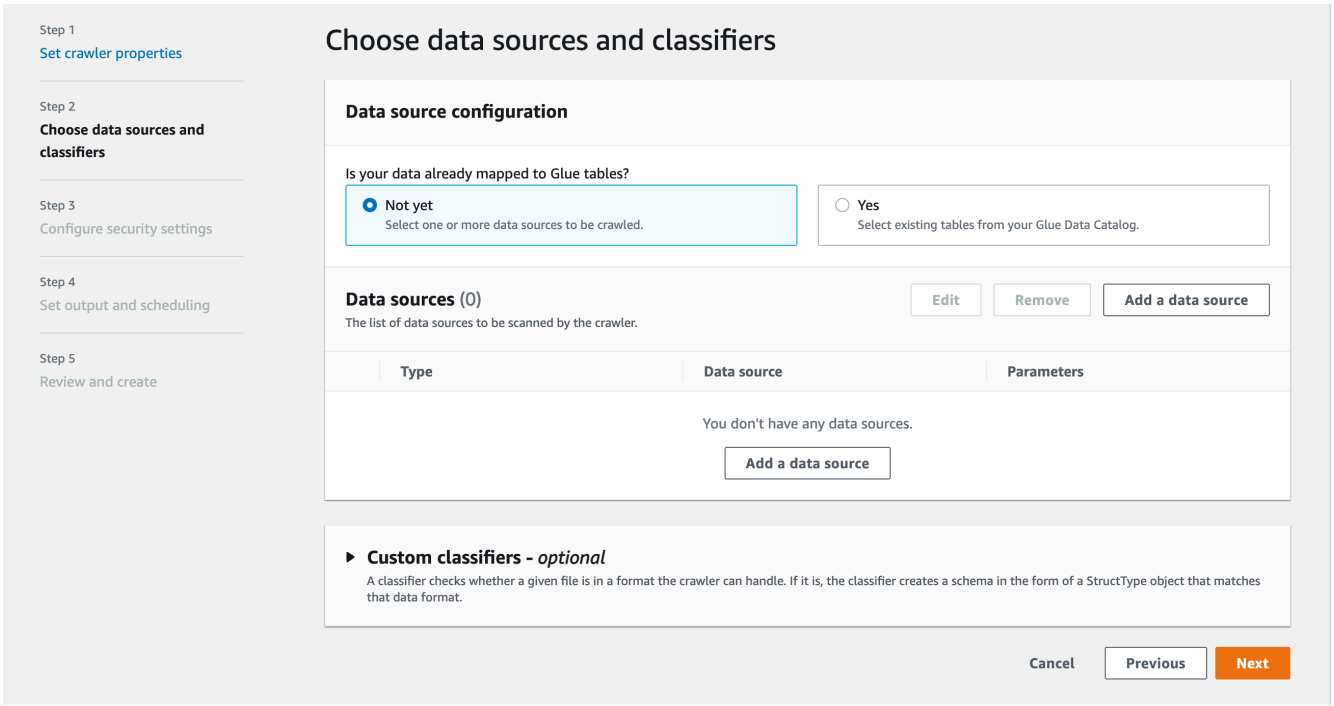
Configurare un crawler per le notifiche eventi Amazon S3 per una destinazione Amazon S3

Segui questi passaggi per configurare un crawler per le notifiche degli eventi di Amazon S3 per un target Amazon S3 utilizzando o. Console di gestione AWS AWS CLI

Console di gestione AWS

1. Accedi a Console di gestione AWS e apri la console all'indirizzo. GuardDuty https://console.aws.amazon.com/guardduty/
2. Imposta le proprietà del crawler. Per ulteriori informazioni, consulta la pagina Impostazione delle opzioni di configurazione del crawler nella console AWS Glue.
3. Nella sezione Data source configuration (Configurazione origine dei dati) viene chiesto Is your data already mapped to AWS Glue tables? .

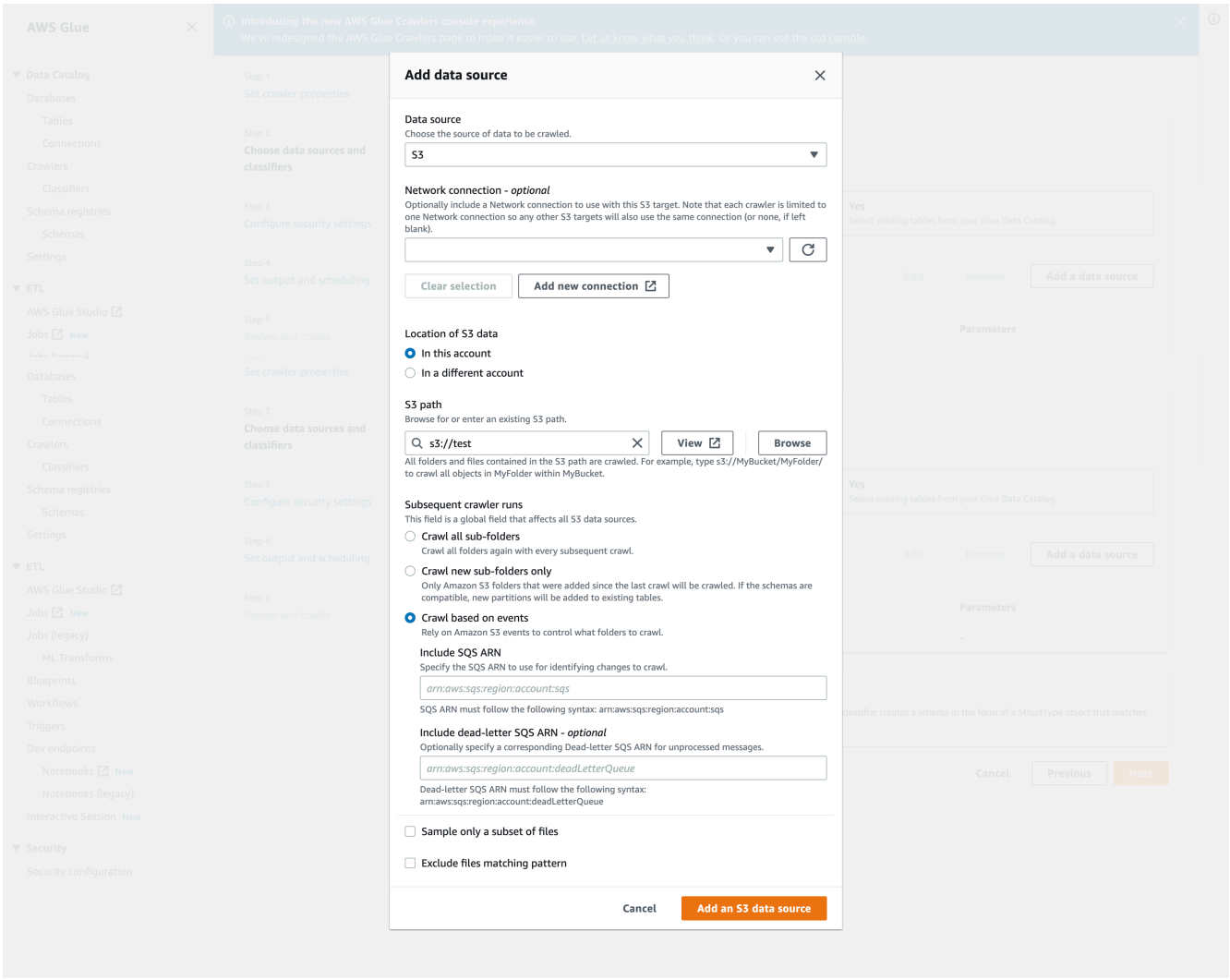
   Per impostazione predefinita, la risposta Not yet (Non ancora) è già selezionata. Lascia questa impostazione come predefinita poiché stai utilizzando un'origine dei dati Amazon S3 e i dati non sono ancora mappati su tabelle AWS Glue.
4. Nella sezione Data sources (Origini dei dati), scegli Add a data source (Aggiungi un'origine dei dati).




5. Nella modalità Add data source (Aggiungi origine dei dati), configura l'origine dati di Amazon S3:

   - Data source (Origine dei dati): per impostazione predefinita, è selezionato Amazon S3.
   - Network connection (Connessione di rete) (Facoltativo): seleziona Add new connection (Aggiungi una nuova connessione).

- Location of Amazon S3 data (Posizione dei dati Amazon S3): per impostazione predefinita, è selezionata l'opzione In this account (In questo account).
- Amazon S3 path (Percorso Amazon S3): specifica il percorso Amazon S3 in cui effettuare il crawling in cartelle e file.
- Subsequent crawler runs (Esecuzione successiva del crawler): seleziona Crawl based on events (Crawling in base agli eventi) per utilizzare le notifiche degli eventi di Amazon S3 per il crawler.
- Include SQS ARN (Includi ARN SQS): specifica i parametri del datastore, incluso un ARN SQS valido. Ad esempio, `arn:aws:sqs:region:account:sqs`.
- Include dead-letter SQS ARN (Includi ARN SQS non recapitabili): specifica un ARN SQS non recapitabile di Amazon valido. Ad esempio, `arn:aws:sqs:region:account:deadLetterQueue`.
- Scegli Add an Amazon S3 data source (Aggiungi un'origine dei dati Amazon S3).




## AWS CLI

Di seguito è riportato un esempio di AWS CLI chiamata Amazon S3 per configurare un crawler per utilizzare le notifiche di eventi per eseguire la scansione di un bucket di destinazione Amazon S3.

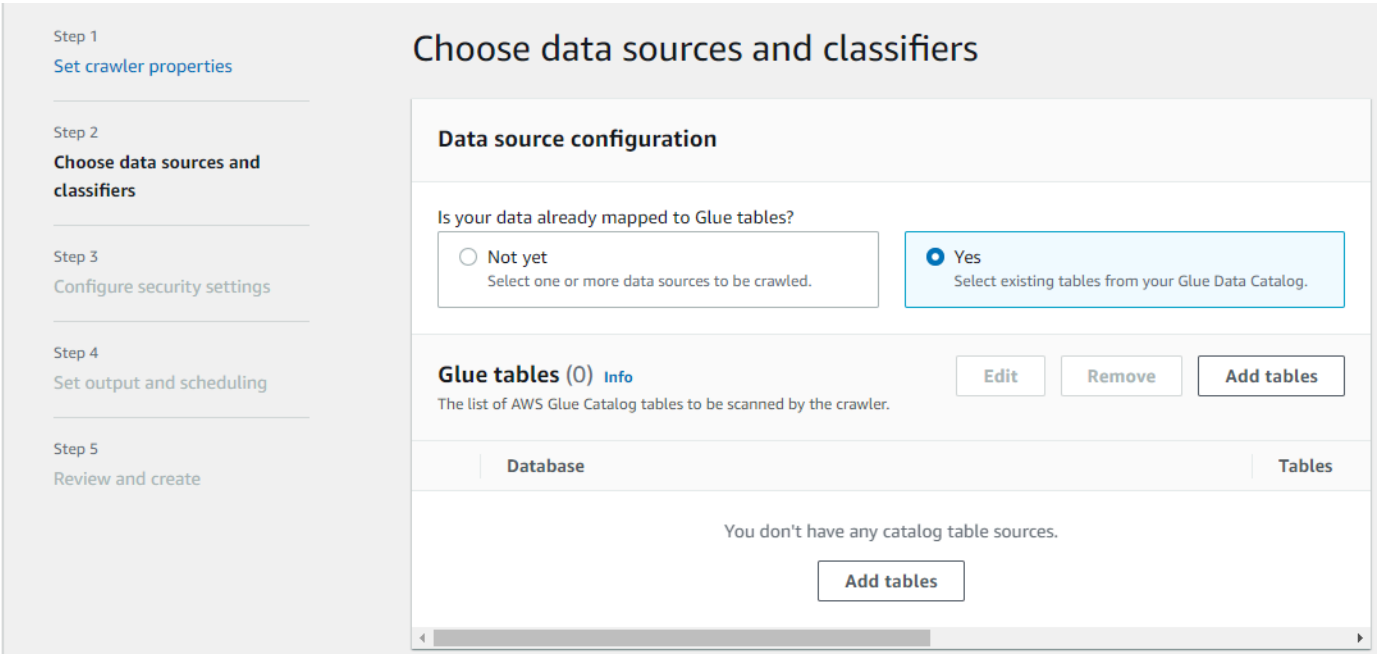
```
Create Crawler:
aws glue update-crawler \
    --name myCrawler \
    --recrawl-policy RecrawlBehavior=CRAWL_EVENT_MODE \
    --schema-change-policy UpdateBehavior=UPDATE_IN_DATABASE,DeleteBehavior=LOG
    --targets '{"S3Targets":[{"Path":"s3://amzn-s3-demo-bucket/", "EventQueueArn":
 "arn:aws:sqs:us-east-1:012345678910:MyQueue"}]}'
```

Configurazione di un crawler per le notifiche degli eventi Amazon S3 per una tabella del Catalogo dati

In presenza di una tabella del Catalogo dati, configura un crawler per le notifiche eventi Amazon S3 utilizzando la console AWS Glue:

1. Imposta le proprietà del crawler. Per ulteriori informazioni, consulta la pagina Impostazione delle opzioni di configurazione del crawler nella console AWS Glue.
2. Nella sezione Data source configuration (Configurazione origine dei dati) viene chiesto Is your data already mapped to AWS Glue tables? .

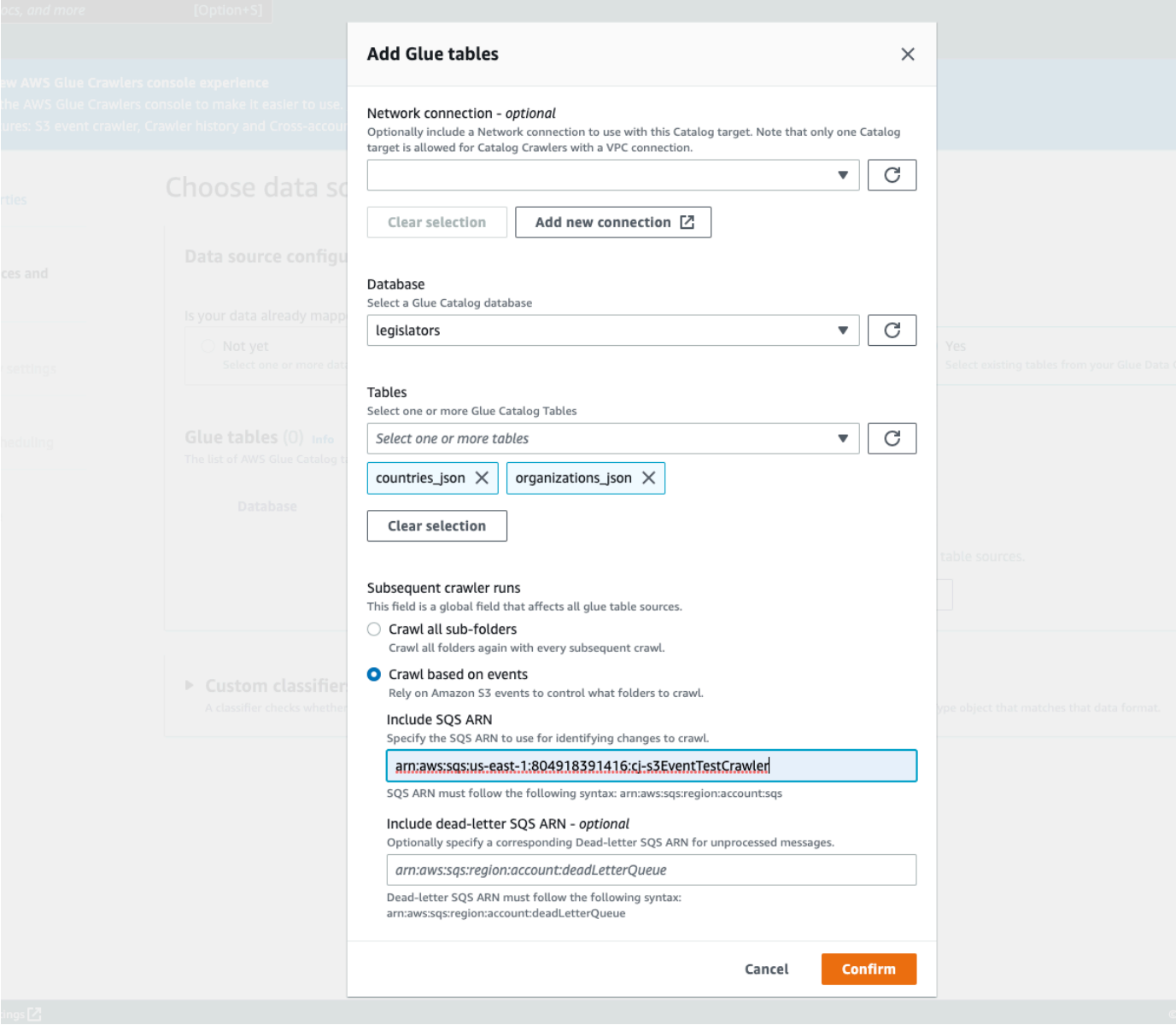
   Seleziona Yes (Sì) per selezionare le tabelle esistenti dal catalogo dati come origine dati.
3. Nella sezione Glue tables, (tabelle Glue) scegli Add tables (aggiungi tabelle).




4. Nella modalità Add table (aggiungi tabella), configura il database e le tabelle:

   - Network connection (Connessione di rete) (Facoltativo): seleziona Add new connection (Aggiungi una nuova connessione).
   - Database: selezionare un database nel catalogo dati.
   - Tabelle: seleziona una o più tabelle da quel database nel catalogo dati.

- Subsequent crawler runs (Esecuzione successiva del crawler): seleziona Crawl based on events (Crawling in base agli eventi) per utilizzare le notifiche degli eventi di Amazon S3 per il crawler.
- Include SQS ARN (Includi ARN SQS): specifica i parametri del datastore, incluso un ARN SQS valido. Ad esempio, `arn:aws:sqs:region:account:sqs`.
- Include dead-letter SQS ARN (Includi ARN SQS non recapitabili): specifica un ARN SQS non recapitabile di Amazon valido. Ad esempio, `arn:aws:sqs:region:account:deadLetterQueue`.
- Scegli Conferma.



# Tutorial: aggiunta di un crawler AWS Glue

Per questo AWS Glue, ti viene chiesto di analizzare i dati degli arrivi dei principali vettori aerei per calcolare la popolarità degli aeroporti di partenza mese dopo mese. Hai i dati dei voli per l'anno 2016 in formato CSV memorizzati in Amazon S3. Prima di trasformare e analizzare i dati, è necessario catalogarne i metadati in AWS Glue Data Catalog.

In questo tutorial, aggiungiamo un crawler che deduce i metadati da questi registri di volo in Amazon S3 e creiamo una tabella nel catalogo dati.

Argomenti


- Prerequisiti
- Fase 1: aggiunta di un crawler
- Fase 2: esecuzione del crawler
- Fase 3: visualizzazione degli oggetti AWS Glue Data Catalog


## Prerequisiti

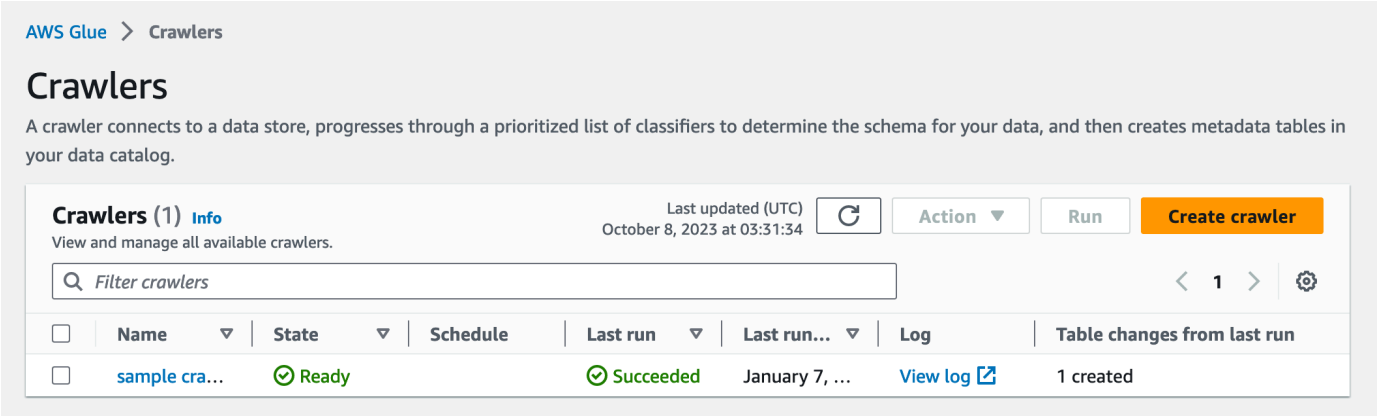
Questo tutorial presuppone che tu abbia un AWS account e un accesso aAWS Glue.

## Fase 1: aggiunta di un crawler

Segui questa procedura per configurare ed eseguire un crawler che estrae i metadati da un file CSV archiviato in Amazon S3.

Per creare un crawler in grado di leggere i file archiviati su Amazon S3

1. Nella console AWS Glue di servizio, nel menu a sinistra, scegli Crawlers.
2. Nella pagina Crawler, scegli Crea crawler. In questo modo viene avviata una serie di pagine che richiedono di specificare i dettagli del crawler.




3. Rinomina il crawler Crawler name (Nome crawler), inserisci **`Flights Data Crawler`**, quindi scegli Next (Avanti).

   I crawler invocano classificatori per dedurre lo schema dei dati. Questo tutorial utilizza il classificatore incorporato per CSV per impostazione predefinita.

4. Per il tipo di origine crawler, scegli Data stores (Datastore) e scegli Next (Avanti).

5. Ora puntiamo il crawler ai dati. Nella pagina Add a data store (Aggiungi datasore), scegli il datastore Amazon S3. Questa esercitazione non usa una connessione, quindi lascia il campo Connection (Connessione) vuoto se è visibile.

   Per l'opzione Crawl data in (Crawling dati), scegli Specified path in another account (Percorso specificato in un altro account). Quindi, nel campo Include path (Percorso di inclusione), inserisci il percorso in cui il crawler può trovare i dati dei voli, che è **`s3://crawler-public-us-east-1/flight/2016/csv`**. Dopo aver inserito il percorso, il titolo di questo campo cambia in Include path (Percorso di inclusione). Seleziona Next (Successivo).

6. È possibile eseguire il crawling di più datasore con un crawler singolo. Tuttavia, in questa esercitazione, stiamo utilizzando un solo datastore, quindi scegli No e poi Next (Successivo).

7. Il crawler ha bisogno delle autorizzazioni per accedere al Data Store e creare oggetti in AWS Glue Data Catalog. Per configurare queste autorizzazioni, scegli Create an IAM role (Crea un ruolo IAM). Il nome del ruolo IAM inizia con `AWSGlueServiceRole-` e, nel campo, inserisci l'ultima parte del nome del ruolo. Inserisci **`CrawlerTutorial`**, quindi seleziona Save (Salva).

> **Note**
>
> Per creare un ruolo IAM, il tuo utente AWS deve avere le autorizzazioni `CreateRole`, `CreatePolicy` e `AttachRolePolicy`.

La procedura guidata crea un ruolo IAM denominato`AWSGlueServiceRole-CrawlerTutorial`, associa la policy AWS gestita `AWSGlueServiceRole` a questo ruolo e aggiunge una policy in linea che consente l'accesso in lettura alla posizione Amazon S3. `s3://crawler-public-us-east-1/flight/2016/csv`

8. Crea una pianificazione per il crawler. Per Frequency (Frequenza), scegli Run on demand (Esegui on demand) e scegli Next (Successivo).
9. I crawler creano le tabelle nel catalogo dati. Un database nel catalogo dati contiene le tabelle. Per prima cosa, scegli Add database (Aggiungi database) per creare un database. Nella finestra popup, inserisci **`test-flights-db`** per il nome del database, quindi scegli Create (crea).

   Quindi, inserisci **`flights`** per Prefix added to tables (Prefisso aggiunto alle tabelle). Utilizza i valori predefiniti per il resto delle opzioni e scegli Next (Successivo).
10. Controlla le selezioni eseguite nella procedura guidata Add crawler (Aggiungi crawler). Se vedi errori, puoi scegliere Back (Indietro) per tornare alle pagine precedenti e apportare modifiche.

    Dopo aver esaminato le informazioni, scegli Finish (Termina) per creare il crawler.

Fase 2: esecuzione del crawler

Dopo aver creato un crawler, la procedura guidata ti reindirizza alla pagina di visualizzazione del crawler. Poiché crei il crawler con una pianificazione on demand, ti viene data la possibilità di eseguirlo.

Per eseguire il crawler

1. Il banner nella parte superiore di questa pagina ti permette di sapere che il crawler è stato creato e chiede se si desidera eseguirlo ora. Seleziona Run it now? (Eseguirlo adesso?) per eseguire il crawler.

   Il banner cambia e mostra i messaggi "Attempting to run" (Tentativo di esecuzione) e "Running" (In esecuzione) per il crawler. Dopo l'avvio del crawler, il banner scompare e la visualizzazione del crawler viene aggiornata per mostrare lo stato avvio del crawler. Dopo un minuto, puoi fare clic sull'icona Refresh (Aggiorna) per aggiornare lo stato del crawler visualizzato nella tabella.

2. Al completamento del crawler, viene visualizzato un nuovo banner che descrive le modifiche apportate dal crawler. Puoi scegliere il test-flights-dblink per visualizzare gli oggetti del Data Catalog.

Fase 3: visualizzazione degli oggetti AWS Glue Data Catalog

Il crawler legge i dati nella posizione di origine e crea tabelle nel catalogo dati. Una tabella è la definizione di metadati che rappresentano i tuoi dati, incluso il relativo schema. Le tabelle del catalogo dati non contengono dati. Vengono invece utilizzate come origine o destinazione in una definizione di processo.

Per visualizzare gli oggetti del catalogo dati creati dal crawler

1. Nel pannello di navigazione a sinistra, sotto Data catalog (Catalogo dati), scegli Database. Qui è possibile visualizzare database `flights-db` creato dal crawler.
2. Nel pannello di navigazione a sinistra, sotto Data catalog (Catalogo dati) e sotto Databases (Database), scegli Tables (Tabelle). Qui è possibile visualizzare la tabella `flightscsv` creata dal crawler. Scegliendo il nome della tabella, è possibile visualizzare le impostazioni, i parametri e le proprietà della tabella. Scorrendo verso il basso nella visualizzazione, puoi visualizzare lo schema, ovvero informazioni sulle colonne e sui tipi di dati della tabella.
3. Se scegli View partitions (Visualizza le partizioni) nella pagina di visualizzazione della tabella, puoi vedere le partizioni create per i dati. La prima colonna è la chiave di partizione.

## Definizione manuale dei metadati

Il AWS Glue Data Catalog è un archivio centrale che archivia i metadati sulle fonti di dati e sui set di dati. Sebbene un crawler sia in grado di eseguire automaticamente il crawling e la compilazione dei metadati per le origini dati supportate, in alcuni scenari potrebbe essere necessario definire manualmente i metadati manualmente nel catalogo dati:

- Formati di dati non supportati: se si dispone di origini dati non supportate dal crawler, occorre definire manualmente i metadati per tali origini dati nel catalogo dati.
- Requisiti personalizzati per i metadati: Crawler di AWS Glue deduce i metadati in base a regole e convenzioni predefinite. Se hai requisiti di metadati specifici che non sono coperti dai metadati Crawler di AWS Glue dedotti, puoi definire manualmente i metadati per soddisfare le tue esigenze
- Standardizzazione e governance dei dati: in alcuni casi, potrebbe essere necessario un maggiore controllo sulle definizioni dei metadati per motivi di governance, conformità o sicurezza dei dati. La

definizione manuale dei metadati consente di garantire che i metadati rispettino gli standard e le politiche dell'organizzazione.

- Segnaposto per una importazione dei dati futura: se si dispone di origini dati che non sono immediatamente disponibili o accessibili, è possibile creare tabelle di schema vuote come segnaposto. Una volta che le origini dati diventano disponibili, è possibile popolare le tabelle con i dati effettivi, mantenendo al contempo la struttura predefinita.

Per definire i metadati manualmente, puoi utilizzare la AWS Glue console, la console Lake Formation, l' AWS Glue API o AWS Command Line Interface (AWS CLI). È possibile creare database, tabelle e partizioni, e specificare proprietà dei metadati come nomi di colonne, tipi di dati, descrizioni e altri attributi.

## Creazione di database

I database vengono utilizzati per organizzare le tabelle dei metadati nella AWS Glue. Quando si definisce una tabella in AWS Glue Data Catalog, la si aggiunge a un database. Una tabella può essere in un solo database.

Il tuo database può contenere tabelle che definiscono dati provenienti da datastore diversi. Questi dati possono includere oggetti in Amazon Simple Storage Service (Amazon S3) e tabelle relazionali in Amazon Relational Database Service.

> **Note**
>
> Quando elimini un database da AWS Glue Data Catalog, tutte le tabelle in esso contenute vengono eliminate.

Per visualizzare l'elenco dei database, accedi Console di gestione AWS e apri la AWS Glue console all'indirizzo https://console.aws.amazon.com/glue/. Scegli Databases (Database) e quindi un nome di database nell'elenco per visualizzarne i dettagli.

Nella scheda Databases (Database) nella console AWS Glue puoi aggiungere, modificare ed eliminare database:

- Per creare un nuovo database, scegli Add database (Aggiungi database) e fornisci un nome e una descrizione. Per compatibilità con altri store di metadati, ad esempio Apache Hive, il nome è in caratteri minuscoli.

> **Note**
>
> Se prevedi di accedere al database da Amazon Athena, fornisci un nome con solo caratteri alfanumerici e di sottolineatura. Per ulteriori informazioni, consulta Nomi di Athena.

- Per modificare la descrizione di un database, seleziona la casella di controllo accanto al nome del database e scegli Edit (Modifica).
- Per eliminare un database, seleziona la casella di controllo accanto al nome del database e scegli Remove (Rimuovi).
- Per visualizzare l'elenco delle tabelle contenute nel database, scegli il nome del database e le proprietà del database mostreranno tutte le tabelle.

Per modificare il database in un crawler scrive devi modificare la definizione del crawler. Per ulteriori informazioni, consulta Utilizzo dei crawler per compilare il Catalogo dati.

Collegamenti di risorsa al database

La console AWS Glue è stata recentemente aggiornata. La versione corrente della console non supporta i collegamenti di risorsa al database.

Il catalogo dati può anche contenere collegamenti di risorsa ai database. Un collegamento di risorsa al database è un collegamento a un database locale o condiviso. Al momento, puoi creare collegamenti di risorsa solo in AWS Lake Formation. Dopo aver creato un collegamento di risorsa a un database, è possibile utilizzare il nome del collegamento di risorsa ovunque desideri utilizzare il nome del database. Insieme ai database di cui si è proprietari o che sono condivisi con l'utente, i collegamenti di risorsa al database vengono restituiti da `glue:GetDatabases()` e vengono visualizzati come voci nella pagina Database della console AWS Glue.

Il catalogo dati può anche contenere collegamenti di risorsa alla tabella.

Per ulteriori informazioni sui collegamenti di risorsa, consulta Creazione di collegamenti di risorsa nella Guida per gli sviluppatori di AWS Lake Formation .

## Creazione di tabelle

Anche se l'esecuzione di un crawler è il metodo consigliato per fare l'inventario dei dati negli archivi dati, puoi aggiungere tabelle di metadati manualmente. AWS Glue Data Catalog Questo approccio

consente di avere un maggiore controllo sulle definizioni dei metadati e di personalizzarle in base a requisiti specifici.

È anche possibile aggiungere manualmente definizioni di tabella nel catalogo dati in questi modi:

- Usando la console AWS Glue per creare manualmente una tabella nel AWS Glue Data Catalog. Per ulteriori informazioni, consulta Creazione di tabelle mediante la console.
- Usando l'operazione `CreateTable` nell'AWS Glue API per creare una tabella nel AWS Glue Data Catalog Per ulteriori informazioni, consulta CreateTable azione (Python: create_table).
- Utilizza modelli. CloudFormation Per ulteriori informazioni, consulta AWS CloudFormation per AWS Glue.

Quando definisci manualmente una tabella utilizzando la console o un'API, specifichi lo schema della tabella e il valore di un campo di classificazione che indica il tipo e il formato dei dati nell'origine dati. Se un crawler crea la tabella, lo schema e il formato dei dati sono determinati da un classificatore incorporato o da uno personalizzato. Per ulteriori informazioni sulla creazione di una tabella tramite la console AWS Glue, consulta Creazione di tabelle mediante la console.

Argomenti

- Partizioni tabella
- Collegamenti di risorsa della tabella
- Creazione di tabelle mediante la console
- Creazione di indici di partizione
- Aggiornamento delle tabelle del catalogo dati create manualmente usando i crawler
- Proprietà della tabella del catalogo dati

## Partizioni tabella

Un a definizione di tabella AWS Glue di una cartella Amazon Simple Storage Service (Amazon S3) può descrivere una tabella partizionata. Ad esempio, per migliorare le prestazioni delle query, una tabella partizionata potrebbe separare i dati mensili in diversi file utilizzando il nome del mese come chiave. Le definizioni di tabella in AWS Glue includono la chiave di partizionamento di una tabella. Quando AWS Glue valuta i dati nelle cartelle Amazon S3 per catalogare una tabella, determina se viene aggiunta una singola tabella o una tabella partizionata.

È possibile creare indici delle partizioni su una tabella per recuperare un sottoinsieme delle partizioni invece di caricare tutte le partizioni nella tabella. Per ulteriori informazioni sull'utilizzo degli indici delle partizioni, consulta Creazione di indici di partizione.

Perché AWS Glue possa creare una tabella partizionata per una cartella Amazon S3, devono essere vere tutte le condizioni seguenti:

- Gli schemi dei file sono simili, come stabilito da AWS Glue.
- Il formato dati dei file è lo stesso.
- Il formato di compressione dei file è lo stesso.

Ad esempio, puoi avere un tuo bucket Amazon S3 denominato `my-app-bucket`, in cui vengono memorizzati i dati di vendita delle app iOS e Android. I dati sono partizionati in base ad anno, mese e giorno. I file di dati per le vendite iOS e Android hanno lo stesso schema, formato dei dati e formato di compressione. In AWS Glue Data Catalog, il AWS Glue crawler crea una definizione di tabella con chiavi di partizionamento per anno, mese e giorno.

Il seguente elenco Amazon S3 di `my-app-bucket` mostra alcune delle partizioni. Il simbolo = viene utilizzato per assegnare i valori di chiave di partizione.

```
my-app-bucket/Sales/year=2010/month=feb/day=1/iOS.csv
my-app-bucket/Sales/year=2010/month=feb/day=1/Android.csv
my-app-bucket/Sales/year=2010/month=feb/day=2/iOS.csv
my-app-bucket/Sales/year=2010/month=feb/day=2/Android.csv
...
my-app-bucket/Sales/year=2017/month=feb/day=4/iOS.csv
my-app-bucket/Sales/year=2017/month=feb/day=4/Android.csv
```

Collegamenti di risorsa della tabella

> La console AWS Glue è stata recentemente aggiornata. La versione corrente della console non supporta i collegamenti di risorsa alla tabella.

Il catalogo dati può anche contenere collegamenti di risorsa della tabella. Un collegamento di risorsa della tabella è un collegamento a un database locale o condiviso. Al momento, puoi creare collegamenti di risorsa solo in AWS Lake Formation. Dopo aver creato un collegamento di risorsa

a una tabella, è possibile utilizzare il nome del collegamento di risorsa ovunque desideri utilizzare il nome della tabella. Insieme alle tabelle di cui si è proprietari o che sono condivisi con l'utente, i collegamenti di risorsa alla tabella vengono restituiti da `glue:GetTables()` e vengono visualizzati come voci nella pagina Tables (Tabelle) della console AWS Glue.

Il catalogo dati può anche contenere collegamenti di risorsa ai database.

Per ulteriori informazioni sui collegamenti di risorsa, consulta Creazione di collegamenti di risorsa nella Guida per gli sviluppatori di AWS Lake Formation .

Creazione di tabelle mediante la console

Una tabella in AWS Glue Data Catalog è la definizione di metadati che rappresenta i dati in un archivio dati. Puoi creare tabelle quando esegui un crawler oppure puoi creare una tabella manualmente nella console AWS Glue . L'elenco Tables (Tabelle) nella console AWS Glue mostra i valori dei metadati della tabella. Usa le definizioni di tabella per specificare origini e destinazioni al momento della creazione di processi ETL (estrazione, trasformazione e caricamento).

> **Note**
>
> A seguito delle recenti modifiche alla console di AWS gestione, potrebbe essere necessario modificare i ruoli IAM esistenti per disporre dell'`SearchTables`autorizzazione. Per la creazione di nuovi ruoli, l'autorizzazione dell'API `SearchTables` è già stata aggiunta come impostazione predefinita.

Per iniziare, accedi Console di gestione AWS e apri la AWS Glue console all'indirizzo https://console.aws.amazon.com/glue/. Scegli la scheda Tables (Tabelle) e usa il pulsante Add tables (Aggiungi tabelle) per creare tabelle con un crawler o digitando manualmente gli attributi.

Aggiunta di tabelle nella console

Per usare un crawler per aggiungere tabelle, scegli Add tables (Aggiungi tabelle), Add tables using a crawler (Aggiungi tabelle utilizzando un crawler). Quindi segui le istruzioni nella procedura guidata Add crawler (Aggiungi crawler). Quando il crawler viene eseguito, vengono aggiunte tabelle al AWS Glue Data Catalog. Per ulteriori informazioni, consulta Utilizzo dei crawler per compilare il Catalogo dati.

Se conosci gli attributi necessari per creare una definizione di tabella Amazon Simple Storage Service (Amazon S3) nel catalogo dati, puoi crearla con la procedura guidata di creazione di tabelle.

Scegli Add tables (Aggiungi tabelle), Add table manually (Aggiungi tabella manualmente) e segui le istruzioni della procedura guidata Add tables (Aggiungi tabella).

Quando aggiungi una tabella manualmente attraverso la console, considera quanto segue:

- Se prevedi di accedere alla tabella da Amazon Athena, fornisci un nome con solo caratteri alfanumerici e di sottolineatura. Per ulteriori informazioni, consulta Nomi di Athena.
- L'ubicazione dei dati di origine deve essere un percorso Amazon S3.
- Il formato dei dati deve corrispondere a uno dei formati elencati nella procedura guidata. La classificazione corrispondente e SerDe le altre proprietà della tabella vengono compilate automaticamente in base al formato scelto. È possibile definire tabelle con i seguenti formati:

  Avro

  Formato binario Apache Avro JSON.

  CSV

  Character separated values. È inoltre possibile specificare il delimitatore di virgola, barra verticale, punto e virgola, tabulazione o Ctrl-A.

  JSON

  JavaScript Notazione dell'oggetto.

  XML

  Formato Extensible Markup Language. Specificate il tag XML che definisce una riga nei dati. Le colonne sono definite all'interno dei tag di riga.

  Parquet

  Storage a colonne Apache Parquet.

  ORC

  Formato di file Optimized Row Columnar (ORC). Un formato progettato per archiviare in modo efficiente i dati Hive.

- È possibile definire una chiave di partizione per la tabella.
- Attualmente, le tabelle partizionate create con la console non possono essere utilizzate nei lavori ETL.

### Attributi della tabella

Di seguito sono elencati alcuni importanti attributi della tua tabella:

Name

Il nome viene stabilito in fase di creazione della tabella e non può essere modificato. Dovrai fare riferimento a un nome di tabella in molte operazioni AWS Glue.

Database

L'oggetto container in cui la tabella risiede. Questo oggetto contiene un'organizzazione delle tabelle che esiste all'interno dell'archivio dati AWS Glue Data Catalog e potrebbe differire da quella dell'organizzazione del data store. Quando elimini un database, anche tutte le tabelle in esso contenute vengono eliminate dal catalogo dati.

Description

La descrizione della tabella. È possibile scrivere una descrizione per aiutare a comprendere i contenuti della tabella.

Formato della tabella

Specificate la creazione di una AWS Glue tabella standard o di una tabella in formato Apache Iceberg.

Il catalogo dati offre le seguenti opzioni di ottimizzazione delle tabelle per gestire l'archiviazione delle tabelle e migliorare le prestazioni delle query per le tabelle Iceberg.

- Compattazione: I file di dati vengono uniti e riscritti per rimuovere i dati obsoleti e consolidare i dati frammentati in file più grandi ed efficienti.
- Conservazione degli snapshot: gli snapshot sono versioni con data e ora di una tabella Iceberg. Le configurazioni di conservazione degli snapshot consentono ai clienti di stabilire per quanto tempo conservare gli snapshot e quanti snapshot conservare. La configurazione di un ottimizzatore di conservazione degli snapshot può aiutare a gestire il sovraccarico di archiviazione rimuovendo gli snapshot più vecchi e non necessari e i relativi file sottostanti.
- Eliminazione di file orfani: i file orfani sono file a cui non fanno più riferimento i metadati della tabella Iceberg. Questi file possono accumularsi nel tempo, soprattutto dopo operazioni come l'eliminazione di tabelle o i processi ETL non riusciti. L'abilitazione dell'eliminazione dei file orfani consente di AWS Glue identificare e rimuovere periodicamente questi file non necessari, liberando spazio di archiviazione.

Per ulteriori informazioni, consulta Ottimizzazione delle tabelle Iceberg.

Configurazione di ottimizzazione

È possibile utilizzare le impostazioni predefinite o personalizzare le impostazioni per abilitare gli ottimizzatori di tabella.

Ruolo IAM

Per eseguire gli ottimizzatori delle tabelle, il servizio assume un ruolo IAM per conto dell'utente. Puoi scegliere un ruolo IAM utilizzando il menu a discesa. Assicurati che il ruolo disponga delle autorizzazioni necessarie per abilitare la compattazione.

Consulta Prerequisiti per l'ottimizzazione delle tabelle per ulteriori informazioni sulle autorizzazioni necessarie per il ruolo IAM.

Location (Ubicazione)

Il puntatore all'ubicazione dei dati in un datastore rappresentato da questa definizione di tabella.

Classificazione

Un valore di categorizzazione fornito al momento della creazione della tabella. Solitamente viene scritto quando un crawler viene eseguito e specifica il formato dei dati di origine.

Ultimo aggiornamento

Data e ora (UTC) in cui questa tabella è stata aggiornata nel catalogo dati.

Data aggiunta

Data e ora (UTC) in cui questa tabella è stata aggiunta al catalogo dati.

Deprecated

Se AWS Glue rileva che una tabella nel catalogo dati non è più presente nel datastore originale, la contrassegna come obsoleta nel catalogo dati. Se esegui un processo che fa riferimento a una tabella obsoleta, il processo potrebbe fallire. Modifica processi che fanno riferimento a tabelle obsolete per rimuoverle come origini e destinazioni. Consigliamo di eliminare le tabelle obsolete quando non sono più necessarie.

Connessione

Se AWS Glue richiede una connessione al datastore, il nome della connessione è associato alla tabella.

Visualizzazione e gestione dei dettagli della tabella

Per vedere i dettagli di una tabella esistente, scegli il nome tabella nell'elenco quindi scegli Action, View details (Operazione, Mostra dettagli).

I dettagli tabella includono le proprietà della tabella e del relativo schema. Questa vista mostra lo schema della tabella, inclusi i nomi colonna nell'ordine definito per la tabella, i tipi di dati e le colonne chiave per le partizioni. Se una colonna è di tipo complesso, puoi scegliere View properties (Visualizza proprietà) per visualizzare i dettagli della struttura di tale campo, come mostrato nell'esempio seguente:

```
{
"StorageDescriptor":
    {
      "cols": {
         "FieldSchema": [
           {
             "name": "primary-1",
             "type": "CHAR",
             "comment": ""
           },
           {
             "name": "second ",
             "type": "STRING",
             "comment": ""
           }
         ]
      },
      "location": "s3://aws-logs-111122223333-us-east-1",
      "inputFormat": "",
      "outputFormat": "org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat",
      "compressed": "false",
      "numBuckets": "0",
      "SerDeInfo": {
          "name": "",
          "serializationLib": "org.apache.hadoop.hive.serde2.OpenCSVSerde",
          "parameters": {
              "separatorChar": "|"
           }
      },
      "bucketCols": [],
      "sortCols": [],
      "parameters": {},
```

```
      "SkewedInfo": {},
      "storedAsSubDirectories": "false"
    },
    "parameters": {
       "classification": "csv"
    }
}
```

Per ulteriori informazioni sulle proprietà di una tabella, come `StorageDescriptor`, consulta StorageDescriptor struttura.

Per modificare lo schema di una tabella, scegli Edit schema (Modifica schema) per aggiungere o rimuovere colonne, modificarne i nomi e modificare i tipi di dati.

Per confrontare diverse versioni di una tabella, incluso lo schema, scegli Confronta versioni per visualizzare un side-by-side confronto tra due versioni dello schema per una tabella. Per ulteriori informazioni, consulta Confronto delle versioni dello schema della tabella.

Per visualizzare i file che costituiscono una partizione Amazon S3, scegli View partition (Visualizza partizione). Per tabelle Amazon S3, la colonna Key (Chiave) visualizza le chiavi di partizione usate per partizionare la tabella nel datastore di origine. Il partizionamento è un modo per dividere una tabella in parti correlate in base ai valori di una colonna chiave, ad esempio data, posizione o reparto. Per ulteriori informazioni sulle partizioni, cercare "hive partitioning." su Internet.

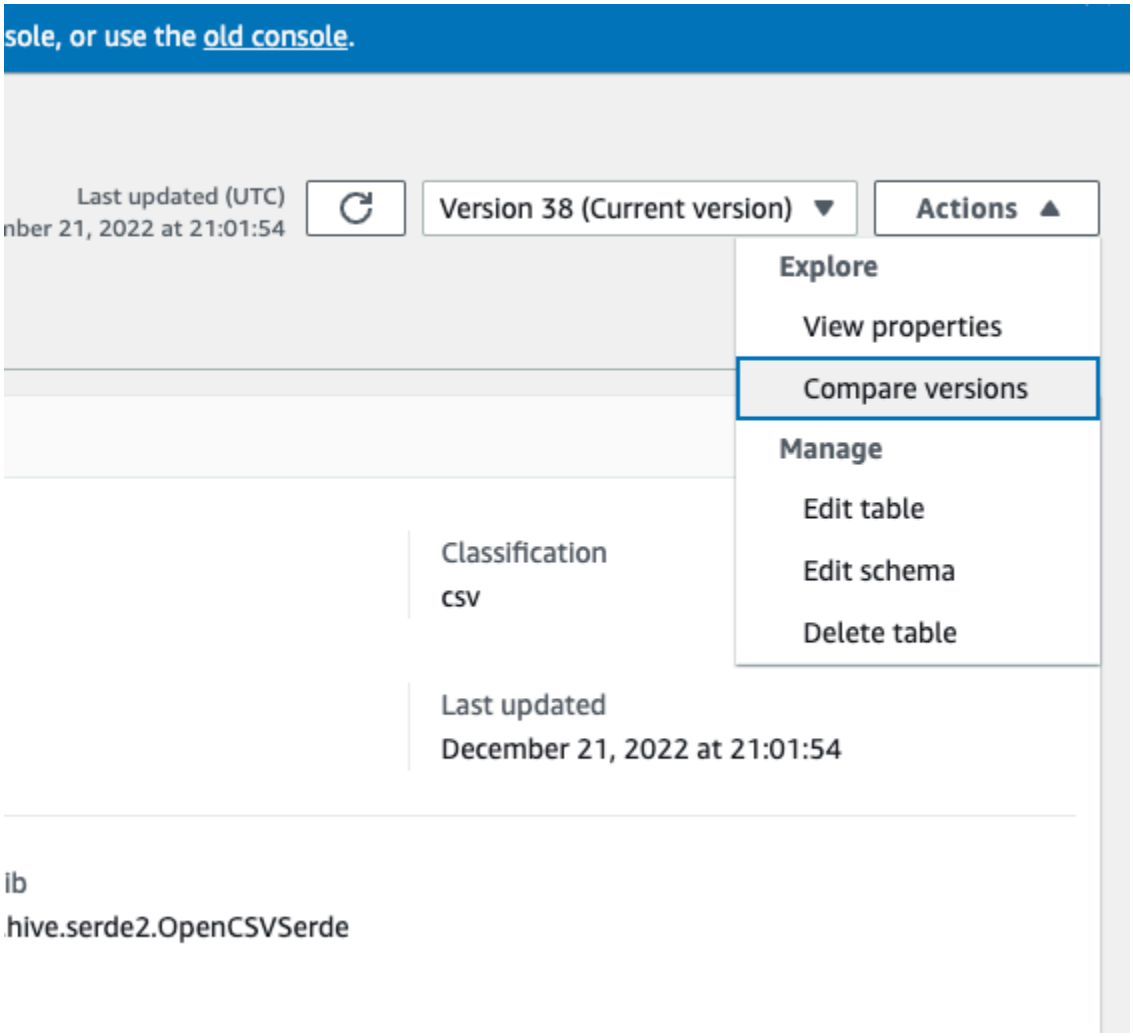
> **Note**
>
> Per ottenere step-by-step indicazioni sulla visualizzazione dei dettagli di una tabella, consulta il tutorial Esplora la tabella nella console.

### Confronto delle versioni dello schema della tabella

Quando si confrontano due versioni di schemi di tabelle, è possibile confrontare le modifiche delle righe nidificate espandendo e comprimendo le righe nidificate, confrontare gli schemi di due versioni e visualizzare le side-by-side proprietà delle tabelle. side-by-side

#### Come confrontare le versioni

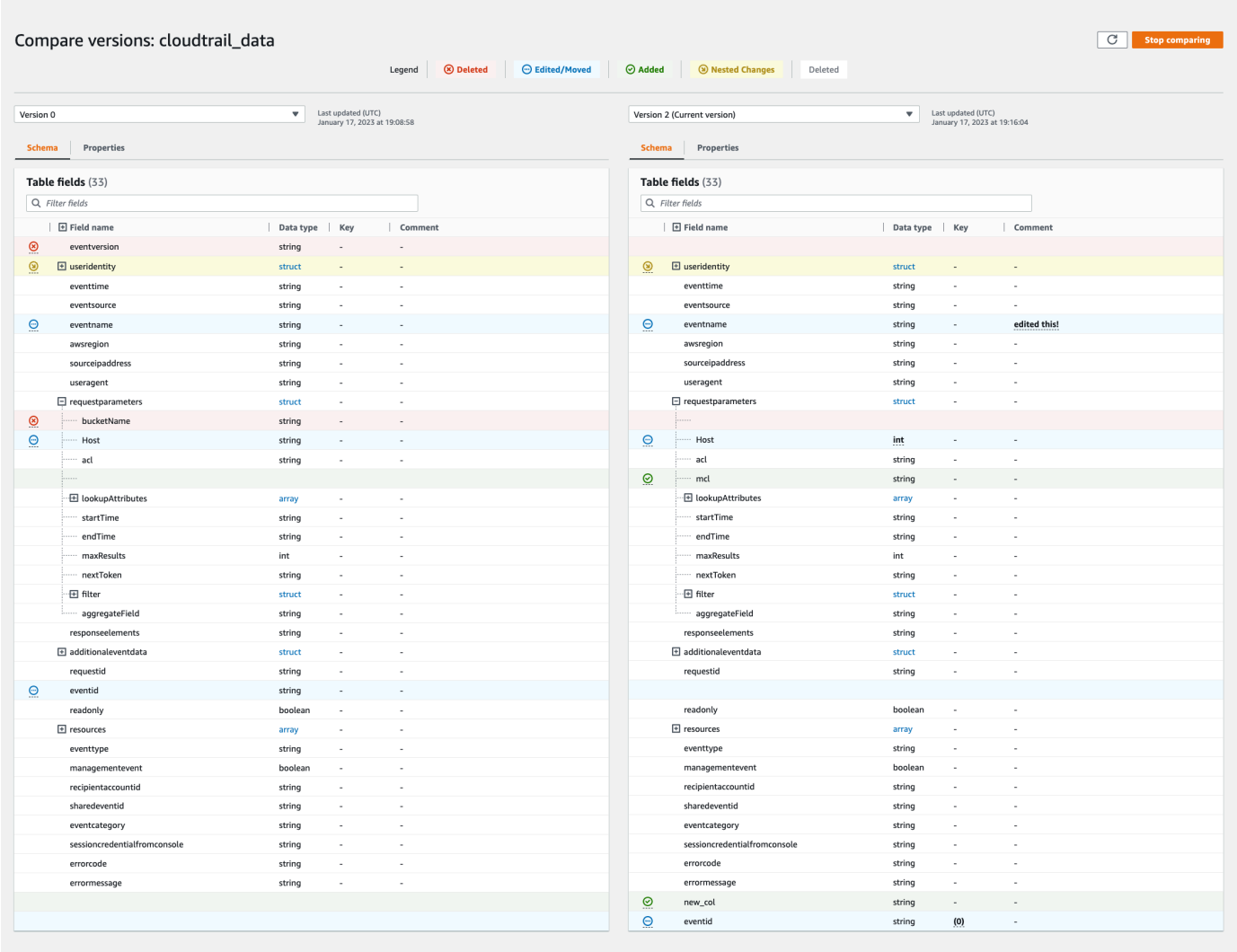
1. Dalla console AWS Glue, scegli Tabelle, quindi Azioni e scegli Confronta versioni.




2. Scegli una versione da confrontare scegliendo il menu a discesa delle versioni. Quando si confrontano gli schemi, la scheda Schema è evidenziata in arancione.
3. Quando si confrontano le tabelle tra due versioni, gli schemi delle tabelle vengono visualizzati sul lato sinistro e destro dello schermo. Ciò consente di determinare visivamente le modifiche confrontando il nome della colonna, il tipo di dati, la chiave e i campi di commento. side-by-side Quando viene apportata una modifica, un'icona colorata mostra il tipo di modifica apportata.

   - Eliminata: contrassegnata da un'icona rossa, indica dove la colonna è stata rimossa da una versione precedente dello schema della tabella.
   - Modificata o spostata: contrassegnata da un'icona blu, indica dove la colonna è stata modificata o spostata in una versione più recente dello schema della tabella.
   - Aggiunta: contrassegnata da un'icona verde, indica dove la colonna è stata aggiunta a una versione più recente dello schema della tabella.

- Modifiche annidate: contrassegnata da un'icona gialla, indica le modifiche nella colonna annidata. Scegli la colonna da espandere e visualizza le colonne che sono state eliminate, modificate, spostate o aggiunte.




4. Utilizza la barra di ricerca dei campi di filtro per visualizzare i campi in base ai caratteri che inserisci qui. Se immetti un nome di colonna in una delle versioni della tabella, i campi filtrati vengono visualizzati in entrambe le versioni della tabella per mostrare dove sono state apportate le modifiche.
5. Per confrontare le proprietà, scegli la scheda Proprietà.
6. Per interrompere il confronto tra le versioni, scegli Interrompi confronto per tornare all'elenco delle tabelle.

## Creazione di indici di partizione

Nel corso del tempo, centinaia di migliaia di partizioni vengono aggiunte a una tabella. L'GetPartitions API viene utilizzata per recuperare le partizioni nella tabella. L'API restituisce partizioni che corrispondono all'espressione fornita nella richiesta.

Prendiamo come esempio una tabella sales_data che è partizionata dalle chiavi Country, Category, Year, Month and creationDate. Se si desidera ottenere i dati di vendita per tutti gli articoli venduti per la categoria Books nell'anno 2020, dopo il 2020-08-15, occorre inviare una richiesta `GetPartitions` con l'espressione “Category = 'Books' and creationDate > '2020-08-15'” al catalogo dati.

Se nella tabella non sono presenti indici delle partizioni, AWS Glue carica tutte le partizioni della tabella e quindi filtra le partizioni caricate utilizzando l'espressione di query fornita dall'utente nella richiesta `GetPartitions`. L'esecuzione della query richiede più tempo man mano che il numero di partizioni aumenta in una tabella senza indici. Con un indice, la query `GetPartitions` cercherà di recuperare un sottoinsieme delle partizioni invece di caricare tutte le partizioni nella tabella.

Argomenti


- Informazioni sugli indici delle partizioni
- Creazione di una tabella con indici delle partizioni
- Aggiunta di un indice di partizione a una tabella esistente
- Descrizione degli indici delle partizioni su una tabella
- Limitazioni all'utilizzo degli indici delle partizioni
- Utilizzo degli indici per una chiamata ottimizzata GetPartitions
- Integrazione con i motori


### Informazioni sugli indici delle partizioni

Quando crei un indice di partizione, specifichi un elenco di chiavi di partizione già esistenti in una determinata tabella. L'indice delle partizioni è un sottoelenco di chiavi di partizione definite nella tabella. Un indice di partizione può essere creato su qualsiasi permutazione delle chiavi di partizione definite nella tabella. Per la tabella sales_data di cui sopra, gli indici possibili sono (country, category, creationDate), (country, category, year), (country, category), (country), (category, country, year, month) e così via.

Il catalogo dati concatenerà i valori delle partizioni nell'ordine fornito al momento della creazione dell'indice. L'indice viene creato in modo coerente man mano che le partizioni vengono aggiunte alla

tabella. Gli indici possono essere creati per i tipi di colonna String (string, char e varchar), Numeric (int, bigint, long, tinyint e smallint) e Date (yyyy-MM-dd).

Tipi di dati supportati

- Data: una data in formato ISO, ad esempio `YYYY-MM-DD`. Ad esempio, la data `2020-08-15`. Il formato utilizza i trattini (-) per separare l'anno, il mese e il giorno. L'intervallo consentito per le date di indicizzazione va da `0000-01-01` a `9999-12-31`.
- String: una stringa letterale racchiusa tra virgolette singole o doppie.
- Char: lunghezza fissa dei dati dei caratteri specificata tra 1 e 255, ad esempio char(10).
- Varchar: lunghezza variabile dei dati dei caratteri specificata tra 1 e 65.535, ad esempio varchar(10).
- Numeric (numerico): int, bigint, long, tinyint e smallint

Gli indici sui tipi di dati Numeric, String e Date supportano gli operatori =, >, >=, <, <= e between. La soluzione di indicizzazione attualmente supporta solo l'operatore logico `AND`. Le sottoespressioni con gli operatori "LIKE", "IN", "OR" e "NOT" vengono ignorate nell'espressione il filtraggio con indice. Il filtraggio per la sottoespressione ignorata viene eseguito sulle partizioni recuperate dopo aver applicato il filtraggio dell'indice.

Per ogni partizione aggiunta a una tabella, viene creato un elemento indice corrispondente. Per una tabella con partizioni "n", l'indice di partizione 1 risulterà come elementi indice di partizione "n". L'indice di partizione "m" sulla stessa tabella risulterà come elementi di indice di partizione "m*n". Ogni elemento dell'indice della partizione verrà addebitato in base alla policy dei prezzi AWS Glue corrente per l'archiviazione del catalogo dati. Per informazioni dettagliate sui prezzi degli oggetti di archiviazione, consulta Prezzi di AWS Glue.

Creazione di una tabella con indici delle partizioni

È possibile creare un indice di partizione durante la creazione di una tabella. La richiesta `CreateTable` utilizza un elenco di oggetti `PartitionIndex` come input. È possibile creare un massimo di 3 indici delle partizioni su una determinata tabella. Ogni indice di partizione richiede un nome e un elenco di `partitionKeys` definite per la tabella. Gli indici creati su una tabella possono essere recuperati utilizzando l'API `GetPartitionIndexes`

### Aggiunta di un indice di partizione a una tabella esistente

Per aggiungere un indice di partizione a una tabella esistente, utilizza l'operazione `CreatePartitionIndex`. Puoi creare un solo `PartitionIndex` per ogni operazione `CreatePartitionIndex`. L'aggiunta di un indice non influisce sulla disponibilità di una tabella, poiché durante la creazione degli indici la tabella continua a essere disponibile.

Lo stato dell'indice per una partizione aggiunta è impostato su CREATING (CREAZIONE IN CORSO) e la creazione dei dati di indice viene avviata. Se il processo per la creazione degli indici ha esito positivo, IndexStatus viene aggiornato in ACTIVE (ATTIVO) e, per un processo non riuscito, lo stato dell'indice viene aggiornato a FAILED (NON RIUSCITO). La creazione dell'indice può avere esito negativo per diversi motivi ed è possibile utilizzare l'operaizione `GetPartitionIndexes` per recuperare i dettagli dell'errore. I possibili errori sono:

- ENCRYPTED_PARTITION_ERROR: la creazione di indici su una tabella con partizioni crittografate non è supportata.
- INVALID_PARTITION_TYPE_DATA_ERROR: riscontrato quando il parametro `partitionKey` non è un valore valido per il tipo di dati della `partitionKey` corrispondente. Ad esempio: una `partitionKey` con il tipo di dati 'int' ha un valore 'foo'.
- MISSING_PARTITION_VALUE_ERROR: riscontrato quando il `partitionValue` per un `indexedKey` non è presente. Ciò può accadere quando una tabella non è partizionata in modo coerente.
- UNSUPPORTED_PARTITION_CHARACTER_ERROR: riscontrato quando il valore di una chiave di partizione indicizzata contiene i caratteri \u0000, \u0001 o \u0002
- INTERNAL_ERROR: si è verificato un errore interno durante la creazione degli indici.

### Descrizione degli indici delle partizioni su una tabella

Per recuperare gli indici delle partizioni creati su una tabella, utilizza l'operazione `GetPartitionIndexes`. La parte risposta mostra tutti gli indici della tabella insieme allo stato corrente di ciascun indice (`IndexStatus`).

L'`IndexStatus` di un indice può essere uno dei seguenti:

- `CREATING`: l'indice è in fase di creazione e non è ancora disponibile per l'uso.
- `ACTIVE`: l'indice è pronto per l'uso. Le richieste possono utilizzare l'indice per eseguire una query ottimizzata.

- `DELETING`: l'indice è attualmente in fase di eliminazione e non può essere più utilizzato. Un indice nello stato attivo può essere eliminato utilizzando la richiesta `DeletePartitionIndex`, che sposta lo stato da ACTIVE (ATTIVO) a DELETING (ELIMINAZIONE IN CORSO).
- `FAILED`: la creazione dell'indice su una tabella esistente non è riuscita. Ogni tabella memorizza gli ultimi 10 indici non riusciti.

Le possibili transizioni di stato per gli indici creati su una tabella esistente sono le seguenti:

- CREATING → ACTIVE → DELETING (CREAZIONE → ATTIVO → ELIMINAZIONE IN CORSO)
- CREATING → FAILED (IN FASE DI CREAZIONE → NON RIUSCITO)

Limitazioni all'utilizzo degli indici delle partizioni

Dopo aver creato un indice delle partizioni, prendi nota delle seguenti modifiche alla funzionalità di tabella e partizione:

Creazione di una nuova partizione (dopo l'aggiunta dell'indice)

Dopo aver creato un indice delle partizioni su una tabella, tutte le nuove partizioni aggiunte alla tabella verranno convalidate per i controlli del tipo di dati per le chiavi indicizzate. Il valore di partizione delle chiavi indicizzate verrà convalidato per il formato del tipo di dati. Se il controllo del tipo di dati non riesce, l'operazione di creazione della partizione avrà esito negativo. Per la tabella sales_data, se viene creato un indice per le chiavi (category [categoria], year [anno]) in cui la categoria è di tipo `string` e l'anno di tipo `int`, la creazione della nuova partizione con un valore YEAR come “foo” non riuscirà.

Dopo che gli indici sono abilitati, l'aggiunta di partizioni con valori chiave indicizzati aventi i caratteri U +0000, U+00001 e U+0002 inizierà ad avere esito negativo.

Aggiornamenti tabella

Una volta creato un indice delle partizioni in una tabella, non è possibile modificare i nomi delle chiavi di partizione esistenti né modificare il tipo o l'ordine delle chiavi registrate con l'indice.

Utilizzo degli indici per una chiamata ottimizzata GetPartitions

Quando chiami `GetPartitions` su una tabella con un indice, puoi includere un'espressione e, se possibile, il catalogo dati utilizzerà un indice. La prima chiave dell'indice deve essere trasmessa

nell'espressione per utilizzare gli indici nel filtro. L'ottimizzazione dell'indice nel filtraggio viene applicata come miglior tentativo. Il catalogo dati tenta di utilizzare il più possibile l'ottimizzazione dell'indice, ma in caso di indice mancante o di operatore non supportato, ritorna all'implementazione esistente di caricamento di tutte le partizioni.

Per la tabella sales_data di cui sopra, aggiungiamo l'indice [Country, Category, Year]. Se "Country" non viene trasmesso nell'espressione, l'indice registrato non sarà in grado di filtrare le partizioni utilizzando gli indici. È possibile aggiungere fino a 3 indici per supportare vari modelli di query.

Prendiamo alcune espressioni di esempio e vediamo come funzionano gli indici:

| Espressioni | Come verrà utilizzato l'indice |
|---|---|
| Country = 'US' | L'indice verrà utilizzato per filtrare le partizioni. |
| Country = 'US' and Category = 'Shoes' | L'indice verrà utilizzato per filtrare le partizioni. |
| Category = 'Shoes' | Gli indici non utilizzati come "country" non verranno specificati nell'espressione. Tutte le partizioni verranno caricate per restituire una risposta. |
| Country = 'US' and Category = 'Shoes' and Year > '2018' | L'indice verrà utilizzato per filtrare le partizioni. |
| Country = 'US' and Category = 'Shoes' and Year > '2018' and month = 2 | L'indice verrà utilizzato per recuperare tutte le partizioni con country = "US" e category = "shoes" e year > 2018. Quindi, verrà eseguito il filtraggio sull'espressione del mese. |
| Country = 'US' AND Category = 'Shoes' OR Year > '2018' | Gli indici non verranno utilizzati poiché nell'espressone è presente l'operatore OR. |
| Country = 'US' AND Category = 'Shoes' AND (Year = 2017 OR Year = '2018') | L'indice verrà utilizzato per recuperare tutte le partizioni con country = "US" e category = "shoes" e quindi verrà eseguito il filtraggio sull'espressione del mese. |

| Espressioni | Come verrà utilizzato l'indice |
| --- | --- |
| Country in ('US', 'UK') AND Category = 'Shoes' | Gli indici non verranno utilizzati per il filtraggi o poiché l'operatore `IN` al momento non è supportato. |
| Country = 'US' AND Category in ('Shoes', 'Books') | L'indice verrà utilizzato per recuperare tutte le partizioni con country = "US", quindi verrà eseguito il filtro sull'espressione Category. |
| Country = 'US' AND Category in ('Shoes', 'Books') AND (creationDate > '2023-9-01') | L'indice verrà utilizzato per recuperare tutte le partizioni con country = "US", con creationDate > '2023-9-01', quindi verrà eseguito il filtro sull'espressione Category. |

Integrazione con i motori

Redshift Spectrum, Amazon EMR ed AWS Glue ETL Spark DataFrames sono in grado di utilizzare gli indici per recuperare le partizioni dopo che gli indici sono in uno stato ATTIVO in. AWS Glue Athena e i frame dinamici AWS Glue ETL richiedono di eseguire ulteriori passaggi per utilizzare gli indici per il miglioramento delle query.

Abilitare la filtrazione delle partizioni

Per abilitare la filtrazione delle partizioni in Athena, è necessario aggiornare le proprietà della tabella come segue:

1. Nella console, in Data Catalog, scegli Tabelle. AWS Glue
2. Scegliere una tabella .
3. In Azioni, scegliere Modifica tabella.
4. In Proprietà della tabella, aggiungere quanto segue:
   - Chiave – `partition_filtering.enabled`
   - Valore: `true`
5. Scegli Applica.

In alternativa, è possibile impostare questo parametro eseguendo una query MODIFICA DELLE PROPRIETÀ DEL SET DI TABELLE in Athena.

```
ALTER TABLE partition_index.table_with_index
SET TBLPROPERTIES ('partition_filtering.enabled' = 'true')
```

Aggiornamento delle tabelle del catalogo dati create manualmente usando i crawler

Potresti voler creare le AWS Glue Data Catalog tabelle manualmente e poi mantenerle aggiornate con i AWS Glue crawler. I crawler in esecuzione su una pianificazione possono aggiungere nuove partizioni e aggiornare le tabelle con qualsiasi modifica dello schema. Questo vale anche per le tabelle migrate da un metastore Apache Hive.

Per fare ciò, quando definisci un crawler, invece di specificare uno o più datastoe come origine di un crawling, puoi specificare una o più tabelle del catalogo dati esistenti. Il crawler esegue quindi il crawling dei datastore specificati dalle tabelle del catalogo. In questo caso, non vengono create nuove tabelle mentre le tabelle create manualmente vengono aggiornate.

Di seguito sono riportati altri motivi per cui puoi creare manualmente le tabelle di catalogo e specificarle come origini del crawler:

- Desideri scegliere il nome della tabella del catalogo e non fare affidamento sull'algoritmo di denominazione della tabella del catalogo.
- Desideri impedire la creazione di nuove tabelle nel caso in cui i file con un formato che potrebbe interrompere il rilevamento della partizione vengano erroneamente salvati nel percorso dell'origine dati.

Per ulteriori informazioni, consulta Fase 2: Scelta delle origini dei dati e dei classificatori.

Proprietà della tabella del catalogo dati

Le proprietà della tabella, o parametri, come sono noti nella AWS CLI, sono stringhe di chiavi e valori non convalidate. È possibile impostare le proprie proprietà sulla tabella per supportare gli usi del catalogo dati all'esterno di AWS Glue. È possibile che lo facciano anche altri servizi che utilizzano il Data Catalog. AWS Glue imposta alcune proprietà della tabella durante l'esecuzione di job o crawler. Salvo diversa indicazione, queste proprietà sono per uso interno, non supportiamo il fatto che continuino a esistere nella loro forma attuale o che supportino il comportamento del prodotto se queste proprietà vengono modificate manualmente.

Per ulteriori informazioni sulle proprietà delle tabelle impostate dai AWS Glue crawler, consultate. the section called "Parametri impostati sulle tabelle del catalogo dati dal crawler"

# Integrazione con le tabelle Amazon S3

AWS Glue L'integrazione di Data Catalog con Amazon S3 Tables ti consente di scoprire, interrogare e unire S3 Tables ai dati nei data lake Amazon S3 utilizzando un unico catalogo. Quando integri S3 Tables con Data Catalog, il servizio crea una struttura di catalogo federata che mappa le risorse di S3 Tables agli oggetti del catalogo: AWS Glue

- Un bucket di tabelle S3 diventa un catalogo nel Data Catalog
- Un namespace S3 diventa un database AWS Glue
- Una tabella S3 diventa una tabella AWS Glue

## Controlli sugli accessi

Il Data Catalog supporta due modalità di controllo degli accessi per l'integrazione di S3 Tables:

- Controllo degli accessi IAM: utilizza le policy IAM per controllare l'accesso a S3 Tables e al Data Catalog. In questo approccio, sono necessarie le autorizzazioni IAM sia sulle risorse S3 Tables che sugli oggetti Data Catalog per accedere alle risorse.
- AWS Lake Formation controllo degli accessi: utilizza le AWS Lake Formation concessioni oltre alle autorizzazioni AWS Glue IAM per controllare l'accesso alle tabelle S3 tramite il Data Catalog. In questa modalità, i principali richiedono le autorizzazioni IAM per interagire con il Data Catalog e le AWS Lake Formation concessioni determinano a quali risorse del catalogo (database, tabelle, colonne, righe) il principale può accedere. Questa modalità supporta sia il controllo degli accessi a grana grossolana (concessioni a livello di database e tabella) che il controllo degli accessi granulare (sicurezza a livello di colonna e di riga). Quando viene configurato un ruolo registrato e la vendita di credenziali è abilitata, le autorizzazioni IAM di S3 Tables non sono richieste per il principale, in quanto vende le credenziali per conto del principale che utilizza il ruolo registrato. AWS Lake Formation AWS Lake Formation il controllo degli accessi supporta anche la vendita di credenziali per motori di analisi di terze parti. Per ulteriori informazioni, consulta Creazione di un catalogo di tabelle S3 nella Guida per gli AWS Lake Formation sviluppatori.

Puoi migrare tra le modalità di controllo degli accessi man mano che le tue esigenze evolvono.

## Gerarchia del catalogo per il montaggio automatico

Quando integri S3 Tables con il Data Catalog utilizzando la console di gestione Amazon S3, la console crea un catalogo federato `s3tablescatalog` chiamato Data Catalog nel tuo account in

quella regione. AWS Questo catalogo federato funge da catalogo principale per tutti i bucket di tabelle S3 esistenti e futuri in quell'account e nella regione. L'integrazione mappa le risorse del table bucket di Amazon S3 nella seguente gerarchia:

- Catalogo federato: `s3tablescatalog` (creato automaticamente)
- Cataloghi per bambini: ogni bucket da tavolo S3 diventa un catalogo per bambini sotto `s3tablescatalog`
- Database: ogni namespace S3 all'interno di un bucket di tabella diventa un database
- Tabelle: ogni tabella S3 all'interno di un namespace diventa una tabella

Ad esempio, se disponi di un bucket di tabella S3 denominato «analytics-bucket» con uno spazio dei nomi «sales» contenente una tabella «transazioni», il percorso completo nel Data Catalog sarebbe: `s3tablescatalog/analytics-bucket/sales/transactions`

Questa gerarchia in quattro parti si applica agli scenari con lo stesso account in cui S3 Tables e Data Catalog si trovano nello stesso account. AWS Per gli scenari con più account, monti manualmente singoli bucket di tabelle S3 nel Data Catalog, il che crea una gerarchia in tre parti.

## Regioni supportate

L'integrazione di S3 Tables con il Data Catalog è disponibile nelle seguenti regioni: AWS

| Codice regione | Nome della Regione |
|---|---|
| us-east-1 | Stati Uniti orientali (Virginia settentrionale) |
| us-east-2 | Stati Uniti orientali (Ohio) |
| us-west-1 | Stati Uniti occidentali (California settentrionale) |
| us-west-2 | Stati Uniti occidentali (Oregon) |
| af-south-1 | Africa (Città del Capo) |
| ap-east-1 | Asia Pacifico (Hong Kong) |
| ap-east-2 | Asia Pacifico (Taipei) |
| ap-northeast-1 | Asia Pacifico (Tokyo) |

| Codice regione | Nome della Regione |
|---|---|
| ap-northeast-2 | Asia Pacifico (Seul) |
| ap-northeast-3 | Asia Pacifico (Osaka) |
| ap-south-1 | Asia Pacifico (Mumbai) |
| ap-south-2 | Asia Pacifico (Hyderabad) |
| ap-southeast-1 | Asia Pacifico (Singapore) |
| ap-southeast-2 | Asia Pacifico (Sydney) |
| ap-southeast-3 | Asia Pacifico (Giacarta) |
| ap-southeast-4 | Asia Pacifico (Melbourne) |
| ap-southeast-5 | Asia Pacifico (Malesia) |
| ap-southeast-6 | Asia Pacifico (Nuova Zelanda) |
| ap-southeast-7 | Asia Pacifico (Thailandia) |
| ca-central-1 | Canada (Centrale) |
| ca-west-1 | Canada occidentale (Calgary) |
| eu-central-1 | Europa (Francoforte) |
| eu-central-2 | Europa (Zurigo) |
| eu-north-1 | Europa (Stoccolma) |
| eu-south-1 | Europa (Milano) |
| eu-south-2 | Europa (Spagna) |
| eu-west-1 | Europa (Irlanda) |
| eu-west-2 | Europa (Londra) |

| Codice regione | Nome della Regione |
|---|---|
| eu-west-3 | Europa (Parigi) |
| il-central-1 | Israele (Tel Aviv) |
| mx-central-1 | Messico (centrale) |
| sa-east-1 | Sud America (San Paolo) |

Argomenti


- Prerequisiti
- Abilitazione dell'integrazione di S3 Tables con il Data Catalog
- Aggiungere database e tabelle al catalogo S3 Tables
- Condivisione degli oggetti del catalogo S3 Tables
- Gestione dell'integrazione con S3 Tables


## Prerequisiti

Prima di creare un catalogo federato per le tabelle S3 nel AWS Glue Data Catalog, assicurati che il tuo principale IAM (utente o ruolo) disponga delle autorizzazioni richieste.

Autorizzazioni IAM richieste

Il tuo responsabile IAM necessita delle seguenti autorizzazioni per abilitare l'integrazione con S3 Tables:

AWS Glue autorizzazioni:

- `glue:CreateCatalog`— Necessario per creare il catalogo `s3tablescatalog` federato
- `glue:GetCatalog`— Necessario per visualizzare i dettagli del catalogo
- `glue:GetDatabase`— Necessario per visualizzare i namespace S3 come database
- `glue:GetTable`— Necessario per visualizzare le tabelle S3
- `glue:passConnection`— Concede al principale chiamante il diritto di delegare la `aws:s3tables` connessione al servizio AWS Glue

Autorizzazioni S3 Tables (per il controllo degli accessi IAM):

- `s3tables:CreateTableBucket`
- `s3tables:GetTableBucket`
- `s3tables:CreateNamespace`
- `s3tables:GetNamespace`
- `s3tables:ListNamespaces`
- `s3tables:CreateTable`
- `s3tables:GetTable`
- `s3tables:ListTables`
- `s3tables:UpdateTableMetadataLocation`
- `s3tables:GetTableMetadataLocation`
- `s3tables:GetTableData`
- `s3tables:PutTableData`

### Esempio di policy IAM

La seguente policy IAM fornisce le autorizzazioni minime richieste per abilitare l'integrazione di S3 Tables con il Data Catalog in modalità IAM:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "GlueDataCatalogPermissions",
      "Effect": "Allow",
      "Action": [
        "glue:CreateCatalog",
        "glue:GetCatalog",
        "glue:GetDatabase",
        "glue:GetTable"
      ],
      "Resource": [
        "arn:aws:glue:region:account-id:catalog/s3tablescatalog",
        "arn:aws:glue:region:account-id:database/s3tablescatalog/*/*",
        "arn:aws:glue:region:account-id:table/s3tablescatalog/*/*/*"
      ]
    },
```

```
    {
      "Sid": "S3TablesDataAccessPermissions",
      "Effect": "Allow",
      "Action": [
        "s3tables:GetTableBucket",
        "s3tables:GetNamespace",
        "s3tables:GetTable",
        "s3tables:GetTableMetadataLocation",
        "s3tables:GetTableData"
      ],
      "Resource": [
        "arn:aws:s3tables:region:account-id:bucket/*",
        "arn:aws:s3tables:region:account-id:bucket/*/table/*"
      ]
    }
  ]
}
```

## Abilitazione dell'integrazione di S3 Tables con il Data Catalog

Puoi abilitare l'integrazione di S3 Tables con il AWS Glue Data Catalog utilizzando la console di gestione Amazon S3 oppure. AWS CLI Quando abiliti l'integrazione tramite la console, AWS crea un catalogo federato denominato `s3tablescatalog` che rileva e monta automaticamente tutti i table bucket S3 nel tuo account e nella tua regione. AWS

Abilita l'integrazione di S3 Tables utilizzando la console di gestione Amazon S3

1. Apri la console Amazon S3 all'indirizzo. https://console.aws.amazon.com/s3/
2. Nel pannello di navigazione a sinistra, scegli Bucket di tabelle.
3. Seleziona Crea bucket di tabelle.
4. Inserisci un Nome del bucket di tabelle e assicurati che sia selezionata l'opzione Abilita integrazione.
5. Seleziona Crea bucket di tabelle.

Amazon S3 integra automaticamente i tuoi table bucket in quella regione. La prima volta che integri i table bucket in una regione, Amazon S3 `s3tablescatalog` crea il catalogo dati di quella regione.

Dopo la creazione del catalogo, tutti i table bucket S3 presenti nell'account e nella regione vengono montati automaticamente come cataloghi secondari. Puoi visualizzare i database (namespace) e le tabelle accedendo al catalogo nel Data Catalog.

Abilita l'integrazione di S3 Tables utilizzando AWS CLI

Usa il `glue create-catalog` comando per creare il `s3tablescatalog` catalogo.

```
aws glue create-catalog \
  --name "s3tablescatalog" \
  --catalog-input '{
    "Description": "Federated catalog for S3 Tables",
    "FederatedCatalog": {
      "Identifier": "arn:aws:s3tables:region:account-id:bucket/*",
      "ConnectionName": "aws:s3tables"
    },
    "CreateDatabaseDefaultPermissions": [{
      "Principal": {
        "DataLakePrincipalIdentifier": "IAM_ALLOWED_PRINCIPALS"
      },
      "Permissions": ["ALL"]
    }],
    "CreateTableDefaultPermissions": [{
      "Principal": {
        "DataLakePrincipalIdentifier": "IAM_ALLOWED_PRINCIPALS"
      },
      "Permissions": ["ALL"]
    }]
  }'
```

*`region`*Sostituiscilo con la tua AWS regione e *`account-id`* con l'ID AWS del tuo account.

Verifica dell'integrazione

Dopo aver creato il catalogo, puoi verificare che i bucket da tavolo S3 siano montati elencando i cataloghi secondari:

```
aws glue get-catalogs \
  --parent-catalog-id s3tablescatalog
```

## Aggiungere database e tabelle al catalogo S3 Tables

Assicurati di disporre delle autorizzazioni necessarie per elencare e creare cataloghi, database e tabelle nel Data Catalog della tua regione. Assicurati che l'integrazione con S3 Tables sia abilitata nel tuo AWS account e nella tua regione.

### Aggiungere un database al catalogo S3 Tables

#### Aggiungere un database (console)

1. Apri la AWS Glue console a https://console.aws.amazon.com/glue/casa.
2. Nel pannello di navigazione a sinistra, scegliere Database.
3. Scegli Aggiungi database.
4. Scegli Glue Database in S3 Tables Federated Catalog.
5. Inserisci un nome univoco per il database.
6. Seleziona il catalogo di destinazione mappato a un bucket di tabelle in S3 Tables.
7. Scegliere Create Database (Crea database).

#### Aggiungere un database ()AWS CLI

```
aws glue create-database \
  --region region \
  --catalog-id "account-id:s3tablescatalog/my-catalog" \
  --database-input '{"Name": "my-database"}'
```

### Aggiungere una tabella al catalogo S3 Tables

#### Aggiungere una tabella (console)

1. Apri la AWS Glue console a https://console.aws.amazon.com/glue/casa.
2. Nel riquadro di navigazione a sinistra, selezionare Tables (Tabelle).
3. Seleziona il catalogo S3 Tables appropriato nel menu a discesa del catalogo.
4. Scegli Aggiungi tabella.
5. Inserisci un nome univoco per la tua tabella.
6. Verifica che il catalogo S3 Tables corretto sia selezionato nel menu a discesa del catalogo.
7. Seleziona il database nel menu a discesa del database.
8. Inserisci lo schema della tabella inserendo un codice JSON o aggiungendo ogni colonna singolarmente.
9. Scegliere Create table (Crea tabella).

Aggiungere una tabella ()AWS CLI

```
aws glue create-table \
  --region region \
  --catalog-id "account-id:s3tablescatalog/my-catalog" \
  --database-name "my-database" \
  --table-input '{
    "Name": "my-table",
    "Parameters": {
      "classification": "",
      "format": "ICEBERG"
    },
    "StorageDescriptor": {
      "Columns": [
        {"Name": "id", "Type": "int", "Parameters": {}},
        {"Name": "val", "Type": "string", "Parameters": {}}
      ]
    }
  }'
```

## Condivisione degli oggetti del catalogo S3 Tables

Quando utilizzi il controllo degli accessi IAM, puoi condividere gli oggetti del catalogo S3 Tables con altri utenti utilizzando i link alle AWS Glue risorse per la condivisione dello stesso account. Per la condivisione tra account, puoi condividere i bucket di tabella S3 con un altro AWS account e il ruolo o l'utente IAM nell'account del destinatario può creare un oggetto di AWS Glue catalogo utilizzando il bucket di tabella condiviso.

Condivisione all'interno dello stesso account utilizzando i link alle risorse

I link alle risorse consentono di creare riferimenti a AWS Glue database e tabelle `s3tablescatalog` presenti nel catalogo AWS Glue predefinito. Ciò è utile per organizzare l'accesso ai dati o creare raggruppamenti logici di tabelle.

Crea un collegamento a una risorsa (console)

1. Apri la AWS Glue console all'indirizzo https://console.aws.amazon.com/glue/.
2. Nel riquadro di navigazione, scegli Cataloghi.
3. Nell'elenco dei cataloghi, seleziona s3tablescatalog.
4. Seleziona la tabella che desideri condividere da. `s3tablescatalog`
5. Scegli Azioni, quindi scegli Crea link alla risorsa.

6. Per Nome del link alla risorsa, inserisci un nome per il link alla risorsa.
7. Per Database Target, seleziona il database in cui desideri creare il link alle risorse.
8. (Facoltativo) In Descrizione, immetti una descrizione.
9. Scegli Create (Crea).

Il link alla risorsa viene visualizzato nel database di destinazione e punta alla tabella originale in`s3tablescatalog`.

Crea collegamenti alle risorse (AWS CLI)

Crea un collegamento a una risorsa del database:

```
aws glue create-database \
  --database-name "my-database-resource-link" \
  --database-input '{
    "Name": "sales_data_link",
    "TargetDatabase": {
      "CatalogId": "account-id:s3tablescatalog/analytics-bucket",
      "DatabaseName": "sales"
    }
  }'
```

Crea un collegamento alle risorse della tabella:

```
aws glue create-table \
  --table-name "my-table-resource-link" \
  --table-input '{
    "Name": "sales_data_link",
    "TargetTable": {
      "CatalogId": "account-id:s3tablescatalog/analytics-bucket",
      "DatabaseName": "sales",
      "Name": "transactions"
    }
  }'
```

## Gestione dell'integrazione con S3 Tables

### Abilita AWS Lake Formation

Puoi abilitarlo AWS Lake Formation per il tuo catalogo S3 Tables quando desideri scalare i requisiti di governance dei dati. AWS Lake Formation fornisce concessioni in stile database per gestire accessi

dettagliati, scalare le autorizzazioni utilizzando l'accesso basato su tag e concedere autorizzazioni basate su attributi utente come le associazioni di gruppo alle tabelle in S3 Tables.

Vai alla console di gestione per abilitare il tuo catalogo S3 Tables AWS Lake Formation . AWS Lake Formation AWS Glue Per ulteriori informazioni, consulta Creazione di un catalogo S3 Tables nella Guida per gli AWS Lake Formation sviluppatori.

Elimina l'integrazione con S3 Tables

Puoi eliminare l'integrazione di S3 Tables eliminando l'integrazione del catalogo nel Data Catalog. Questa operazione elimina solo i metadati nel Data Catalog e non le risorse in S3 Tables.

Assicurati di disporre delle autorizzazioni necessarie per elencare, modificare ed eliminare gli oggetti del catalogo. AWS Glue

Eliminare l'integrazione (console)

1. Apri la AWS Glue console a https://console.aws.amazon.com/glue/casa.
2. Nel riquadro di navigazione, scegli Cataloghi.
3. Nell'elenco dei cataloghi, seleziona s3tablescatalog.
4. Scegli Elimina.
5. Conferma che l'eliminazione del catalogo elimina anche tutti gli oggetti del catalogo associati nel Data Catalog.
6. Scegli Elimina.

Elimina integrazione ()AWS CLI

```
aws glue delete-catalog \
  --region region \
  --catalog-id "s3tablescatalog"
```

## Integrazione con altri servizi AWS

Sebbene sia possibile utilizzare Crawler di AWS Glue s per compilarli AWS Glue Data Catalog, esistono diversi AWS servizi che possono integrarsi e popolare automaticamente nel catalogo. Le sezioni seguenti forniscono ulteriori informazioni sui casi d'uso specifici supportati dai AWS servizi che possono popolare il Data Catalog.

Argomenti

- AWS Lake Formation
- Amazon Athena

## AWS Lake Formation

AWS Lake Formation è un servizio che semplifica la configurazione di un data lake sicuro. AWS Lake Formation è costruita su AWS Glue, e Lake Formation AWS Glue condividiamo la stessa cosa AWS Glue Data Catalog. Puoi registrare la tua posizione dati Amazon S3 con Lake Formation e utilizzare la console Lake Formation per creare database e tabelle nel AWS Glue Data Catalog, definire politiche di accesso ai dati e controllare l'accesso ai dati attraverso il tuo data lake da una posizione centrale. È possibile utilizzare il controllo granulare degli accessi di Lake Formation per gestire le risorse esistenti nel catalogo dati e le posizioni di dati di Amazon S3.

Con i dati registrati con Lake Formation, puoi condividere in sicurezza le risorse del Data Catalog tra responsabili, AWS account, AWS organizzazioni e unità organizzative IAM.

Per ulteriori informazioni sulla creazione di risorse Data Catalog utilizzando Lake Formation, consulta Creating Data Catalog tables and database nella AWS Lake Formation Developer Guide.

## Amazon Athena

Amazon Athena utilizza il Data Catalog per archiviare e recuperare i metadati delle tabelle per i dati Amazon S3 nel tuo account. AWS I metadati della tabella consentono al motore di query Athena di sapere come trovare, leggere ed elaborare i dati che si desidera interrogare.

È possibile compilare il file AWS Glue Data Catalog utilizzando direttamente le istruzioni `CREATE TABLE` Athena. È possibile definire e popolare manualmente lo schema e i metadati delle partizioni nel catalogo dati senza dover eseguire un crawler.

1. Nella console Athena, creare un database che memorizzerà i metadati della tabella nel catalogo dati.
2. Usare l'istruzione `CREATE EXTERNAL TABLE` per definire lo schema dell'origine dati.
3. Usare la clausola `PARTITIONED BY` per definire eventuali chiavi di partizione se i dati sono partizionati.
4. Usare la clausola `LOCATION` per specificare il percorso di Amazon S3 in cui vengono archiviati i file di dati effettivi.
5. Eseguire l'istruzione `CREATE TABLE`.

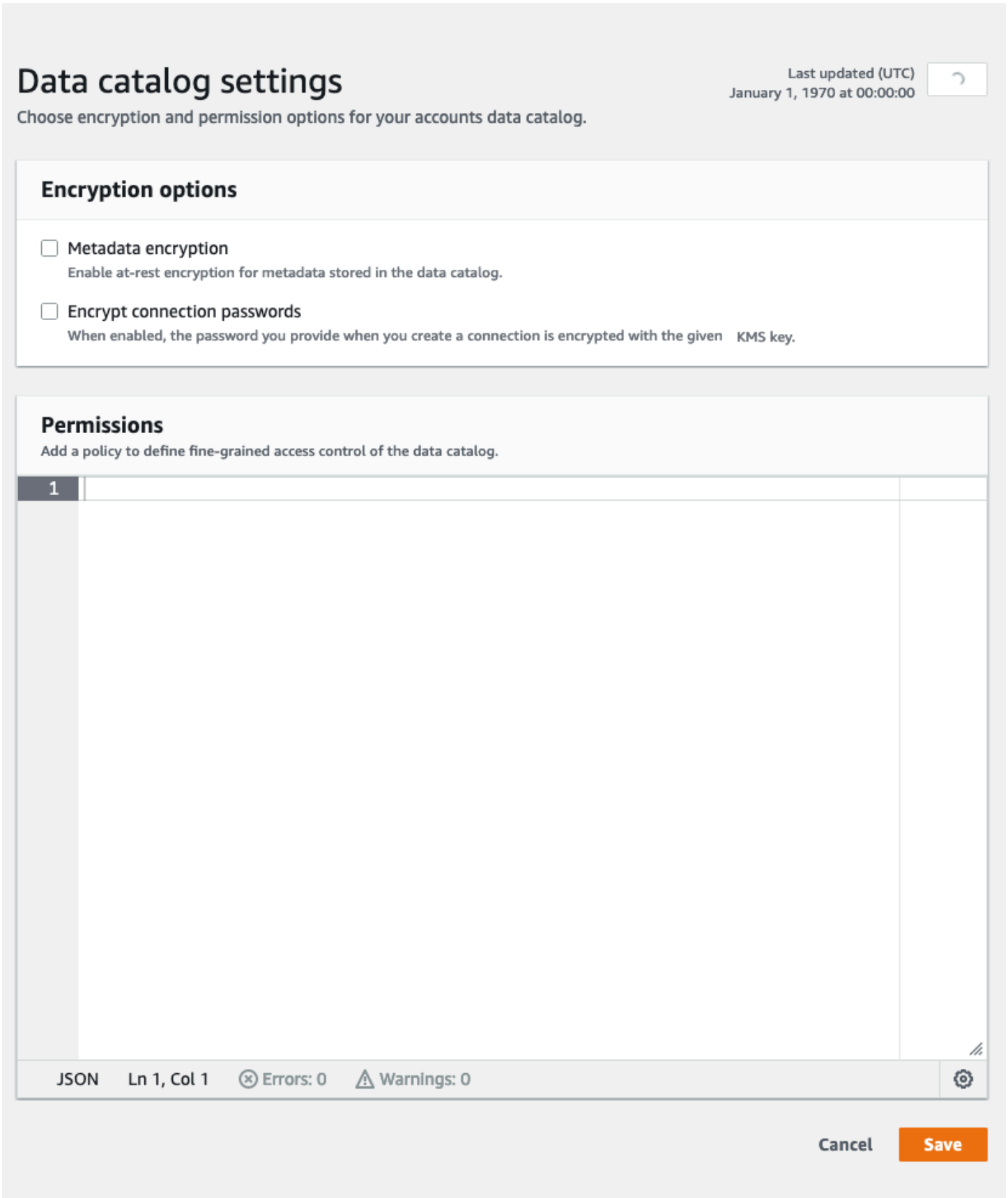
Questa query crea i metadati della tabella nel catalogo dati in base allo schema e alle partizioni definiti, senza svolgere effettivamente il crawling dei dati.

È possibile interrogare la tabella in Athena, che utilizzerà i metadati dal catalogo dati per accedere e interrogare i file di dati in Amazon S3.

Per ulteriori informazioni, consultare Creare database e tabelle nella guida per l'utente di Amazon Athena.

## Impostazioni del Catalogo dati

Le impostazioni del Catalogo dati contengono opzioni per impostare le opzioni relative alla crittografia e alle autorizzazioni per il Catalogo dati nell'account.


Data catalog settings
Choose encryption and permission options for your accounts data catalog.
Last updated (UTC)
January 1, 1970 at 00:00:00
Encryption options
Metadata encryption
Enable at-rest encryption for metadata stored in the data catalog.
Encrypt connection passwords
When enabled, the password you provide when you create a connection is encrypted with the given KMS key.
Permissions
Add a policy to define fine-grained access control of the data catalog.
1
JSON
Ln 1, Col 1
Errors: 0
Warnings: 0
Cancel
Save

Per modificare il controllo granulare degli accessi del catalogo dati

1. Accedi a Console di gestione AWS e apri la AWS Glue console all'indirizzo https://console.aws.amazon.com/glue/.
2. Scegli un'opzione di crittografia.

   - Crittografia dei metadati – Seleziona questa casella di controllo per crittografare i metadati nel catalogo dati. I metadati vengono crittografati quando sono inattivi utilizzando la chiave AWS Key Management Service (AWS KMS) specificata. Per ulteriori informazioni, consulta Crittografia del catalogo dati.
   - Crittografa le password di connessione – Seleziona questa casella di controllo per crittografare le password nell'oggetto di connessione AWS Glue quando la connessione viene creata o aggiornata. Le password vengono crittografate utilizzando la AWS KMS chiave specificata. Quando vengono restituite, le password vengono crittografate. Questa opzione è un'impostazione globale per tutte le connessioni AWS Glue nel catalogo dati. Se si deseleziona questa casella di controllo, le password precedentemente crittografate rimangono crittografate utilizzando la chiave utilizzata al momento della creazione o dell'aggiornamento. Per ulteriori informazioni sulle connessioni AWS Glue, consulta Connessione ai dati.

     Quando abiliti questa opzione, scegli una AWS KMS chiave o scegli Inserisci un ARN per la chiave e fornisci l'Amazon Resource Name (ARN) per la chiave. Immetti l'ARN usando questo formato: `arn:aws:kms:region:account-id:key/key-id` . Puoi specificare l'ARN anche sotto forma di alias di chiavi, ad esempio `arn:aws:kms:region:account-id:alias/alias-name` .

     > **Important**
     >
     > Se si seleziona questa opzione, qualsiasi utente o ruolo che crea o aggiorna una connessione deve avere l'autorizzazione `kms:Encrypt` sulla chiave KMS specificata.

     Per ulteriori informazioni, consulta Crittografia delle password di connessione.
3. Scegli Settings (Impostazioni), quindi nell'editor Permissions (Autorizzazioni) aggiungi l'istruzione di policy per modificare il controllo granulare degli accessi del catalogo dati per il tuo account. Al catalogo dati è possibile collegare una sola policy per volta. Puoi incollare una policy sulle risorse JSON in questo controllo. Per ulteriori informazioni, consulta Politiche basate sulle risorse all'interno di Glue AWS.

4. Scegliere Save (Salva) per aggiornare il catalogo dati con le modifiche apportate.

È inoltre possibile utilizzare le operazioni API AWS Glue per archiviare, recuperare ed eliminare le policy delle risorse. Per ulteriori informazioni, consulta Sicurezza APIs in AWS Glue.

# Popolamento e gestione delle tabelle transazionali

Apache Iceberg, Apache Hudi e Linux Foundation Delta Lake sono formati di tabelle open source progettati per gestire analisi dei dati su larga scala e carichi di lavoro di data lake in Apache Spark.

È possibile popolare le tabelle Iceberg, Hudi e Delta Lake AWS Glue Data Catalog utilizzando i seguenti metodi:

- Crawler di AWS Glue; — Crawler di AWS Glue s può scoprire e popolare automaticamente i metadati delle tabelle Iceberg, Hudi e Delta Lake nel Data Catalog. Per ulteriori informazioni, consulta Utilizzo dei crawler per compilare il Catalogo dati.
- AWS Glue Processi ETL: puoi creare lavori ETL per scrivere dati nelle tabelle Iceberg, Hudi e Delta Lake e popolare i relativi metadati nel Data Catalog. Per ulteriori informazioni, consulta Using Data Lake Frameworks with ETL jobs. AWS Glue
- AWS Glue console, AWS Lake Formation console AWS CLI o API: puoi utilizzare la AWS Glue console, la console Lake Formation o l'API per creare e gestire le definizioni delle tabelle Iceberg nel Data Catalog.

Argomenti


- Creazione di tabelle Apache Iceberg
- Ottimizzazione delle tabelle Iceberg
- Ottimizzazione delle prestazioni delle query per le tabelle Iceberg


## Creazione di tabelle Apache Iceberg

È possibile creare tabelle Apache Iceberg che utilizzano il formato di dati Apache Parquet in AWS Glue Data Catalog con dati che risiedono in Amazon S3. Una tabella nel catalogo dati è la definizione dei metadati che rappresenta i dati in un datastore. Per impostazione predefinita, AWS Glue crea tabelle Iceberg v2. Per la differenza tra le tabelle v1 e v2, consulta Modifiche al tipo di formato nella documentazione di Apache Iceberg.

Apache Iceberg è un formato a tabella aperta per set di dati analitici di dimensioni molto grandi. Iceberg consente di modificare facilmente lo schema, operazione nota anche come evoluzione dello schema, il che significa che gli utenti possono aggiungere, rinominare o rimuovere colonne da una tabella di dati senza pregiudicare i dati sottostanti. Iceberg fornisce anche supporto per il controllo delle versioni dei dati, che consente agli utenti di tenere traccia delle modifiche apportate ai dati nel tempo. Ciò abilita la funzionalità di query temporale, che consente agli utenti di accedere e interrogare le versioni storiche dei dati e analizzare le modifiche ai dati tra aggiornamenti ed eliminazioni.

Puoi utilizzare AWS Glue la console Lake Formation o l'`CreateTable`operazione nell' AWS Glue API per creare una tabella Iceberg nel Data Catalog. Per ulteriori informazioni, vedere CreateTable action (Python: create_table).

Quando si crea una tabella Iceberg nel catalogo dati, occorre specificare il formato della tabella e il percorso del file dei metadati in Amazon S3 per poter eseguire letture e scritture.

Puoi usare Lake Formation per proteggere la tua tabella Iceberg utilizzando autorizzazioni di controllo degli accessi granulari quando registri la posizione dati di Amazon S3 con. AWS Lake Formation Per i dati di origine in Amazon S3 e i metadati non registrati con Lake Formation, l'accesso è determinato dalle politiche di autorizzazione IAM per Amazon S3 e dalle azioni. AWS Glue Per ulteriori informazioni, consultare Gestione delle autorizzazioni.

> **Note**
>
> Il catalogo dati non supporta la creazione di partizioni e l'aggiunta di proprietà delle tabelle Iceberg.

## Prerequisiti

Per creare tabelle Iceberg nel catalogo dati e configurare le autorizzazioni di accesso ai dati di Lake Formation, occorre soddisfare i seguenti requisiti:

1. Autorizzazioni richieste per creare tabelle Iceberg senza i dati registrati con Lake Formation.

   Oltre alle autorizzazioni richieste per creare una tabella nel catalogo dati, il creatore della tabella richiede le seguenti autorizzazioni:

   - `s3:PutObject` sulla risorsa arn:aws:s3:::{bucketName}
   - `s3:GetObject` sulla risorsa arn:aws:s3:::{bucketName}

- `s3:DeleteObject` sulla risorsa arn:aws:s3:::{bucketName}

2. Autorizzazioni richieste per creare tabelle Iceberg con dati registrati con Lake Formation:

   Per utilizzare Lake Formation per gestire e proteggere i dati nel data lake, registrare la posizione Amazon S3 che contiene i dati per le tabelle con Lake Formation. In questo modo Lake Formation può fornire credenziali a servizi di AWS analisi come Athena, Redshift Spectrum e Amazon EMR per accedere ai dati. Per ulteriori informazioni su come registrare una posizione Amazon S3, consultare Aggiungere una posizione Amazon S3 al data lake.

   Un principale che legge e scrive i dati sottostanti registrati con Lake Formation richiede le seguenti autorizzazioni:

   - `lakeformation:GetDataAccess`
   - `DATA_LOCATION_ACCESS`

     Un principale che dispone di autorizzazioni di localizzazione dei dati in una sede dispone anche delle autorizzazioni di localizzazione su tutte le sedi secondarie.

     Per ulteriori informazioni sulle autorizzazioni di localizzazione dei dati, consultare il link Controlli degli accessi ai dati sottostanti.

Per abilitare la compattazione, il servizio deve assumere un ruolo IAM con le autorizzazioni per aggiornare le tabelle nel catalogo dati. Per maggiori dettagli, consultare Prerequisiti per l'ottimizzazione delle tabelle.

## Creazione di una tabella Iceberg

Puoi creare tabelle Iceberg v1 e v2 utilizzando AWS Glue la console Lake Formation o AWS Command Line Interface come documentato in questa pagina. È anche possibile creare tabelle Iceberg usando Crawler di AWS Glue. Per ulteriori informazioni, consultare Catalogo dati e crawler nella Guida per gli sviluppatori di AWS Glue .

Per creare una tabella Iceberg

Console

1. Accedi Console di gestione AWS e apri la console all'indirizzo. AWS Glue https://console.aws.amazon.com/glue/

2. In catalogo dati, scegliere Tabelle e utilizzare il pulsante Crea tabella per specificare i seguenti attributi:
   - Nome tabella: inserire un nome per la tabella. Se si utilizza Athena per accedere alle tabelle, utilizzare questi suggerimenti per la denominazione della Guida per l'utente di Amazon Athena.
   - Database: scegliere un database esistente o crearne uno nuovo.
   - Descrizione: la descrizione della tabella. È possibile scrivere una descrizione per aiutare a comprendere i contenuti della tabella.
   - Formato tabella: come Formato tabella, scegliere Apache Iceberg.
   - Abilita compattazione: scegliere Abilita compattazione per compattare piccoli oggetti Amazon S3 nella tabella in oggetti più grandi.
   - Ruolo IAM: per eseguire la compattazione, il servizio assume un ruolo IAM per tuo conto. Puoi scegliere un ruolo IAM utilizzando il menu a discesa. Assicurati che il ruolo disponga delle autorizzazioni necessarie per abilitare la compattazione.

     Per ulteriori informazioni sulle autorizzazioni necessarie per questo ruolo, consultare Prerequisiti per l'ottimizzazione delle tabelle.
   - Posizione: specificare il percorso della cartella in Amazon S3 che memorizza la tabella dei metadati. Iceberg ha bisogno di un file di metadati e di una posizione nel catalogo dati per poter eseguire letture e scritture.
   - Schema: scegliere Aggiungi colonne per aggiungere colonne e tipi di dati delle colonne. Si dispone della possibilità di creare una tabella vuota e di aggiornare lo schema in un secondo momento. Il catalogo dati supporta i tipi di dati Hive. Per ulteriori informazioni, consultare Tipi di dati Hive.

     Iceberg consente di evolvere lo schema e la partizione dopo aver creato la tabella. È possibile usare le query di Athena per aggiornare lo schema della tabella e le query Spark per aggiornare le partizioni.

AWS CLI

```
aws glue create-table \
    --database-name iceberg-db \
    --region us-west-2 \
    --open-table-format-input '{
      "IcebergInput": {
```

```
          "MetadataOperation": "CREATE",
          "Version": "2"
       }
    }' \
--table-input '{"Name":"test-iceberg-input-demo",
          "TableType": "EXTERNAL_TABLE",
          "StorageDescriptor":{
             "Columns":[
                 {"Name":"col1", "Type":"int"},
                 {"Name":"col2", "Type":"int"},
                 {"Name":"col3", "Type":"string"}
              ],
             "Location":"s3://DOC_EXAMPLE_BUCKET_ICEBERG/"
          }
       }'
```

## Ottimizzazione delle tabelle Iceberg

AWS Glue supporta diverse opzioni di ottimizzazione delle tabelle per migliorare la gestione e le prestazioni delle tabelle Apache Iceberg utilizzate dai motori AWS analitici e dai job ETL. Questi ottimizzatori offrono un utilizzo efficiente dello storage, prestazioni di query migliorate e una gestione efficace dei dati. Sono disponibili tre tipi di ottimizzatori di tabelle in: AWS Glue

- Compattazione: la compattazione dei dati compatta file di dati di piccole dimensioni per ridurre l'utilizzo dell'archiviazione e migliorare le prestazioni di lettura. I file di dati vengono uniti e riscritti per rimuovere i dati obsoleti e consolidare i dati frammentati in file più grandi ed efficienti. È possibile configurare la compattazione in modo che venga eseguita automaticamente.

  Binpack è la strategia di compattazione predefinita in Apache Iceberg. Combina file di dati più piccoli in file più grandi per prestazioni ottimali. La compattazione supporta anche strategie di Sort e Z-order che raggruppano dati simili. Sort organizza i dati in base a colonne specifiche, migliorando le prestazioni delle query per le operazioni filtrate. Z-order crea set di dati ordinati che migliorano le prestazioni delle query quando vengono eseguite query su più colonne contemporaneamente. Tutte e tre le strategie di compattazione (Binpack, Sort e Z-order) riducono la quantità di dati scansionati dai motori di query, riducendo così i costi di elaborazione delle query.
- Conservazione degli snapshot: gli snapshot sono versioni con data e ora di una tabella Iceberg. Le configurazioni di conservazione degli snapshot consentono ai clienti di stabilire per quanto tempo conservare gli snapshot e quanti snapshot conservare. La configurazione di un ottimizzatore di

conservazione degli snapshot può aiutare a gestire il sovraccarico di archiviazione rimuovendo gli snapshot più vecchi e non necessari e i relativi file sottostanti.

- Eliminazione di file orfani: i file orfani sono file a cui non fanno più riferimento i metadati della tabella Iceberg. Questi file possono accumularsi nel tempo, soprattutto dopo operazioni come l'eliminazione di tabelle o i processi ETL non riusciti. L'abilitazione dell'eliminazione dei file orfani consente di AWS Glue identificare e rimuovere periodicamente questi file non necessari, liberando spazio di archiviazione.

La configurazione di ottimizzazione a livello di catalogo è disponibile tramite la console Lake Formation e utilizzando l'operazione AWS Glue `UpdateCatalog` API. Puoi abilitare o disabilitare gli ottimizzatori di compattazione, conservazione delle istantanee e cancellazione di file orfani per le singole tabelle Iceberg nel Data Catalog utilizzando la console o le AWS Glue operazioni API. AWS CLI AWS Glue

Argomenti


- Prerequisiti per l'ottimizzazione delle tabelle
- Ottimizzatori di tabelle a livello di catalogo
- Ottimazione della compattazione
- Ottimizzazione di conservazione degli snapshot
- Eliminazione di file orfani
- Visualizzazione dei dettagli dell'ottimizzazione
- Visualizzazione delle Amazon CloudWatch metriche
- Eliminazione di un ottimizzatore
- Considerazioni e limitazioni
- Regioni supportate per gli ottimizzatori di tabelle


## Prerequisiti per l'ottimizzazione delle tabelle

L'ottimizzatore di tabelle presuppone le autorizzazioni del ruolo AWS Identity and Access Management (IAM) specificato quando abiliti le opzioni di ottimizzazione (compattazione, conservazione delle istantanee ed eliminazione di file orfani) per una tabella. È possibile creare un singolo ruolo per tutti gli ottimizzatori o creare ruoli separati per ogni ottimizzatore.

> **Note**
>
> L'ottimizzatore per l'eliminazione dei file orfani non richiede le autorizzazioni `glue:updateTable` o `s3:putObject`. Gli ottimizzatori di scadenza e compattazione degli snapshot richiedono lo stesso set di autorizzazioni.

Il ruolo IAM deve disporre delle autorizzazioni per leggere i dati e aggiornare i metadati nel Catalogo dati. Puoi creare un ruolo IAM e collegare le seguenti policy in linea:

- Aggiungi la seguente politica in linea che concede le autorizzazioni di Amazon read/write S3 sulla posizione per i dati che non sono registrati con. AWS Lake Formation Questa politica include anche le autorizzazioni per aggiornare la tabella nel Data Catalog e per consentire l'aggiunta di log nei log e AWS Glue la pubblicazione di metriche. Amazon CloudWatch Per i dati di origine in Amazon S3 che non sono registrati con Lake Formation, l'accesso è determinato dalle policy di autorizzazione IAM per Amazon S3 e dalle operazioni AWS Glue .

  Nelle seguenti policy in linea, sostituisci `bucket-name` con il nome del bucket Amazon S3, `aws-account-id` e `region` con un numero di account AWS valido e una regione del Catalogo dati, `database_name` con il nome del database e `table_name` con il nome della tabella.

  JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObject",
                "s3:DeleteObject"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket"
```

```
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "glue:UpdateTable",
                "glue:GetTable"
            ],
            "Resource": [
                "arn:aws:glue:us-east-1:111122223333:table/<database-
name>/<table-name>",
                "arn:aws:glue:us-east-1:111122223333:database/<database-name>",
                "arn:aws:glue:us-east-1:111122223333:catalog"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": [
                "arn:aws:logs:us-east-1:111122223333:log-group:/aws-glue/
iceberg-compaction/logs:*",
                "arn:aws:logs:us-east-1:111122223333:log-group:/aws-glue/
iceberg-retention/logs:*",
                "arn:aws:logs:us-east-1:111122223333:log-group:/aws-glue/
iceberg-orphan-file-deletion/logs:*"
            ]
        }
    ]
}
```

- Utilizza la seguente policy per abilitare la compattazione dei dati registrati con Lake Formation.

  Se al ruolo di ottimizzazione non sono concesse le autorizzazioni di gruppo `IAM_ALLOWED_PRINCIPALS` sulla tabella, il ruolo richiede autorizzazioni `ALTER`, `DESCRIBE`, `INSERT` e `DELETE` di Lake Formation sulla tabella.

Per ulteriori informazioni su come registrare un bucket Amazon S3 con Lake Formation, consultare Aggiungere una posizione Amazon S3 al data lake.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "lakeformation:GetDataAccess"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "glue:UpdateTable",
        "glue:GetTable"
      ],
      "Resource": [
        "arn:aws:glue:us-east-1:111122223333:table/databaseName/tableName",
        "arn:aws:glue:us-east-1:111122223333:database/databaseName",
        "arn:aws:glue:us-east-1:111122223333:catalog"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": [
        "arn:aws:logs:us-east-1:111122223333:log-group:/aws-glue/iceberg-
compaction/logs:*",
        "arn:aws:logs:us-east-1:111122223333:log-group:/aws-glue/iceberg-
retention/logs:*",
        "arn:aws:logs:us-east-1:111122223333:log-group:/aws-glue/iceberg-
orphan-file-deletion/logs:*"
      ]
    }
```

```
  ]
}
```

- (Facoltativo) Per ottimizzare le tabelle Iceberg con i dati nei bucket Amazon S3 crittografati utilizzando la Crittografia lato server, il ruolo di compattazione richiede le autorizzazioni per decrittografare gli oggetti Amazon S3 e per generare una nuova chiave dati per scrivere oggetti nei bucket crittografati. Aggiungi la seguente politica alla chiave desiderata. AWS KMS Supportiamo solo la crittografia a livello di bucket.

```
{
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::<aws-account-id>:role/<optimizer-role-name>"
    },
    "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey"
    ],
    "Resource": "*"
}
```

- (Facoltativo) Per la posizione dei dati registrati con Lake Formation, il ruolo utilizzato per registrare la posizione richiede le autorizzazioni per decrittografare gli oggetti Amazon S3 e generare una nuova chiave dati per scrivere oggetti nei bucket crittografati. Per ulteriori informazioni, consulta la pagina Registrazione di una posizione crittografata Amazon S3.
- (Facoltativo) Se la AWS KMS chiave è archiviata in un altro AWS account, è necessario includere le seguenti autorizzazioni per il ruolo di compattazione.

  JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt",
        "kms:GenerateDataKey"
      ],
      "Resource": [
        "arn:aws:kms:us-east-1:111122223333:key/key-id"
      ]
```

```
    }
  ]
}
```

- Il ruolo utilizzato per eseguire la compattazione deve disporre dell'autorizzazione `iam:PassRole` relativa al ruolo.

  JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": [
        "arn:aws:iam::111122223333:role/<optimizer-role-name>"
      ]
    }
  ]
}
```

- Aggiungi la seguente policy di fiducia al ruolo affinché il AWS Glue servizio assuma il ruolo IAM per eseguire il processo di compattazione.

  JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "glue.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

-

(Facoltativo) Per aggiornare le impostazioni del Data Catalog e abilitare le ottimizzazioni delle tabelle a livello di catalogo, il ruolo IAM utilizzato deve disporre dell'`glue:UpdateCatalog`autorizzazione o dell' AWS Lake Formation `ALTER CATALOG`autorizzazione sul catalogo principale. È possibile utilizzare l'API `GetCatalog` per verificare le proprietà del catalogo.

## Ottimizzatori di tabelle a livello di catalogo

Con una configurazione unica del catalogo, puoi impostare ottimizzatori automatici come la compattazione, la conservazione degli snapshot e l'eliminazione dei file orfani per tutte le tabelle Apache Iceberg nuove e aggiornate di AWS Glue Data Catalog. Le configurazioni degli ottimizzatori a livello di catalogo consentono di applicare impostazioni di ottimizzazione coerenti su tutte le tabelle all'interno di un catalogo, eliminando la necessità di configurare gli ottimizzatori singolarmente per ogni tabella.

Gli amministratori di Data Lake possono configurare gli ottimizzatori di tabelle selezionando il catalogo predefinito nella console Lake Formation e abilitando gli ottimizzatori utilizzando l'opzione `Table optimization`. Quando si creano nuove tabelle o si aggiornano le tabelle esistenti nel Catalogo dati, questo esegue automaticamente le ottimizzazioni delle tabelle per ridurre il carico operativo.

Se hai configurato l'ottimizzazione a livello di tabella o se hai precedentemente eliminato le impostazioni di ottimizzazione per una tabella, tali impostazioni specifiche della tabella hanno la precedenza sulle impostazioni predefinite del catalogo. Se un parametro di configurazione non è definito né a livello di tabella né di catalogo, verrà applicato il valore della proprietà della tabella Iceberg. Questa impostazione è applicabile all'ottimizzatore per la conservazione degli snapshot e l'eliminazione dei file orfani.

Quando si abilitano gli ottimizzatori a livello di catalogo, considerare quanto segue:

- Quando si configurano le impostazioni di ottimizzazione al momento della creazione del catalogo e successivamente si disabilitano le ottimizzazioni tramite una richiesta di aggiornamento del catalogo, l'operazione viene eseguita a cascata su tutte le tabelle all'interno del catalogo.
- Se sono già stati configurati degli ottimizzatori per una determinata tabella, l'operazione di disabilitazione a livello di catalogo non avrà alcun impatto su questa tabella.
- Quando si disabilitano gli ottimizzatori a livello di catalogo, le tabelle con configurazioni di ottimizzazione esistenti manterranno le relative impostazioni specifiche e non verranno alterate

dalla modifica a livello di catalogo. Tuttavia, le tabelle senza le proprie configurazioni di ottimizzazione erediteranno lo stato di disabilitazione dal livello di catalogo.

- Poiché gli ottimizzatori per la conservazione degli snapshot e l'eliminazione dei file orfani possono essere basate sulla pianificazione, gli aggiornamenti introdurranno un ritardo casuale all'inizio della pianificazione. In questo modo, ogni ottimizzatore si avvia in un momento leggermente diverso dagli altri, così da suddividere il carico e ridurre la probabilità di superare i limiti di servizio.
- Le impostazioni dell'ottimizzatore a livello di catalogo non vengono ereditate automaticamente dalle tabelle quando AWS Glue Data Catalog la crittografia è abilitata. Se nel catalogo è abilitata la crittografia dei metadati, è necessario configurare gli ottimizzatori di tabella singolarmente per ogni tabella. Per utilizzare l'ereditarietà degli ottimizzatori a livello di catalogo, la crittografia dei metadati deve essere disabilitata nel catalogo.

Argomenti


- Abilitazione dell'ottimizzazione automatica delle tabelle a livello di catalogo
- Visualizzazione delle ottimizzazioni a livello di catalogo
- Disabilitazione dell'ottimizzazione delle tabelle a livello di catalogo


## Abilitazione dell'ottimizzazione automatica delle tabelle a livello di catalogo

Puoi abilitare l'ottimizzazione automatica delle tabelle per tutte le nuove tabelle Apache Iceberg nel Catalogo dati. Dopo aver creato la tabella, puoi anche aggiornare in modo esplicito le impostazioni dell'ottimizzazione della tabella manualmente.

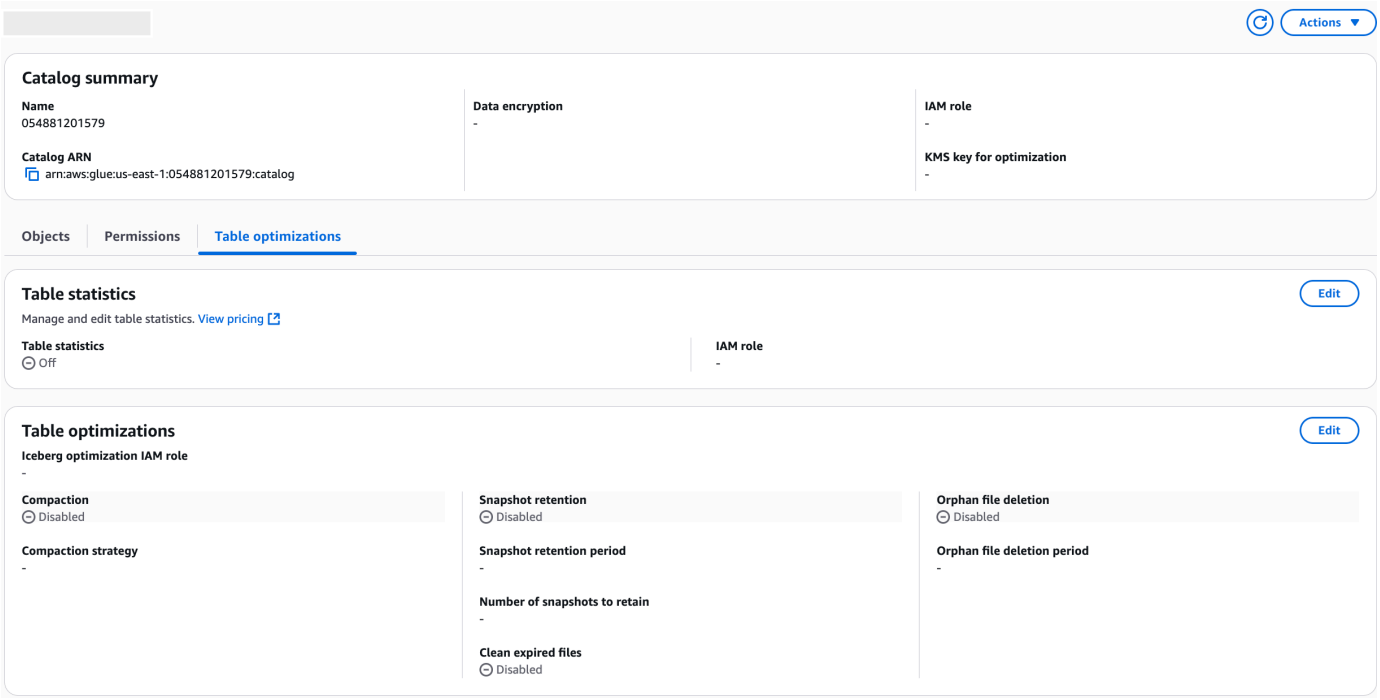
Per aggiornare le impostazioni dei cataloghi dati per abilitare l'ottimizzazione a livello di catalogo, il ruolo IAM utilizzato deve avere `glue:UpdateCatalog` l'autorizzazione sul catalogo principale. È possibile utilizzare `GetCatalog` l'API per verificare le proprietà del catalogo.

Per le tabelle gestite da Lake Formation, il ruolo IAM selezionato durante la configurazione di ottimizzazione del catalogo richiede le autorizzazioni Lake Formation `ALTER`, `DESCRIBE`, `INSERT` e `DELETE` per eventuali nuove tabelle o tabelle aggiornate.

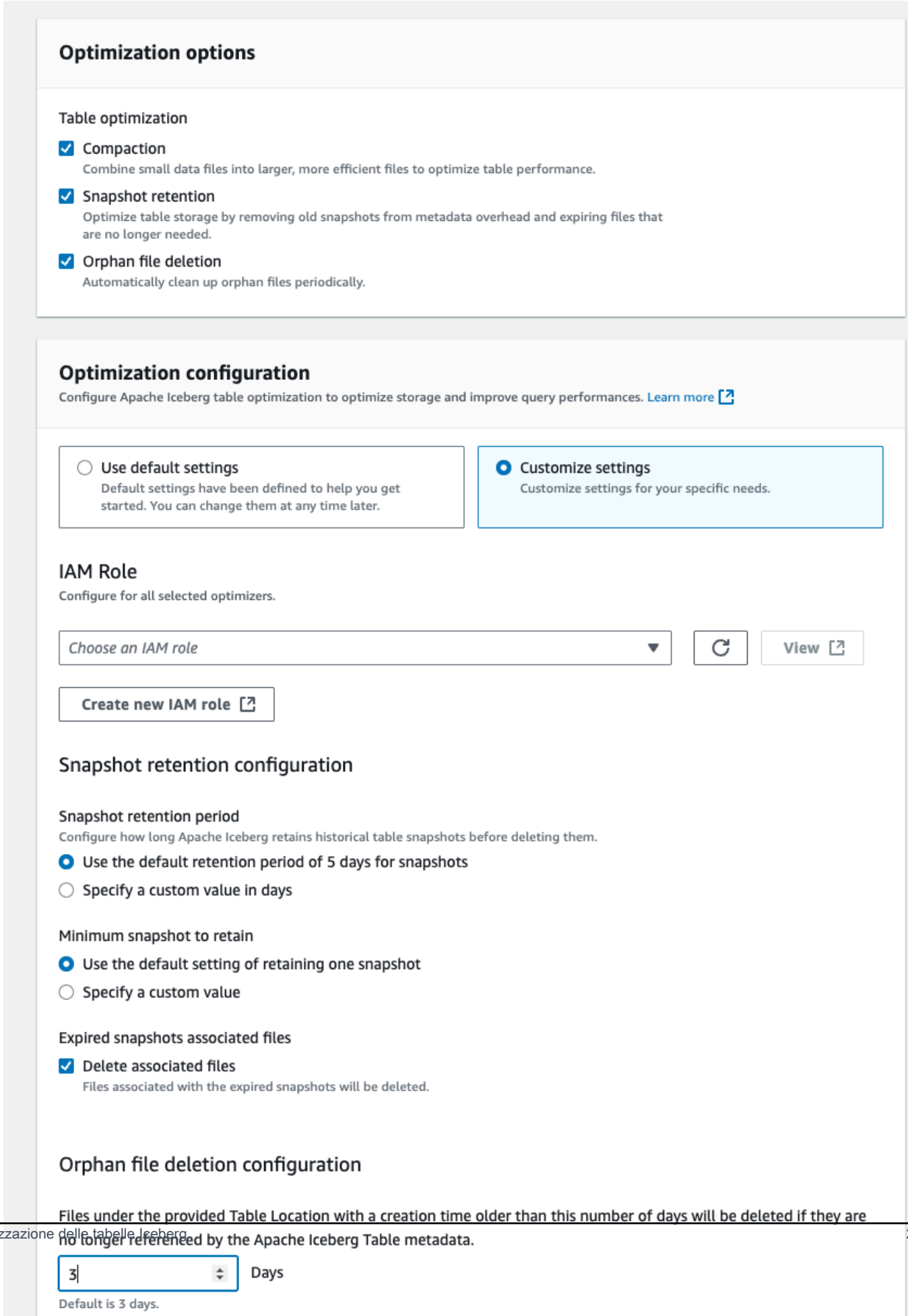
### Abilitazione degli ottimizzatori a livello di catalogo (console)

1. Apri la console Lake Formation all'indirizzo https://console.aws.amazon.com/lakeformation/.
2. Nel pannello di navigazione, seleziona Catalogo dati.
3. Seleziona la scheda Cataloghi.

4. Scegli il catalogo a livello di account.
5. Scegli Ottimizzazioni della tabella, Modifica nella scheda Ottimizzazioni della tabella. Puoi anche scegliere Modifica ottimizzazioni dalla sezione Azioni.




6. Nella pagina Ottimizzazione della tabella, configura le seguenti opzioni:


Optimization options
Table optimization
Compaction
Combine small data files into larger, more efficient files to optimize table performance.
Snapshot retention
Optimize table storage by removing old snapshots from metadata overhead and expiring files that are no longer needed.
Orphan file deletion
Automatically clean up orphan files periodically.
Optimization configuration
Configure Apache Iceberg table optimization to optimize storage and improve query performances. Learn more
Use default settings
Default settings have been defined to help you get started. You can change them at any time later.
Customize settings
Customize settings for your specific needs.
IAM Role
Configure for all selected optimizers.
Choose an IAM role
View
Create new IAM role
Snapshot retention configuration
Snapshot retention period
Configure how long Apache Iceberg retains historical table snapshots before deleting them.
Use the default retention period of 5 days for snapshots
Specify a custom value in days
Minimum snapshot to retain
Use the default setting of retaining one snapshot
Specify a custom value
Expired snapshots associated files
Delete associated files
Files associated with the expired snapshots will be deleted.
Orphan file deletion configuration
Files under the provided Table Location with a creation time older than this number of days will be deleted if they are no longer referenced by the Apache Iceberg Table metadata.
3
Days
Default is 3 days.

a. Configura le impostazioni di Compattazione:

- Abilita/disabilita la compattazione.
- Scegli il ruolo IAM che dispone delle autorizzazioni necessarie per eseguire gli ottimizzatori.

  Per ulteriori informazioni sui requisiti di autorizzazione per il ruolo IAM, consulta Prerequisiti per l'ottimizzazione delle tabelle.

b. Configura le impostazioni di Conservazione degli snapshot:

- Abilita/disabilita la conservazione.
- Imposta il periodo di conservazione dello snapshot in giorni: l'impostazione predefinita è 5 giorni.
- Imposta il numero di snapshot da mantenere: l'impostazione predefinita è 1 snapshot.
- Abilita/disabilita la pulizia dei file scaduti.

c. Configura le impostazioni di eliminazione dei file orfani:

- Abilita/disabilita l'eliminazione dei file orfani.
- Imposta il periodo di conservazione del file orfano in giorni: l'impostazione predefinita è 3 giorni.

7. Scegli Save (Salva).

Abilitazione degli ottimizzatori a livello di catalogo tramite AWS CLI

Utilizza il seguente comando CLI per aggiornare un catalogo esistente con le impostazioni dell'ottimizzatore:

Example Aggiorna il catalogo con le impostazioni dell'ottimizzatore

```
aws glue update-catalog \
   --name catalog-id \
  --catalog-input \
  '{
    "CatalogId": "111122223333",
    "CatalogInput": {
        "CatalogProperties": {
            "CustomProperties": {
                "ColumnStatistics.Enabled": "false",
```

```
                "ColumnStatistics.RoleArn": "arn:aws:iam::111122223333:role/service-
role/stats-role-name"
            },
            "IcebergOptimizationProperties": {
                "RoleArn": "arn:aws:iam::111122223333:role/optimizer-role-name",
                "Compaction": {
                    "enabled": "true"
                },
                "Retention": {
                    "enabled": "true",
                    "snapshotRetentionPeriodInDays": "10",
                    "numberOfSnapshotsToRetain": "5",
                    "cleanExpiredFiles": "true"
                },
                "OrphanFileDeletion": {
                    "enabled": "true",
                    "orphanFileRetentionPeriodInDays": "3"
                }
            }
        }
    }
}'
```

Se riscontri problemi con gli ottimizzatori a livello di catalogo, controlla quanto segue:

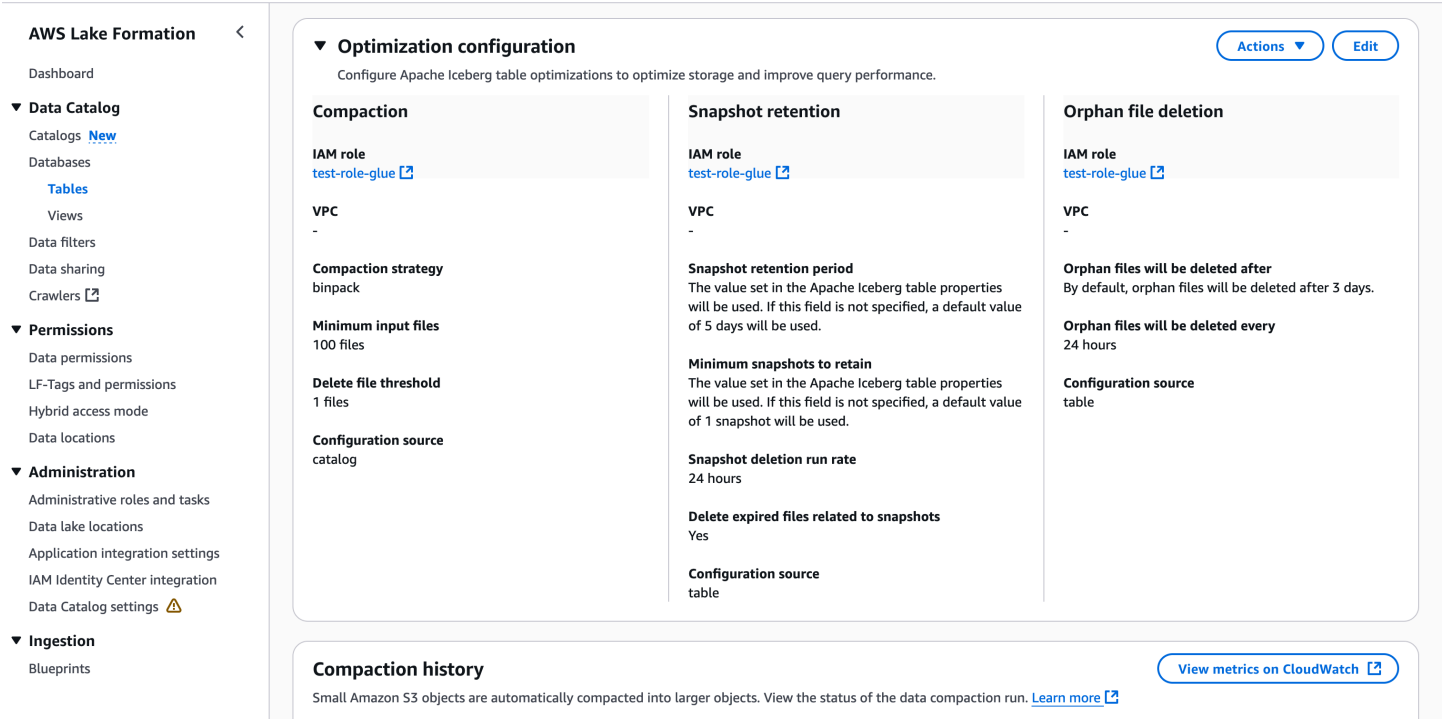
- Assicurati che il ruolo IAM disponga delle autorizzazioni corrette, come indicato nella sezione Prerequisiti.
- Controlla CloudWatch i log per eventuali messaggi di errore relativi alle operazioni dell'ottimizzatore.

  Per ulteriori informazioni, consultare Visualizzazione di parametri disponibili nella Guida per l'utente di Amazon CloudWatch .
- Verifica che le impostazioni del catalogo siano state applicate correttamente controllando la configurazione del catalogo.
- Per gli errori di accesso alle tabelle, controlla i CloudWatch log e le EventBridge notifiche per informazioni dettagliate sugli errori.

Visualizzazione delle ottimizzazioni a livello di catalogo

Quando l'ottimizzazione della tabella a livello di catalogo è abilitata, ogni volta che viene creata o aggiornata una tabella Apache Iceberg tramite `CreateTable` o `UpdateTable` APIs tramite SDK o Console di gestione AWS Crawler di AWS Glue, viene creata un'impostazione a livello di tabella equivalente per quella tabella.

Dopo aver creato o aggiornato una tabella, è possibile verificare i dettagli della tabella per confermarne l'ottimizzazione. `Table optimization` mostra la proprietà `Configuration source` impostata come `Catalog`.




Disabilitazione dell'ottimizzazione delle tabelle a livello di catalogo

Puoi disabilitare l'ottimizzazione delle tabelle per le nuove tabelle utilizzando la AWS Lake Formation console, l'`glue:UpdateCatalog`API.

Per disabilitare le ottimizzazioni delle tabelle a livello di catalogo

1. Apri la console Lake Formation all'indirizzo https://console.aws.amazon.com/lakeformation/.
2. Nella barra di navigazione a sinistra, scegli Cataloghi.
3. Nella pagina di Riepilogo del catalogo, scegli Modifica in Ottimizzazioni delle tabelle.
4. Nella pagina Modifica ottimizzazione, deselezionare le opzioni di ottimizzazione.

5. Scegli Save (Salva).

## Ottimizzazione della compattazione

I data lake Amazon S3 che utilizzano formati di tabelle aperte come Apache Iceberg archiviano i dati come oggetti S3. La presenza di migliaia di piccoli oggetti Amazon S3 in una tabella di data lake aumenta il sovraccarico dei metadati e influisce sulle prestazioni di lettura. AWS Glue Data Catalog fornisce una compattazione gestita per le tabelle Iceberg, compattando oggetti di piccole dimensioni in oggetti più grandi per migliorare le prestazioni di lettura tramite servizi di AWS analisi come Amazon Athena Amazon EMR ed ETL. AWS Glue Catalogo dati esegue la compattazione senza interferire con le query simultanee e supporta la compattazione solo per le tabelle in formato Parquet.

L'ottimizzatore delle tabelle monitora continuamente le partizioni delle tabelle e avvia il processo di compattazione quando viene superata la soglia per il numero di file e le dimensioni dei file.

Nel Catalogo dati, il processo di compattazione inizia quando una tabella o una delle sue partizioni contiene più di 100 file. Ogni file deve essere inferiore al 75% della dimensione del file di destinazione. La dimensione del file di destinazione è definita dalla proprietà della tabella `write.target-file-size-bytes`, che per impostazione predefinita è 512 MB, se non è impostata in modo esplicito.

Per le limitazioni, consulta Formati supportati e limitazioni per la compattazione gestita dei dati.

Argomenti

- Abilitazione dell'ottimizzatore di compattazione
- Disabilitazione dell'ottimizzatore di compattazione

### Abilitazione dell'ottimizzatore di compattazione

Puoi utilizzare la AWS Glue console o l' AWS CLI AWS API per abilitare la compattazione delle tabelle Apache Iceberg nel AWS Glue Data Catalog. Per le nuove tabelle, puoi scegliere Apache Iceberg come formato di tabella e abilitare la compattazione quando crei la tabella. La compattazione è disabilitata per impostazione predefinita per le nuove tabelle.

Console

Per abilitare la compattazione

1. Apri la AWS Glue console all'indirizzo https://console.aws.amazon.com/glue/e accedi come amministratore del data lake, creatore della tabella o utente a cui sono state concesse `lakeformation:GetDataAccess` le autorizzazioni `glue:UpdateTable` e sulla tabella.
2. Nel pannello di navigazione, in Catalogo dati, seleziona Tabelle.
3. Nella pagina Tabelle, scegli una tabella in formato tabella aperta per la quale desideri abilitare la compattazione, quindi nel menu Azioni, scegli Ottimizzazione e poi Abilita.

   Puoi anche abilitare la compattazione selezionando l'Ottimizzazione della tabella nella pagina dei Dettagli della tabella. Scegli la scheda Ottimizzazione della tabella nella sezione inferiore della pagina e scegli Abilita compattazione.

   L'opzione Abilita ottimizzazione è disponibile anche quando crei una nuova tabella Iceberg nel Catalogo dati.
4. Nella pagina Abilita ottimizzazione, scegli Compattazione in Opzioni di ottimizzazione.

## Optimization configuration

Configure Apache Iceberg table optimization to optimize storage and improve query performances. Learn more

- Use default settings
  Default settings have been defined to help you get started. You can change them at any time later.
- Customize settings
  Customize settings for your specific needs.

### IAM Role

Configure for all selected optimizers.

- [x] Apply the selected IAM role to all selected optimizers. Update individual roles later as needed.
  Create a single role with necessary permissions to configure all table optimizers. Learn more

**IAM role**

*Choose an IAM role* | View

Create new IAM role

**Virtual private cloud (VPC) - *optional***

Choose a network connection to access a data store in your VPC.

*Choose a Glue connection* | Create Glue network connection

### Compaction configuration

**Compaction strategy**

Choose a compaction strategy Learn more

- Binpack
  Combines small files into larger files, typically targeting sizes over 100MB, while applying any pending deletes. This is the default and recommended compaction strategy for most use cases.
- Sort
  Organizes data based on specified columns which are sorted hierarchically during compaction, improving query performance for filtered operations. This strategy is recommended when your queries frequently filter on specific columns. To use this strategy, you must first define a sort order in your Iceberg table properties using the 'sort_order' table property.
- Z-Order
  Optimizes data organization by blending multiple attributes into a single scalar value that can be used for sorting, allowing efficient querying across multiple dimensions. This strategy is recommended when you need to query data across multiple dimensions simultaneously. To use this strategy, you must first define a sort order in your Apache Iceberg table properties using the 'sort_order' table property.

**Minimum input files**

Minimum number of data-files to be present in a partition before compaction will actually compact files.

100 files

Default is 100 files.

**Delete file threshold**

Minimum number of deletes in a data-file to be eligible for compaction.

1 deletes

Default is 1 delete.

5. Successivamente, seleziona un ruolo IAM dal menu a discesa con le autorizzazioni mostrate nella sezione Prerequisiti per l'ottimizzazione delle tabelle.

   Puoi anche scegliere l'opzione Crea un nuovo ruolo IAM per creare un ruolo personalizzato con le autorizzazioni necessarie per eseguire la compattazione.

   Segui la procedura riportata di seguito per aggiornare un ruolo IAM esistente:

   a. Per aggiornare la politica di autorizzazione per il ruolo IAM, nella console IAM, vai al ruolo IAM utilizzato per eseguire la compattazione.

   b. Nella sezione Aggiungi autorizzazioni, scegli Crea policy. Nella finestra del browser appena aperta, crea una nuova policy da utilizzare con il tuo ruolo.

   c. Nella pagina Crea policy, scegli la `JSON` scheda. Copia il codice JSON mostrato nei Prerequisiti nel campo dell'editor delle policy.

6. Se hai configurazioni di policy di sicurezza in cui l'ottimizzatore di tabelle Iceberg deve accedere ai bucket Amazon S3 da uno specifico Virtual Private Cloud (VPC), crea una connessione di rete o usane una esistente. AWS Glue

   Se non hai già configurato una connessione AWS Glue VPC, creane una nuova seguendo i passaggi nella sezione Creazione di connessioni per connettori utilizzando la AWS Glue console o /SDK. AWS CLI

7. Scegliete una strategia di compattazione. Le opzioni disponibili sono:

   - Binpack: è la strategia di compattazione predefinita in Apache Iceberg. Combina file di dati più piccoli in file più grandi per prestazioni ottimali.
   - Ordinamento: l'ordinamento in Apache Iceberg è una tecnica di organizzazione dei dati che raggruppa le informazioni all'interno dei file in base a colonne specifiche, migliorando significativamente le prestazioni delle query riducendo il numero di file da elaborare. L'ordinamento viene definito nei metadati di Iceberg utilizzando il campo di ordinamento e, quando vengono specificate più colonne, i dati vengono ordinati nella sequenza in cui le colonne appaiono nell'ordine di ordinamento, assicurando che i record con valori simili vengano archiviati insieme all'interno dei file. La strategia di ordinamento e compattazione migliora l'ottimizzazione, ordinando i dati tra tutti i file all'interno di una partizione.
   - Ordinamento Z: è un modo per organizzare i dati quando è necessario ordinarli in base a più colonne con uguale importanza. A differenza dell'ordinamento tradizionale che dà priorità a una colonna rispetto alle altre, l'ordinamento Z attribuisce un peso equilibrato a ciascuna colonna, aiutando il motore di query a leggere meno file durante la ricerca di dati.

La tecnica funziona intrecciando le cifre binarie dei valori di colonne diverse. Ad esempio, se hai i numeri 3 e 4 di due colonne, l'ordinamento Z li converte prima in binario (3 diventa 011 e 4 diventa 100), quindi esegue l'interleave di queste cifre per creare un nuovo valore: 011010. Attraverso l'interleave si crea uno schema che mantiene i dati correlati fisicamente vicini tra loro.

L'ordinamento Z è particolarmente efficace per le query multidimensionali. Ad esempio, una tabella clienti ordinata in base a reddito, stato e codice postale può offrire prestazioni superiori rispetto all'ordinamento gerarchico quando si eseguono query su più dimensioni. Questa organizzazione consente di eseguire query mirate a combinazioni specifiche di reddito e posizione geografica per individuare rapidamente i dati pertinenti riducendo al minimo le scansioni dei file non necessarie.

8. Numero minimo di file di input: il numero di file di dati necessari in una partizione prima che venga attivata la compattazione.
9. Soglia di eliminazione dei file: operazioni di eliminazione minime richieste in un file di dati prima che diventi idoneo alla compattazione.
10. Scegli Abilita ottimizzazione.

AWS CLI

L'esempio seguente mostra come abilitare la compattazione. Sostituisci l'ID dell'account con un ID account valido. AWS Sostituisci il nome del database e della tabella con quello effettivo della tabella Iceberg e del database. Sostituisci `roleArn` con il nome della risorsa (ARN) AWS del ruolo IAM e il nome del ruolo IAM che dispone delle autorizzazioni necessarie per eseguire la compattazione. Puoi sostituire la strategia di compattazione `sort` con altre strategie supportate come `z-order` o `binpack`.

ordina" in base alle tue esigenze.

```
aws glue create-table-optimizer \
  --catalog-id 123456789012 \
  --database-name iceberg_db \
  --table-name iceberg_table \
  --table-optimizer-configuration '{
    "roleArn": "arn:aws:iam::123456789012:role/optimizer_role",
    "enabled": true,
    "vpcConfiguration": {"glueConnectionName": "glue_connection_name"},
    "compactionConfiguration": {
```

```
      "icebergConfiguration": {"strategy": "sort"}
    }
  }'\
--type compaction
```

AWS API

Chiama l'operazione CreateTableOptimizer per abilitare la compattazione di una tabella.

Dopo aver abilitato la compattazione, la scheda di Ottimizzazione della tabella mostra i seguenti dettagli di compattazione, una volta completata la compattazione:

Ora di inizio

L'ora in cui è iniziato il processo di compattazione all'interno del Catalogo dati. Il valore è un timestamp in formato UTC.

Ora di fine

L'ora in cui è terminato il processo di compattazione nel Catalogo dati. Il valore è un timestamp in formato UTC.

Status

Lo stato dell'esecuzione di compattazione. I valori sono esito positivo o negativo.

File compattati

Numero totale di file compattati

Byte compattati

Numero totale di byte compattati

### Disabilitazione dell'ottimizzatore di compattazione

È possibile disabilitare la compattazione automatica per una particolare tabella Apache Iceberg utilizzando AWS Glue la console o. AWS CLI

Console

1. Accedi a Console di gestione AWS e apri la console all' AWS Glue indirizzo. https://console.aws.amazon.com/glue/
2. Nella barra di navigazione a sinistra, in Catalogo dati, seleziona Tabelle.

3. Dall'elenco delle tabelle, seleziona la tabella Iceberg di cui desideri disabilitare la compattazione.
4. Seleziona la scheda Ottimizzazione delle tabelle nella sezione inferiore della pagina Dettagli delle tabelle.
5. Da Azioni, seleziona Disabilita, quindi scegli Compattazione.
6. Scegli Disabilita la compattazione nel messaggio di conferma. È possibile abilitare nuovamente la compattazione in un secondo momento.

   Dopo la conferma, la compattazione viene disabilitata e il relativo stato torna a `Disabled`.

AWS CLI

Nell'esempio seguente, sostituisci l'ID account con un ID AWS account valido. Sostituisci il nome del database e della tabella con quello effettivo della tabella Iceberg e del database. Sostituisci `roleArn` con il nome della AWS risorsa (ARN) del ruolo IAM e il nome effettivo del ruolo IAM che dispone delle autorizzazioni necessarie per eseguire la compattazione.

```
aws glue update-table-optimizer \
  --catalog-id 123456789012 \
  --database-name iceberg_db \
  --table-name iceberg_table \
  --table-optimizer-configuration
 '{"roleArn":"arn:aws:iam::123456789012:role/optimizer_role", "enabled":'false',
 "vpcConfiguration":{"glueConnectionName":"glue_connection_name"}}'\
  --type compaction
```

AWS API

Chiama l'operazione UpdateTableOptimizer per disabilitare la compattazione per una tabella specifica.

## Ottimizzazione di conservazione degli snapshot

La funzionalità di conservazione degli snapshot di Apache Iceberg consente agli utenti di effettuare query sui dati storici in momenti specifici e ripristinare le modifiche indesiderate alle tabelle. Nel AWS Glue Data Catalog, la configurazione di conservazione delle istantanee controlla per quanto tempo queste istantanee (versioni dei dati della tabella) vengono conservate prima di essere scadute e rimosse. Ciò aiuta a gestire i costi di archiviazione e il sovraccarico dei metadati rimuovendo

automaticamente gli snapshot meno recenti in base a un periodo di conservazione configurato o al numero massimo di snapshot da conservare.

È possibile configurare il periodo di conservazione in giorni e il numero massimo di istantanee da conservare per una tabella. AWS Glue rimuove le istantanee più vecchie del periodo di conservazione specificato dai metadati della tabella, mantenendo le istantanee più recenti fino al limite configurato. Dopo aver rimosso le vecchie istantanee dai metadati, AWS Glue elimina i file di dati e metadati corrispondenti che non sono più referenziati e univoci per le istantanee scadute. In questo modo, è possibile effettuare query temporali solo fino ai restanti snapshot conservati, recuperando al contempo lo spazio di archiviazione utilizzato dai dati degli snapshot scaduti.

Argomenti


- Abilitazione dell'ottimizzatore per la conservazione degli snapshot
- Ottimizzatore per la conservazione degli snapshot
- Disabilitazione dell'ottimizzatore per la conservazione degli snapshot


## Abilitazione dell'ottimizzatore per la conservazione degli snapshot

Puoi utilizzare la AWS Glue console o l' AWS API per abilitare gli ottimizzatori di conservazione delle istantanee per le tue tabelle Apache Iceberg nel Data Catalog. AWS CLI Per le nuove tabelle, puoi scegliere Apache Iceberg come formato di tabella e abilitare l'ottimizzatore dell'eliminazione dei file orfani quando crei la tabella. La conservazione degli snapshot è disabilitata per impostazione predefinita per le nuove tabelle.

### Console

#### Abilitazione dell'ottimizzatore per la conservazione degli snapshot

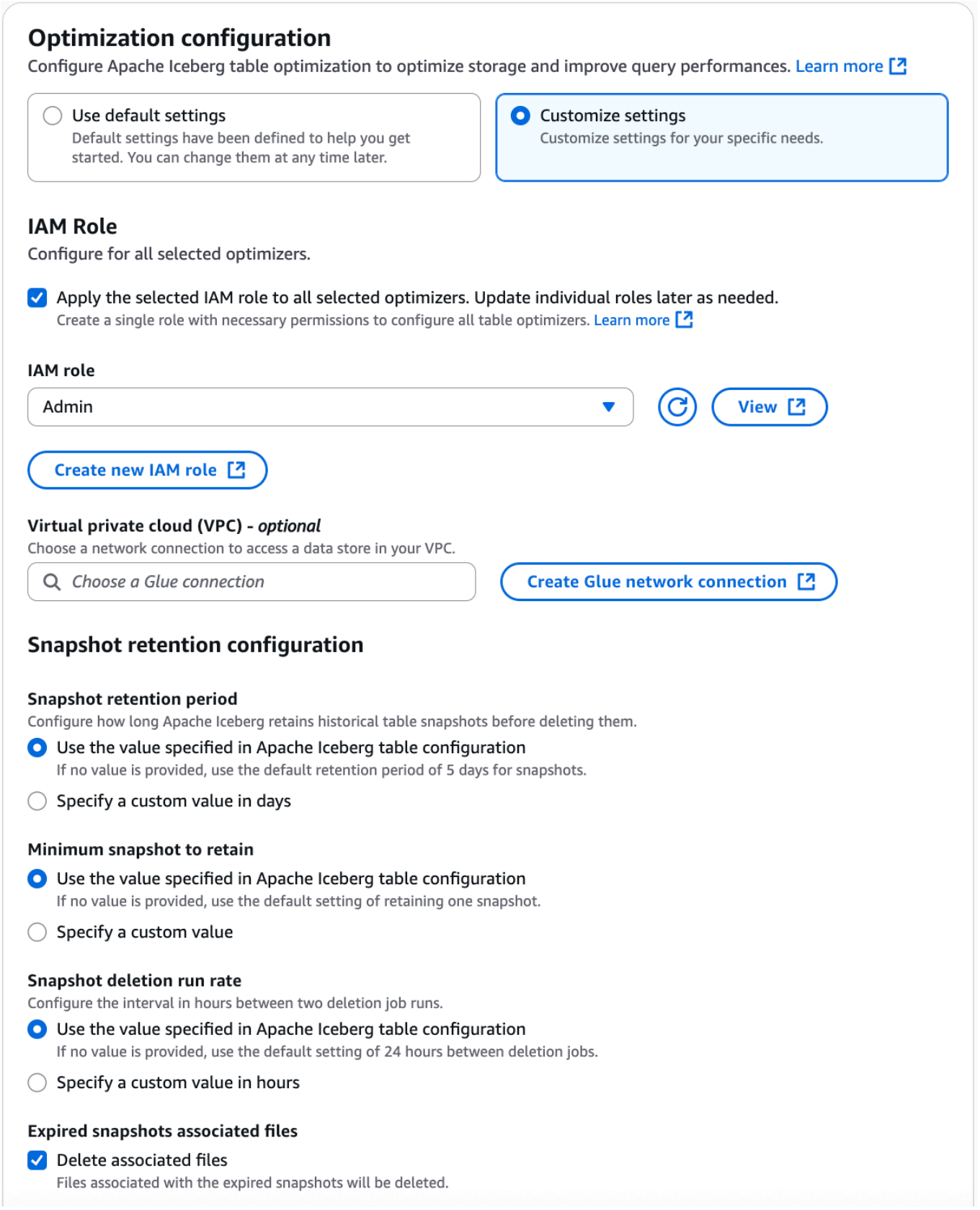
1. Apri la AWS Glue console all'indirizzo https://console.aws.amazon.com/glue/e accedi come amministratore del data lake, creatore della tabella o utente a cui sono state concesse le `lakeformation:GetDataAccess` autorizzazioni `glue:UpdateTable` e sulla tabella.
2. Nel pannello di navigazione, in Catalogo dati, seleziona Tabelle.
3. Nella pagina Tabelle, scegli una tabella Iceberg per la quale desideri abilitare l'ottimizzatore di conservazione degli snapshot, quindi nel menu Azioni, scegli Abilita in Ottimizzazione.

   Puoi anche abilitare l'ottimizzazione selezionando la tabella e aprendo la pagina dei Dettagli della tabella. Scegli la scheda Ottimizzazione della tabella nella sezione inferiore della pagina e scegli Abilita la conservazione degli snapshot.

4. Nella pagina Abilita ottimizzazione , in Configurazione dell'ottimizzazione, sono disponibili due opzioni: Usa l'impostazione predefinita o Personalizza le impostazioni. Se scegli di utilizzare le impostazioni predefinite, AWS Glue utilizza le proprietà definite nella configurazione della tabella Iceberg per determinare il periodo di conservazione delle istantanee e il numero di istantanee da conservare. In assenza di questa configurazione, AWS Glue conserva un'istantanea per cinque giorni ed elimina i file associati alle istantanee scadute.
5. Quindi, scegli un ruolo IAM che AWS Glue possa assumere per tuo conto per eseguire l'ottimizzatore. Per ulteriori informazioni sulle autorizzazioni necessarie per il ruolo IAM, consulta la sezione Prerequisiti per l'ottimizzazione delle tabelle.

   Segui la procedura riportata di seguito per aggiornare un ruolo IAM esistente:

   a. Per aggiornare la politica di autorizzazione per il ruolo IAM, nella console IAM, vai al ruolo IAM utilizzato per eseguire la compattazione.
   b. Nella sezione Aggiungi autorizzazioni, scegli Crea policy. Nella finestra del browser appena aperta, crea una nuova policy da utilizzare con il tuo ruolo.
   c. Nella pagina Crea policy, scegli la scheda JSON. Copia il codice JSON mostrato nei Prerequisiti nel campo dell'editor delle policy.
6. Se preferisci impostare manualmente i valori per la Configurazione della conservazione degli snapshot, scegli Personalizza impostazioni.


Optimization configuration
Configure Apache Iceberg table optimization to optimize storage and improve query performances. Learn more
Use default settings
Default settings have been defined to help you get started. You can change them at any time later.
Customize settings
Customize settings for your specific needs.
IAM Role
Configure for all selected optimizers.
Apply the selected IAM role to all selected optimizers. Update individual roles later as needed.
Create a single role with necessary permissions to configure all table optimizers. Learn more
IAM role
Admin
View
Create new IAM role
Virtual private cloud (VPC) - optional
Choose a network connection to access a data store in your VPC.
Choose a Glue connection
Create Glue network connection
Snapshot retention configuration
Snapshot retention period
Configure how long Apache Iceberg retains historical table snapshots before deleting them.
Use the value specified in Apache Iceberg table configuration
If no value is provided, use the default retention period of 5 days for snapshots.
Specify a custom value in days
Minimum snapshot to retain
Use the value specified in Apache Iceberg table configuration
If no value is provided, use the default setting of retaining one snapshot.
Specify a custom value
Snapshot deletion run rate
Configure the interval in hours between two deletion job runs.
Use the value specified in Apache Iceberg table configuration
If no value is provided, use the default setting of 24 hours between deletion jobs.
Specify a custom value in hours
Expired snapshots associated files
Delete associated files
Files associated with the expired snapshots will be deleted.

7. Scegli la casella Applica il ruolo IAM selezionato agli ottimizzatori selezionati per utilizzare un singolo ruolo IAM per tutti, abilitando tutti gli ottimizzatori.
8. Se hai configurazioni di policy di sicurezza in cui l'ottimizzatore di tabelle Iceberg deve accedere ai bucket Amazon S3 da uno specifico Virtual Private Cloud (VPC), crea una connessione di rete o usane una esistente. AWS Glue

   Se non hai già configurato una connessione AWS Glue VPC, creane una nuova seguendo i passaggi nella sezione Creazione di connessioni per connettori utilizzando la AWS Glue console o /SDK. AWS CLI
9. Successivamente, in Configurazione di conservazione delle istantanee, scegli di utilizzare i valori specificati nella configurazione della tabella Iceberg o specifica valori personalizzati per il periodo di conservazione delle istantanee (history.expire). max-snapshot-age-ms), numero minimo di istantanee (history.expire). min-snapshots-to-keep) da conservare e il tempo in ore tra l'esecuzione di un processo di eliminazione consecutiva delle istantanee.
10. Scegli Elimina file associati per eliminare i file sottostanti quando l'ottimizzatore di tabella elimina i vecchi snapshot dai metadati della tabella.

    Se non scegli questa opzione, quando gli snapshot più vecchie vengono rimosse dai metadati della tabella, i file associati rimarranno nell'archivio come file orfani.
11. Quindi, leggi l'avviso di avvertenza e scegli Confermo per procedere.

    > **Note**
    >
    > Nel Catalogo dati, l'ottimizzatore di conservazione degli snapshot rispetta il ciclo di vita controllato da politiche di conservazione a livello di filiale e tag. Per ulteriori informazioni, consultare la sezione Ramificazione e tagging nella documentazione di Apache Iceberg.
12. Rivedi la configurazione e scegli Abilita ottimizzazione.

    Attendi qualche minuto che l'ottimizzatore di conservazione venga eseguito e i vecchi snapshot scadano in base alla configurazione.

AWS CLI

Per abilitare la conservazione delle istantanee per le nuove tabelle Iceberg in AWS Glue, è necessario creare un ottimizzatore di tabelle di tipo `retention` e impostare il `enabled`

campo su in. `true` `table-optimizer-configuration` È possibile farlo utilizzando il AWS CLI comando o. `create-table-optimizer` `update-table-optimizer` Inoltre, è necessario specificare i campi di configurazione della conservazione in modo analogo `snapshotRetentionPeriodInDays` e `numberOfSnapshotsToRetain` in base alle proprie esigenze.

L'esempio seguente mostra come abilitare l'ottimizzatore per la conservazione degli snapshot. Sostituisci l'ID dell'account con un ID AWS account valido. Sostituisci il nome del database e della tabella con quello effettivo della tabella Iceberg e del database. Sostituisci `roleArn` con il nome della AWS risorsa (ARN) del ruolo IAM e il nome del ruolo IAM che dispone delle autorizzazioni necessarie per eseguire lo snapshot retention optimizer.

```
aws glue create-table-optimizer \
  --catalog-id 123456789012 \
  --database-name iceberg_db \
  --table-name iceberg_table \
  --table-optimizer-configuration
 '{"roleArn":"arn:aws:iam::123456789012:role/optimizer_role","enabled":'true',
 "vpcConfiguration":{
"glueConnectionName":"glue_connection_name"}, "retentionConfiguration":
{"icebergConfiguration":
{"snapshotRetentionPeriodInDays":7,"numberOfSnapshotsToRetain":3,"cleanExpiredFiles":'true'}
  --type retention
```

Questo comando crea un ottimizzatore di conservazione per la tabella Iceberg nel catalogo, nel database e nella regione specificati. table-optimizer-configurationspecifica il ruolo IAM ARN da utilizzare, abilita l'ottimizzatore e imposta la configurazione di conservazione. In questo esempio, conserva gli snapshot per 7 giorni, mantiene almeno 3 snapshot e pulisce i file scaduti.

- snapshotRetentionPeriodInDays —Il numero di giorni in cui conservare le istantanee prima della loro scadenza. Il valore predefinito è 5.
- numberOfSnapshotsToRetain — Il numero minimo di istantanee da conservare, anche se sono più vecchie del periodo di conservazione. Il valore predefinito è 1.
- cleanExpiredFiles — Un valore booleano che indica se eliminare i file di dati scaduti dopo la scadenza delle istantanee. Il valore predefinito è `true`.

  Se impostato su true, gli snapshot più vecchi vengono rimossi dai metadati della tabella e i relativi file sottostanti vengono eliminati. Se questo parametro è impostato su false, gli snapshot

più vecchi vengono rimossi dai metadati della tabella ma i relativi file sottostanti rimangono nell'archivio come file orfani.

AWS API

CreateTableOptimizerOperazione di chiamata per abilitare l'ottimizzatore della conservazione delle istantanee per una tabella.

Dopo aver abilitato la compattazione, la scheda di Ottimizzazione della tabella mostra i seguenti dettagli di compattazione, dopo circa 15-20 minuti:

Ora di inizio

L'ora in cui è stato avviato l'ottimizzatore per la conservazione degli snapshot. Il valore è un timestamp in formato UTC.

Tempo di esecuzione

Il tempo indica il tempo impiegato dall'ottimizzatore per completare l'operazione. Il valore è un timestamp in formato UTC.

Status

Lo stato dell'esecuzione dell'ottimizzatore. I valori sono esito positivo o negativo.

File di dati eliminati

Numero totale di file eliminati.

File manifesto eliminati

Numero totale di file manifesto eliminati.

Elenchi manifesto eliminati

Numero totale di elenchi manifesto eliminati.

Ottimizzatore per la conservazione degli snapshot

È possibile aggiornare la configurazione esistente di un ottimizzatore di conservazione delle istantanee per una particolare tabella Apache Iceberg utilizzando la AWS Glue console o l' AWS CLI API. UpdateTableOptimizer

### Console

Per aggiornare la configurazione di conservazione degli snapshot

1. Accedi Console di gestione AWS e apri la console all'indirizzo. AWS Glue https://console.aws.amazon.com/glue/
2. Scegliere Catalogo dati e poi Tabelle. Dall'elenco delle tabelle, selezionare la tabella Iceberg in cui aggiornare la configurazione dell'ottimizzatore per la conservazione degli snapshot.
3. Nella parte inferiore della pagina Dettagli tabella, selezionare la scheda Ottimizzazione tabella, poi scegliere Modifica. È anche possibile scegliere Modifica in Ottimizzazione dal menu Operazioni situato nell'angolo in alto a destra della pagina.
4. Nella pagina Modifica ottimizzazione, apportare le modifiche desiderate.
5. Scegli Save (Salva).

### AWS CLI

Per aggiornare un ottimizzatore di conservazione delle istantanee utilizzando AWS CLI, è possibile utilizzare il seguente comando:

```
aws glue update-table-optimizer \
 --catalog-id 123456789012 \
 --database-name iceberg_db \
 --table-name iceberg_table \
 --table-optimizer-configuration
 '{"roleArn":"arn:aws:iam::123456789012:role/optimizer_role"","enabled":'true',
 "vpcConfiguration":
{"glueConnectionName":"glue_connection_name"},"retentionConfiguration":
{"icebergConfiguration":
{"snapshotRetentionPeriodInDays":7,"numberOfSnapshotsToRetain":3,"cleanExpiredFiles":'true'}
 \
 --type retention
```

Questo comando aggiorna la configurazione della conservazione per la tabella specificata nel dato catalogo, nel database e nella regione. I parametri chiave sono:

- snapshotRetentionPeriodInDays —Il numero di giorni in cui conservare le istantanee prima della scadenza. Il valore predefinito è 1.
- numberOfSnapshotsToRetain — Il numero minimo di istantanee da conservare, anche se sono più vecchie del periodo di conservazione. Il valore predefinito è 5.

- cleanExpiredFiles — Un valore booleano che indica se eliminare i file di dati scaduti dopo la scadenza delle istantanee. Il valore predefinito è `true`.

  Se impostato su true, gli snapshot più vecchi vengono rimossi dai metadati della tabella e i relativi file sottostanti vengono eliminati. Se questo parametro è impostato su false, gli snapshot più vecchi vengono rimossi dai metadati della tabella ma i relativi file sottostanti rimangono nell'archivio come file orfani.

API

Per aggiornare un ottimizzatore di tabelle, si può utilizzare l'API `UpdateTableOptimizer`. Questa API consente di aggiornare la configurazione di un ottimizzatore di tabelle esistente per la compattazione, la conservazione o la rimozione di file orfani. I parametri della richiesta includono:

- catalogId (obbligatorio): l'ID del catalogo contenente la tabella
- databaseName (opzionale): il nome del database contenente la tabella
- tableName (opzionale): il nome della tabella
- type (obbligatorio): il tipo di ottimizzatore della tabella (compattazione, conservazione o orphan_file_delete)
- retentionConfiguration (obbligatorio): la configurazione aggiornata per l'ottimizzatore delle tabelle, che include l'ARN del ruolo, lo stato di abilitazione, la configurazione di conservazione e la configurazione per la rimozione dei file orfani.

### Disabilitazione dell'ottimizzatore per la conservazione degli snapshot

È possibile disabilitare l'ottimizzatore di conservazione delle istantanee per una particolare tabella Apache Iceberg utilizzando la console o. AWS Glue AWS CLI

Console

Per disabilitare la conservazione degli snapshot

1. Accedi a Console di gestione AWS e apri la console all'indirizzo. AWS Glue https://console.aws.amazon.com/glue/
2. Scegliere Catalogo dati e poi Tabelle. Dall'elenco delle tabelle, seleziona la tabella Iceberg per la quale vuoi disabilitare l'ottimizzatore per la conservazione degli snapshot.

3. Nella parte inferiore della pagina Dettagli tabella, seleziona Ottimizzazione tabella e Disabilita, Conservazione snapshot in Azioni.

   Puoi anche scegliere Disabilita in Ottimizzazione dal menu Azioni situato nell'angolo in alto a destra della pagina.
4. Scegli Disabilita nel messaggio di conferma. Potrai riattivare l'ottimizzatore di conservazione degli snapshot in un secondo momento.

   Dopo la conferma, l'ottimizzatore di conservazione degli snapshot viene disabilitato e lo stato di conservazione degli snapshot torna su `Not enabled`.

AWS CLI

Nell'esempio seguente, sostituisci l'ID account con un ID AWS account valido. Sostituisci il nome del database e della tabella con quello effettivo della tabella Iceberg e del database. Sostituisci `roleArn` con il nome della AWS risorsa (ARN) del ruolo IAM e il nome effettivo del ruolo IAM che dispone delle autorizzazioni necessarie per eseguire l'ottimizzatore della conservazione.

```
aws glue update-table-optimizer \
  --catalog-id 123456789012 \
  --database-name iceberg_db \
  --table-name iceberg_table \
  --table-optimizer-configuration
 '{"roleArn":"arn:aws:iam::123456789012:role/optimizer_role", "vpcConfiguration":
{"glueConnectionName":"glue_connection_name"}, "enabled":'false'}'\
  --type retention
```

AWS API

UpdateTableOptimizerOperazione di chiamata per disabilitare l'ottimizzatore di conservazione delle istantanee per una tabella specifica.

## Eliminazione di file orfani

AWS Glue Data Catalog ti permette di rimuovere file orfani dalle tue tabelle Iceberg. I file orfani sono file non referenziati presenti nell'origine dati di Amazon S3 nella posizione della tabella specificata, non sono tracciati dai metadati della tabella Iceberg e sono più vecchi del limite di età configurato. Questi file orfani possono accumularsi nel tempo a causa di errori in operazioni come compattazione, cali di partizione o riscritture di tabelle, e occupare spazio di archiviazione non necessario.

L'ottimizzatore per l'eliminazione dei file AWS Glue orfani analizza i metadati della tabella e i file di dati effettivi, identifica i file orfani e li elimina per recuperare spazio di archiviazione. L'ottimizzatore rimuove solo i file creati dopo la data di creazione dell'ottimizzatore che soddisfano anche i criteri di eliminazione configurati. I file creati prima o alla data di creazione dell'ottimizzatore non vengono mai eliminati.

Logica di eliminazione dei file orfani

1. Controllo della data: confronta la data di creazione del file con quella di creazione dell'ottimizzatore. Se il file è precedente o contemporaneo alla data di creazione dell'ottimizzatore, il file viene ignorato.
2. Controllo della configurazione dell'ottimizzatore: se il file è più recente della data di creazione dell'ottimizzatore, valuta il file rispetto al limite di età configurato. L'ottimizzatore elimina il file se soddisfa i criteri di eliminazione. Ignora il file, se non corrisponde ai criteri.

È possibile avviare l'eliminazione dei file orfani creando un ottimizzatore di tabelle per l'eliminazione dei file orfani nel catalogo dati.

> **Important**
>
> Per impostazione predefinita, l'eliminazione dei file orfani valuta i file in tutta la posizione della tabella. AWS Glue Sebbene sia possibile configurare un prefisso secondario per limitare l'ambito di valutazione utilizzando il parametro API, è necessario assicurarsi che la posizione della tabella non contenga file provenienti da altre origini dati o tabelle. Se la posizione della tabella si sovrappone ad altre origini dati, il servizio potrebbe identificare ed eliminare i file non correlati come orfani.

Argomenti


- Attivazione dell'eliminazione di file orfani
- Aggiornamento dell'ottimizzatore per l'eliminazione dei file orfani
- Disabilitazione dell'eliminazione di file orfani


Attivazione dell'eliminazione di file orfani

Puoi utilizzare la AWS Glue console o l' AWS API per abilitare l'eliminazione di file orfani per le tue tabelle Apache Iceberg nel Data Catalog. AWS CLI Per le nuove tabelle, puoi scegliere Apache

Iceberg come formato di tabella e abilitare l'ottimizzatore dell'eliminazione dei file orfani quando crei la tabella. La conservazione degli snapshot è disabilitata per impostazione predefinita per le nuove tabelle.

Console

Abilitazione dell'eliminazione di file orfani

1. Apri la AWS Glue console all'indirizzo https://console.aws.amazon.com/glue/e accedi come amministratore del data lake, creatore della tabella o utente a cui sono state concesse `lakeformation:GetDataAccess` le autorizzazioni `glue:UpdateTable` e sulla tabella.
2. Nel pannello di navigazione, in Catalogo dati, seleziona Tabelle.
3. Nella pagina Tabelle, scegli una tabella Iceberg in cui desideri abilitare l'eliminazione dei file orfani.

   Scegli la scheda Ottimizzazione della tabella nella sezione inferiore della pagina e scegli Abilita, Eliminazione dei file orfani da Azioni.

   Puoi anche scegliere Abilita in Ottimizzazione dal menu Azioni situato nell'angolo in alto a destra della pagina.
4. Nella pagina Abilita ottimizzazione, scegli Eliminazione dei file orfani in Opzioni di ottimizzazione.
5. Se scegli di utilizzare le Impostazioni predefinite, tutti i file orfani verranno eliminati dopo 3 giorni. Se desideri conservare i file orfani per un numero specifico di giorni, scegli Personalizza impostazioni.
6. Quindi, scegli un ruolo IAM con le autorizzazioni necessarie per eliminare i file orfani.
7. Se hai configurazioni di policy di sicurezza in cui l'ottimizzatore di tabelle Iceberg deve accedere ai bucket Amazon S3 da uno specifico Virtual Private Cloud (VPC), crea una connessione di rete o usane una esistente. AWS Glue

   Se non hai già configurato una connessione AWS Glue VPC, creane una nuova seguendo i passaggi nella sezione Creazione di connessioni per connettori utilizzando la AWS Glue console o /SDK. AWS CLI
8. Se scegli Personalizza impostazioni, inserisci il numero di giorni durante cui mantenere i file prima dell'eliminazione in Configurazione dell'eliminazione dei file orfani. Puoi anche specificare l'intervallo tra due esecuzioni consecutive dell'ottimizzatore. Il valore predefinito è 24 ore.

9. Scegli Abilita ottimizzazione.

AWS CLI

Per abilitare l'eliminazione di file orfani per una tabella Iceberg in AWS Glue, devi creare un ottimizzatore di tabella di tipo `orphan_file_deletion` e impostare il campo su true. `enabled` Per creare un ottimizzatore di eliminazione di file orfano per una tabella Iceberg utilizzando il AWS CLI, è possibile utilizzare il seguente comando:

```
aws glue create-table-optimizer \
 --catalog-id 123456789012 \
 --database-name iceberg_db \
 --table-name iceberg_table \
 --table-optimizer-configuration
 '{"roleArn":"arn:aws:iam::123456789012:role/optimizer_role","enabled":true,
 "vpcConfiguration":{
"glueConnectionName":"glue_connection_name"}, "orphanFileDeletionConfiguration":
{"icebergConfiguration":{"orphanFileRetentionPeriodInDays":3, "location":'S3
 location'}}}'\
 --type orphan_file_deletion
```

Questo comando crea un ottimizzatore per l'eliminazione dei file orfani per la tabella Iceberg specificata. I parametri chiave sono:

- roleARN: l'ARN del ruolo IAM con autorizzazioni di accesso al bucket S3 e alle risorse Glue.
- enabled: impostare su true per abilitare l'ottimizzatore.
- orphanFileRetentionPeriodInDays — Il numero di giorni per conservare i file orfani prima di eliminarli (minimo 1 giorno).
- type: imposta su orphan_file_deletion per creare un ottimizzatore per l'eliminazione dei file orfani.

Dopo averlo creato, l'ottimizzatore di tabelle, eseguirà periodicamente l'eliminazione dei file orfani (una volta al giorno se lasciato abilitato). Puoi controllare le esecuzioni utilizzando l'`list-table-optimizer-runs`API. Il processo di eliminazione dei file orfani identificherà ed eliminerà i file che non sono tracciati nei metadati Iceberg per la tabella.

API

CreateTableOptimizerOperazione di chiamata per creare l'ottimizzatore per l'eliminazione dei file orfani per una tabella specifica.

### Aggiornamento dell'ottimizzatore per l'eliminazione dei file orfani

È possibile modificare la configurazione dell'ottimizzatore per l'eliminazione dei file orfani, ad esempio modificando il periodo di conservazione per i file orfani o il ruolo IAM utilizzato dall'ottimizzatore tramite AWS Glue console o l'operazione AWS CLI. `UpdateTableOptimizer`

Console di gestione AWS

Per aggiornare l'ottimizzatore per l'eliminazione dei file orfani

1. Scegliere Catalogo dati e poi Tabelle. Dall'elenco delle tabelle, selezionare la tabella in cui aggiornare la configurazione dell'ottimizzatore per l'eliminazione dei file orfani.
2. Nella parte inferiore della pagina Dettagli tabella, scegliere Ottimizzazione tabella, poi scegliere Modifica.
3. Nella pagina Modifica ottimizzazione, apportare le modifiche desiderate.
4. Scegli Save (Salva).

AWS CLI

È possibile utilizzare la `update-table-optimizer` chiamata per aggiornare l'ottimizzatore per l'eliminazione dei file orfani in AWS Glue, è possibile utilizzare call. Ciò consente di modificare `OrphanFileDeletionConfiguration` nel campo `icebergConfiguration` in cui è possibile specificare il `OrphanFileRetentionPeriodInDays` aggiornato per impostare il numero di giorni in cui mantenere i file orfani, per specificare la posizione della tabella Iceberg da cui eliminare questi ultimi.

```
aws glue update-table-optimizer \
 --catalog-id 123456789012 \
 --database-name iceberg_db \
 --table-name Iceberg_table \
 --table-optimizer-configuration
 '{"roleArn":"arn:aws:iam::123456789012:role/optimizer_role","enabled":true,
 "vpcConfiguration":
```

```
{"glueConnectionName":"glue_connection_name"},"orphanFileDeletionConfiguration":
{"icebergConfiguration":{"orphanFileRetentionPeriodInDays":5}}}' \
 --type orphan_file_deletion
```

API

Richiama l'UpdateTableOptimizeroperazione per aggiornare l'ottimizzatore per la cancellazione dei file orfani per una tabella.

### Disabilitazione dell'eliminazione di file orfani

È possibile disabilitare l'ottimizzatore di eliminazione dei file orfani per una particolare tabella Apache Iceberg utilizzando la console o. AWS Glue AWS CLI

Console

Per disabilitare l'eliminazione di file orfani

1. Scegliere Catalogo dati e poi Tabelle. Dall'elenco delle tabelle, seleziona la tabella Iceberg per la quale vuoi disabilitare l'ottimizzatore per l'eliminazione dei file orfani.
2. Nella parte inferiore della pagina Dettagli tabella, seleziona la scheda Ottimizzazione tabella.
3. Scegli Azioni, quindi seleziona Disabilita, Eliminazione file orfani.

   Puoi anche selezionare Disabilita in Ottimizzazione dal menu Azioni.
4. Scegli Disabilita nel messaggio di conferma. Potrai riattivare l'ottimizzatore per l'eliminazione dei file orfani in un secondo momento.

   Dopo la conferma, l'ottimizzatore per l'eliminazione dei file orfani viene disabilitato e lo stato dell'eliminazione dei file orfani torna su `Not enabled`.

AWS CLI

Nell'esempio seguente, sostituisci l'ID dell'account con un ID account valido. AWS Sostituisci il nome del database e della tabella con quello effettivo della tabella Iceberg e del database. Sostituisci `roleArn` con il nome della AWS risorsa (ARN) del ruolo IAM e il nome effettivo del ruolo IAM che dispone delle autorizzazioni necessarie per disabilitare l'ottimizzatore.

```
aws glue update-table-optimizer \
  --catalog-id 123456789012 \
```

```
  --database-name iceberg_db \
  --table-name iceberg_table \
  --table-optimizer-configuration
'{"roleArn":"arn:aws:iam::123456789012:role/optimizer_role", "enabled":'false'}'\
  --type orphan_file_deletion
```

API

Richiama l'UpdateTableOptimizeroperazione per disabilitare l'ottimizzatore di conservazione delle istantanee per una tabella specifica.

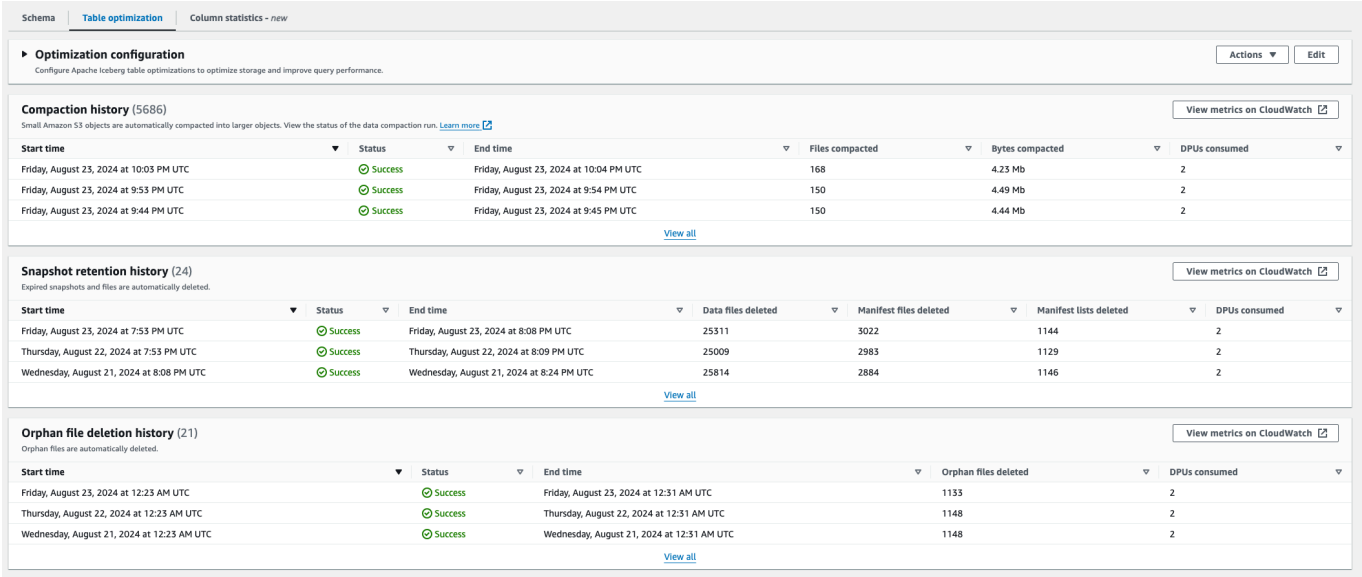
## Visualizzazione dei dettagli dell'ottimizzazione

È possibile visualizzare lo stato di ottimizzazione per le tabelle Apache Iceberg nella AWS Glue console o utilizzando AWS CLI le operazioni AWS API.

Console

Per visualizzare lo stato di ottimizzazione per le tabelle Iceberg (console)

- È possibile visualizzare lo stato di ottimizzazione per le tabelle Iceberg sulla AWS Glue console scegliendo una tabella Iceberg dall'elenco Tabelle in Data Catalog. In Ottimizzazione delle tabelle. Scegliere Visualizza tutto




AWS CLI

È possibile visualizzare i dettagli di ottimizzazione utilizzando. AWS CLI

Negli esempi seguenti, sostituisci l'ID account con un ID AWS account valido, il nome del database e il nome della tabella con il nome effettivo della tabella Iceberg. Per `type`, fornire un tipo di ottimizzazione. I valori accettabili sono `compaction`, `retention` e `orphan-file-deletion`.

- Per ottenere i dettagli dell'ultima esecuzione di compattazione per una tabella

```
aws get-table-optimizer \
  --catalog-id 123456789012 \
  --database-name iceberg_db \
  --table-name iceberg_table \
  --type compaction
```

- Utilizza l'esempio seguente per recuperare la cronologia di un ottimizzatore per una tabella specifica.

```
aws list-table-optimizer-runs \
  --catalog-id 123456789012 \
  --database-name iceberg_db \
  --table-name iceberg_table \
  --type compaction
```

- L'esempio seguente mostra come recuperare l'esecuzione dell'ottimizzazione e i dettagli di configurazione per più ottimizzatori. È possibile specificare un massimo di 20 ottimizzatori.

```
aws glue batch-get-table-optimizer \
--entries '[{"catalogId":"123456789012", "databaseName":"iceberg_db",
 "tableName":"iceberg_table", "type":"compaction"}]'
```

API

- Usare l'operazione `GetTableOptimizer` per recuperare i dettagli dell'ultima esecuzione di un ottimizzatore.
- Usare l'operazione `ListTableOptimizerRuns` per recuperare la cronologia di un determinato ottimizzatore su una tabella specifica. È possibile specificare 20 ottimizzatori in una singola chiamata API.
- Utilizzate l'BatchGetTableOptimizeroperazione per recuperare i dettagli di configurazione per più ottimizzatori nel vostro account.

## Visualizzazione delle Amazon CloudWatch metriche

Dopo aver eseguito correttamente gli ottimizzatori delle tabelle, il servizio crea parametri Amazon CloudWatch sulle prestazioni del processo di ottimizzazione. Puoi andare su CloudWatch Metriche e scegliere Metriche, Tutte le metriche. Puoi filtrare le metriche in base allo spazio dei nomi specifico (ad esempio AWS Glue), al nome della tabella o al nome del database.

Per ulteriori informazioni, consultare Visualizzazione di parametri disponibili nella Guida per l'utente di Amazon CloudWatch .

Compattazione

- Numero di byte compattati
- Numero di file compattati
- Numero di DPU assegnate al processo
- Durata del processo (ore)

Conservazione degli snapshot

- Numero di file di dati eliminati
- Numero di file manifest eliminati
- Numero di elenchi Manifest eliminati
- Durata del processo (ore)

Eliminazione di file orfani

- Numero di file orfani eliminati
- Durata del processo (ore)

## Eliminazione di un ottimizzatore

È possibile eliminare un ottimizzatore e i metadati associati per la tabella utilizzando AWS CLI l'operazione AWS API.

Esegui il AWS CLI comando seguente per eliminare la cronologia di ottimizzazione per una tabella. Devi specificare l'ottimizzatore `type` insieme all'ID del catalogo, al nome del database e al nome della tabella. I valori accettabili sono: `compaction`, `retention` e `orphan_file_deletion`.

```
aws glue delete-table-optimizer \
  --catalog-id 123456789012 \
  --database-name iceberg_db \
  --table-name iceberg_table \
  --type compaction
```

Usa l'operazione `DeleteTableOptimizer` per eliminare un ottimizzatore per una tabella.

## Considerazioni e limitazioni

Questa sezione include gli aspetti da considerare quando si utilizzano ottimizzatori delle tabelle all'interno di AWS Glue Data Catalog.

Durata e correttezza

Posizioni delle tabelle S3:

Quando più AWS Glue Data Catalog tabelle condividono la stessa posizione Amazon S3 e dispongono di ottimizzatori abilitati, l'ottimizzatore per la conservazione degli snapshot o l'eliminazione di file orfani per una tabella può eliminare i file a cui fa ancora riferimento l'altra tabella. Assicurati che ogni tabella con ottimizzatori abilitati abbia una posizione Amazon S3 unica che non sia condivisa con nessun'altra tabella, incluse le tabelle in database diversi.

Scadenza del ciclo di vita S3:

Le regole di scadenza del ciclo di vita di Amazon S3 che si applicano alle posizioni di archiviazione delle tabelle Iceberg possono eliminare file manifest e di dati a cui fanno ancora riferimento gli snapshot attivi. Se il tuo bucket ha regole di scadenza del ciclo di vita, assicurati che escludano il percorso di storage della tabella Iceberg.

Problemi noti

La documentazione sugli ottimizzatori di tabelle a livello di catalogo afferma che «le tabelle senza le proprie configurazioni di ottimizzazione erediteranno lo stato disabilitato dal livello di catalogo». Esiste un problema noto per cui alcune tabelle senza la propria configurazione di ottimizzazione potrebbero non ereditare correttamente lo stato di disabilitazione dalla configurazione a livello di catalogo. Utilizza i registri di esecuzione della AWS Glue console e dell'ottimizzatore per verificare quali ottimizzatori sono attualmente abilitati e in esecuzione nel tuo account e disabilita quelli che non ti servono.

Formati supportati e limitazioni per la compattazione gestita dei dati

La compattazione dei dati supporta diversi tipi di dati e formati di compressione per la lettura e la scrittura di dati, inclusa la lettura di dati da tabelle crittografate.

Controllo della concorrenza:

Apache Iceberg supporta il controllo ottimistico della concorrenza, che consente a più scrittori di eseguire operazioni contemporaneamente. I conflitti vengono rilevati e risolti al momento del commit. Quando lavori con pipeline di streaming, configura le impostazioni di riprova appropriate tramite le proprietà delle tabelle e le impostazioni di compattazione per gestire efficacemente le scritture simultanee. Per una guida dettagliata, consulta il AWS Big Data Blog sulla gestione delle scritture simultanee nelle tabelle Iceberg.

Tentativi di compattazione:

Quando le operazioni di compattazione falliscono quattro volte consecutive, l'ottimizzazione della tabella AWS Glue del catalogo sospende automaticamente l'ottimizzatore per evitare un consumo non necessario di risorse di elaborazione. Analizzate innanzitutto i log e cercate di capire perché la compattazione fallisce ripetutamente. Per riprendere l'ottimizzazione della compattazione, puoi riattivare l'ottimizzatore tramite la console o l'API. AWS Glue

La compattazione dei dati supporta:

- Crittografia: la compattazione dei dati supporta solo la crittografia Amazon S3 (SSE-S3) e la crittografia KMS lato server (SSE-KMS).
- Strategie di compattazione: ordinamento in formato Binpack, ordinamento e ordinamento in ordine Z
- Puoi eseguire la compattazione dall'account in cui risiede il Catalogo dati quando il bucket Amazon S3 che archivia i dati sottostanti si trova in un altro account. Per eseguire questa azione, il ruolo di compattazione richiede l'accesso al bucket Amazon S3.

La compattazione dei dati attualmente non supporta:

- Compattazione su tabelle con più account: non è possibile eseguire la compattazione su tabelle con più account.
- Compattazione su tabelle con più regioni: non è possibile eseguire la compattazione su tabelle con più regioni.
- Abilitazione della compattazione sui link alle risorse

- Tabelle nella classe di storage Amazon S3 Express One Zone: non è possibile eseguire la compattazione su Amazon S3 Express One Zone Iceberg Tables.
- La strategia di compattazione con ordine Z non supporta i seguenti tipi di dati:
  - Decimale
  - TimestampWithoutZone

Considerazioni sulla conservazione degli snapshot e sugli ottimizzatori di eliminazione di file orfani

Le seguenti considerazioni si applicano alla conservazione degli snapshot e agli ottimizzatori di eliminazione di file orfani.

- I processi di conservazione degli snapshot e di eliminazione dei file orfani hanno un limite massimo di eliminazione di 1.000.000 di file per esecuzione. Quando si eliminano gli snapshot scaduti, se il numero di file idonei all'eliminazione supera 1.000.000, tutti i file rimanenti oltre tale soglia continueranno a esistere nella tabella di archiviazione come file orfani.
- Gli snapshot verranno conservati dall'ottimizzatore di conservazione degli snapshot solo quando entrambi i criteri saranno soddisfatti: il numero minimo di snapshot da conservare e il periodo di conservazione specificato.
- L'ottimizzatore di conservazione degli snapshot elimina i metadati degli snapshot scaduti da Apache Iceberg, evitando le query temporali per gli snapshot scaduti e, facoltativamente, eliminando i file di dati associati.
- L'ottimizzatore di eliminazione di file orfani elimina file di metadati e dati orfani a cui non fanno più riferimento i metadati Iceberg se la data di creazione è precedente al periodo di conservazione per l'eliminazione di file orfani dal momento dell'esecuzione dell'ottimizzatore.
- Apache Iceberg facilita il controllo delle versioni tramite rami e tag, che sono puntatori denominati a stati specifici degli snapshot. Ogni ramo e tag segue il proprio ciclo di vita indipendente, regolato da policy di conservazione definite ai rispettivi livelli. Gli AWS Glue Data Catalog ottimizzatori tengono conto di queste politiche del ciclo di vita, garantendo il rispetto delle regole di conservazione specificate. Le policy di conservazione a livello di ramo e tag hanno la precedenza sulle configurazioni dell'ottimizzatore.

  Per ulteriori informazioni, consultare Ramificazione e tagging nella documentazione di Apache Iceberg.
- Gli ottimizzatori di conservazione degli snapshot e di eliminazione dei file orfani elimineranno i file idonei alla pulizia in base ai parametri configurati. Migliorare il controllo sull'eliminazione dei file implementando le policy di controllo delle versioni di S3 e del ciclo di vita nei bucket appropriati.

Per istruzioni dettagliate sulla configurazione del controllo delle versioni e sulla creazione di regole del ciclo di vita, consultare https://docs.aws.amazon.com/AmazonS3/latest/userguide/Versioning.html.

- Per una corretta determinazione dei file orfani, assicurarsi che la posizione della tabella e gli eventuali percorsi secondari forniti non si sovrappongano o contengano dati provenienti da altre tabelle o origini dati. Se i percorsi si sovrappongono, si rischia una perdita irreversibile dei dati a causa dell'eliminazione involontaria dei file.

Eccezione di debug OversizedAllocationException

Per risolvere un'eccezione `OversizedAllocationException`:

- Ridurre la dimensione del batch del lettore vettorializzato e controllare. La dimensione predefinita del batch è 5000. Questo è controllato in `read.parquet.vectorization.batch-size`.
  - Se non funziona anche dopo molteplici varianti, disattivare la vettorizzazione. Questo è controllato in `read.parquet.vectorization.enabled`.

## Regioni supportate per gli ottimizzatori di tabelle

Le funzionalità di ottimizzazione delle tabelle (compattazione, conservazione delle istantanee ed eliminazione di file orfani) per AWS Glue Data Catalog sono disponibili nelle seguenti versioni: Regioni AWS

- Asia Pacifico (Tokyo)
- Asia Pacifico (Seul)
- Asia Pacifico (Mumbai)
- Asia Pacifico (Singapore)
- Asia Pacifico (Sydney)
- Asia Pacifico (Giacarta)
- Canada (Centrale)
- Europa (Irlanda)
- Europa (Londra)
- Europa (Francoforte)
- Europa (Stoccolma)

- Stati Uniti orientali (Virginia settentrionale)
- Stati Uniti orientali (Ohio)
- Stati Uniti occidentali (Oregon)
- Sud America (San Paolo)

## Ottimizzazione delle prestazioni delle query per le tabelle Iceberg

Apache Iceberg è un formato di tabella aperta ad alte prestazioni per enormi set di dati analitici. AWS Glue supporta il calcolo e l'aggiornamento del numero di valori distinti (NDVs) per ogni colonna nelle tabelle Iceberg. Queste statistiche possono facilitare una migliore ottimizzazione delle query, la gestione dei dati e l'efficienza delle prestazioni per gli ingegneri e gli scienziati che lavorano con set di dati su larga scala.

AWS Glue stima il numero di valori distinti in ogni colonna della tabella Iceberg e li memorizza in file Puffin su Amazon S3 associati agli snapshot delle tabelle Iceberg. Puffin è un formato di file Iceberg progettato per archiviare metadati come indici, statistiche e schizzi. L'archiviazione degli schizzi in file Puffin collegati alle istantanee garantisce la coerenza transazionale e l'aggiornamento delle statistiche NDV.

Puoi configurare l'esecuzione di un'attività di generazione di statistiche sulle colonne utilizzando la console o. AWS Glue AWS CLI Quando avvii il processo, AWS Glue avvia un job Spark in background e aggiorna i metadati della AWS Glue tabella nel Data Catalog. Puoi visualizzare le statistiche delle colonne utilizzando la AWS Glue console AWS CLI o chiamando l'GetColumnStatisticsForTableoperazione API.

> **Note**
>
> Se utilizzi AWS Lake Formation le autorizzazioni per controllare l'accesso alla tabella, il ruolo assunto dall'attività di statistica delle colonne richiede l'accesso completo alla tabella per generare statistiche.

Argomenti

- Prerequisiti per la generazione delle statistiche delle colonne
- Generazione delle statistiche delle colonne delle tabelle Iceberg
- Consulta anche

## Prerequisiti per la generazione delle statistiche delle colonne

Per generare o aggiornare le statistiche delle colonne per le tabelle Iceberg, l'attività di generazione delle statistiche assume un ruolo AWS Identity and Access Management (IAM) per conto dell'utente. In base alle autorizzazioni concesse al ruolo, l'attività di generazione delle statistiche delle colonne può leggere i dati dal datastore di Amazon S3.

Quando si configura l'attività di generazione delle statistiche sulle colonne, AWS Glue consente di creare un ruolo che include la politica `AWSGlueServiceRole` AWS gestita più la politica in linea richiesta per l'origine dati specificata.

Se si specifica un ruolo esistente per la generazione di statistiche delle colonne, bisogna assicurarsi che includa la policy `AWSGlueServiceRole` o equivalente (o una versione ridotta di questa policy), oltre alle policy inline richieste.

Per ulteriori informazioni sulle autorizzazioni richieste, consulta Prerequisiti per la generazione delle statistiche delle colonne.

## Generazione delle statistiche delle colonne delle tabelle Iceberg

Segui questi passaggi per configurare una pianificazione per la generazione di statistiche nel Data Catalog utilizzando la AWS Glue console AWS CLI o o esegui l'StartColumnStatisticsTaskRunoperazione.

Per generare statistiche sulle colonne

1. Accedi alla AWS Glue console all'indirizzo https://console.aws.amazon.com/glue/.
2. Scegli Tabelle nel Catalogo dati.
3. Scegli una tabella Iceberg dall'elenco.
4. Scegli Statistiche colonna, Genera su richiesta, nel menu Azioni.

   Puoi anche scegliere il pulsante Genera statistiche nella scheda Statistiche di colonna nella sezione inferiore della pagina Tabelle.
5. Nella pagina Genera statistiche, fornisci i dettagli sulla generazione delle statistiche. Segui i passaggi 6-11 della sezione Generazione di statistiche a colonne in base a una pianificazione per configurare una pianificazione per la generazione di statistiche per le tabelle Iceberg.

   Puoi anche scegliere di generare statistiche sulle colonne su richiesta seguendo le istruzioni contenute in Generazione di statistiche delle colonne on demand

> **Note**
>
> L'opzione di campionamento non è disponibile per le tabelle Iceberg.

AWS Glue calcola il numero di valori distinti per ogni colonna della tabella Iceberg in un nuovo file Puffin salvato nell'ID snapshot specificato nella tua posizione Amazon S3.

### Consulta anche

- Visualizzazione delle statistiche delle colonne
- Visualizzazione dell'attività relativa alle statistiche delle colonne
- Interruzione dell'esecuzione relativa alle statistiche delle colonne
- Eliminazione delle statistiche delle colonne

## Gestione del Catalogo dati

AWS Glue Data Catalog È un repository di metadati centrale che archivia i metadati strutturali e operativi per i set di dati Amazon S3. La gestione efficace del Catalogo dati è fondamentale per mantenere la qualità, le prestazioni, la sicurezza e la governance dei dati.

Comprendendo e applicando queste pratiche di gestione del Catalogo dati, puoi garantire che i metadati rimangano accurati, performanti, sicuri e ben governati man mano che il panorama dei dati si evolve.

Questa sezione tratta i seguenti aspetti della gestione del Catalogo dati:

- Aggiornamento dello schema della tabella e delle partizioni  Man mano che i dati evolvono, potrebbe essere necessario aggiornare lo schema della tabella o la struttura delle partizioni definiti nel Catalogo dati. Per ulteriori informazioni su come effettuare questi aggiornamenti a livello di codice utilizzando l'ETL, consulta. AWS Glue Aggiornamento dello schema e aggiunta di nuove partizioni nel catalogo dati usando processi ETL AWS Glue
- Gestione delle statistiche sulle colonne: le statistiche accurate sulle colonne aiutano a ottimizzare i piani di query e a migliorare le prestazioni. Per ulteriori informazioni su come generare, aggiornare e gestire le statistiche sulle colonne, vedereOttimizzazione delle prestazioni delle query con le statistiche delle colonne.

- Crittografia del catalogo dati Per proteggere i metadati sensibili, puoi crittografare il catalogo dati utilizzando (). AWS Key Management Service AWS KMS In questa sezione viene descritto come abilitare e gestire la crittografia per il catalogo dati.
- Proteggere il Catalogo dati con AWS Lake Formation  Lake Formation offre un approccio completo alla sicurezza dei data lake e al controllo degli accessi. Puoi utilizzare Lake Formation per proteggere e gestire l'accesso al tuo Catalogo dati e ai dati sottostanti.

Argomenti


- Aggiornamento dello schema e aggiunta di nuove partizioni nel catalogo dati usando processi ETL AWS Glue
- Ottimizzazione delle prestazioni delle query con le statistiche delle colonne
- Crittografia del catalogo dati
- Protezione del catalogo dati con Lake Formation
- Lavorare con AWS Glue Data Catalog le viste in AWS Glue


# Aggiornamento dello schema e aggiunta di nuove partizioni nel catalogo dati usando processi ETL AWS Glue

Il processo di estrazione, trasformazione e caricamento (ETL) potrebbe creare nuove partizioni di tabella nell'archivio dati di destinazione. Lo schema del set di dati può evolversi e divergere dallo schema del catalogo dati AWS Glue nel corso del tempo. AWS Glue I processi ETL ora forniscono diverse funzionalità che puoi utilizzare all'interno dello script ETL per aggiornare lo schema e le partizioni nel catalogo dati. Queste caratteristiche ti consentono di vedere i risultati del processo ETL nel catalogo dati, senza dover eseguire nuovamente il crawler.

## Nuove partizioni

Se desideri visualizzare le nuove partizioni in AWS Glue Data Catalog, puoi effettuare una delle seguenti operazioni:

- Al termine del processo, esegui nuovamente il crawler e visualizza le nuove partizioni sulla console al termine del crawler.
- Al termine del processo, visualizza immediatamente le nuove partizioni sulla console, senza dover eseguire nuovamente il crawler. Puoi abilitare questa caratteristica aggiungendo alcune righe di codice allo script ETL, come mostrato negli esempi seguenti. Il codice utilizza l'argomento

`enableUpdateCatalog` per indicare che il catalogo dati deve essere aggiornato durante l'esecuzione del processo quando vengono create nuove partizioni.

### Metodo 1

Passare `enableUpdateCatalog` e `partitionKeys` in un argomento opzioni.

Python

```
additionalOptions = {"enableUpdateCatalog": True}
additionalOptions["partitionKeys"] = ["region", "year", "month", "day"]


sink = glueContext.write_dynamic_frame_from_catalog(frame=last_transform,
 database=<target_db_name>,

 table_name=<target_table_name>, transformation_ctx="write_sink",

 additional_options=additionalOptions)
```

Scala

```
val options = JsonOptions(Map(
    "path" -> <S3_output_path>,
    "partitionKeys" -> Seq("region", "year", "month", "day"),
    "enableUpdateCatalog" -> true))
val sink = glueContext.getCatalogSink(
    database = <target_db_name>,
    tableName = <target_table_name>,
    additionalOptions = options)sink.writeDynamicFrame(df)
```

### Metodo 2

Passare `enableUpdateCatalog` e `partitionKeys` in `getSink()` e chiamare `setCatalogInfo()` sull'oggetto `DataSink`.

Python

```
sink = glueContext.getSink(
    connection_type="s3",
    path="<S3_output_path>",
    enableUpdateCatalog=True,
    partitionKeys=["region", "year", "month", "day"])
```

```
sink.setFormat("json")
sink.setCatalogInfo(catalogDatabase=<target_db_name>,
 catalogTableName=<target_table_name>)
sink.writeFrame(last_transform)
```

Scala

```
val options = JsonOptions(
   Map("path" -> <S3_output_path>,
       "partitionKeys" -> Seq("region", "year", "month", "day"),
       "enableUpdateCatalog" -> true))
val sink = glueContext.getSink("s3", options).withFormat("json")
sink.setCatalogInfo(<target_db_name>, <target_table_name>)
sink.writeDynamicFrame(df)
```

Ora, puoi creare nuove tabelle di catalogo, aggiornare le tabelle esistenti con schema modificato e aggiungere nuove partizioni di tabella nel catalogo dati utilizzando direttamente un processo ETL di AWS Glue, senza la necessità di eseguire nuovamente i crawler.

## Aggiornamento dello schema della tabella

Se desideri sovrascrivere lo schema della tabella del catalogo dati, puoi eseguire una delle seguenti operazioni:

- Al termine del processo, esegui nuovamente il crawler e assicurati che il crawler sia configurato per aggiornare anche la definizione della tabella. Visualizza le nuove partizioni sulla console insieme agli eventuali aggiornamenti dello schema, al termine del crawler. Per maggiori informazioni, consulta Configurazione di un crawler utilizzando l'API.
- Al termine del processo, visualizza immediatamente lo schema modificato sulla console, senza dover eseguire nuovamente il crawler. Puoi abilitare questa caratteristica aggiungendo alcune righe di codice allo script ETL, come mostrato negli esempi seguenti. Il codice utilizza `enableUpdateCatalog` impostato su true, e anche `updateBehavior` impostato su `UPDATE_IN_DATABASE`, il che indica di sovrascrivere lo schema e aggiungere nuove partizioni nel catalogo dati durante l'esecuzione del processo.

Python

```
additionalOptions = {
    "enableUpdateCatalog": True,
```

```
    "updateBehavior": "UPDATE_IN_DATABASE"}
additionalOptions["partitionKeys"] = ["partition_key0", "partition_key1"]

sink = glueContext.write_dynamic_frame_from_catalog(frame=last_transform,
 database=<dst_db_name>,
    table_name=<dst_tbl_name>, transformation_ctx="write_sink",
    additional_options=additionalOptions)
job.commit()
```

Scala

```
val options = JsonOptions(Map(
    "path" -> outputPath,
    "partitionKeys" -> Seq("partition_0", "partition_1"),
    "enableUpdateCatalog" -> true))
val sink = glueContext.getCatalogSink(database = nameSpace, tableName = tableName,
 additionalOptions = options)
sink.writeDynamicFrame(df)
```

Puoi inoltre impostare il valore `updateBehavior` su `LOG` se desideri impedire che lo schema di tabella venga sovrascritto, ma se desidera comunque aggiungere le nuove partizioni. Il valore predefinito di `updateBehavior` è `UPDATE_IN_DATABASE`, quindi se non lo definisci esplicitamente, lo schema della tabella verrà sovrascritto.

Se `enableUpdateCatalog` non è impostato su true, indipendentemente da qualsiasi opzione selezionata per `updateBehavior`, il processo ETL non aggiornerà la tabella nel catalogo dati.

## Creazione di nuove tabelle

Puoi inoltre utilizzare le stesse opzioni per creare una nuova tabella nel catalogo dati. Puoi specificare il database e il nome della nuova tabella utilizzando `setCatalogInfo`.

Python

```
sink = glueContext.getSink(connection_type="s3", path="s3://path/to/data",
    enableUpdateCatalog=True, updateBehavior="UPDATE_IN_DATABASE",
    partitionKeys=["partition_key0", "partition_key1"])
sink.setFormat("<format>")
sink.setCatalogInfo(catalogDatabase=<dst_db_name>, catalogTableName=<dst_tbl_name>)
sink.writeFrame(last_transform)
```

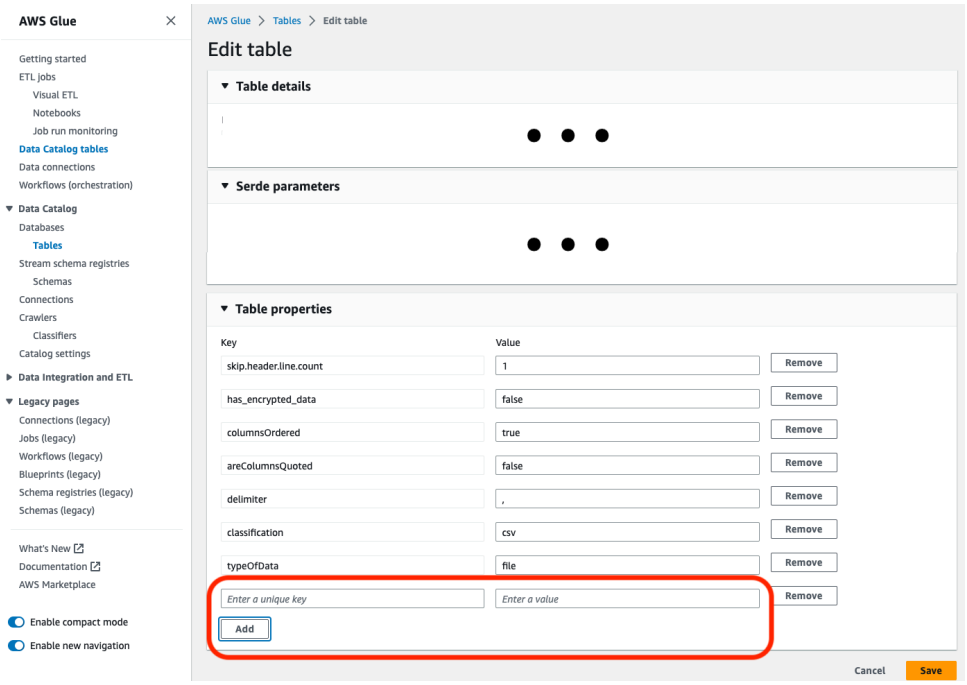
Scala

```
val options = JsonOptions(Map(
    "path" -> outputPath,
    "partitionKeys" -> Seq("<partition_1>", "<partition_2>"),
    "enableUpdateCatalog" -> true,
    "updateBehavior" -> "UPDATE_IN_DATABASE"))
val sink = glueContext.getSink(connectionType = "s3", connectionOptions =
 options).withFormat("<format>")
sink.setCatalogInfo(catalogDatabase = "<dst_db_name>", catalogTableName =
 "<dst_tbl_name>")
sink.writeDynamicFrame(df)
```

## Restrizioni

Prestare attenzione alle seguenti restrizioni:

- Sono supportate solo le destinazioni di Amazon Simple Storage Service (Amazon S3).
- La funzionalità `enableUpdateCatalog` non è supportata per le tabelle governate.
- Sono supportati solo i seguenti formati: `json`, `csv`, `avro`, e `parquet`.
- Per creare o aggiornare tabelle con la `parquet` classificazione, è necessario utilizzare il parquet writer AWS Glue ottimizzato per. DynamicFrames È possibile farlo in uno dei modi seguenti:
  - Se stai aggiornando una tabella esistente nel catalogo con la classificazione `parquet`, la proprietà della tabella `"useGlueParquetWriter"` deve essere impostata su `true` prima di aggiornarla. È possibile impostare questa proprietà tramite AWS Glue APIs /SDK, tramite la console o tramite un'istruzione Athena DDL.




Una volta impostata la proprietà della tabella del catalogo, puoi utilizzare il seguente frammento di codice per aggiornare la tabella del catalogo con i nuovi dati:

```
glueContext.write_dynamic_frame.from_catalog(
    frame=frameToWrite,
    database="dbName",
    table_name="tableName",
    additional_options={
        "enableUpdateCatalog": True,
        "updateBehavior": "UPDATE_IN_DATABASE"
    }
)
```

- Se la tabella non esiste ancora nel catalogo, puoi utilizzare il metodo `getSink()` nello script con `connection_type="s3"` per aggiungere la tabella e le sue partizioni al catalogo, oltre a scrivere i dati su Amazon S3. Fornisci i valori appropriati di `partitionKeys` e `compression` per il tuo flusso di lavoro.

```
s3sink = glueContext.getSink(
    path="s3://bucket/folder/",
    connection_type="s3",
    updateBehavior="UPDATE_IN_DATABASE",
    partitionKeys=[],
    compression="snappy",
```

```
    enableUpdateCatalog=True
)

s3sink.setCatalogInfo(
    catalogDatabase="dbName", catalogTableName="tableName"
)

s3sink.setFormat("parquet", useGlueParquetWriter=True)
s3sink.writeFrame(frameToWrite)
```

  - Il valore di `glueparquet` formato è un metodo obsoleto per abilitare il AWS Glue parquet writer.
- Quando `updateBehavior` è impostato su `LOG`, nuove partizioni verranno aggiunte solo se lo schema `DynamicFrame` è equivalente o contiene un sottoinsieme delle colonne definite nello schema della tabella del catalogo dati.
- Gli aggiornamenti dello schema non sono supportati per le tabelle non partizionate (che non utilizzano l'opzione "partitionKeys").
- Le PartitionKeys devono essere equivalenti, e nello stesso ordine, tra il parametro passato nello script ETL e le PartitionKeys nello schema della tabella del catalogo dati.
- Questa funzionalità attualmente non supporta ancora le updating/creating tabelle in cui sono annidati gli schemi di aggiornamento (ad esempio, gli array all'interno delle strutture).

Per ulteriori informazioni, consulta the section called "AWS Glue per Spark".

## Lavorare con le connessioni MongoDB nei processi ETL

Puoi creare una connessione per MongoDB e usare quella connessione nel tuo processo AWS Glue. Per ulteriori informazioni, consultare the section called "Connessioni MongoDB" la guida alla AWS Glue programmazione. La connessione`url`, `username` e `password` sono archiviati nella connessione MongoDB. Altre opzioni possono essere specificate nello script del processo ETL utilizzando il parametro `additionalOptions` di `glueContext.getCatalogSource`. Le altre opzioni possono includere:

- `database`: (Obbligatorio) Il database MongoDB da cui leggere.
- `collection`: (Obbligatorio) La raccolta MongoDB da cui leggere.

Posizionando le informazioni di `database` e `collection` all'interno dello script del processo ETL, puoi utilizzare la stessa connessione in più processi.

1. Crea una AWS Glue Data Catalog connessione per l'origine dati MongoDB. Consulta "connectionType": "mongodb" per una descrizione dei parametri di connessione. È possibile creare la connessione utilizzando la console APIs o la CLI.
2. Crea un database in AWS Glue Data Catalog per memorizzare le definizioni delle tabelle per i tuoi dati MongoDB. Per ulteriori informazioni, consulta Creazione di database.
3. Crea un crawler che esegue il crawling dati in MongoDB utilizzando le informazioni nella connessione per connettersi a MongoDB. Il crawler crea le tabelle AWS Glue Data Catalog che descrivono le tabelle del database MongoDB che usi nel tuo job. Per ulteriori informazioni, consulta Utilizzo dei crawler per compilare il Catalogo dati.
4. Crea un processo con uno script personalizzato. È possibile creare il lavoro utilizzando la console APIs o la CLI. Per ulteriori informazioni, consulta Aggiunta di processi in AWS Glue.
5. Scegli le destinazioni dati per il tuo processo. Le tabelle che rappresentano la destinazione dei dati possono essere definite nel catalogo dati oppure il processo può creare le tabelle di destinazione quando viene eseguito. Puoi scegliere una posizione di destinazione al momento della creazione del processo. Se la destinazione richiede una connessione, anche la connessione ha un riferimento nel tuo processo. Se il processo richiede più destinazioni dati, in seguito potrai aggiungerle modificando lo script.
6. Personalizza l'ambiente di elaborazione del processo fornendo gli argomenti per il tuo processo e lo script generato.

   Di seguito è illustrato un esempio di creazione di un `DynamicFrame` dal database MongoDB in base alla struttura della tabella definita nel catalogo dati. Il codice utilizza `additionalOptions` per fornire le informazioni aggiuntive sull'origine dati:

   Scala

   ```
   val resultFrame: DynamicFrame = glueContext.getCatalogSource(
           database = catalogDB,
           tableName = catalogTable,
           additionalOptions = JsonOptions(Map("database" -> DATABASE_NAME,
                   "collection" -> COLLECTION_NAME))
         ).getDynamicFrame()
   ```

   Python

   ```
   glue_context.create_dynamic_frame_from_catalog(
           database = catalogDB,
           table_name = catalogTable,
   ```

```
        additional_options = {"database":"database_name",
            "collection":"collection_name"})
```

7. Esegui il processo, on demand o tramite un trigger.

## Ottimizzazione delle prestazioni delle query con le statistiche delle colonne

Puoi calcolare statistiche a livello di colonna per AWS Glue Data Catalog tabelle in formati di dati come Parquet, ORC, JSON, ION, CSV e XML senza configurare pipeline di dati aggiuntive. Le statistiche delle colonne consentono di comprendere i profili di dati ottenendo informazioni dettagliate sui valori all'interno di una colonna.

Data Catalog supporta la generazione di statistiche per valori di colonna come valore minimo, valore massimo, valori nulli totali, valori distinti totali, lunghezza media dei valori e occorrenze totali di valori reali. AWS servizi di analisi come Amazon Redshift Amazon Athena possono utilizzare queste statistiche a colonne per generare piani di esecuzione delle query e scegliere il piano ottimale che migliori le prestazioni delle query.

Sono possibili tre scenari per generare le statistiche delle colonne:

Automatico

AWS Glue supporta la generazione automatica di statistiche sulle colonne a livello di catalogo in modo da poter generare automaticamente statistiche per nuove tabelle in. AWS Glue Data Catalog

Pianificato

AWS Glue supporta la generazione di statistiche sulle colonne di pianificazione in modo che possa essere eseguita automaticamente in base a una pianificazione ricorrente.

Con il calcolo pianificato delle statistiche, l'attività di statistica delle colonne aggiorna le statistiche complessive a livello di tabella, ad esempio min, max e avg, con le nuove statistiche, fornendo ai motori di query statistiche accurate e per ottimizzare l'esecuzione delle query. up-to-date

Su richiesta

Utilizzare questa opzione per generare statistiche delle colonne on demand ogni volta che è necessario. Questo è utile per analisi ad hoc o quando le statistiche devono essere calcolate immediatamente.

È possibile configurare l'esecuzione di attività di generazione di statistiche sulle colonne utilizzando le operazioni AWS Glue della console e AWS Glue dell'API. AWS CLI Quando avvii il processo, AWS Glue avvia un job Spark in background e aggiorna i metadati della AWS Glue tabella nel Data Catalog. Puoi visualizzare le statistiche delle colonne utilizzando la AWS Glue console AWS CLI o chiamando l'GetColumnStatisticsForTableoperazione API.

> **Note**
>
> Se utilizzi le autorizzazioni di Lake Formation per controllare l'accesso alla tabella, il ruolo assunto dall'attività di statistica delle colonne richiede l'accesso completo alla tabella per generare statistiche.

Argomenti


- Prerequisiti per la generazione delle statistiche delle colonne
- Generazione automatica delle statistiche delle colonne
- Generazione di statistiche a colonne in base a una pianificazione
- Generazione di statistiche delle colonne on demand
- Visualizzazione delle statistiche delle colonne
- Visualizzazione dell'attività relativa alle statistiche delle colonne
- Interruzione dell'esecuzione relativa alle statistiche delle colonne
- Eliminazione delle statistiche delle colonne
- Considerazioni e limitazioni


## Prerequisiti per la generazione delle statistiche delle colonne

Per generare o aggiornare le statistiche delle colonne, l'attività di generazione delle statistiche assume un ruolo (IAM) AWS Identity and Access Management . In base alle autorizzazioni concesse al ruolo, l'attività di generazione delle statistiche delle colonne può leggere i dati dal datastore di Amazon S3.

Quando si configura l'attività di generazione delle statistiche sulle colonne, AWS Glue consente di creare un ruolo che include la politica `AWSGlueServiceRole` AWS gestita più la politica in linea richiesta per l'origine dati specificata.

Se si specifica un ruolo esistente per la generazione di statistiche delle colonne, bisogna assicurarsi che includa la policy `AWSGlueServiceRole` o equivalente (o una versione ridotta di questa policy), oltre alle policy inline richieste. Per creare un nuovo ruolo IAM, attenersi alla seguente procedura:

> **Note**
>
> Per generare statistiche per le tabelle gestite da Lake Formation, il ruolo IAM utilizzato per generare le statistiche richiede l'accesso completo alla tabella.

Quando si configura l'attività di generazione delle statistiche sulle colonne, AWS Glue consente di creare un ruolo che include la politica `AWSGlueServiceRole` AWS gestita più la politica in linea richiesta per l'origine dati specificata. È inoltre possibile creare un ruolo, allegare le autorizzazioni elencate nella policy riportata di seguito e aggiungere il ruolo all'attività di generazione delle statistiche delle colonne.

Per creare un ruolo IAM per la generazione delle statistiche delle colonne

1. Per creare un ruolo IAM, consulta l'argomento relativo alla creazione di ruoli IAM per AWS Glue.
2. Per aggiornare un ruolo esistente, nella console IAM, vai al ruolo IAM utilizzato dal processo di generazione delle statistiche delle colonne.
3. Nella sezione Autorizzazioni, scegli Collega policy. Nella finestra del browser appena aperta, scegli policy `AWSGlueServiceRole` AWS gestita.
4. È necessario includere anche le autorizzazioni di lettura dei dati dalla posizione dei dati Amazon S3.

   Nella sezione Autorizzazioni, scegli Aggiungi policy bucket. Nella finestra del browser appena aperta, crea una nuova policy da utilizzare con il tuo ruolo.
5. Nella pagina Crea policy seleziona la scheda JSON. Copia il codice seguente `JSON` nel campo dell'editor di policy.

   > **Note**
   >
   > Nelle seguenti politiche, sostituisci l'ID dell'account con un nome valido Account AWS, la regione della tabella e `bucket-name` il nome del bucket Amazon S3. `region`

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "S3BucketAccess",
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket",
                "s3:GetObject"
            ],
            "Resource": [
             "arn:aws:s3:::amzn-s3-demo-bucket/*",
       "arn:aws:s3:::amzn-s3-demo-bucket"
            ]
        }
     ]
}
```

6. (Facoltativo) Se utilizzi le autorizzazioni di Lake Formation per fornire l'accesso ai tuoi dati, il ruolo IAM richiede le autorizzazioni `lakeformation:GetDataAccess`.

   JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "LakeFormationDataAccess",
      "Effect": "Allow",
      "Action": "lakeformation:GetDataAccess",
      "Resource": [
        "*"
      ]
    }
  ]
}
```

Se la posizione dei dati di Amazon S3 è registrata con Lake Formation e il ruolo IAM assunto dall'attività di generazione delle statistiche delle colonne non dispone delle autorizzazioni di gruppo `IAM_ALLOWED_PRINCIPALS` concesse sulla tabella, il ruolo richiede le autorizzazioni `ALTER` e `DESCRIBE` di Lake Formation sulla tabella. Il ruolo utilizzato per la registrazione del bucket Amazon S3 richiede le autorizzazioni `INSERT` e `DELETE` di Lake Formation sulla tabella.

Se la posizione dei dati di Amazon S3 non è registrata con Lake Formation e il ruolo IAM non dispone delle autorizzazioni di gruppo `IAM_ALLOWED_PRINCIPALS` concesse sulla tabella, il ruolo richiede le autorizzazioni `ALTER`, `DESCRIBE`, `INSERT` e `DELETE` di Lake Formation sulla tabella.

7. Se è stata abilitata l'opzione `Automatic statistics generation` a livello di catalogo, il ruolo IAM deve avere l'autorizzazione `glue:UpdateCatalog` o l'autorizzazione `ALTER CATALOG` di Lake Formation sul Catalogo dati predefinito. Puoi utilizzare l'operazione `GetCatalog` per verificare le proprietà del catalogo.
8. (Facoltativo) L'attività di generazione delle statistiche delle colonne che scrive Amazon CloudWatch Logs crittografati necessita delle autorizzazioni seguenti nella policy della chiave.

   JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "CWLogsKmsPermissions",
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:AssociateKmsKey"
      ],
      "Resource": [
        "arn:aws:logs:us-east-1:111122223333:log-group:/aws-glue:*"
      ]
    },
    {
      "Sid": "KmsPermissions",
      "Effect": "Allow",
```

```
      "Action": [
        "kms:GenerateDataKey",
        "kms:Decrypt",
        "kms:Encrypt"
      ],
      "Resource": [
        "arn:aws:kms:us-east-1:111122223333:key/arn of key used for ETL
 cloudwatch encryption"
      ],
      "Condition": {
        "StringEquals": {
          "kms:ViaService": [
            "glue.us-east-1.amazonaws.com"
          ]
        }
      }
    }
  ]
}
```

9. Il ruolo utilizzato per eseguire le statistiche delle colonne deve disporre dell'autorizzazione `iam:PassRole` relativa al ruolo.

   JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": [
        "arn:aws:iam::111122223333:role/columnstats-role-name"
      ]
    }
  ]
}
```

10. Quando crei un ruolo IAM per la generazione delle statistiche delle colonne, tale ruolo deve disporre anche della policy di attendibilità seguente che consente al servizio di assumere il ruolo.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "TrustPolicy",
      "Effect": "Allow",
      "Principal": {
        "Service": "glue.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

## Generazione automatica delle statistiche delle colonne

La generazione automatica delle statistiche delle colonne consente di pianificare e calcolare automaticamente le statistiche su nuove tabelle nel AWS Glue Data Catalog. Quando abiliti la generazione automatica di statistiche, il Catalogo dati rileva nuove tabelle con formati di dati specifici come Parquet, JSON, CSV, XML, ORC, ION e Apache Iceberg, insieme ai rispettivi percorsi bucket individuali. Con una configurazione del catalogo una tantum, il catalogo dati genera statistiche per queste tabelle.

Gli amministratori di data Lake possono configurare la generazione delle statistiche selezionando il catalogo predefinito nella console Lake Formation e abilitando le statistiche delle tabelle utilizzando l'opzione `Optimization configuration`. Quando crei nuove tabelle o aggiorni tabelle esistenti nel Data Catalog, il Data Catalog raccoglie il numero di valori distinti (NDVs) per le tabelle Apache Iceberg e statistiche aggiuntive come il numero di null, la lunghezza massima, minima e media per altri formati di file supportati su base settimanale.

Se hai configurato la generazione delle statistiche a livello di tabella o se hai precedentemente eliminato le impostazioni di generazione delle statistiche per una tabella, tali impostazioni specifiche hanno la precedenza sulle impostazioni predefinite del catalogo per la generazione automatica delle statistiche delle colonne.

L'attività di generazione automatica delle statistiche analizza il 50% dei record nelle tabelle per calcolare le statistiche. La generazione automatica delle statistiche delle colonne garantisce che il Catalogo dati mantenga metriche settimanali che possono essere utilizzate da motori di query come Amazon Athena e Amazon Redshift Spectrum per migliorare le prestazioni delle query e favorire potenziali risparmi sui costi. Consente di pianificare la generazione di statistiche utilizzando AWS Glue APIs o la console, fornendo un processo automatizzato senza intervento manuale.

Argomenti


- Abilitazione della generazione automatica di statistiche a livello di catalogo
- Visualizzazione delle impostazioni automatiche a livello di tabella
- Disattivazione della generazione delle statistiche delle colonne a livello di catalogo


## Abilitazione della generazione automatica di statistiche a livello di catalogo

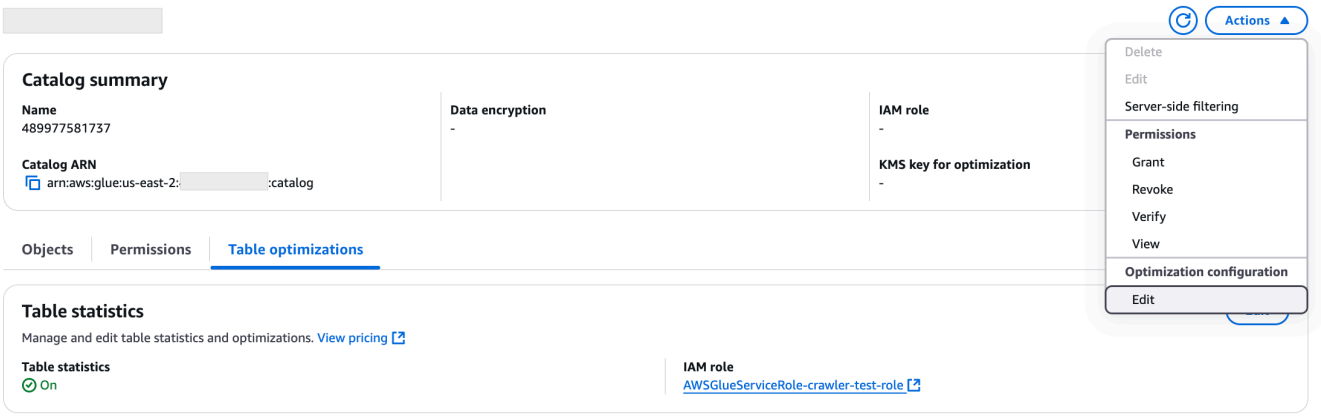
Puoi abilitare la generazione automatica di statistiche sulle colonne per tutte le nuove tabelle Apache Iceberg e le tabelle in formati di tabella non OTF (Parquet, JSON, CSV, XML, ORC, ION) nel Catalogo dati. Dopo aver creato la tabella, puoi anche aggiornare in modo esplicito le impostazioni delle statistiche delle colonne manualmente.

Per aggiornare le impostazioni del Data Catalog per abilitare il livello di catalogo, il ruolo IAM utilizzato deve disporre dell'`glue:UpdateCatalog`autorizzazione o dell' AWS Lake Formation `ALTER CATALOG`autorizzazione sul catalogo principale. È possibile utilizzare `GetCatalog` l'API per verificare le proprietà del catalogo.

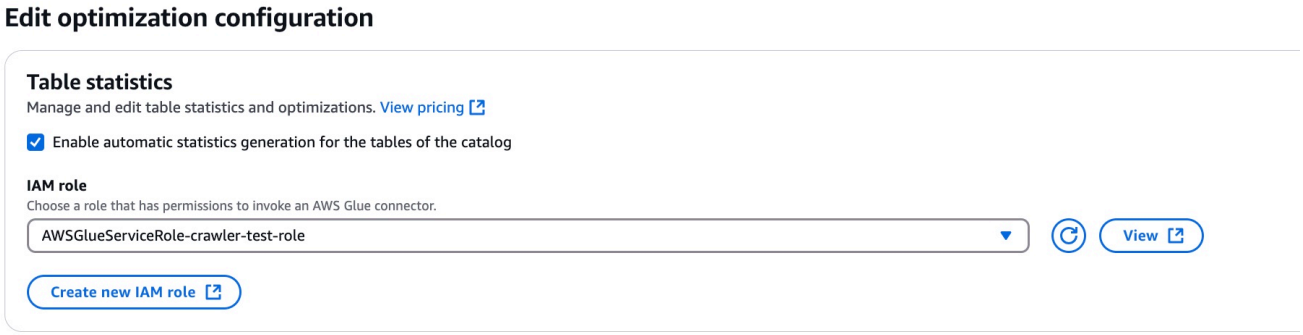
Console di gestione AWS

Abilitazione della generazione automatica delle statistiche delle colonne a livello di account

1. Apri la console Lake Formation all'indirizzo https://console.aws.amazon.com/lakeformation/.
2. Nella barra di navigazione a sinistra, scegli Cataloghi.
3. Nella pagina di Riepilogo del catalogo, scegli Modifica in Configurazione dell'ottimizzazione.




4. Nella pagina di Configurazione dell'ottimizzazione della tabella, scegli l'opzione Abilita la generazione automatica delle statistiche per le tabelle del catalogo.




5. Scegli un ruolo IAM esistente o creane uno nuovo con le autorizzazioni necessarie per eseguire l'attività relativa alle statistiche relative alle colonne.
6. Seleziona Invia.

## AWS CLI

Puoi anche abilitare la raccolta di statistiche a livello di catalogo tramite AWS CLI. Per configurare la raccolta di statistiche a livello di tabella utilizzando AWS CLI, esegui il seguente comando:

```
aws glue update-catalog --cli-input-json '{
    "name": "123456789012",
    "catalogInput": {
        "description": "Updating root catalog with role arn",
        "catalogProperties": {
            "customProperties": {
                "ColumnStatistics.RoleArn": "arn:aws:iam::"123456789012":role/
service-role/AWSGlueServiceRole",
```

```
                "ColumnStatistics.Enabled": "true"
            }
        }
    }
}'
```

Il comando precedente richiama AWS Glue l'`UpdateCatalog`operazione, che prevede una `CatalogProperties` struttura con le seguenti coppie chiave-valore per la generazione di statistiche a livello di catalogo:

- ColumnStatistics. RoleArn — ARN del ruolo IAM da utilizzare per tutte le attività attivate per la generazione di statistiche a livello di catalogo
- ColumnStatistics.Enabled: valore booleano che indica se le impostazioni a livello di catalogo sono abilitate o disabilitate

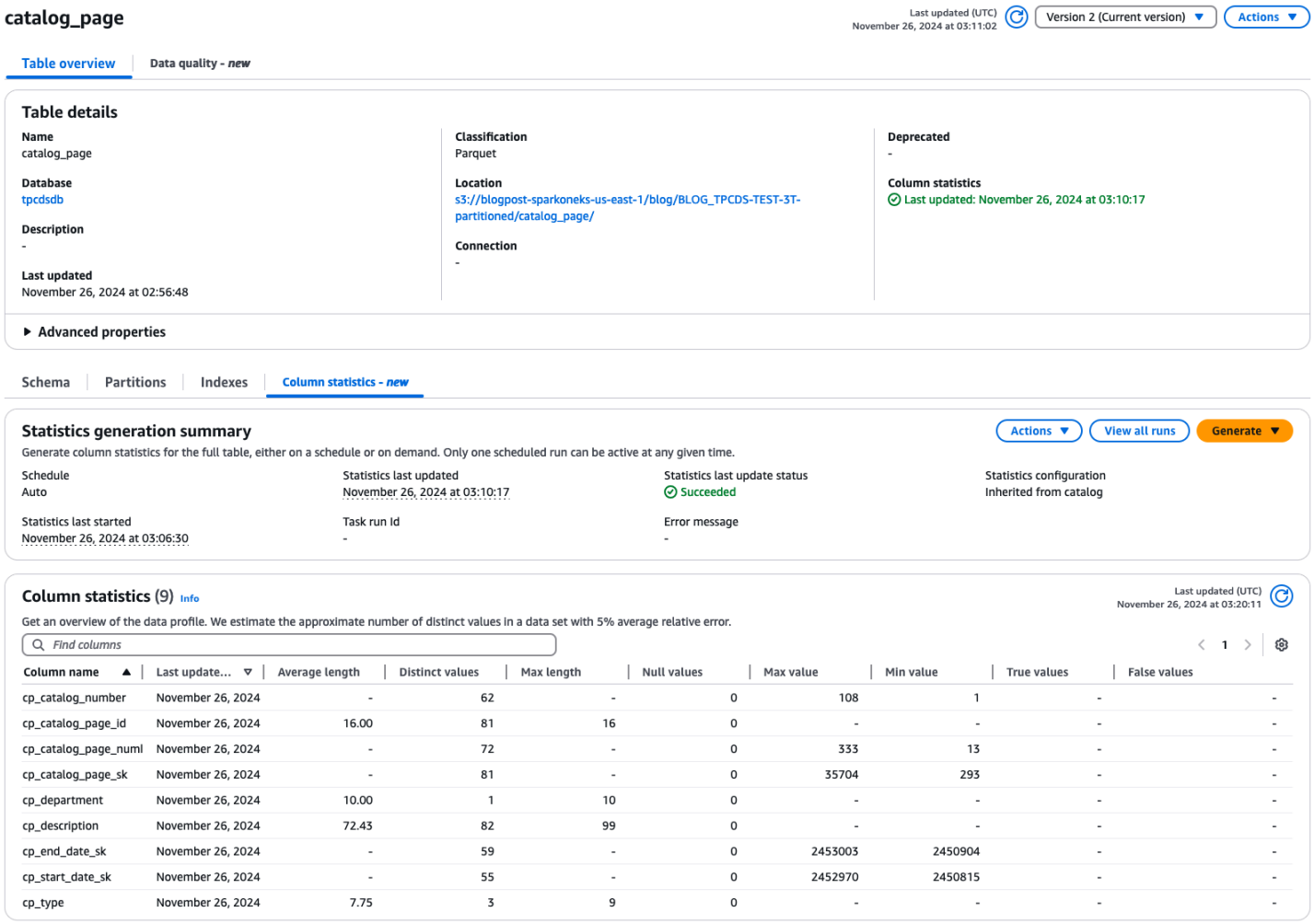
Visualizzazione delle impostazioni automatiche a livello di tabella

Quando la raccolta di statistiche a livello di catalogo è abilitata, ogni volta che una tabella Apache Hive o Apache Iceberg viene creata o aggiornata tramite o tramite SDK `CreateTable` o `UpdateTable` APIs Console di gestione AWS, viene creata un'impostazione a livello di tabella equivalente per quella tabella. Crawler di AWS Glue

Le tabelle con la generazione automatica delle statistiche abilitata devono seguire una delle seguenti proprietà:

- Usa un comando `InputSerdeLibrary` che inizia con org.apache.hadoop e `TableType` è uguale a `EXTERNAL_TABLE`
- Usa un comando `InputSerdeLibrary` che inizia con `com.amazon.ion` e `TableType` è uguale a `EXTERNAL_TABLE`
- Contiene table_type: “ICEBERG” nella sua struttura dei parametri.

Dopo aver creato o aggiornato una tabella, puoi verificare i dettagli della tabella per confermare la generazione delle statistiche. `Statistics generation summary` mostra la proprietà `Schedule` impostata come `AUTO` e il valore di `Statistics configuration` è `Inherited from catalog`. Qualsiasi configurazione della tabella con la seguente impostazione verrebbe attivata automaticamente da Glue internamente.




Disattivazione della generazione delle statistiche delle colonne a livello di catalogo

È possibile disabilitare la generazione automatica di statistiche sulle colonne per nuove tabelle utilizzando la AWS Lake Formation console, l'API o l'API. `glue:UpdateCatalogSettings` `glue:DeleteColumnStatisticsTaskSettings`

Disabilitazione della generazione automatica delle statistiche delle colonne a livello di account

1. Apri la console Lake Formation all'indirizzo https://console.aws.amazon.com/lakeformation/.
2. Nella barra di navigazione a sinistra, scegli Cataloghi.
3. Nella pagina Riepilogo del catalogo, scegli Modifica in Configurazione dell'ottimizzazione.
4. Nella pagina Configurazione dell'ottimizzazione della tabella, deseleziona l'opzione Abilita la generazione automatica delle statistiche per le tabelle del catalogo.
5. Seleziona Invia.

## Generazione di statistiche a colonne in base a una pianificazione

Segui questi passaggi per configurare una pianificazione per la generazione di statistiche sulle colonne AWS Glue Data Catalog utilizzando la AWS Glue console AWS CLI, l'o l'CreateColumnStatisticsTaskSettingsoperazione.

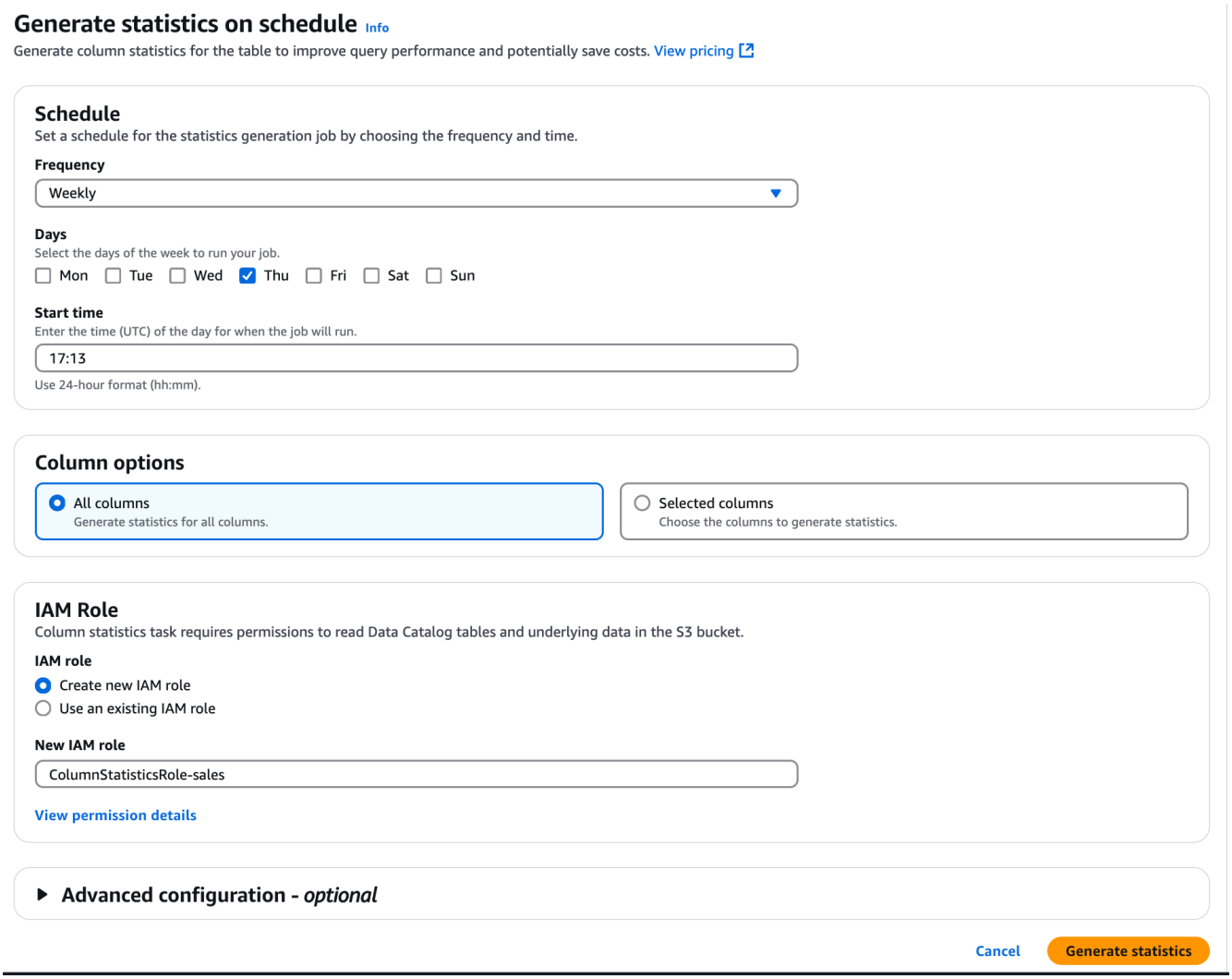
Console

Per generare statistiche delle colonne utilizzando la console

1. Accedi alla AWS Glue console all'indirizzo https://console.aws.amazon.com/glue/.
2. Seleziona le tabelle del Catalogo dati.
3. Scegliere una tabella dall'elenco.
4. Scegli la scheda Statistiche delle colonne nella sezione inferiore della pagina Tabelle.
5. Puoi anche scegliere Genera in base alla pianificazione nelle Statistiche delle colonne da Azioni.
6. Nella pagina Genera statistiche in base alla pianificazione, configura una pianificazione ricorrente per l'esecuzione dell'attività di statistica a colonne scegliendo la frequenza e l'ora di inizio. Puoi scegliere la frequenza oraria, giornaliera, settimanale o definire un'espressione Cron per specificare la pianificazione.

   Un'espressione Cron è una stringa che rappresenta uno schema di pianificazione, composto da 6 campi separati da spazi: * * * * * * <minute> <hour> <day of month> <month> <day of week> <year> Ad esempio, per eseguire un'attività ogni giorno a mezzanotte, l'espressione Cron sarebbe: 0 0 * *? *

   Per ulteriori informazioni consulta Espressioni Cron.




7. Quindi, scegli l'opzione della colonna per generare statistiche.

   - Tutte le colonne: scegli questa opzione per generare statistiche per tutte le colonne della tabella.
   - Colonne selezionate: scegliere questa opzione per generare statistiche per colonne specifiche. È possibile selezionare le colonne dall'elenco a discesa.

8. Scegli un ruolo IAM o crea un ruolo esistente con le autorizzazioni per generare statistiche. AWS Glue assume questo ruolo per generare statistiche sulle colonne.

   Un approccio più rapido consiste nel lasciare che la AWS Glue console crei un ruolo per te. Il ruolo che crea è specifico per la generazione di statistiche sulle colonne e include la politica `AWSGlueServiceRole` AWS gestita più la politica in linea richiesta per l'origine dati specificata.

Se si specifica un ruolo esistente per la generazione di statistiche delle colonne, bisogna assicurarsi che includa la policy `AWSGlueServiceRole` o equivalente (o una versione ridotta di questa policy), oltre alle policy inline richieste.

9. (Facoltativo) Scegli quindi una configurazione di sicurezza per abilitare la crittografia dei dati inattivi per i log.
10. (Facoltativo) È possibile scegliere una dimensione del campione indicando solo una percentuale specifica di righe dalla tabella per generare statistiche. Il valore predefinito è Tutte le righe. Utilizzate le frecce su e giù per aumentare o diminuire il valore percentuale.

    Includi tutte le righe nella tabella per calcolare statistiche accurate. Utilizza righe di esempio per generare statistiche delle colonne solo quando i valori approssimativi sono accettabili.
11. Scegli Genera statistiche per eseguire l'attività di generazione delle statistiche sulle colonne.

AWS CLI

È possibile utilizzare l' AWS CLI esempio seguente per creare una pianificazione della generazione di statistiche su colonne. Il nome del database, il nome della tabella e il ruolo sono parametri obbligatori, mentre i parametri facoltativi sono schedule, catalog-id column-name-list, sample-size e security-configuration.

```
aws glue create-column-statistics-task-settings \
 --database-name 'database_name' \
 --table-name table_name \
 --role 'arn:aws:iam::123456789012:role/stats-role' \
 --schedule 'cron(0 0-5 14 * * ?)' \
 --column-name-list 'col-1' \
 --catalog-id '123456789012' \
 --sample-size '10.0 ' \
 --security-configuration 'test-security'
```

È possibile generare statistiche StartColumnStatisticsTaskRunsulle colonne anche chiamando l'operazione.

## Gestione della pianificazione per la generazione delle statistiche delle colonne

È possibile gestire le operazioni di pianificazione come l'aggiornamento, l'avvio, l'interruzione e l'eliminazione delle pianificazioni per la generazione delle statistiche delle colonne in AWS Glue.

Puoi utilizzare le operazioni API di statistica AWS Glue della AWS Glue console o delle colonne per eseguire queste attività. AWS CLI

Argomenti

- Aggiornamento del programma di generazione delle statistiche delle colonne
- Interruzione della pianificazione per la generazione delle statistiche delle colonne
- Ripresa della pianificazione per la generazione delle statistiche delle colonne
- Eliminazione del programma di generazione delle statistiche delle colonne

### Aggiornamento del programma di generazione delle statistiche delle colonne

Dopo la creazione, è possibile aggiornare la pianificazione per attivare l'attività di generazione delle statistiche delle colonne È possibile utilizzare la AWS Glue console o eseguire l'UpdateColumnStatisticsTaskSettingsoperazione per aggiornare la pianificazione di una tabella. AWS CLIÈ possibile modificare i parametri di una pianificazione esistente, ad esempio il tipo di pianificazione (su richiesta o pianificata) e altri parametri opzionali.

Console di gestione AWS

Per aggiornare le impostazioni di un'attività di generazione delle statistiche delle colonne

1. Accedi alla AWS Glue console all'indirizzo https://console.aws.amazon.com/glue/.
2. Scegli la tabella da aggiornare dall'elenco delle tabelle.
3. Nella sezione inferiore della pagina Tabelle, scegli Statistiche delle colonne.
4. In Azioni, scegli Modifica per aggiornare la pianificazione.
5. Apporta le modifiche desiderate alla pianificazione e scegli Salva.

AWS CLI

Se non utilizzi AWS Glue la funzionalità di generazione delle statistiche nella console, puoi aggiornare manualmente la pianificazione utilizzando il `update-column-statistics-task-settings` comando. L'esempio seguente mostra come aggiornare le statistiche delle colonne utilizzando AWS CLI.

```
aws glue update-column-statistics-task-settings \
 --database-name 'database_name' \
 --table-name 'table_name' \
```

```
 --role arn:aws:iam::123456789012:role/stats_role \
 --schedule 'cron(0 0-5 16 * * ?)' \
 --column-name-list 'col-1' \
 --sample-size '20.0' \
 --catalog-id '123456789012'\
 --security-configuration 'test-security'
```

Interruzione della pianificazione per la generazione delle statistiche delle colonne

Se non hai più bisogno delle statistiche incrementali, puoi interrompere la generazione pianificata per risparmiare risorse e costi. Mettere in pausa la pianificazione non influisce sulle statistiche generate in precedenza. Puoi riprendere la pianificazione quando preferisci.

Console di gestione AWS

Per interrompere la pianificazione di un'attività di generazione di statistiche delle colonne

1. Sulla AWS Glue console, scegli Tabelle in Data Catalog.
2. Seleziona una tabella con le statistiche delle colonne.
3. Nella pagina dei Dettagli della tabella, scegli Statistiche delle colonne.
4. In Azioni, scegli Generazione pianificata, Pausa.
5. Scegli Pausa per confermare.

AWS CLI

Per interrompere la pianificazione dell'esecuzione di un'attività di statistica su colonne utilizzando il AWS CLI, puoi utilizzare il seguente comando:

```
aws glue stop-column-statistics-task-run-schedule \
 --database-name ''database_name' \
 --table-name 'table_name'
```

Sostituire `database_name` e `table_name` con i nomi effettivi del database e della tabella per i quali si desidera interrompere la pianificazione dell'esecuzione dell'attività di statistiche delle colonne.

### Ripresa della pianificazione per la generazione delle statistiche delle colonne

Se hai messo in pausa la pianificazione della generazione delle statistiche, ti AWS Glue consente di riprenderla quando preferisci. Puoi riprendere la pianificazione utilizzando la AWS Glue console o AWS CLI l'operazione. StartColumnStatisticsTaskRunSchedule

#### Console di gestione AWS

Per riprendere la pianificazione per la generazione delle statistiche delle colonne

1. Sulla AWS Glue console, scegli Tabelle in Data Catalog.
2. Seleziona una tabella con le statistiche delle colonne.
3. Nella pagina dei Dettagli della tabella, scegli Statistiche delle colonne.
4. In Azioni, scegli Generazione pianificata, quindi scegli Riprendi.
5. Scegli Riprendi per confermare.

#### AWS CLI

Sostituire `database_name` e `table_name` con i nomi effettivi del database e della tabella per i quali si desidera interrompere la pianificazione dell'esecuzione dell'attività di statistiche delle colonne.

```
aws glue start-column-statistics-task-run-schedule \
 --database-name 'database_name' \
 --table-name 'table_name'
```

### Eliminazione del programma di generazione delle statistiche delle colonne

Sebbene la gestione delle up-to-date statistiche sia generalmente consigliata per prestazioni ottimali delle query, esistono casi d'uso specifici in cui la rimozione della pianificazione di generazione automatica potrebbe essere utile.

- Se i dati rimangono relativamente statici, le statistiche delle colonne esistenti possono rimanere accurate per un periodo prolungato, riducendo la necessità di aggiornamenti frequenti. L'eliminazione della pianificazione può evitare il consumo inutile di risorse e il sovraccarico associato alla rigenerazione delle statistiche su dati immutati.
- Quando è preferibile il controllo manuale sulla generazione delle statistiche. Eliminando la pianificazione automatica, gli amministratori possono aggiornare selettivamente le statistiche delle

colonne a intervalli specifici o dopo modifiche significative dei dati, allineando il processo alle strategie di manutenzione e alle esigenze di allocazione delle risorse.

Console di gestione AWS

Per eliminare la pianificazione per la generazione delle statistiche delle colonne

1. Sulla AWS Glue console, scegli Tabelle in Data Catalog.
2. Seleziona una tabella con le statistiche delle colonne.
3. Nella pagina dei Dettagli della tabella, scegli Statistiche delle colonne.
4. In Azioni, scegli Generazione pianificata, Elimina.
5. Seleziona Elimina per confermare.

AWS CLI

Sostituire `database_name` e `table_name` con i nomi effettivi del database e della tabella per i quali si desidera interrompere la pianificazione dell'esecuzione dell'attività di statistiche delle colonne.

Puoi eliminare la pianificazione delle statistiche delle colonne utilizzando l'operazione DeleteColumnStatisticsTaskSettingsAPI o AWS CLI. L'esempio seguente mostra come eliminare la pianificazione per la generazione di statistiche sulle colonne utilizzando AWS Command Line Interface (AWS CLI).

```
aws glue delete-column-statistics-task-settings \
    --database-name 'database_name' \
    --table-name 'table_name'
```

## Generazione di statistiche delle colonne on demand

È possibile eseguire l'attività di statistica delle colonne per l'attività AWS Glue Data Catalog tabelle su richiesta senza una pianificazione prestabilita. Questa opzione è utile per analisi ad hoc o quando le statistiche devono essere calcolate immediatamente.

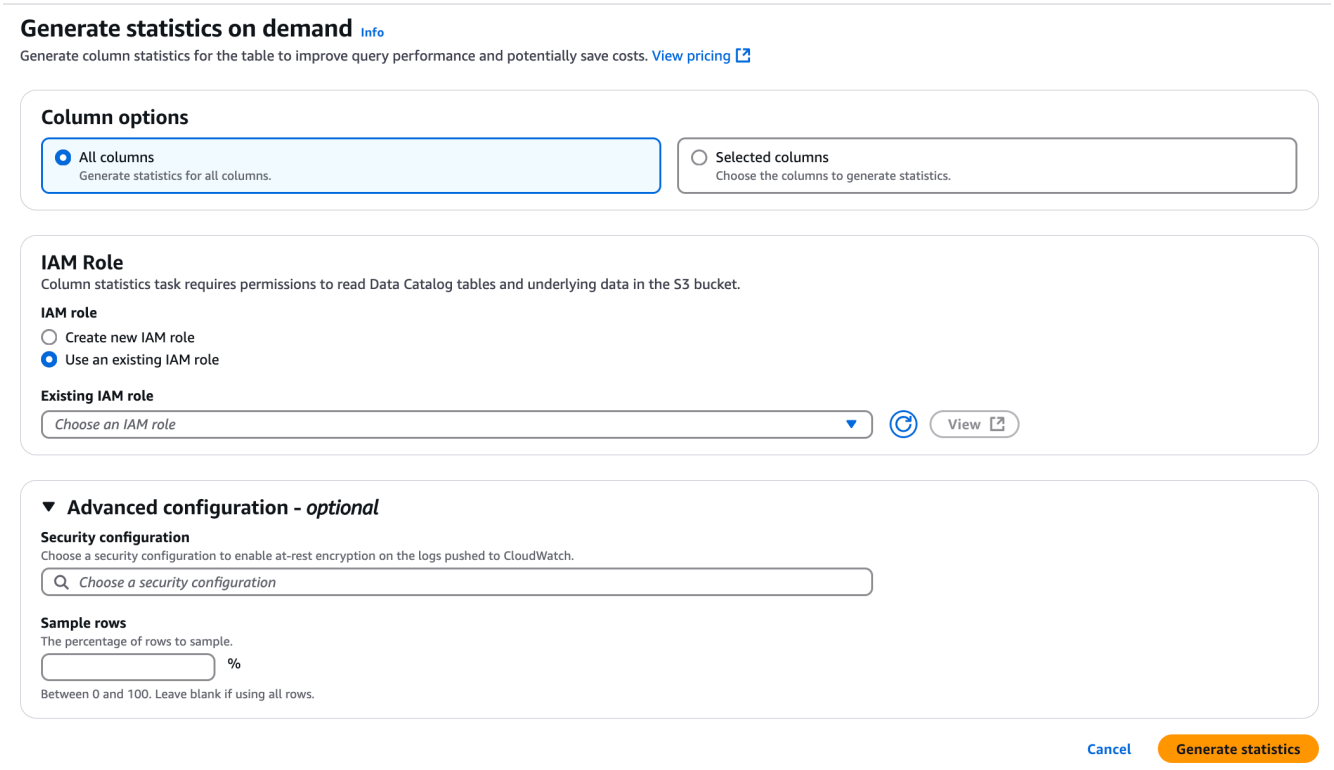
Segui questi passaggi per generare statistiche sulle colonne su richiesta per le tabelle del Data Catalog utilizzando la AWS Glue console o AWS CLI.

## Console di gestione AWS

### Per generare statistiche delle colonne utilizzando la console

1. Accedi alla AWS Glue console all'indirizzo https://console.aws.amazon.com/glue/.
2. Seleziona le tabelle del Catalogo dati.
3. Scegliere una tabella dall'elenco.
4. Scegli Genera statistiche nel menu Azioni.

   È possibile anche scegliere le opzioni Genera statistiche e Genera on demand nella scheda Statistiche delle colonne nella sezione inferiore della pagina Tabelle.
5. Seguire i passaggi da 7 a 11 in Generazione di statistiche a colonne in base a una pianificazione per generare statistiche delle colonne per la tabella.
6. Nella pagina Genera statistiche, specifica le seguenti opzioni:




   - Tutte le colonne: scegliere questa opzione per generare statistiche per tutte le colonne della tabella.
   - Colonne selezionate: scegliere questa opzione per generare statistiche per colonne specifiche. È possibile selezionare le colonne dall'elenco a discesa.

- Ruolo IAM: scegliere Crea un nuovo ruolo IAM con le policy di autorizzazione necessarie per eseguire l'attività di generazione delle statistiche delle colonne. Scegliere Visualizza i dettagli delle autorizzazioni per rivedere le istruzioni della policy. È possibile anche selezionare un ruolo IAM dall'elenco. Per ulteriori informazioni sulle autorizzazioni richieste, consulta Prerequisiti per la generazione delle statistiche delle colonne.

  AWS Glue assume le autorizzazioni del ruolo specificato per generare statistiche.

  Per ulteriori informazioni sulla fornitura di ruoli per AWS Glue, consulta Politiche basate sull'identità per. AWS Glue.
- (Facoltativo) Scegli quindi una configurazione di sicurezza per abilitare la crittografia dei dati inattivi per i log.
- Righe di esempio: scegli solo una percentuale specifica di righe dalla tabella per generare statistiche. Il valore predefinito è Tutte le righe. Utilizzate le frecce su e giù per aumentare o diminuire il valore percentuale.

  > **Note**
  >
  > Includi tutte le righe nella tabella per calcolare statistiche accurate. Utilizza righe di esempio per generare statistiche delle colonne solo quando i valori approssimativi sono accettabili.

Scegliere Genera statistiche per eseguire l'attività.

AWS CLI

Questo comando attiverà l'esecuzione di un'operazione di statistiche delle colonne per la tabella specificata. È necessario fornire il nome del database, il nome della tabella, un ruolo IAM con le autorizzazioni per generare statistiche e, facoltativamente, i nomi delle colonne e una percentuale di dimensione del campione per il calcolo delle statistiche.

```
aws glue start-column-statistics-task-run \
    --database-name 'database_name \
    --table-name 'table_name' \
    --role 'arn:aws:iam::123456789012:role/stats-role' \
    --column-name 'col1','col2'  \
    --sample-size 10.0
```

Questo comando avvierà un'attività per generare statistiche delle colonne per la tabella specificata.

### Aggiornamento delle statistiche delle colonne on demand

Il mantenimento delle statistiche sulle up-to-date colonne è fondamentale per l'ottimizzazione delle query per generare piani di esecuzione efficienti, garantendo migliori prestazioni delle query, riduzione del consumo di risorse e migliori prestazioni complessive del sistema. Questo processo è particolarmente importante dopo modifiche significative dei dati, come blocco di dati o modifiche estese, che possono rendere obsolete le statistiche esistenti.

È necessario eseguire in modo esplicito l'attività Genera statistiche dalla AWS Glue console per aggiornare le statistiche delle colonne. Il Catalogo dati non aggiorna automaticamente le statistiche.

Se non si utilizza AWS Glue la funzionalità di generazione delle statistiche nella console, è possibile aggiornare manualmente le statistiche delle colonne utilizzando l'operazione UpdateColumnStatisticsForTableAPI o. AWS CLI L'esempio seguente mostra come aggiornare le statistiche delle colonne utilizzando AWS CLI.

```
aws glue update-column-statistics-for-table --cli-input-json:

{
    "CatalogId": "111122223333",
    "DatabaseName": "database_name",
    "TableName": "table_name",
    "ColumnStatisticsList": [
        {
            "ColumnName": "col1",
            "ColumnType": "Boolean",
            "AnalyzedTime": "1970-01-01T00:00:00",
            "StatisticsData": {
                "Type": "BOOLEAN",
                "BooleanColumnStatisticsData": {
                    "NumberOfTrues": 5,
                    "NumberOfFalses": 5,
                    "NumberOfNulls": 0
                }
            }
        }
```

```
    ]
}
```

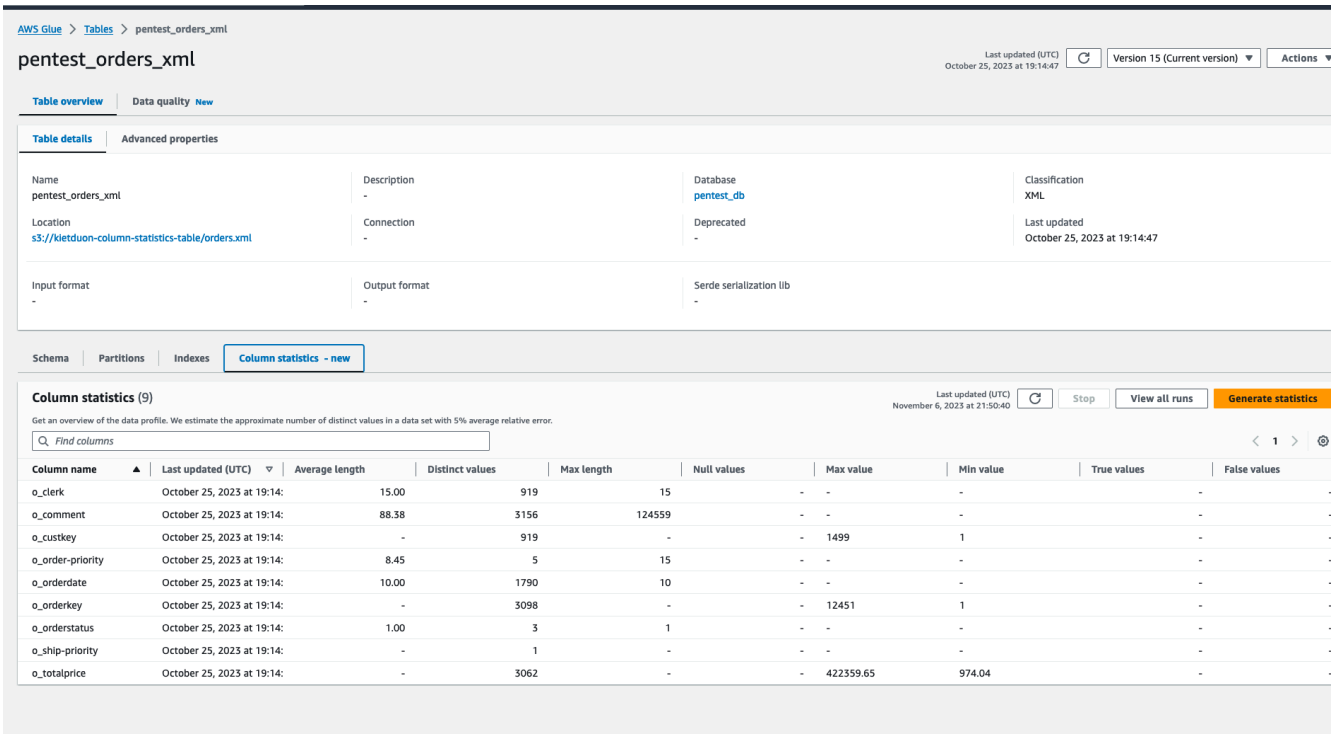
## Visualizzazione delle statistiche delle colonne

Dopo aver generato correttamente le statistiche, Data Catalog memorizza queste informazioni per gli ottimizzatori basati sui costi in Amazon Redshift Amazon Athena e in Amazon Redshift per effettuare scelte ottimali durante l'esecuzione delle query. Le statistiche variano in base al tipo di colonna.

### Console di gestione AWS

Per visualizzare le statistiche delle colonne per una tabella

- Dopo l'esecuzione dell'attività di statistica delle colonne, la scheda Statistiche delle colonne della pagina dei Dettagli della tabella mostra le statistiche relative alla tabella.




Sono disponibili le seguenti statistiche:

- Nome colonna: nome della colonna utilizzato per generare statistiche
- Ultimo aggiornamento: data e ora in cui sono state generate le statistiche
- Lunghezza media: lunghezza media dei valori nella colonna
- Valori distinti: il numero totale di valori distinti nella colonna. Eseguiamo una stima del numero di valori distinti in una colonna con un errore relativo del 5%.

- Valore massimo: il valore più alto nella colonna.
- Valore minimo: il valore più basso nella colonna.
- Lunghezza massima: la lunghezza del valore più alto nella colonna.
- Valori null: il numero di valori null nella colonna.
- Valori true: il numero di valori true nella colonna.
- Valori false: il numero di valori false nella colonna.
- numFiles: il numero totale di file nella tabella. Questo valore è disponibile nella scheda Proprietà avanzate.

AWS CLI

L'esempio seguente mostra come recuperare le statistiche delle colonne utilizzando AWS CLI.

```
aws glue get-column-statistics-for-table \
    --database-name database_name \
    --table-name table_name \
    --column-names <column_name>
```

Puoi anche visualizzare le statistiche delle colonne utilizzando l'operazione API. GetColumnStatisticsForTable

## Visualizzazione dell'attività relativa alle statistiche delle colonne

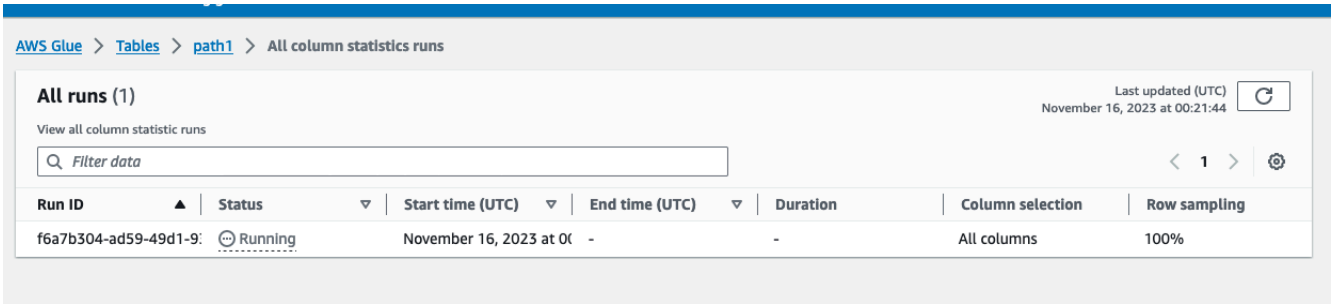
Dopo aver eseguito un'attività di statistica delle colonne, è possibile esplorare i dettagli dell'esecuzione dell'attività per una tabella utilizzando la AWS Glue console AWS CLI o utilizzando GetColumnStatisticsTaskRunsoperation.

Console

Per visualizzare i dettagli dell'attività relativa alle statistiche sulle colonne

1. Sulla AWS Glue console, scegli Tabelle in Data Catalog.
2. Seleziona una tabella con le statistiche delle colonne.
3. Nella pagina dei Dettagli della tabella, scegli Statistiche delle colonne.
4. Scegli Visualizza esecuzioni.

   Puoi visualizzare le informazioni su tutte le esecuzioni associate alla tabella specificata.




AWS CLI

Nell'esempio seguente, sostituisci i valori per `DatabaseName` e `TableName` con i nomi effettivi di database e tabelle.

```
aws glue get-column-statistics-task-runs --input-cli-json file://input.json
{
    "DatabaseName": "database_name",
    "TableName": "table_name"
}
```

## Interruzione dell'esecuzione relativa alle statistiche delle colonne

È possibile interrompere l'esecuzione di un'attività di statistica delle colonne per una tabella utilizzando la AWS Glue console AWS CLI o utilizzando StopColumnStatisticsTaskRunoperation.

Console

Per interrompere un'attività di statistica delle colonne, esegui:

1. Sulla AWS Glue console, scegli Tabelle in Data Catalog.
2. Seleziona la tabella con la colonna "Attività statistiche". L'operazione è in corso.
3. Nella pagina dei Dettagli della tabella, scegli Statistiche delle colonne.
4. Scegli Stop (Arresta).

   Se interrompi l'attività prima del completamento dell'esecuzione, le statistiche delle colonne non verranno generate per la tabella.

AWS CLI

Nell'esempio seguente, sostituisci i valori per `DatabaseName` e `TableName` con i nomi effettivi di database e tabelle.

```
aws glue stop-column-statistics-task-run --input-cli-json file://input.json
{
    "DatabaseName": "database_name",
    "TableName": "table_name"
}
```

## Eliminazione delle statistiche delle colonne

È possibile eliminare le statistiche delle colonne utilizzando l'operazione DeleteColumnStatisticsForTableAPI o AWS CLI. L'esempio seguente mostra come eliminare le statistiche delle colonne utilizzando AWS Command Line Interface (AWS CLI).

```
aws glue delete-column-statistics-for-table \
    --database-name 'database_name' \
    --table-name 'table_name' \
    --column-name 'column_name'
```

## Considerazioni e limitazioni

Le seguenti considerazioni e limitazioni si applicano alla generazione di statistiche delle colonne.

Considerazioni

- L'utilizzo del campionamento per generare statistiche riduce il tempo di esecuzione ma può generare statistiche imprecise.
- Il Catalogo dati non memorizza versioni diverse delle statistiche.
- È possibile eseguire solo un'attività alla volta per la generazione di statistiche per tabella.
- Se una tabella è crittografata utilizzando la AWS KMS chiave cliente registrata con Data Catalog, AWS Glue utilizza la stessa chiave per crittografare le statistiche.

L'attività di Statistiche delle colonne supporta la generazione di statistiche:

- Quando il ruolo IAM dispone delle autorizzazioni complete per la tabella (IAM o Lake Formation).
- Quando il ruolo IAM dispone di autorizzazioni sulla tabella utilizzando la modalità di accesso ibrida di Lake Formation.

L'attività di Statistiche delle colonne non supporta la generazione di statistiche per:

- Tabelle con controllo degli accessi basato su celle di Lake Formation
- Data lake transazionali: Linux Foundation Delta Lake, Apache Hudi
- Tabelle in database federati - Hive metastore, unità di condivisione dati Amazon Redshift
- Colonne, matrici e tipi di dati di struttura nidificati.
- Tabella condivisa con te da un altro account

## Crittografia del catalogo dati

Puoi proteggere i tuoi metadati archiviati in AWS Glue Data Catalog at rest utilizzando chiavi di crittografia gestite da AWS Key Management Service (AWS KMS). Puoi abilitare la crittografia del Catalogo dati per il nuovo Catalogo dati utilizzando le impostazioni del Catalogo dati. Puoi abilitare o disabilitare la crittografia per il Catalogo dati esistente in base alle esigenze. Se abilitata, AWS Glue crittografa tutti i nuovi metadati scritti nel catalogo, mentre i metadati esistenti rimangono non crittografati.

Per informazioni dettagliate sulla crittografia del Catalogo dati, consulta Crittografia del catalogo dati.

## Protezione del catalogo dati con Lake Formation

AWS Lake Formation è un servizio che semplifica la configurazione di un data lake sicuro AWS. Fornisce una posizione centrale per creare e gestire in modo sicuro i data lake definendo autorizzazioni di controllo granulare degli accessi. Lake Formation utilizza il catalogo dati per archiviare e recuperare i metadati relativi al data lake, come le definizioni delle tabelle, le informazioni sullo schema e le impostazioni di controllo dell'accesso ai dati.

È possibile registrare la posizione dei dati Amazon S3 della tabella o del database di metadati con Lake Formation e utilizzarla per definire le autorizzazioni a livello di metadati sulle risorse del catalogo dati. È possibile utilizzare Lake Formation per gestire le autorizzazioni di accesso allo storage sui dati sottostanti archiviati in Amazon S3 per conto di motori analitici integrati.

Per ulteriori informazioni, consulta What is AWS Lake Formation? .

# Lavorare con AWS Glue Data Catalog le viste in AWS Glue

È possibile creare e gestire le viste in AWS Glue Data Catalog, comunemente note come AWS Glue Data Catalog visualizzazioni. Queste viste sono utili perché supportano più motori di query SQL e consentono di accedere alla stessa vista su AWS servizi diversi, ad esempio Amazon Athena Amazon Redshift, e AWS Glue. È possibile utilizzare le viste basate su Apache Iceberg, Apache Hudi e Delta Lake.

Creando una vista nel Data Catalog, puoi utilizzare le concessioni di risorse e i controlli di accesso basati su tag AWS Lake Formation per concedere l'accesso ad essa. Utilizzando questo metodo di controllo degli accessi, non è necessario configurare un accesso aggiuntivo alle tabelle a cui si fa riferimento durante la creazione della vista. Questo metodo di concessione delle autorizzazioni è chiamato semantica, mentre queste viste sono chiamate viste del definitore. Per ulteriori informazioni sul controllo degli accessi in AWS Lake Formation, consulta Concessione e revoca delle autorizzazioni sulle risorse di Data Catalog nella Guida per gli sviluppatori. AWS Lake Formation

Le viste del Catalogo dati sono utili per i seguenti casi d'uso:

- Controllo granulare degli accessi: è possibile creare una vista che limiti l'accesso ai dati in base alle autorizzazioni necessarie per l'utente. Ad esempio, puoi utilizzare le viste nel Catalogo dati per impedire ai dipendenti che non lavorano nel reparto delle risorse umane di visualizzare le informazioni di identificazione personale (PII).
- Definizione completa della vista: applicando determinati filtri alla vista di Catalogo dati, è possibile assicurare che i record di dati disponibili siano sempre completi.
- Sicurezza avanzata: la definizione delle query utilizzata per creare la vista deve essere completa, in modo che le viste di Catalogo dati siano meno suscettibili all'intervento di soggetti malintenzionati nei comandi SQL.
- Condivisione semplice dei dati: condividi i dati con altri AWS account senza spostarli, utilizzando la condivisione dei dati tra account in. AWS Lake Formation

## Creazione di una vista di Catalogo Dati

Puoi creare viste del catalogo dati utilizzando AWS CLI gli script AWS Glue ETL utilizzando Spark SQL. La sintassi per la creazione di una vista di Catalogo dati include la specificazione del tipo di vista come `MULTI DIALECT` e del predicato `SECURITY` come `DEFINER`, che indica una vista del definitore.

Istruzione SQL di esempio per creare una vista nel Catalogo dati:

```
CREATE PROTECTED MULTI DIALECT VIEW database_name.catalog_view SECURITY DEFINER
AS SELECT order_date, sum(totalprice) AS price
FROM source_table
GROUP BY order_date;
```

Dopo aver creato una vista del catalogo dati, puoi utilizzare un ruolo IAM con l'autorizzazione AWS Lake Formation `SELECT` sulla vista per interrogarla da servizi come Amazon Athena, Amazon Redshift o lavori ETL. AWS Glue Non è necessario concedere l'accesso alle tabelle sottostanti a cui si fa riferimento nella vista.

Per ulteriori informazioni sulla creazione e la configurazione delle viste del catalogo dati, consulta Building AWS Glue Data Catalog views nella Developer Guide. AWS Lake Formation

## Operazioni supportate per le viste

I seguenti frammenti di comandi mostrano vari modi di lavorare con le viste nel Catalogo dati:

CREA VISUALIZZAZIONE

Crea una vista data-catalog. Di seguito è riportato un esempio che mostra la creazione di una vista da una tabella esistente:

```
CREATE PROTECTED MULTI DIALECT VIEW catalog_view
SECURITY DEFINER AS SELECT * FROM my_catalog.my_database.source_table
```

ALTER VIEW

Sintassi disponibile:

```
ALTER VIEW view_name [FORCE] ADD DIALECT AS query
ALTER VIEW view_name [FORCE] UPDATE DIALECT AS query
ALTER VIEW view_name DROP DIALECT
```

È possibile utilizzare l'opzione `FORCE ADD DIALECT` per forzare l'aggiornamento dello schema e degli oggetti secondari secondo il nuovo dialetto del motore. Tieni presente che questa operazione può causare errori di query se non utilizzi anche `FORCE` per aggiornare altri dialetti del motore. Di seguito è riportato un esempio:

```
ALTER VIEW catalog_view FORCE ADD DIALECTAS
```

```
SELECT order_date, sum(totalprice) AS priceFROM source_tableGROUP BY orderdate;
```

Quello che segue è un esempio di come modificare una vista per aggiornare il dialetto:

```
ALTER VIEW catalog_view UPDATE DIALECT AS
SELECT count(*) FROM my_catalog.my_database.source_table;
```

DESCRIVI LA VISTA

Sintassi disponibile per descrivere una vista:

`SHOW COLUMNS {FROM|IN} view_name [{FROM|IN} database_name]`— Se l'utente dispone dei requisiti AWS Glue e AWS Lake Formation delle autorizzazioni per descrivere la vista, può elencare le colonne. Di seguito sono riportati un paio di comandi di esempio per la vista delle colonne:

```
SHOW COLUMNS FROM my_database.source_table;
SHOW COLUMNS IN my_database.source_table;
```

`DESCRIBE view_name`— Se l'utente dispone dei requisiti AWS Glue e AWS Lake Formation delle autorizzazioni per descrivere la vista, può elencare le colonne della vista insieme ai relativi metadati.

ELIMINA VISUALIZZAZIONE

Sintassi disponibile:

```
DROP VIEW [ IF EXISTS ] view_name
```

L'esempio seguente mostra un'istruzione DROP che verifica l'esistenza di una vista prima di eliminarla:

```
DROP VIEW IF EXISTS catalog_view;
```

`SHOW CREATE VIEW view_name`: mostra l'istruzione SQL che crea la vista specificata. Di seguito è riportato un esempio che mostra la creazione di una vista data-catalog:

```
SHOW CREATE TABLE my_database.catalog_view;CREATE PROTECTED MULTI DIALECT VIEW
 my_catalog.my_database.catalog_view (
  net_profit,
  customer_id,
  item_id,
```

```
  sold_date)
TBLPROPERTIES (
  'transient_lastDdlTime' = '1736267222')
SECURITY DEFINER AS SELECT * FROM
my_database.store_sales_partitioned_lf WHERE customer_id IN (SELECT customer_id from
 source_table limit 10)
```

MOSTRA VISUALIZZAZIONI

Elenca tutte le viste del catalogo, ad esempio le viste normali, le viste multi-dialettali (MDV) e le MDV senza il dialetto Spark. La sintassi disponibile è la seguente:

```
SHOW VIEWS [{ FROM | IN } database_name] [LIKE regex_pattern]:
```

Il seguente è un comando di esempio per mostrare le viste:

```
SHOW VIEWS IN marketing_analytics LIKE 'catalog_view*';
```

Per ulteriori informazioni sulla creazione e la configurazione delle viste del catalogo dati, consulta Building AWS Glue Data Catalog views nella Developer Guide. AWS Lake Formation

## Interrogazione di una vista di Catalogo Dati

Dopo aver creato una vista in Catalogo dati, è possibile eseguirvi una query. Il ruolo IAM configurato nei tuoi AWS Glue job deve avere l'autorizzazione Lake Formation SELECT nella vista Data Catalog. Non è necessario concedere l'accesso alle tabelle sottostanti a cui si fa riferimento nella vista.

Dopo aver impostato tutto, è possibile eseguire query sulla vista. Ad esempio, è possibile eseguire la query seguente per accedere a una vista.

```
SELECT * from my_database.catalog_view LIMIT 10;
```

## Limitazioni

Quando si utilizzano le viste di Catalogo dati, considerare le seguenti limitazioni.

- Puoi creare viste Data Catalog solo con AWS Glue 5.0 e versioni successive.
- Il definitore della vista di Catalogo dati deve avere l'accesso `SELECT` alle tabelle di base sottostanti a cui la vista accede. La creazione della vista di Catalogo dati non riesce se una tabella base specifica ha dei filtri Lake Formation imposti sul ruolo del definitore.

- Le tabelle di base non devono avere l'autorizzazione `IAMAllowedPrincipals` per il data lake AWS Lake Formation. Se presente, si verifica l'errore Multi Dialect views può fare riferimento solo a tabelle senza le autorizzazioni IAMAllowed Principal.
- La posizione Amazon S3 della tabella deve essere registrata come posizione di AWS Lake Formation data lake. Se la tabella non è registrata, si verifica l'errore `Multi Dialect views may only reference AWS Lake Formation managed tables`. Per informazioni su come registrare le sedi Amazon S3 in AWS Lake Formation, consulta Registrazione di una sede Amazon S3 nella Developer Guide. AWS Lake Formation
- Puoi creare solo viste `PROTECTED` in Catalogo dati. Le viste `UNPROTECTED` non sono supportate.
- Non puoi fare riferimento alle tabelle di un altro AWS account in una definizione di visualizzazione del catalogo dati. Inoltre, non puoi fare riferimento a una tabella nello stesso account che si trova in una Regione separata.
- Per condividere i dati in un account o una Regione, tutta la vista deve essere condivisa in account e Regioni diverse utilizzando i link alle risorse AWS Lake Formation .
- Le funzioni definite dall'utente (UDFs) non sono supportate.
- Le viste di Catalogo Dati non possono fare riferimento ad altre viste.

# Accesso al Catalogo dati

Puoi utilizzare AWS Glue Data Catalog (Data Catalog) per scoprire e comprendere i tuoi dati. Il Catalogo dati offre un modo coerente per mantenere le definizioni degli schemi, i tipi di dati, le posizioni e altri metadati. Puoi accedere al Catalogo dati utilizzando i seguenti metodi:

- AWS Glue console: puoi accedere e gestire il Data Catalog tramite la AWS Glue console, un'interfaccia utente basata sul Web. La console consente di sfogliare e cercare database, tabelle e i relativi metadati associati, nonché creare, aggiornare ed eliminare definizioni di metadati.
- Crawler di AWS Glue — I crawler sono programmi che scansionano automaticamente le fonti di dati e popolano il Data Catalog con metadati. Puoi creare ed eseguire crawler per scoprire e catalogare dati da varie fonti come Amazon S3, Amazon RDS, Amazon DynamoDB Amazon CloudWatch e database relazionali conformi a JDBC come MySQL e PostgreSQL, oltre a diverse fonti non come Snowflake e Google.AWS BigQuery
- AWS Glue APIs — È possibile accedere al AWS Glue APIs Data Catalog in modo programmatico utilizzando. Questi APIs consentono di interagire con il Data Catalog in modo programmatico, abilitando l'automazione e l'integrazione con altre applicazioni e servizi.

- AWS Command Line Interface (AWS CLI) — È possibile utilizzare il AWS CLI per accedere e gestire il Data Catalog dalla riga di comando. L'interfaccia a riga di comando (CLI) fornisce comandi per la creazione, l'aggiornamento e l'eliminazione delle definizioni dei metadati, nonché per l'esecuzione di query e il recupero delle informazioni sui metadati.
- Integrazione con altri AWS servizi: il Data Catalog si integra con vari altri AWS servizi, consentendo di accedere e utilizzare i metadati archiviati nel catalogo. Ad esempio, puoi utilizzare Amazon Athena per interrogare le fonti di dati utilizzando i metadati nel Catalogo dati e utilizzare AWS Lake Formation per gestire l'accesso e la governance dei dati per le risorse del Catalogo dati.

Argomenti


- Connessione al Data Catalog utilizzando l'endpoint REST AWS Glue Iceberg
- Connessione al Data Catalog utilizzando l'endpoint di estensione AWS Glue Iceberg REST
- AWS Glue Specifiche REST APIs per Apache Iceberg
- Connessione al Catalogo dati da un'applicazione Spark standalone
- Mappatura di dati tra Amazon Redshift e Apache Iceberg
- Considerazioni e limitazioni sull'utilizzo di AWS Glue Iceberg REST Catalog APIs


## Connessione al Data Catalog utilizzando l'endpoint REST AWS Glue Iceberg

L'endpoint AWS Glue Iceberg Rest supporta le operazioni API specificate nella specifica di Apache Iceberg REST. Utilizzando un client Iceberg REST, è possibile collegare l'applicazione in esecuzione su un motore di analisi al catalogo REST ospitato nel Catalogo dati.

L'endpoint supporta entrambe le specifiche delle tabelle Apache Iceberg: v1 e v2, con l'impostazione predefinita su v2. Quando si utilizza la specifica della tabella Iceberg v1, occorre specificare v1 nella chiamata API. Usando l'operazione API, è possibile accedere alle tabelle Iceberg archiviate sia nello storage degli oggetti Amazon S3 che nello storage delle Tabelle Amazon S3.

Configurazione dell'endpoint

È possibile accedere al catalogo AWS Glue Iceberg REST utilizzando l'endpoint del servizio. Per l'endpoint specifico della regione, consulta la guida di riferimento agli endpoint del servizio AWS Glue. Ad esempio, quando ti connetti ad AWS Glue nella regione us-east-1, devi configurare la proprietà URI dell'endpoint come segue:

```
Endpoint : https://glue.us-east-1.amazonaws.com/iceberg
```

Proprietà di configurazione aggiuntive: quando usi il client Iceberg per connettere un motore di analisi come Spark all'endpoint del servizio, devi specificare le seguenti proprietà di configurazione dell'applicazione:

```
catalog_name = "mydatacatalog"
aws_account_id = "123456789012"
aws_region = "us-east-1"
spark = SparkSession.builder \
    ... \
    .config("spark.sql.defaultCatalog", catalog_name) \
    .config(f"spark.sql.catalog.{catalog_name}",
 "org.apache.iceberg.spark.SparkCatalog") \
    .config(f"spark.sql.catalog.{catalog_name}.type", "rest") \
    .config(f"spark.sql.catalog.{catalog_name}.uri", "https://glue.
{aws_region}.amazonaws.com/iceberg") \
    .config(f"spark.sql.catalog.{catalog_name}.warehouse", "{aws_account_id}") \
    .config(f"spark.sql.catalog.{catalog_name}.rest.sigv4-enabled", "true") \
    .config(f"spark.sql.catalog.{catalog_name}.rest.signing-name", "glue") \

 .config("spark.sql.extensions","org.apache.iceberg.spark.extensions.IcebergSparkSessionExtensi
 \
    .getOrCreate()
```

AWS Glue L'endpoint Iceberg `https://glue.us-east-1.amazonaws.com/iceberg` supporta i seguenti Iceberg REST: APIs

- GetConfig
- ListNamespaces
- CreateNamespace
- LoadNamespaceMetadata
- UpdateNamespaceProperties
- DeleteNamespace
- ListTables
- CreateTable
- LoadTable

- TableExists
- UpdateTable
- DeleteTable

## Parametri del percorso del catalogo e del prefisso

Il catalogo Iceberg REST APIs ha un prefisso in formato libero nella richiesta. URLs Ad esempio, la chiamata `ListNamespaces` API utilizza il formato URL. `GET/v1/{prefix}/namespaces` AWS Glue il prefisso segue sempre la `/catalogs/{catalog}` struttura per garantire che il percorso REST allinei la gerarchia AWS Glue multicatalogo. Il parametro di percorso `{catalog}` può essere derivato in base alle seguenti regole:

| Schema di accesso | Stile ID Catalogo Glue | Stile del prefisso | Esempio di ID catalogo predefinito | Esempio di percorso REST |
|---|---|---|---|---|
| Accedi al catalogo predefinito nell'account corrente | non obbligatorio | : | non applicabile | GET /v1/catalogs/:/namespaces |
| Accesso al catalogo predefinito in un account specifico | accountID | accountID | 111122223333 | GET /v1/catalogs/111122223333/namespaces |
| Accesso a un catalogo annidato nell'account corrente | catalog1/catalog2 | catalog1/catalog2 | rmscatalog1:db1 | OTTIENI / v1/ catalogs/rmscatalog1:db1/namespaces |
| Accesso a un catalogo annidato in uno specifico account | accountId:catalog1/catalog2 | accountId:catalog1/catalog2 | 123456789012/rmscatalog1:db1 | GET /v1/catalogs/123456789012:r |

| Schema di accesso | Stile ID Catalogo Glue | Stile del prefisso | Esempio di ID catalogo predefinito | Esempio di percorso REST |
|---|---|---|---|---|
| | | | | mscatalog1:db1/namespaces |

Questa mappatura tra l'ID del catalogo e il prefisso è richiesta solo quando si chiama direttamente REST. APIs Quando lavori con il catalogo REST di AWS Glue Iceberg APIs tramite un motore, devi specificare l'ID del AWS Glue catalogo nel `warehouse` parametro per l'impostazione dell'API del catalogo Iceberg REST o nel `glue.id` parametro per l' AWS Glue impostazione dell'API delle estensioni. Scopri come utilizzarlo ad esempio con EMR Spark in Utilizza un cluster Iceberg con Spark.

### Parametro del percorso namespace

I namespace nel percorso del catalogo REST di Iceberg possono avere più livelli APIs . Tuttavia, AWS Glue supporta solo namespace a livello singolo. Per accedere a un namespace in una gerarchia di cataloghi multilivello, è possibile connettersi a un catalogo multilivello sopra il namespace per fare riferimento al namespace. Ciò consente a qualsiasi motore di query che supporti la notazione in 3 parti di accedere `catalog.namespace.table` agli oggetti nella gerarchia del catalogo a più livelli senza problemi AWS Glue di compatibilità rispetto all'utilizzo del namespace a più livelli.

## Connessione al Data Catalog utilizzando l'endpoint di estensione AWS Glue Iceberg REST

AWS Glue L'endpoint di estensione Iceberg REST fornisce funzionalità aggiuntive APIs, non presenti nella specifica Apache Iceberg REST, e fornisce funzionalità di pianificazione della scansione lato server. Questi elementi aggiuntivi APIs vengono utilizzati quando accedi alle tabelle archiviate nello storage gestito di Amazon Redshift. L'endpoint è accessibile da un'applicazione che utilizza le estensioni Apache Iceberg. AWS Glue Data Catalog

Configurazione dell'endpoint: è possibile accedere a un catalogo con tabelle nello storage gestito Redshift usando l'endpoint del servizio. Per l'endpoint specifico della regione, consulta la guida di riferimento agli endpoint del servizio AWS Glue. Ad esempio, quando ci si connette alla AWS Glue regione us-east-1, è necessario configurare la proprietà URI dell'endpoint come segue:

```
Endpoint : https://glue.us-east-1.amazonaws.com/extensions
```

```
catalog_name = "myredshiftcatalog"
aws_account_id = "123456789012"
aws_region = "us-east-1"
spark = SparkSession.builder \
    .config("spark.sql.defaultCatalog", catalog_name) \
    .config(f"spark.sql.catalog.{catalog_name}",
 "org.apache.iceberg.spark.SparkCatalog") \
    .config(f"spark.sql.catalog.{catalog_name}.type", "glue") \
    .config(f"spark.sql.catalog.{catalog_name}.glue.id",
 "{123456789012}:redshiftnamespacecatalog/redshiftdb") \

 .config("spark.sql.extensions","org.apache.iceberg.spark.extensions.IcebergSparkSessionExtensi
 \
    .getOrCreate()
```

## AWS Glue Specifiche REST APIs per Apache Iceberg

Questa sezione contiene le specifiche sul catalogo e sull' AWS Glue estensione AWS Glue APIs Iceberg REST e le considerazioni relative al loro utilizzo. APIs

Le richieste API agli AWS Glue Data Catalog endpoint vengono autenticate utilizzando AWS Signature Version 4 (SigV4). Consulta AWS la sezione Signature Version 4 per le richieste API per saperne di più su SigV4. AWS

Quando accede all'endpoint del AWS Glue servizio e ai AWS Glue metadati, l'applicazione assume un ruolo IAM che richiede l'intervento di IAM. `glue:getCatalog`

L'accesso al Catalogo dati e ai relativi oggetti può essere gestito utilizzando le autorizzazioni IAM, Lake Formation o Lake Formation in modalità ibrida.

I cataloghi federati nel Catalogo dati hanno posizioni di dati registrate da Lake Formation. Lake Formation collabora con Catalogo dati per fornire autorizzazioni in stile database per gestire l'accesso degli utenti agli oggetti del Catalogo dati.

Puoi utilizzare le autorizzazioni in modalità ibrida IAM o Lake Formation per gestire l'accesso al Data Catalog predefinito e ai relativi oggetti. AWS Lake Formation

Per creare, inserire o eliminare i dati negli oggetti gestiti di Lake Formation, devi impostare autorizzazioni specifiche per il ruolo o l'utente IAM.

- CREATE_CATALOG: necessario per creare cataloghi
- CREATE_DATABASE: necessario per creare database
- CREATE_TABLE: necessario per creare tabelle
- DELETE: necessario per eliminare i dati da una tabella
- DESCRIBE: necessario per leggere i metadati
- DROP: obbligatorio per drop/delete una tabella o un database
- INSERT: necessario quando il principale deve inserire dati in una tabella
- SELECT: necessario quando il principale deve selezionare i dati da una tabella

Per ulteriori informazioni, consulta il riferimento alle autorizzazioni di Lake Formation nella Guida per gli AWS Lake Formation sviluppatori.

GetConfig

Informazioni generali

| | |
|---|---|
| Nome operazione | GetConfig |
| Tipo | API del catalogo REST Iceberg |
| Percorso REST | OTTIENI/iceberg/v1/config |
| Operazione IAM | colla: GetCatalog |
| Autorizzazioni Lake Formation | Non applicabile |
| CloudTrail Evento . | colla: GetCatalog |
| Definizione Open API | https://github.com/apache/iceberg/blob/apache-iceberg-1.6.1/open-api/rest- catalog-open-api .yaml #L67 |

Considerazioni e limitazioni

- Il parametro di `warehouse` interrogazione deve essere impostato sull'ID del AWS Glue catalogo. Se non è impostato, il catalogo principale dell'account attuale viene utilizzato per restituire la

risposta. Per ulteriori informazioni, consulta the section called "Parametri del percorso del catalogo e del prefisso".

### GetCatalog

#### Informazioni generali

| | |
|---|---|
| Nome operazione | GetCatalog |
| Tipo | AWS Glue API di estensione |
| Percorso REST | GET/extensions/v1/catalogs/{catalogo} |
| Operazione IAM | colla: GetCatalog |
| Autorizzazioni Lake Formation | DESCRIBE |
| CloudTrail Evento . | colla: GetCatalog |
| Definizione Open API | https://github.com/awslabs/glue-extensions-for-iceberg/blob/main/glue-extensions-api.yaml #L40 |

#### Considerazioni e limitazioni

- Il parametro del percorso del catalogo deve seguire lo stile descritto nella sezione the section called "Parametri del percorso del catalogo e del prefisso".

### ListNamespaces

#### Informazioni generali

| | |
|---|---|
| Nome operazione | ListNamespaces |
| Tipo | API del catalogo REST Iceberg |
| Percorso REST | GET/iceberg/v1/catalogs/{catalog}/namespaces |
| Operazione IAM | colla: GetDatabase |
| Autorizzazioni Lake Formation | ALL, DESCRIBE, SELECT |

| | |
|---|---|
| CloudTrail Evento . | colla: GetDatabase |
| Definizione Open API | https://github.com/apache/iceberg/blob/apache-iceberg-1.6.1/open-api/rest- catalog-open-api .yaml #L205 |

Considerazioni e limitazioni

- Il parametro del percorso del catalogo deve seguire lo stile descritto nella sezione the section called "Parametri del percorso del catalogo e del prefisso".
- Vengono visualizzati solo i namespace del livello successivo. Per elencare i namespace dei livelli più avanzati, specifica l'ID del catalogo annidato nel parametro del percorso del catalogo.

CreateNamespace

Informazioni generali

| | |
|---|---|
| Nome operazione | CreateNamespace |
| Tipo | API del catalogo REST Iceberg |
| Percorso REST | POST/iceberg/v1/catalogs/{catalog}/namespaces |
| Operazione IAM | colla: CreateDatabase |
| Autorizzazioni Lake Formation | ALL, DESCRIBE, SELECT |
| CloudTrail Evento . | colla: CreateDatabase |
| Definizione Open API | https://github.com/apache/iceberg/blob/apache-iceberg-1.6.1/open-api/rest- catalog-open-api .yaml #L256 |

Considerazioni e limitazioni

- Il parametro del percorso del catalogo deve seguire lo stile descritto nella sezione the section called "Parametri del percorso del catalogo e del prefisso".
- È possibile creare solo namespace a livello singolo. Per creare un namespace multilivello, è necessario creare iterativamente ogni livello e connettersi al livello utilizzando il parametro del percorso del catalogo.

StartCreateNamespaceTransaction

Informazioni generali

| | |
|---|---|
| Nome operazione | StartCreateNamespaceTransaction |
| Tipo | AWS Glue estensioni API |
| Percorso REST | POST/extensions/v1/catalogs/{catalog}/namespaces |
| Operazione IAM | colla: CreateDatabase |
| Autorizzazioni Lake Formation | ALL, DESCRIBE, SELECT |
| CloudTrail Evento . | colla: CreateDatabase |
| Definizione Open API | https://github.com/apache/iceberg/blob/apache-iceberg-1.6.1/open-api/rest- catalog-open-api .yaml #L256 |

Considerazioni e limitazioni

- Il parametro del percorso del catalogo deve seguire lo stile descritto nella sezione the section called "Parametri del percorso del catalogo e del prefisso".
- È possibile creare un solo namespace a livello singolo. Per creare namespace a più livelli, è necessario creare in modo iterativo ogni livello e connettersi al livello utilizzando il parametro del percorso del catalogo.
- L'API è asincrona e restituisce un ID di transazione che è possibile utilizzare per il tracciamento tramite la chiamata API `CheckTransactionStatus`.
- È possibile chiamare questa API solo se la chiamata API `GetCatalog` contiene il parametro `use-extensions=true` nella risposta.

LoadNamespaceMetadata

Informazioni generali

| | |
|---|---|
| Nome operazione | LoadNamespaceMetadata |
| Tipo | API del catalogo REST Iceberg |

| | |
|---|---|
| Percorso REST | GET/iceberg/v1/catalogs/{catalog}/namespaces/{ns} |
| Operazione IAM | colla: GetDatabase |
| Autorizzazioni Lake Formation | ALL, DESCRIBE, SELECT |
| CloudTrail Evento . | colla: GetDatabase |
| Definizione Open API | https://github.com/apache/iceberg/blob/apache-iceberg-1.6.1/open-api/rest- catalog-open-api .yaml #L302 |

Considerazioni e limitazioni

- Il parametro del percorso del catalogo deve seguire lo stile descritto nella sezione the section called "Parametri del percorso del catalogo e del prefisso".
- È possibile specificare solo un namespace a livello singolo nel parametro del percorso REST. Per ulteriori informazioni, consultare la sezione the section called "Parametro del percorso namespace".

UpdateNamespaceProperties

Informazioni generali

| | |
|---|---|
| Nome operazione | UpdateNamespaceProperties |
| Tipo | API del catalogo REST Iceberg |
| Percorso REST | POSTA/iceberg/v1/catalogs/{catalog}/namespaces/{ns}/properties |
| Operazione IAM | colla: UpdateDatabase |
| Autorizzazioni Lake Formation | ALL, ALTER |
| CloudTrail Evento . | colla: UpdateDatabase |
| Definizione Open API | https://github.com/apache/iceberg/blob/apache-iceberg-1.6.1/open-api/rest- catalog-open-api .yaml #L400 |

Considerazioni e limitazioni

- Il parametro del percorso del catalogo deve seguire lo stile descritto nella sezione the section called "Parametri del percorso del catalogo e del prefisso".
- È possibile specificare solo un namespace a livello singolo nel parametro del percorso REST. Per ulteriori informazioni, consultare la sezione the section called "Parametro del percorso namespace".

DeleteNamespace

Informazioni generali

| | |
|---|---|
| Nome operazione | DeleteNamespace |
| Tipo | API del catalogo REST Iceberg |
| Percorso REST | DELETE/iceberg/v1/catalogs/{catalog}/namespces/{ns} |
| Operazione IAM | colla: DeleteDatabase |
| Autorizzazioni Lake Formation | ALL, DROP |
| CloudTrail Evento . | colla: DeleteDatabase |
| Definizione Open API | https://github.com/apache/iceberg/blob/apache-iceberg-1.6.1/open-api/rest- catalog-open-api .yaml #L365 |

Considerazioni e limitazioni

- Il parametro del percorso del catalogo deve seguire lo stile descritto nella sezione the section called "Parametri del percorso del catalogo e del prefisso".
- È possibile specificare solo un namespace a livello singolo nel parametro del percorso REST. Per ulteriori informazioni, consulta la sezione the section called "Parametro del percorso namespace".
- Se nel database sono presenti oggetti, l'operazione non andrà a buon fine.
- L'API è asincrona e restituisce un ID di transazione che è possibile utilizzare per il tracciamento tramite la chiamata API `CheckTransactionStatus`.
- L'API può essere utilizzata solo se la chiamata API `GetCatalog` indica una risposta `use-extensions=true`.

StartDeleteNamespaceTransaction

Informazioni generali

| | |
|---|---|
| Nome operazione | StartDeleteNamespaceTransaction |
| Tipo | AWS Glue estensioni API |
| Percorso REST | ELIMINA/extensions/v1/catalogs/{catalog}/namespces/{ns} |
| Operazione IAM | colla: DeleteDatabase |
| Autorizzazioni Lake Formation | ALL, DROP |
| CloudTrail Evento . | colla: DeleteDatabase |
| Definizione Open API | https://github.com/awslabs/glue-extensions-for-iceberg/blob/main/glue-extensions-api.yaml #L85 |

Considerazioni e limitazioni

- Il parametro del percorso del catalogo deve seguire lo stile descritto nella sezione the section called "Parametri del percorso del catalogo e del prefisso".
- È possibile specificare solo un namespace a livello singolo nel parametro del percorso REST. Per ulteriori informazioni, consultare la sezione the section called "Parametro del percorso namespace".
- Se nel database sono presenti oggetti, l'operazione non andrà a buon fine.
- L'API è asincrona e restituisce un ID di transazione che è possibile utilizzare per il tracciamento tramite la chiamata API `CheckTransactionStatus`.
- L'API può essere utilizzata solo se la chiamata API `GetCatalog` indica una risposta `use-extensions=true`.

ListTables

Informazioni generali

| | |
|---|---|
| Nome operazione | ListTables |
| Tipo | API del catalogo REST Iceberg |

| | |
|---|---|
| Percorso REST | OTTIENI/iceberg/v1/catalogs/{catalog}/namespaces/{ns}/tables |
| Operazione IAM | colla: GetTables |
| Autorizzazioni Lake Formation | ALL, SELECT, DESCRIBE |
| CloudTrail Evento . | colla: GetTables |
| Definizione Open API | https://github.com/apache/iceberg/blob/apache-iceberg-1.6.1/open-api/rest- catalog-open-api .yaml #L463 |

Considerazioni e limitazioni

- Il parametro del percorso del catalogo deve seguire lo stile descritto nella sezione the section called "Parametri del percorso del catalogo e del prefisso".
- È possibile specificare solo un namespace a livello singolo nel parametro del percorso REST. Per ulteriori informazioni, consulta la sezione the section called "Parametro del percorso namespace".
- Verranno elencate tutte le tabelle, incluse quelle non Iceberg. Per determinare se una tabella può essere caricata o meno come tabella Iceberg, chiama l'operazione `LoadTable`.

CreateTable

Informazioni generali

| | |
|---|---|
| Nome operazione | CreateTable |
| Tipo | API del catalogo REST Iceberg |
| Percorso REST | OTTIENI/iceberg/v1/catalogs/{catalog}/namespaces/{ns}/tables |
| Operazione IAM | colla: CreateTable |
| Autorizzazioni Lake Formation | ALL, CREATE_TABLE |
| CloudTrail Evento . | colla: CreateTable |

| | |
|---|---|
| Definizione Open API | https://github.com/apache/iceberg/blob/apache-iceberg-1.6.1/open-api/rest- catalog-open-api .yaml #L497 |

Considerazioni e limitazioni

- Il parametro del percorso del catalogo deve seguire lo stile descritto nella sezione the section called "Parametri del percorso del catalogo e del prefisso".
- È possibile specificare solo un namespace a livello singolo nel parametro del percorso REST. Per ulteriori informazioni, consultare la sezione the section called "Parametro del percorso namespace".
- `CreateTable` con gestione temporanea non è supportato. Se viene specificato il parametro di query `stageCreate`, l'operazione avrà esito negativo. Ciò significa che un'operazione come `CREATE TABLE AS SELECT` non è supportata e che devi utilizzare una combinazione di `CREATE TABLE` e `INSERT INTO` per risolvere il problema.
- L'operazione API `CreateTable` non supporta l'opzione `state-create = TRUE`.

StartCreateTableTransaction

Informazioni generali

| | |
|---|---|
| Nome operazione | CreateTable |
| Tipo | AWS Glue estensioni API |
| Percorso REST | POST/extensions/v1/catalogs/{catalog}/namespaces/{ns}/tables |
| Operazione IAM | colla: CreateTable |
| Autorizzazioni Lake Formation | ALL, CREATE_TABLE |
| CloudTrail Evento . | colla: CreateTable |
| Definizione Open API | https://github.com/awslabs/glue-extensions-for-iceberg/blob/main/glue-extensions-api.yaml #L107 |

Considerazioni e limitazioni

- Il parametro del percorso del catalogo deve seguire lo stile descritto nella sezione the section called "Parametri del percorso del catalogo e del prefisso".
- È possibile specificare solo un namespace a livello singolo nel parametro del percorso REST. Per ulteriori informazioni, consultare la sezione the section called "Parametro del percorso namespace".
- `CreateTable` con gestione temporanea non è supportato. Se viene specificato il parametro di query `stageCreate`, l'operazione avrà esito negativo. Ciò significa che un'operazione come `CREATE TABLE AS SELECT` non è supportata e che l'utente deve utilizzare una combinazione di `CREATE TABLE` e `INSERT INTO` per risolvere il problema.
- L'API è asincrona e restituisce un ID di transazione che è possibile utilizzare per il tracciamento tramite la chiamata API `CheckTransactionStatus`.
- L'API può essere utilizzata solo se la chiamata API `GetCatalog` indica una risposta `use-extensions=true`.

LoadTable

Informazioni generali

| | |
|---|---|
| Nome operazione | LoadTable |
| Tipo | API del catalogo REST Iceberg |
| Percorso REST | GETiceberg/v1/catalogs/{catalog}/namespaces/{ns}/tables/{tabella} |
| Azione IAM | colla: GetTable |
| Autorizzazioni Lake Formation | ALL, SELECT, DESCRIBE |
| CloudTrail evento | colla: GetTable |
| Definizione Open API | https://github.com/apache/iceberg/blob/apache-iceberg-1.6.1/open-api/rest- catalog-open-api .yaml #L616 |

Considerazioni

- Il parametro del percorso del catalogo deve seguire lo stile descritto nella sezione the section called "Parametri del percorso del catalogo e del prefisso".
- È possibile specificare solo un namespace a livello singolo nel parametro del percorso REST. Per ulteriori informazioni, consulta la sezione the section called "Parametro del percorso namespace".
- `CreateTable` con gestione temporanea non è supportato. Se viene specificato il parametro di query `stageCreate`, l'operazione avrà esito negativo. Ciò significa che un'operazione come `CREATE TABLE AS SELECT` non è supportata e che l'utente deve utilizzare una combinazione di `CREATE TABLE` e `INSERT INTO` per risolvere il problema.
- L'API è asincrona e restituisce un ID di transazione che è possibile utilizzare per il tracciamento tramite la chiamata API `CheckTransactionStatus`.
- L'API può essere utilizzata solo se la chiamata API `GetCatalog` indica una risposta `use-extensions=true`.

ExtendedLoadTable

Informazioni generali

| | |
|---|---|
| Nome operazione | LoadTable |
| Tipo | AWS Glue estensioni API |
| Percorso REST | GET/extensions/v1/catalogs/{catalog}/namespaces/{ns}/tables/{table} |
| Azione IAM | colla: GetTable |
| Autorizzazioni Lake Formation | ALL, SELECT, DESCRIBE |
| CloudTrail evento | colla: GetTable |
| Definizione Open API | https://github.com/awslabs/glue-extensions-for-iceberg/blob/main/glue-extensions-api.yaml #L134 |

Considerazioni

- Il parametro del percorso del catalogo deve seguire lo stile descritto nella sezione the section called "Parametri del percorso del catalogo e del prefisso".
- È possibile specificare solo un namespace a livello singolo nel parametro del percorso REST. Per ulteriori informazioni, consulta la sezione the section called "Parametro del percorso namespace".
- È supportata solo la modalità `all` per il parametro delle query degli snapshot.
- Rispetto all'API `LoadTable`, l'API `ExtendedLoadTable` si differenzia nei seguenti modi:
  - Non impone rigorosamente che tutti i campi siano disponibili.
  - fornisce i seguenti parametri aggiuntivi nel campo di configurazione della risposta:

    Parametri aggiuntivi

| Chiave di configurazione | Description |
|---|---|
| leggi. server-side-capabilities.pianificazione della scansione | Indica se la tabella può essere scansionata utilizzando la tabella e. PreparePlan PlanTable APIs |
| leggi. server-side-capabilities.data-commit | Indica se è possibile eseguire il commit della tabella utilizzando la transazione. StartUpdateTable |
| aws.glue.staging.location | Utilizzato per la pianificazione della scansione lato server o il data commit, una posizione temporanea gestita dal servizio che può essere utilizzata dal motore per scrivere file di dati temporanei |
| aws.glue.staging. access-key-id | Utilizzato per la pianificazione della scansione lato server o il data commit, una parte delle AWS credenziali temporane e per accedere alla posizione temporanea gestita dal servizio |
| aws.glue.staging. secret-access-key | Utilizzato per la pianificazione della scansione lato server o il data commit, parte delle AWS credenziali temporane e per accedere alla posizione temporanea gestita dal servizio. |
| aws.glue.staging.session-token | Utilizzato per la pianificazione della scansione lato server o il data commit, fa parte delle AWS credenziali temporane |

| Chiave di configurazione | Description |
| --- | --- |
| | e per accedere alla posizione temporanea gestita dal servizio. |
| aws.glue.staging.expiration-ms | Utilizzato per la pianificazione della scansione lato server o il data commit, il tempo di scadenza delle credenziali per accedere alla posizione temporanea gestita dal servizio. |
| aws.glue.staging. data-transfer-role-arn | Utilizzato per la pianificazione della scansione lato server o il data commit, un ruolo IAM che può essere assunto per accedere alla posizione di staging gestita dal servizio. |

PreplanTable

Informazioni generali

| | |
| --- | --- |
| Nome operazione | PreplanTable |
| Tipo | AWS Glue estensioni API |
| Percorso REST | POSTA/extensions/v1/catalogs/{catalog}/namespaces/{ns}/tables/{table}/preplan |
| Azione IAM | colla: GetTable |
| Autorizzazioni Lake Formation | ALL, SELECT, DESCRIBE |
| CloudTrail evento | colla: GetTable |
| Definizione Open API | https://github.com/awslabs/glue-extensions-for-iceberg/blob/main/glue-extensions-api.yaml #L211 |

Considerazioni

- Il parametro del percorso del catalogo deve seguire lo stile descritto nella sezione the section called "Parametri del percorso del catalogo e del prefisso".
- È possibile specificare solo un namespace a livello singolo nel parametro del percorso REST. Per ulteriori informazioni, consultare la sezione the section called "Parametro del percorso namespace".

- Il chiamante di questa API deve sempre determinare se ci sono risultati rimanenti da recuperare in base al token della pagina. Una risposta con un elemento di pagina vuoto ma un token di impaginazione è possibile se il lato server è ancora in fase di elaborazione ma non è in grado di produrre alcun risultato nel tempo di risposta specificato.
- È possibile utilizzare questa API solo se la risposta dell'API `ExtendedLoadTable` contiene `aws.server-side-capabilities.scan-planning=true`.

PlanTable

Informazioni generali

| | |
|---|---|
| Nome operazione | PlanTable |
| Tipo | AWS Glue estensioni API |
| Percorso REST | POSTA/extensions/v1/catalogs/{catalog}/namespaces/{ns}/tables/{table}/plan |
| Azione IAM | colla: GetTable |
| Autorizzazioni Lake Formation | ALL, SELECT, DESCRIBE |
| CloudTrail evento | colla: GetTable |
| Definizione Open API | https://github.com/awslabs/glue-extensions-for-iceberg/blob/main/glue-extensions-api.yaml #L243 |

Considerazioni

- Il parametro del percorso del catalogo deve seguire lo stile descritto nella sezione the section called "Parametri del percorso del catalogo e del prefisso".
- È possibile specificare solo un namespace a livello singolo nel parametro del percorso REST. Per ulteriori informazioni, consultare la sezione the section called "Parametro del percorso namespace".
- Il chiamante di questa API deve sempre determinare se ci sono risultati rimanenti da recuperare in base al token della pagina. Una risposta con un elemento di pagina vuoto ma un token di impaginazione è possibile se il lato server è ancora in fase di elaborazione ma non è in grado di produrre alcun risultato nel tempo di risposta specificato.

- È possibile utilizzare questa API solo se la risposta dell'API `ExtendedLoadTable` contiene `aws.server-side-capabilities.scan-planning=true`.

TableExists

Informazioni generali

| | |
|---|---|
| Nome operazione | TableExists |
| Tipo | API del catalogo REST Iceberg |
| Percorso REST | HEAD/iceberg/v1/catalogs/{catalog}/namespaces/{ns}/tables/{tabella} |
| Azione IAM | colla: GetTable |
| Autorizzazioni Lake Formation | ALL, SELECT, DESCRIBE |
| CloudTrail evento | colla: GetTable |
| Definizione Open API | https://github.com/apache/iceberg/blob/apache-iceberg-1.6.1/open-api/rest- catalog-open-api .yaml #L833 |

Considerazioni

- Il parametro del percorso del catalogo deve seguire lo stile descritto nella sezione the section called "Parametri del percorso del catalogo e del prefisso".
- È possibile specificare solo un namespace a livello singolo nel parametro del percorso REST. Per ulteriori informazioni, consultare la sezione the section called "Parametro del percorso namespace".

UpdateTable

Informazioni generali

| | |
|---|---|
| Nome operazione | UpdateTable |
| Tipo | API del catalogo REST Iceberg |

| | |
|---|---|
| Percorso REST | POSTiceberg/v1/catalogs/{catalog}/namespaces/{ns}/tables/{tabella} |
| Azione IAM | colla: UpdateTable |
| Autorizzazioni Lake Formation | ALL, ALTER |
| CloudTrail evento | colla: UpdateTable |
| Definizione Open API | https://github.com/apache/iceberg/blob/apache-iceberg-1.6.1/open-api/rest- catalog-open-api .yaml #L677 |

Considerazioni

- Il parametro del percorso del catalogo deve seguire lo stile descritto nella sezione the section called "Parametri del percorso del catalogo e del prefisso".
- È possibile specificare solo un namespace a livello singolo nel parametro del percorso REST. Per ulteriori informazioni, consultare la sezione the section called "Parametro del percorso namespace".

StartUpdateTableTransaction

Informazioni generali

| | |
|---|---|
| Nome operazione | StartUpdateTableTransaction |
| Tipo | AWS Glue API di estensione |
| Percorso REST | POST/extensions/v1/catalogs/{catalog}/namespaces/{ns}/tables/{tabella} |
| Azione IAM | colla: UpdateTable |
| Autorizzazioni Lake Formation | ALL, ALTER |
| CloudTrail evento | colla: UpdateTable |
| Definizione Open API | https://github.com/awslabs/glue-extensions-for-iceberg/blob/main/glue-extensions-api.yaml #L154 |

Considerazioni

- Il parametro del percorso del catalogo deve seguire lo stile descritto nella sezione the section called "Parametri del percorso del catalogo e del prefisso".
- È possibile specificare solo un namespace a livello singolo nel parametro del percorso REST. Per ulteriori informazioni, consultare la sezione the section called "Parametro del percorso namespace".
- L'API è asincrona e restituisce un ID di transazione che è possibile utilizzare per il tracciamento tramite la chiamata API `CheckTransactionStatus`.
- Un'operazione `RenamTable` può essere eseguita anche tramite questa API. Quando ciò accade, il chiamante deve disporre anche dell'autorizzazione glue: CreateTable o LakeFormation CREATE_TABLE per rinominare la tabella.
- È possibile utilizzare questa API solo se la risposta dell'API `ExtendedLoadTable` contiene `aws.server-side-capabilities.scan-planning=true`.

DeleteTable

Informazioni generali

| | |
|---|---|
| Nome operazione | DeleteTable |
| Tipo | API del catalogo REST Iceberg |
| Percorso REST | DELETE/iceberg/v1/catalogs/{catalog}/namespaces/{ns}/tables/{tabella} |
| Azione IAM | colla: DeleteTable |
| Autorizzazioni Lake Formation | ALL, DROP |
| CloudTrail evento | colla: DeleteTable |
| Definizione Open API | https://github.com/apache/iceberg/blob/apache-iceberg-1.6.1/open-api/rest- catalog-open-api .yaml #L793 |

Considerazioni

- Il parametro del percorso del catalogo deve seguire lo stile descritto nella sezione the section called "Parametri del percorso del catalogo e del prefisso".

- È possibile specificare solo un namespace a livello singolo nel parametro del percorso REST. Per ulteriori informazioni, consulta la sezione the section called "Parametro del percorso namespace".
- L'operazione API `DeleteTable` supporta un'opzione di eliminazione definitiva. Quando l'opzione di eliminazione è impostata su `true`, i dati della tabella vengono eliminati, altrimenti i dati non vengono eliminati. Per le tabelle in Amazon S3, l'operazione non elimina i dati della tabella. L'operazione non va a buon fine quando la tabella è archiviata in Amazon S3 e `purge = TRUE,`.

  Per le tabelle archiviate nello storage gestito Amazon Redshift, l'operazione eliminerà i dati della tabella, in modo simile al comportamento `DROP TABLE` in Amazon Redshift. L'operazione non va a buon fine quando la tabella è archiviata in Amazon Redshift e `purge = FALSE`.
- `purgeRequest=true` non è supportato.

StartDeleteTableTransaction

Informazioni generali

| | |
|---|---|
| Nome operazione | StartDeleteTableTransaction |
| Tipo | AWS Glue estensioni API |
| Percorso REST | ELIMINA/extensions/v1/catalogs/{catalog}/namespaces/{ns}/tables/{tabella} |
| Azione IAM | colla: DeleteTable |
| Autorizzazioni Lake Formation | ALL, DROP |
| CloudTrail evento | colla: DeleteTable |
| Definizione Open API | https://github.com/apache/iceberg/blob/apache-iceberg-1.6.1/open-api/rest- catalog-open-api .yaml #L793 |

Considerazioni

- Il parametro del percorso del catalogo deve seguire lo stile descritto nella sezione the section called "Parametri del percorso del catalogo e del prefisso".
- È possibile specificare solo un namespace a livello singolo nel parametro del percorso REST. Per ulteriori informazioni, consultare la sezione the section called "Parametro del percorso namespace".

- `purgeRequest=false` non è supportato.
- L'API è asincrona e restituisce un ID di transazione che può essere tracciato tramite `CheckTransactionStatus`.

CheckTransactionStatus

Informazioni generali

| | |
|---|---|
| Nome operazione | CheckTransactionStatus |
| Tipo | AWS Glue estensioni API |
| Percorso REST | POST/extensions/v1/transactions/status |
| Azione IAM | La stessa autorizzazione dell'azione che avvia la transazione |
| Autorizzazioni Lake Formation | La stessa autorizzazione dell'azione che avvia la transazione |
| Definizione Open API | https://github.com/awslabs/glue-extensions-for-iceberg/blob/main/glue-extensions-api.yaml #L273 |

Considerazioni

- Il parametro del percorso del catalogo deve seguire lo stile descritto nella sezione the section called “Parametri del percorso del catalogo e del prefisso”.

## Connessione al Catalogo dati da un'applicazione Spark standalone

È possibile connettersi al Catalogo dati da un'applicazione standalone usando un connettore Apache Iceberg.

1. Creazione di un ruolo IAM per l'applicazione Spark.
2. Connettiti all'endpoint AWS Glue Iceberg Rest utilizzando il connettore Iceberg.

```
# configure your application. Refer to https://docs.aws.amazon.com/cli/latest/
userguide/cli-configure-envvars.html for best practices on configuring environment
 variables.
export AWS_ACCESS_KEY_ID=$(aws configure get appUser.aws_access_key_id)
export AWS_SECRET_ACCESS_KEY=$(aws configure get appUser.aws_secret_access_key)
export AWS_SESSION_TOKEN=$(aws configure get appUser.aws_secret_token)

export AWS_REGION=us-east-1
export REGION=us-east-1
export AWS_ACCOUNT_ID = {specify your aws account id here}

~/spark-3.5.3-bin-hadoop3/bin/spark-shell \
    --packages org.apache.iceberg:iceberg-spark-runtime-3.4_2.12:1.6.0 \
    --conf
 "spark.sql.extensions=org.apache.iceberg.spark.extensions.IcebergSparkSessionExtensions"
 \
    --conf "spark.sql.defaultCatalog=spark_catalog" \
    --conf "spark.sql.catalog.spark_catalog=org.apache.iceberg.spark.SparkCatalog" \
    --conf "spark.sql.catalog.spark_catalog.type=rest" \
    --conf "spark.sql.catalog.spark_catalog.uri=https://glue.us-east-1.amazonaws.com/
iceberg" \
    --conf "spark.sql.catalog.spark_catalog.warehouse = {AWS_ACCOUNT_ID}" \
    --conf "spark.sql.catalog.spark_catalog.rest.sigv4-enabled=true" \
    --conf "spark.sql.catalog.spark_catalog.rest.signing-name=glue" \
    --conf "spark.sql.catalog.spark_catalog.rest.signing-region=us-east-1" \
    --conf "spark.sql.catalog.spark_catalog.io-
impl=org.apache.iceberg.aws.s3.S3FileIO" \
    --conf
 "spark.hadoop.fs.s3a.aws.credentials.provider=org.apache.hadoop.fs.s3a.SimpleAWSCredentialP
```

3. Esecuzione di query sui dati nel Catalogo dati.

```
spark.sql("create database myicebergdb").show()
spark.sql("""CREATE TABLE myicebergdb.mytbl (name string) USING iceberg location
 's3://bucket_name/mytbl'""")
spark.sql("insert into myicebergdb.mytbl values('demo') ").show()
```

# Mappatura di dati tra Amazon Redshift e Apache Iceberg

Redshift e Iceberg supportano diversi tipi di dati. La seguente matrice di compatibilità descrive il supporto e le limitazioni nella mappatura dei dati tra questi due sistemi. Consulta le specifiche dei tipi di dati di Amazon Redshift e della tabella di Apache Iceberg per maggiori dettagli sui tipi di dati supportati nei rispettivi sistemi.

| Tipi dati di Redshift | Alias | Tipo di dati di Iceberg |
|---|---|---|
| SMALLINT | INT2 | int |
| INTEGER | INT, INT4 | int |
| BIGINT | INT8 | Long |
| DECIMAL | NUMERIC | decimal |
| REAL | FLOAT4 | virgola mobile |
| REAL | FLOAT4 | virgola mobile |
| DOUBLE PRECISION | FLOAT8, GALLEGGIANTE | virgola mobile a doppia precisione |
| CHAR | CHARACTER, NCHAR | stringa |
| VARCHAR | CHARACTER VARYING, NVARCHAR | stringa |
| BPCHAR | | stringa |
| TEXT | | stringa |
| DATE | | data |
| TIME | TIME WITHOUT TIMEZONE | time |
| TIME | TIME WITH TIMEZONE | non supportato |
| TIMESTAMP | TIMESTAMP WITHOUT TIMEZONE | TIMESTAMP |

| Tipi dati di Redshift | Alias | Tipo di dati di Iceberg |
|---|---|---|
| TIMESTAMPZ | TIMESTAMP WITH TIMEZONE | TIMESTAMPZ |
| INTERVAL YEAR TO MONTH | | Non supportata |
| INTERVAL DAY TO SECOND | | Non supportata |
| BOOLEAN | BOOL | bool |
| HLLSKETCH | | Non supportata |
| SUPER | | Non supportata |
| VARBYTE | VARBINARY, BINARY VARYING | binary |
| GEOMETRY | | Non supportata |
| GEOGRAPHY | | Non supportata |

## Considerazioni e limitazioni sull'utilizzo di AWS Glue Iceberg REST Catalog APIs

Di seguito sono riportate le considerazioni e le limitazioni relative all'utilizzo del comportamento operativo DDL (Rest Catalog Data Definition Language) di Apache Iceberg.

Considerazioni

- Comportamento delle API **`RenameTable`**: l'operazione `RenameTable` è supportata nelle tabelle in Amazon Redshift ma non in Amazon S3.
- Operazioni DDL per namespace e tabelle in Amazon Redshift: le operazioni Crea, Aggiorna ed Elimina per namespace e tabelle in Amazon Redshift sono operazioni asincrone perché dipendono da quando è disponibile il gruppo di lavoro gestito di Amazon Redshift e dal fatto che sia in corso una transazione DDL e DML in conflitto; l'operazione deve attendere il blocco e quindi tentare di apportare modifiche.

Limitazioni

- Le visualizzazioni APIs nella specifica REST di Apache Iceberg non sono supportate nel AWS Glue catalogo REST di Iceberg.
- Limite di dimensione dei metadati delle tabelle Iceberg: AWS Glue Data Catalog supporta le tabelle Iceberg con una dimensione massima dei metadati di 20 MB. AWS Glue Data Catalog rifiuta le richieste alle tabelle con metadati che superano questo limite. Per scoprire tabelle con dimensioni dei AWS Glue Data Catalog metadati superiori a 20 MB, abilita la compattazione e la conservazione delle istantanee per le tabelle. Per ulteriori informazioni, consulta Ottimizzazione delle tabelle Iceberg.

# AWS Glue Procedure ottimali per Data Catalog

Questa sezione descrive le migliori pratiche per la gestione e l' AWS Glue Data Catalog utilizzo efficaci di. Sottolinea pratiche come l'uso efficiente dei crawler, l'organizzazione dei metadati, la sicurezza, l'ottimizzazione delle prestazioni, l'automazione, la governance dei dati e l'integrazione con altri servizi. AWS

- Usa i crawler in modo efficace: esegui i crawler regolarmente per mantenere il Data Catalog aggiornato sulle modifiche delle tue fonti di dati. up-to-date Utilizza il crawling incrementale per modificare frequentemente le origini dati e migliorare le prestazioni. Configura i crawler per aggiungere automaticamente nuove partizioni o aggiornare gli schemi quando vengono rilevate modifiche.
- Organizza e assegna un nome alle tabelle di metadati: si stabilisce una convenzione di denominazione coerente per database e tabelle in Catalogo dati. Raggruppa le origini dati correlate in database o cartelle logici per migliorare l'organizzazione. Utilizza nomi descrittivi che chiariscano lo scopo e il contenuto di ogni tabella.
- Gestisci gli schemi in modo efficace: sfrutta le funzionalità di inferenza degli schemi dei crawler. AWS Glue Rivede e aggiorna le modifiche allo schema prima di applicarle per evitare di interrompere le applicazioni downstream. Utilizza le funzionalità di evoluzione dello schema per gestire le modifiche allo schema in modo corretto.
- Protegge il Catalogo dati: abilita la crittografia dei dati inattivi e in transito per il Catalogo dati. Implementa policy di controllo degli accessi granulari per limitare l'accesso ai dati sensibili. Controlla e rivede regolarmente le autorizzazioni e i registri delle attività del Catalogo dati.
- Integrazione con altri AWS servizi Data Catalog Utilizza Data Catalog come livello di metadati centralizzato per servizi come Amazon Athena, Redshift Spectrum e. AWS Lake Formation Sfrutta

i processi AWS Glue ETL per trasformare e caricare i dati in vari datastore mantenendo i metadati nel Catalogo dati.

- Monitora e ottimizza le prestazioni Data Catalog Monitora le prestazioni dei crawler e dei lavori ETL utilizzando le metriche. Amazon CloudWatch Esegue il partizionamento dei set di dati di grandi dimensioni nel Catalogo dati per migliorare le prestazioni delle query. Implementa ottimizzazioni delle prestazioni per i metadati a cui si accede di frequente.
- Resta aggiornato con AWS Glue la documentazione e le migliori pratiche Data Catalog Controlla regolarmente la AWS Glue documentazione e AWS Glue le risorse per gli aggiornamenti, le migliori pratiche e i consigli più recenti. Partecipa a AWS Glue webinar, workshop e altri eventi per imparare dagli esperti e rimanere informato sulle nuove funzionalità e funzionalità.

# Monitoraggio delle metriche di utilizzo del Data Catalog in Amazon CloudWatch

AWS Glue Data Catalog le metriche di utilizzo sono ora disponibili con una soluzione Amazon CloudWatch che semplifica il monitoraggio e la comprensione dell'utilizzo delle risorse nel Data Catalog. Ora disponi di una visibilità immediata sull'uso di cataloghi, database, tabelle, partizioni e connessioni nell'API Glue Catalog, rendendo più facile mantenere la supervisione del tuo Catalogo dati.

## Panoramica delle metriche del Catalogo dati

AWS Glue Data Catalog pubblica automaticamente le metriche di utilizzo su. Amazon CloudWatch Con l'integrazione CloudWatch delle metriche, puoi tenere traccia degli indicatori di performance critici ogni minuto, tra cui:

- Richieste di tabelle
- Indici di partizione creati
- Connessioni aggiornate
- Statistiche aggiornate

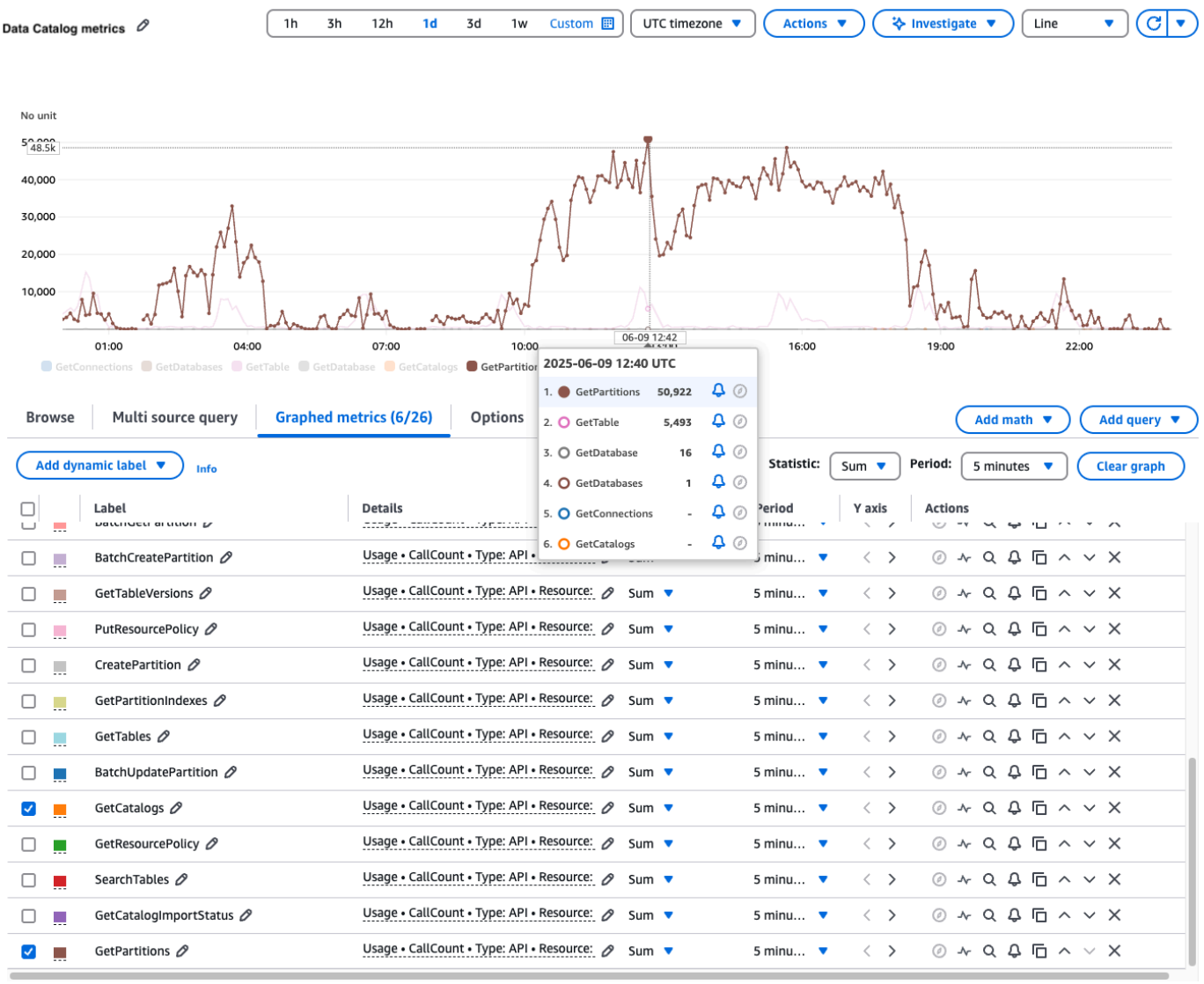
Queste metriche aiutano a identificare i colli di bottiglia, rilevare le anomalie e prendere decisioni basate sui dati per migliorare l'affidabilità complessiva del catalogo dati. Puoi anche impostare CloudWatch allarmi per ricevere notifiche quando le metriche superano le soglie specificate, consentendo una gestione proattiva dell'implementazione.

# Aggiungere metriche alla dashboard CloudWatch

Puoi creare dashboard personalizzate per monitorare AWS Glue Data Catalog le tue risorse e impostare allarmi per ricevere notifiche di qualsiasi attività insolita.

Puoi aggiungere le metriche di Data Catalog alla tua CloudWatch dashboard seguendo questi passaggi:

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, seleziona Parametri.
3. Seleziona Tutte le metriche.
4. Scegli  AWS Utilizzo>Per risorsa.
5. Filtra per Glue per visualizzare le metriche disponibili.
6. Seleziona i parametri che vuoi aggiungere al pannello di controllo.
7. Aggiungi metriche per cataloghi, database, tabelle, partizioni e connessioni al tuo grafico. CloudWatch




È possibile configurare allarmi personalizzati che si attivano automaticamente quando l'uso dell'API supera le soglie definite, al fine di identificare anomalie nell'utilizzo del catalogo dati.

Per istruzioni dettagliate sulla configurazione degli allarmi, consulta Creazione di un allarme Metrics Insights. CloudWatch

# Registro dello schema di AWS Glue

> **Note**
>
> AWS GlueIl registro degli schemi non è supportato nelle seguenti aree della AWS Glue console: Medio Oriente (Emirati Arabi Uniti).

Il registro dello schema di AWS Glue consente l'individuazione, il controllo e l'evoluzione in modo centralizzato degli schemi dei flussi dei dati. Uno schema definisce la struttura e il formato di un registro di dati. Con Registro dello schema AWS Glue è possibile gestire e applicare schemi sulle applicazioni di flussi di dati utilizzando comode integrazioni con Apache Kafka, Amazon Streaming gestito per Apache Kafka, Flusso di dati Amazon Kinesis, Servizio gestito da Amazon per Apache Flink e AWS Lambda.

Il registro degli schemi di supporta il formato dati AVRO (v1.11.4), il formato dati JSON con formato di schemi JSON per lo schema (specifiche Draft-04, Draft-06 e Draft-07) con la convalida dello schema JSON utilizzando la libreria Everit, le versioni proto2 e proto3 di Protocol Buffers (Protobuf) senza supporto per `extensions` o `groups` e il supporto Java, con altri linguaggi e formati di dati disponibili in futuro. Le caratteristiche supportate includono compatibilità, acquisizione dello schema tramite metadati, registrazione automatica degli schemi, compatibilità IAM e compressione ZLIB facoltativa per ridurre l'archiviazione e il trasferimento dei dati. Il registro degli schemi è serverless e gratuito.

L'utilizzo di uno schema come contratto di formato dati tra produttori e consumer comporta una migliore governance dei dati, dati di qualità superiore e consente ai consumer di dati di essere resilienti alle modifiche upstream compatibili.

Il registro degli schemi consente a sistemi diversi di condividere uno schema per la serializzazione e la deserializzazione. Ad esempio, si supponga di avere un produttore e un consumer di dati. Il produttore conosce lo schema quando pubblica i dati. Il registro degli schemi fornisce un serializzatore e un deserializzatore per alcuni sistemi come Amazon MSK o Apache Kafka.

Per ulteriori informazioni, consulta Funzionamento del registro degli schemi.

Argomenti

- Schemi
- Registri
- Controllo delle versioni e compatibilità degli schemi

- Librerie Serde open source
- Quote del registro degli schemi
- Funzionamento del registro degli schemi
- Guida introduttiva al registro degli schemi

## Schemi

Uno schema definisce la struttura e il formato di un registro di dati. Uno schema è una specifica con versioni per la pubblicazione, il consumo o l'archiviazione dei dati in modo affidabile.

In questo schema di esempio per Avro, il formato e la struttura sono definiti dal layout e dai nomi dei campi e il formato dei nomi dei campi è definito dai tipi di dati (ad esempio, `string`, `int`).

```
{
    "type": "record",
    "namespace": "ABC_Organization",
    "name": "Employee",
    "fields": [
        {
            "name": "Name",
            "type": "string"
        },
        {
            "name": "Age",
            "type": "int"
        },
        {
            "name": "address",
            "type": {
                "type": "record",
                "name": "addressRecord",
                "fields": [
                    {
                        "name": "street",
                        "type": "string"
                    },
                    {
                        "name": "zipcode",
                        "type": "int"
                    }
                ]
```

```
            }
        }
    ]
}
```

In questo schema JSON draft-07 per JSON di esempio, il formato è definito dall'organizzazione dello schema JSON.

```
{
 "$id": "https://example.com/person.schema.json",
 "$schema": "http://json-schema.org/draft-07/schema#",
 "title": "Person",
 "type": "object",
 "properties": {
  "firstName": {
   "type": "string",
   "description": "The person's first name."
  },
  "lastName": {
   "type": "string",
   "description": "The person's last name."
  },
  "age": {
   "description": "Age in years which must be equal to or greater than zero.",
   "type": "integer",
   "minimum": 0
  }
 }
}
```

In questo esempio per Protobuf, il formato è definito dalla versione 2 del linguaggio Protocol Buffers (proto2).

```
syntax = "proto2";

package tutorial;

option java_multiple_files = true;
option java_package = "com.example.tutorial.protos";
option java_outer_classname = "AddressBookProtos";

message Person {
```

```
  optional string name = 1;
  optional int32 id = 2;
  optional string email = 3;

  enum PhoneType {
    MOBILE = 0;
    HOME = 1;
    WORK = 2;
  }

  message PhoneNumber {
    optional string number = 1;
    optional PhoneType type = 2 [default = HOME];
  }

  repeated PhoneNumber phones = 4;
}

message AddressBook {
  repeated Person people = 1;
}
```

## Registri

Un registro è un container logico di schemi. I registri consentono di organizzare gli schemi e gestire il controllo degli accessi per le applicazioni. Un registro dispone di un Amazon Resource Name (ARN) che consente di organizzare e impostare diverse autorizzazioni di accesso per le operazioni dello schema all'interno del registro.

È possibile utilizzare il registro di default o creare tutti i nuovi registri necessari.

Gerarchia del registro degli schemi di AWS Glue

- RegistryName: [stringa]
  - RegistryArn: [AWS ARN]
  - CreatedTime: [timestamp]
  - UpdatedTime: [timestamp]
- SchemaName: [stringa]
  - SchemaArn: [AWS ARN]

  - DataFormat: [Avro, Json o Protobuf]
  - Compatibility: (ad es. BACKWARD, BACKWARD_ALL, FORWARD, FORWARD_ALL, FULL, FULL_ALL, NONE, DISABLED]
  - Status: [ad es. PENDING, AVAILABLE, DELETING]
  - SchemaCheckpoint: [numero intero]
  - CreatedTime: [timestamp]
  - UpdatedTime: [timestamp]
- SchemaVersion: [stringa]
  - SchemaVersionNumber: [numero intero]
  - Status: [ad es. PENDING, AVAILABLE, DELETING, FAILURE]
  - SchemaDefinition: [stringa, valore: JSON]
  - CreatedTime: [timestamp]
- SchemaVersionMetadata: [elenco]
  - MetadataKey: [stringa]
  - MetadataInfo
  - MetadataValue: [stringa]
  - CreatedTime: [timestamp]

## Controllo delle versioni e compatibilità degli schemi

Ogni schema può avere più versioni. Il controllo delle versioni è regolato da una regola di compatibilità applicata a uno schema. Le richieste di registrazione di nuove versioni dello schema vengono verificate in base a questa regola dal registro degli schemi prima che possano avere esito positivo.

Una versione dello schema contrassegnata come checkpoint viene utilizzata per determinare la compatibilità della registrazione di nuove versioni di uno schema. Quando uno schema viene creato per la prima volta, il checkpoint di default sarà la prima versione. Man mano che lo schema si evolve con più versioni, puoi utilizzare il CLI/SDK per modificare il checkpoint in una versione di uno schema utilizzando l'`UpdateSchema`API che aderisce a una serie di vincoli. Nella console, la modifica della definizione dello schema o della modalità di compatibilità modificherà il checkpoint alla versione più recente per impostazione predefinita.

Le modalità di compatibilità consentono di controllare come gli schemi possono o non possono evolvere nel tempo. Queste modalità costituiscono il contratto tra le applicazioni che producono e consumano dati. Quando una nuova versione di uno schema viene inviata al registro, la regola di compatibilità applicata al nome dello schema viene utilizzata per determinare se la nuova versione può essere accettata. Esistono 8 modalità di compatibilità: NONE, DISABLED, BACKWARD, BACKWARD_ALL, FORWARD, FORWARD_ALL, FULL, FULL_ALL.

Nel formato dati Avro, i campi possono essere facoltativi o obbligatori. Un campo facoltativo è un campo in cui il `Type` include null. I campi obbligatori non contengono il valore null come `Type`.

Nel formato dati Protobuf, i campi possono essere facoltativi (anche ripetuti) o obbligatori nella sintassi proto2, mentre tutti i campi sono facoltativi (anche ripetuti) nella sintassi proto3. Tutte le regole di compatibilità sono determinate in base alla comprensione delle specifiche di Protocol Buffers e dalle linee guida della Documentazione Google Protocol Buffers.

- NONE: non si applica alcuna modalità di compatibilità. È possibile utilizzare questa opzione negli scenari di sviluppo o se non si conoscono le modalità di compatibilità da applicare agli schemi. Qualsiasi nuova versione aggiunta sarà accettata senza essere sottoposta a un controllo di compatibilità.
- DISABLED: questa scelta di compatibilità impedisce il controllo delle versioni per uno schema specifico. Non è possibile aggiungere nuove versioni.
- BACKWARD: questa scelta di compatibilità è consigliata in quanto consente ai consumer di leggere sia la versione attuale che quella precedente dello schema. È possibile utilizzare questa opzione per verificare la compatibilità con la versione precedente dello schema quando si eliminano campi o si aggiungono campi facoltativi. Un tipico caso d'uso per BACKWARD è quando l'applicazione è stata creata per lo schema più recente.

  AVRO

  Ad esempio, si supponga di avere uno schema definito da nome (obbligatorio), cognome (obbligatorio), e-mail (obbligatorio) e numero di telefono (facoltativo).

  Se la versione successiva dello schema rimuove il campo e-mail obbligatorio, questa operazione verrebbe registrata correttamente. La compatibilità BACKWARD richiede ai consumer di essere in grado di leggere la versione corrente e precedente dello schema. I consumer potranno leggere il nuovo schema in quanto il campo dell'e-mail in più viene ignorato dai vecchi messaggi.

  Se si dispone di una nuova versione dello schema proposta che aggiunge un campo obbligatorio, ad esempio il codice postale, con la compatibilità BACKWARD questo non verrebbe registrato

correttamente. I consumer della nuova versione non sarebbero in grado di leggere i messaggi precedenti alla modifica dello schema, poiché mancherebbe il campo obbligatorio del codice postale. Tuttavia, se il campo del codice postale è impostato come facoltativo nel nuovo schema, la versione proposta viene registrata correttamente poiché i consumer sono in grado di leggere il vecchio schema senza il campo facoltativo del codice postale.

JSON

Ad esempio, si supponga di avere una versione dello schema definita da nome (facoltativo), cognome (facoltativo), e-mail (facoltativa) e numero di telefono (facoltativo).

Se nella versione successiva dello schema viene aggiunta la proprietà facoltativa del numero di telefono, la registrazione viene eseguita correttamente, a condizione che la versione originale dello schema non consenta proprietà aggiuntive impostando il campo `additionalProperties` su false. La compatibilità BACKWARD richiede ai consumer di essere in grado di leggere la versione corrente e precedente dello schema. I consumer saranno in grado di leggere i dati prodotti con lo schema originale in cui la proprietà numero di telefono non esiste.

Se si dispone di una nuova versione dello schema proposta che aggiunge la proprietà facoltativa del numero di telefono, questa operazione non viene registrata correttamente con la compatibilità BACKWARD se la versione originale dello schema imposta il campo `additionalProperties` su true, vale a dire consentendo qualsiasi proprietà aggiuntiva. I consumer della nuova versione non sarebbero in grado di leggere i messaggi precedenti alla modifica dello schema, in quanto non possono leggere i dati con la proprietà numero di telefono in un tipo differente, ad esempio come stringa anziché numero.

PROTOBUF

Ad esempio, supponiamo di avere una versione di uno schema definito da un messaggio `Person` con i campi `first name` (obbligatorio), `last name` (obbligatorio), `email` (obbligatorio), e `phone number` (facoltativo) sotto la sintassi proto2.

Come negli scenari AVRO, se la versione successiva dello schema rimuove il campo `email` obbligatorio, questa operazione verrà registrata correttamente. La compatibilità BACKWARD richiede ai consumer di essere in grado di leggere la versione corrente e precedente dello schema. I consumer potranno leggere il nuovo schema in quanto il campo `email` in più viene ignorato dai vecchi messaggi.

Se si dispone di una nuova versione dello schema proposta che aggiunge un campo obbligatorio, ad esempio `zip code`, con la compatibilità BACKWARD questo non verrebbe registrato correttamente. I consumer della nuova versione non sarebbero in grado di leggere i messaggi precedenti alla modifica dello schema, poiché mancherebbe il campo obbligatorio `zip code`. Tuttavia, se il campo `zip code` è impostato come facoltativo nel nuovo schema, la versione proposta viene registrata correttamente poiché i consumer sono in grado di leggere il vecchio schema senza il campo facoltativo `zip code`.

In caso di utilizzo di gRPC, l'aggiunta di un nuovo servizio RPC o metodo RPC è una modifica compatibile con le versioni precedenti. Ad esempio, supponiamo di avere una versione di uno schema definito da un servizio RPC `MyService` con due metodi RPC `Foo` e `Bar`.

Se la prossima versione dello schema aggiunge un nuovo metodo RPC chiamato `Baz`, questo si verrà registrato correttamente. I consumer saranno in grado di leggere i dati prodotti con lo schema originale in base alla compatibilità BACKWARD dal nuovo metodo RPC `Baz` è facoltativo.

Se si dispone di una nuova versione dello schema proposta che rimuove un campo obbligatorio, ad esempio il metodo RPC `Foo` esistente, con la compatibilità BACKWARD questo non verrebbe registrato correttamente. I consumer della nuova versione non sarebbero in grado di leggere i messaggi precedenti alla modifica dello schema, in quanto non possono comprendere e leggere i dati con il metodo RPC inesistente `Foo` in un'applicazione gRPC.

- BACKWARD_ALL:: questa scelta di compatibilità consente ai consumer di leggere sia la versione attuale che tutte quelle precedenti dello schema. È possibile utilizzare questa opzione per verificare la compatibilità con tutte le versioni precedenti dello schema quando si eliminano campi o si aggiungono campi facoltativi.
- FORWARD: questa scelta di compatibilità consente ai consumer di leggere sia la versione attuale che le versioni successive dello schema, ma non necessariamente le versioni più recenti. È possibile utilizzare questa opzione per verificare la compatibilità con l'ultima versione dello schema quando si aggiungono campi o si eliminano campi facoltativi. Un tipico caso d'uso per FORWARD è quando l'applicazione è stata creata per uno schema precedente e deve poter elaborare uno schema più recente.

  AVRO

  Ad esempio, si supponga di avere una versione di uno schema definito da nome (obbligatorio), cognome (obbligatorio), e-mail (facoltativo).

Se si dispone di una nuova versione dello schema che aggiunge un campo obbligatorio, ad esempio il numero di telefono, la registrazione viene eseguita correttamente. La compatibilità FORWARD richiede ai consumatori di essere in grado di leggere i dati prodotti con il nuovo schema utilizzando la versione precedente.

Se si dispone di una versione dello schema proposta che elimina il campo obbligatorio del nome, con la compatibilità BACKWARD questo non verrebbe registrato correttamente. I consumatori della versione precedente non sarebbero in grado di leggere gli schemi proposti in quanto mancherebbe il campo obbligatorio del nome. Tuttavia, se il campo del nome era originariamente facoltativo, il nuovo schema proposto verrebbe registrato correttamente poiché i consumatori potrebbero leggere i dati in base al nuovo schema che non dispone del campo facoltativo del nome.

JSON

Ad esempio, si supponga di avere una versione dello schema definita da nome (facoltativo), cognome (facoltativo), e-mail (facoltativa) e numero di telefono (facoltativo).

Se si dispone di una nuova versione dello schema in cui viene rimossa la proprietà facoltativa del numero di telefono, la registrazione viene eseguita correttamente, a condizione che la nuova versione dello schema non consenta proprietà aggiuntive impostando il campo `additionalProperties` su false. La compatibilità FORWARD richiede ai consumatori di essere in grado di leggere i dati prodotti con il nuovo schema utilizzando la versione precedente.

Se si dispone di una versione dello schema proposta che elimina la proprietà facoltativa del numero di telefono, questa operazione non viene registrata correttamente con la compatibilità FORWARD se la nuova versione dello schema imposta il campo `additionalProperties` su true, vale a dire consentendo qualsiasi proprietà aggiuntiva. I consumer della versione precedente non sarebbero in grado di leggere lo schema proposto, in quanto potrebbero disporre della proprietà numero di telefono in un tipo differente, ad esempio come stringa anziché numero.

PROTOBUF

Ad esempio, supponiamo di avere una versione di uno schema definito da un messaggio `Person` con i campi `first name` (obbligatorio), `last name` (obbligatorio), `email` (facoltativo) sotto la sintassi proto2.

Se si dispone di una nuova versione dello schema che aggiunge un campo obbligatorio, ad esempio `phone number`, la registrazione viene eseguita correttamente. La compatibilità

FORWARD richiede ai consumatori di essere in grado di leggere i dati prodotti con il nuovo schema utilizzando la versione precedente.

Se si dispone di una versione dello schema proposta che elimina il campo obbligatorio `first name`, con la compatibilità BACKWARD questo non verrebbe registrato correttamente. I consumatori della versione precedente non sarebbero in grado di leggere gli schemi proposti in quanto mancherebbe il campo obbligatorio `first name`. Tuttavia, se il campo `first name` era originariamente facoltativo, il nuovo schema proposto verrebbe registrato correttamente poiché i consumer potrebbero leggere i dati in base al nuovo schema che non dispone del campo facoltativo `first name`.

In caso di utilizzo di gRPC, la rimozione di un nuovo servizio RPC o metodo RPC è una modifica compatibile con le versioni future. Ad esempio, supponiamo di avere una versione di uno schema definito da un servizio RPC `MyService` con due metodi RPC `Foo` e `Bar`.

Se la versione successiva dello schema elimina il metodo RPC esistente denominato `Foo`, si registrerà correttamente in base alla compatibilità FORWARD in quanto i consumer possono leggere i dati prodotti con il nuovo schema utilizzando la versione precedente. Se si dispone di una versione dello schema proposta che elimina il campo obbligatorio `Baz`, con la compatibilità FORWARD questo non verrà registrato correttamente. I consumer della versione precedente non sarebbero in grado di leggere gli schemi proposti in quanto mancherebbe il metodo RPC `Baz`.

- FORWARD_ALL: questa scelta di compatibilità consente ai consumer di leggere dati scritti dai produttori di qualsiasi nuovo schema registrato. È possibile utilizzare questa opzione quando è necessario aggiungere campi o eliminare campi facoltativi e verificare la compatibilità con tutte le versioni precedenti dello schema.
- FULL: questa scelta di compatibilità consente consumer di leggere i dati scritti dai produttori utilizzando la versione precedente o successiva dello schema, ma non versioni precedenti o successive. È possibile utilizzare questa opzione per verificare la compatibilità con l'ultima versione dello schema quando si aggiungono o si eliminano campi facoltativi.
- FULL_ALL: questa scelta di compatibilità consente ai consumer di leggere i dati scritti dai produttori utilizzando tutte le versioni precedenti dello schema. È possibile utilizzare questa opzione per verificare la compatibilità con tutte le versioni precedenti dello schema quando si aggiungono o si eliminano campi facoltativi.

## Librerie Serde open source

AWS fornisce librerie Serde open source come framework per la serializzazione e la deserializzazione dei dati. La progettazione open source di queste librerie permette alle applicazioni e ai framework open source comuni di supportare queste librerie nei loro progetti.

Per ulteriori dettagli sul funzionamento delle librerie Serde, consulta Funzionamento del registro degli schemi.

## Quote del registro degli schemi

Le quote, note anche come limiti in, sono i valori massimi per le risorse AWS, le azioni e gli elementi presenti nell'account. AWS Di seguito sono riportati i limiti flessibili per il registro degli schemi di AWS Glue.

Coppia chiave-valore dei metadati della versione dello schema

Puoi avere fino a 10 coppie chiave-valore per regione. SchemaVersion AWS

È possibile visualizzare o impostare le coppie di metadati chiave-valore utilizzando o. QuerySchemaVersionMetadata azione (Python: query_schema_version_metadata) PutSchemaVersionMetadata azione (Python: put_schema_version_metadata) APIs

Di seguito sono riportati i limiti rigidi per il registro degli schemi di AWS Glue.

Registri

Puoi avere fino a 100 registri per AWS regione per questo account.

SchemaVersion

Puoi avere fino a 10000 versioni dello schema per AWS regione per questo account.

Ogni nuovo schema crea una nuova versione dello schema, quindi in teoria è possibile avere fino a 10.000 schemi per account per ciascuna regione, se ogni schema ha una sola versione.

Payload dello schema

Esiste un limite di dimensioni pari a 170 KB per i payload dello schema.

# Funzionamento del registro degli schemi

Questa sezione descrive il funzionamento dei processi di serializzazione e deserializzazione nel registro degli schemi.

1. Registrare uno schema: se lo schema non esiste già nel registro, può essere registrato con un nome uguale al nome della destinazione (ad esempio, test_topic, test_stream, prod_firehose) oppure il produttore può fornire un nome personalizzato per lo schema. I produttori possono anche aggiungere coppie chiave-valore allo schema come metadati, ad esempio source: MSK_Kafka_topic_A, o applicare tag agli schemi durante la creazione dello schema. AWS Una volta registrato uno schema, il registro degli schemi restituisce l'ID della versione dello schema al serializzatore. Se lo schema esiste ma il serializzatore utilizza una nuova versione non esistente, il registro degli schemi controllerà il riferimento allo schema con una regola di compatibilità per garantire che la nuova versione sia compatibile prima di registrarla.

   Esistono due metodi per registrare uno schema: registrazione manuale e registrazione automatica. È possibile registrare uno schema manualmente tramite la console AWS Glue o la CLI/l'SDK.

   Quando la registrazione automatica è attivata nelle impostazioni del serializzatore, lo schema verrà registrato automaticamente. Se `REGISTRY_NAME` non viene fornito nelle configurazioni del produttore, la registrazione automatica registrerà la nuova versione dello schema nel registro predefinito (default-registry). Consulta Installazione delle SerDe librerie per informazioni su come specificare la proprietà di registrazione automatica.
2. Il serializzatore convalida i record di dati rispetto allo schema: quando l'applicazione che produce i dati ha registrato il proprio schema, il serializzatore del registro degli schemi convalida il record prodotto dall'applicazione, strutturato con i campi e i tipi di dati corrispondenti a uno schema registrato. Se lo schema del record non corrisponde a uno schema registrato, il serializzatore restituirà un'eccezione e l'applicazione non riuscirà a consegnare il record alla destinazione.

   Se non esiste uno schema e se il nome dello schema non viene fornito tramite le configurazioni del produttore, lo schema viene creato con lo stesso nome dell'argomento (se Apache Kafka o Amazon MSK) o del flusso (se Kinesis Data Streams).

   Ogni record comprende dati e una definizione dello schema. Viene eseguita una query sulla definizione dello schema in base agli schemi e alle versioni esistenti nel registro degli schemi.

   Per impostazione predefinita, i produttori memorizzano nella cache le definizioni degli schemi e la versione dello schema registrati. IDs Se la definizione della versione dello schema di un

record non corrisponde a ciò che è disponibile nella cache, il produttore tenterà di convalidare lo schema con il registro degli schemi. Se la versione dello schema è valida, l'ID della versione e la definizione verranno memorizzati nella cache locale del produttore.

È possibile regolare il periodo di cache di default (24 ore) all'interno delle proprietà del produttore facoltative nel passaggio 3 di Installazione delle SerDe librerie.

3. Serializza e distribuisci i record: se il record è conforme allo schema, il serializzatore aggiunge a ogni record l'ID della versione dello schema, serializza il record in base al formato di dati selezionato (AVRO, JSON o Protobuf e altri formati disponibili a breve), comprime il record (configurazione opzionale del produttore) e lo consegna alla destinazione.
4. I consumer deserializzano i dati: i consumer che leggono questi dati utilizzano la libreria del deserializzatore del registro degli schemi che analizza l'ID della versione dello schema dal payload del record.
5. Il deserializzatore può richiedere lo schema dal registro degli schemi: se il deserializzatore vede i record con un particolare ID della versione dello schema per la prima volta, richiederà lo schema dal registro degli schemi utilizzando questo ID e memorizzerà lo schema nella cache locale del consumer. Se il registro degli schemi non è in grado di deserializzare il record, il consumer può registrare i dati dal record e passare oltre o arrestare l'applicazione.
6. Il deserializzatore utilizza lo schema per deserializzare il record: quando il deserializzatore recupera l'ID della versione dello schema dal registro degli schemi, decomprime il record (se il record inviato dal produttore è compresso) e utilizza lo schema per deserializzarlo. L'applicazione elabora il record.

> **Note**
>
> Crittografia: i client comunicano con il registro degli schemi tramite chiamate API che crittografano i dati in transito utilizzando la crittografia TLS su HTTPS. Gli schemi archiviati nel registro degli schemi sono sempre crittografati quando sono inattivi utilizzando una chiave gestita dal servizio () AWS Key Management Service .AWS KMS

> **Note**
>
> Autorizzazione utente: il registro degli schemi supporta le policy IAM basati sull'identità.

# Guida introduttiva al registro degli schemi

Le sezioni seguenti offrono una panoramica e illustrano la configurazione e l'uso del registro degli schemi. Per informazioni su concetti e componenti del registro degli schemi, consultare Registro dello schema di AWS Glue.

Argomenti


- Installazione delle SerDe librerie
- Integrazione con il registro degli schemi di AWS Glue
- Migrazione da un registro degli schemi di terze parti al registro degli schemi di AWS Glue


## Installazione delle SerDe librerie

Le SerDe librerie forniscono un framework per la serializzazione e la deserializzazione dei dati.

Installerai il serializzatore open source per le applicazioni che producono dati (collettivamente i "serializzatori"). Il serializzatore gestisce la serializzazione, la compressione e l'interazione con il registro degli schemi. Il serializzatore estrae automaticamente lo schema da un record in fase di scrittura in una destinazione compatibile con il registro degli schemi, ad esempio Amazon MSK. Allo stesso modo, installerai il deserializzatore open source sulle applicazioni che consumano dati.

implementazione Java

> **Note**
>
> Prerequisiti: prima di completare i passaggi riportati di seguito, dovrai disporre di un cluster in esecuzione Amazon Streaming gestito per Apache Kafka (Amazon MSK) o Apache Kafka. I produttori e i consumer devono essere in esecuzione su Java 8 o versione successiva.

Per installare le librerie su produttori e consumer:

1. All'interno dei file pom.xml dei produttori e dei consumer, aggiungi questa dipendenza tramite il codice qui sotto:

```
<dependency>
    <groupId>software.amazon.glue</groupId>
```

```
    <artifactId>schema-registry-serde</artifactId>
    <version>1.1.5</version>
</dependency>
```

In alternativa, è possibile clonare il repository Github del registro degli schemi di AWS Glue.

2. Imposta i tuoi produttori con le seguenti proprietà obbligatorie:

```
props.put(ProducerConfig.KEY_SERIALIZER_CLASS_CONFIG,
 StringSerializer.class.getName()); // Can replace StringSerializer.class.getName())
 with any other key serializer that you may use
props.put(ProducerConfig.VALUE_SERIALIZER_CLASS_CONFIG,
 GlueSchemaRegistryKafkaSerializer.class.getName());
props.put(AWSSchemaRegistryConstants.AWS_REGION, "us-east-2");
properties.put(AWSSchemaRegistryConstants.DATA_FORMAT, "JSON"); // OR "AVRO"
```

Se non esistono schemi, è necessario attivare la registrazione automatica (passaggio successivo). Se si dispone di uno schema da applicare, sostituire "my-schema" con il nome dello schema. Se la registrazione automatica dello schema è disattivata deve essere fornito anche "registry-name". Se lo schema viene creato sotto "default-registry", il nome del registro può essere omesso.

3. (Facoltativo) Impostare una di queste proprietà facoltative del produttore. Per descrizioni dettagliate delle proprietà, consultate il ReadMe file.

```
props.put(AWSSchemaRegistryConstants.SCHEMA_AUTO_REGISTRATION_SETTING, "true"); // If
 not passed, uses "false"
props.put(AWSSchemaRegistryConstants.SCHEMA_NAME, "my-schema"); // If not passed,
 uses transport name (topic name in case of Kafka, or stream name in case of Kinesis
 Data Streams)
props.put(AWSSchemaRegistryConstants.REGISTRY_NAME, "my-registry"); // If not passed,
 uses "default-registry"
props.put(AWSSchemaRegistryConstants.CACHE_TIME_TO_LIVE_MILLIS, "86400000"); // If
 not passed, uses 86400000 (24 Hours)
props.put(AWSSchemaRegistryConstants.CACHE_SIZE, "10"); // default value is 200
props.put(AWSSchemaRegistryConstants.COMPATIBILITY_SETTING, Compatibility.FULL); //
 Pass a compatibility mode. If not passed, uses Compatibility.BACKWARD
props.put(AWSSchemaRegistryConstants.DESCRIPTION, "This registry is used for several
 purposes."); // If not passed, constructs a description
props.put(AWSSchemaRegistryConstants.COMPRESSION_TYPE,
 AWSSchemaRegistryConstants.COMPRESSION.ZLIB); // If not passed, records are sent
 uncompressed
```

La registrazione automatica registra la versione dello schema nel registro di default ("default-registry"). Se nel passaggio precedente non è stato specificato un SCHEMA_NAME, il nome dell'argomento viene dedotto come SCHEMA_NAME.

Per ulteriori informazioni sulle modalità di compatibilità, consulta Controllo delle versioni e compatibilità degli schemi.

4. Imposta i tuoi consumer con le seguenti proprietà obbligatorie:

```
props.put(ConsumerConfig.KEY_DESERIALIZER_CLASS_CONFIG,
 StringDeserializer.class.getName());
props.put(ConsumerConfig.VALUE_DESERIALIZER_CLASS_CONFIG,
 GlueSchemaRegistryKafkaDeserializer.class.getName());
props.put(AWSSchemaRegistryConstants.AWS_REGION, "us-east-2"); // Pass an Regione AWS
props.put(AWSSchemaRegistryConstants.AVRO_RECORD_TYPE,
 AvroRecordType.GENERIC_RECORD.getName()); // Only required for AVRO data format
```

5. (Facoltativo) Imposta queste proprietà facoltative del consumer. Per descrizioni dettagliate delle proprietà, consultate il ReadMe file.

```
properties.put(AWSSchemaRegistryConstants.CACHE_TIME_TO_LIVE_MILLIS, "86400000"); //
 If not passed, uses 86400000
props.put(AWSSchemaRegistryConstants.CACHE_SIZE, "10"); // default value is 200
props.put(AWSSchemaRegistryConstants.SECONDARY_DESERIALIZER,
 "com.amazonaws.services.schemaregistry.deserializers.external.ThirdPartyDeserializer"); //
 For migration fall back scenario
```

Implementazione C#

> **Note**
>
> Prerequisiti: prima di completare i passaggi riportati di seguito, dovrai disporre di un cluster in esecuzione Amazon Streaming gestito per Apache Kafka (Amazon MSK) o Apache Kafka. I produttori e i consumatori devono essere in esecuzione su.NET 8.0 o versioni successive.

Installazione

Per le applicazioni C#, installa il SerDe NuGet pacchetto AWS Glue Schema Registry utilizzando uno dei seguenti metodi:

.NET CLI:

Usa il seguente comando per installare il pacchetto:

```
dotnet add package Aws.Glue.SchemaRegistry --version 1.0.0-<rid>
```

dove <rid> potrebbe essere`1.0.0-linux-x64`, `1.0.0-linux-musl-x64` oppure `1.0.0-linux-arm64`

PackageReference (nel tuo file.csproj):

Aggiungi quanto segue al tuo file di progetto:

```
<PackageReference Include="Aws.Glue.SchemaRegistry" Version="1.0.0-<rid>" />
```

dove <rid> potrebbe essere`1.0.0-linux-x64`, `1.0.0-linux-musl-x64` oppure `1.0.0-linux-arm64`

Configurazione del file di configurazione

Crea un file delle proprietà di configurazione (ad esempio`gsr-config.properties`) con le impostazioni richieste:

Configurazione minima:

Di seguito viene illustrato un esempio di configurazione minima:

```
region=us-east-1
registry.name=default-registry
dataFormat=AVRO
schemaAutoRegistrationEnabled=true
```

Utilizzo della libreria client C# Glue Schema per Kafka SerDes

Esempio di utilizzo del serializzatore:

L'esempio seguente mostra come utilizzare il serializzatore:

```
private static readonly string PROTOBUF_CONFIG_PATH = "<PATH_TO_CONFIG_FILE>";
var protobufSerializer = new GlueSchemaRegistryKafkaSerializer(PROTOBUF_CONFIG_PATH);
var serialized = protobufSerializer.Serialize(message, message.Descriptor.FullName);
// send serialized bytes to Kafka using producer.Produce(serialized)
```

Esempio di utilizzo del deserializzatore:

L'esempio seguente mostra come utilizzare il deserializzatore:

```
private static readonly string PROTOBUF_CONFIG_PATH = "<PATH_TO_CONFIG_FILE>";
var dataConfig = new GlueSchemaRegistryDataFormatConfiguration(
    new Dictionary<string, dynamic>
    {
        {
            GlueSchemaRegistryConstants.ProtobufMessageDescriptor, message.Descriptor
        }
    }
);
var protobufDeserializer = new
 GlueSchemaRegistryKafkaDeserializer(PROTOBUF_CONFIG_PATH, dataConfig);

// read message from Kafka using serialized = consumer.Consume()
var deserializedObject = protobufDeserializer.Deserialize(message.Descriptor.FullName,
 serialized);
```

Utilizzo della libreria client C# Glue Schema con for KafkaFlow SerDes

Esempio di utilizzo del serializzatore:

L'esempio seguente mostra come configurare KafkaFlow con il serializzatore:

```
services.AddKafka(kafka => kafka
    .UseConsoleLog()
    .AddCluster(cluster => cluster
        .WithBrokers(new[] { "localhost:9092" })
        .AddProducer<CustomerProducer>(producer => producer
            .DefaultTopic("customer-events")
            .AddMiddlewares(m => m

 .AddSerializer<GlueSchemaRegistryKafkaFlowProtobufSerializer<Customer>>(
                    () => new
 GlueSchemaRegistryKafkaFlowProtobufSerializer<Customer>("config/gsr-
config.properties")
                )
            )
        )
    )
);
```

Esempio di utilizzo del deserializzatore:

L'esempio seguente mostra come configurare KafkaFlow con il deserializzatore:

```
.AddConsumer(consumer => consumer
    .Topic("customer-events")
    .WithGroupId("customer-group")
    .WithBufferSize(100)
    .WithWorkersCount(10)
    .AddMiddlewares(middlewares => middlewares
        .AddDeserializer<GlueSchemaRegistryKafkaFlowProtobufDeserializer<Customer>>(
            () => new
 GlueSchemaRegistryKafkaFlowProtobufDeserializer<Customer>("config/gsr-
config.properties")
        )
        .AddTypedHandlers(h => h.AddHandler<CustomerHandler>())
    )
)
```

Proprietà opzionali del produttore

È possibile estendere il file di configurazione con proprietà opzionali aggiuntive:

```
# Auto-registration (if not passed, uses "false")
schemaAutoRegistrationEnabled=true

# Schema name (if not passed, uses topic name)
schema.name=my-schema

# Registry name (if not passed, uses "default-registry")
registry.name=my-registry

# Cache settings
cacheTimeToLiveMillis=86400000
cacheSize=200

# Compatibility mode (if not passed, uses BACKWARD)
compatibility=FULL

# Registry description
description=This registry is used for several purposes.

# Compression (if not passed, records are sent uncompressed)
```

```
compressionType=ZLIB
```

Formati di dati supportati

Entrambe le implementazioni Java e C# supportano gli stessi formati di dati:

- AVRO: formato binario Apache Avro
- JSON: formato dello schema JSON
- PROTOBUF: formato Protocol Buffer

Note

- Per iniziare a usare la libreria, visita https://www.nuget. org/packages/AWS.Colla. SchemaRegistry
- Il codice sorgente è disponibile all'indirizzo: https://github.com/awslabs/aws-glue-schema-registry

Creazione di un registro

È possibile utilizzare il registro predefinito o creare tutti i nuovi registri necessari utilizzando la console AWS Glue APIs oAWS Glue.

AWS Glue APIs

È possibile utilizzare questi passaggi per eseguire questa operazione utilizzando il. AWS Glue APIs

Per utilizzare il AWS CLI registro degli AWS Glue schemi APIs, assicurati di aggiornare il tuo AWS CLI alla versione più recente.

Per aggiungere un nuovo registro, utilizza l'API CreateRegistry azione (Python: create_registry). Specifica `RegistryName` come nome del registro da creare, con una lunghezza massima di 255 caratteri e può contenere solo lettere, numeri, trattini, trattini bassi, simboli del dollaro o cancelletti.

Specificare una `Description` come una stringa non superiore a 2048 byte di lunghezza, corrispondente al modello di stringa su più righe dell'indirizzo URI.

Facoltativamente, puoi specificare uno o più `Tags` per il registro, come una matrice di mappe di coppie chiave-valore.

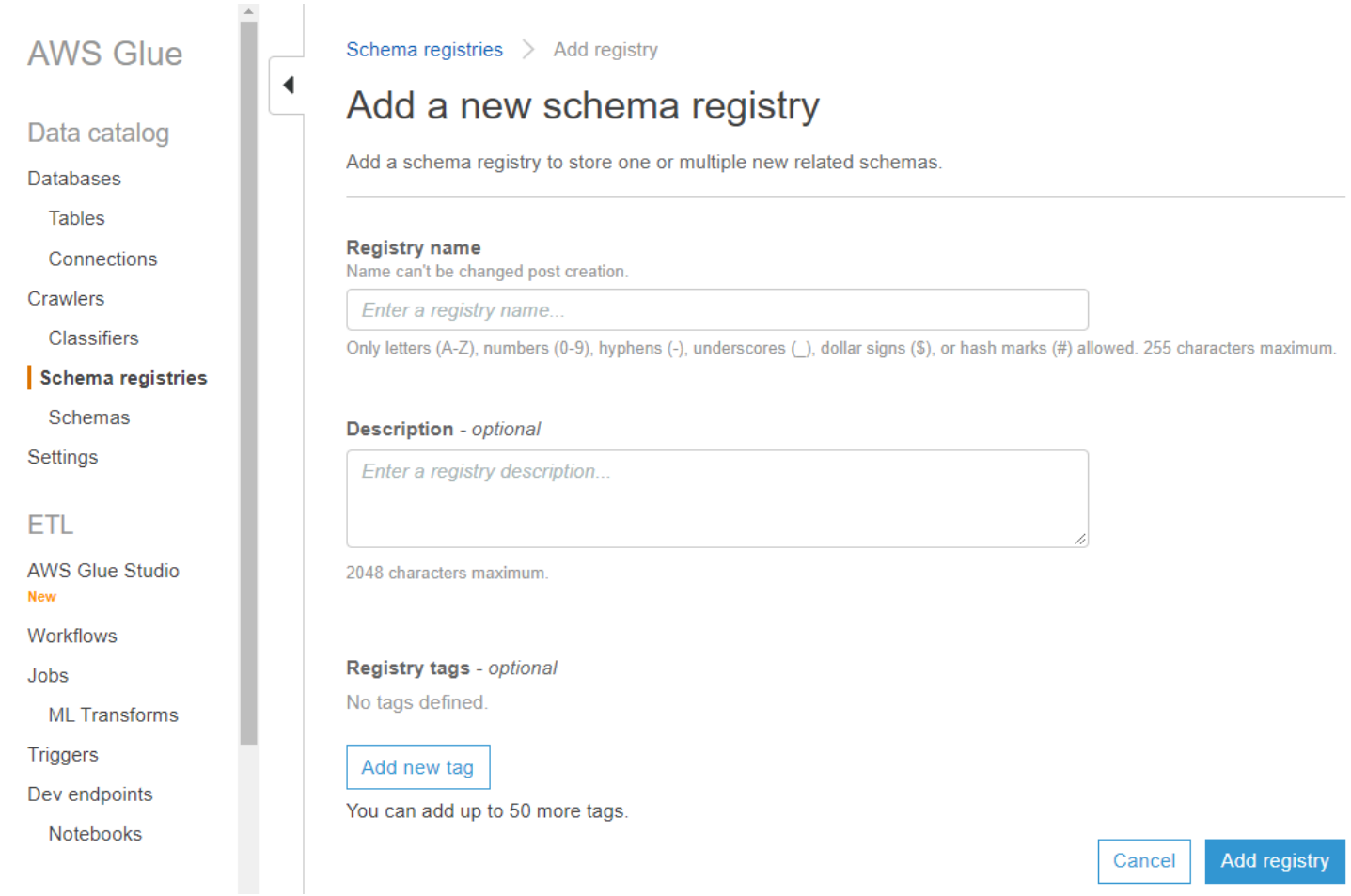
```
aws glue create-registry --registry-name registryName1 --description description
```

Al momento della creazione del registro, gli viene assegnato un Amazon Resource Name (ARN) che puoi visualizzare nel `RegistryArn` della risposta API. Dopo aver creato un registro, crea uno o più schemi per questo registro.

Console AWS Glue

Per aggiungere un nuovo registro nella console AWS Glue:

1. Accedi a Console di gestione AWS e apri la AWS Glue console all'indirizzo https://console.aws.amazon.com/glue/.
2. Nel pannello di navigazione, in Data catalog (Catalogo dati), seleziona Schema registries (Registri degli schemi).
3. Scegli Add registry (Aggiungi registro).
4. Inserisci un Registry name (Nome registro) per il registro, composto da lettere, numeri, trattini o trattini bassi. Questo nome non può essere modificato.
5. Inserisci una Description (Descrizione) (facoltativo) per il registro.
6. Facoltativamente, applica uno o più tag al registro. Seleziona Add new tag (Aggiungi nuovo tag) e immetti una Tag key (Chiave di tag) e facoltativamente un Tag value (Valore di tag).
7. Scegli Add registry (Aggiungi registro).




Al momento della creazione del registro, gli viene assegnato un Amazon Resource Name (ARN) che puoi visualizzare selezionando il registro dall'elenco in Schema registries (Registri degli schemi). Dopo aver creato un registro, crea uno o più schemi per questo registro.

Utilizzo di un record specifico (JAVA POJO) per JSON

È possibile utilizzare un POJO (Plain Old Java Object) e passare l'oggetto come record. È simile alla nozione di un record specifico in AVRO. [mbknor-jackson-jsonschema](mbknor-jackson-jsonschema)Può generare uno schema JSON per il POJO passato. Questa libreria può anche inserire informazioni aggiuntive nello schema JSON.

La libreria del registro degli schemi di AWS Glue utilizza il campo "className" inserito nello schema per fornire un nome di classe completamente classificato. Il campo "className" viene utilizzato dal deserializzatore per eseguire la deserializzazione in un oggetto di quella classe.

```
 Example class :

@JsonSchemaDescription("This is a car")
```

```
@JsonSchemaTitle("Simple Car Schema")
@Builder
@AllArgsConstructor
@EqualsAndHashCode
// Fully qualified class name to be added to an additionally injected property
// called className for deserializer to determine which class to deserialize
// the bytes into
@JsonSchemaInject(
        strings = {@JsonSchemaString(path = "className",
                value =
 "com.amazonaws.services.schemaregistry.integrationtests.generators.Car")}
)
// List of annotations to help infer JSON Schema are defined by https://github.com/
mbknor/mbknor-jackson-jsonSchema
public class Car {
    @JsonProperty(required = true)
    private String make;

    @JsonProperty(required = true)
    private String model;

    @JsonSchemaDefault("true")
    @JsonProperty
    public boolean used;

    @JsonSchemaInject(ints = {@JsonSchemaInt(path = "multipleOf", value = 1000)})
    @Max(200000)
    @JsonProperty
    private int miles;

    @Min(2000)
    @JsonProperty
    private int year;

    @JsonProperty
    private Date purchaseDate;

    @JsonProperty
    @JsonFormat(shape = JsonFormat.Shape.NUMBER)
    private Date listedDate;

    @JsonProperty
    private String[] owners;
```

```
    @JsonProperty
    private Collection<Float> serviceChecks;

    // Empty constructor is required by Jackson to deserialize bytes
    // into an Object of this class
    public Car() {}
}
```

Creazione di uno schema

È possibile creare uno schema utilizzando la AWS Glue APIs o la AWS Glue console.

AWS Glue APIs

È possibile utilizzare questi passaggi per eseguire questa operazione utilizzando il AWS Glue APIs.

Per aggiungere un nuovo schema, utilizza l'API CreateSchema azione (Python: create_schema).

Specifica una struttura `RegistryId` per indicare un registro per lo schema. Oppure, ometti il `RegistryId` per utilizzare il registro di default.

Specifica un `SchemaName` composto da lettere, numeri, trattini o trattini bassi e `DataFormat` come **`AVRO`** o **`JSON`**. Una volta impostato su uno schema, `DataFormat` non è modificabile.

Specifica una modalità di `Compatibility`:

- Backward (consigliato): il consumer può leggere sia la versione attuale che quella precedente.
- Backward all: il consumer può leggere la versione attuale e tutte quelle precedenti.
- Forward: il consumer può leggere sia la versione attuale che quella successiva.
- Forward all: il consumer può leggere sia la versione attuale che tutte quelle successive.
- Full: combinazione di Backward e Forward.
- Full all: combinazione di Backward all e Forward all.
- None: non vengono eseguiti controlli di compatibilità.
- Disabled: impedisce il controllo delle versioni per questo schema.

Facoltativamente, specifica i `Tags` per lo schema.

Specifica un valore di `SchemaDefinition` per definire lo schema in formato dati Avro, JSON o Protobuf. Consulta questi esempi.

Per il formato dati Avro:

```
aws glue create-schema --registry-id RegistryName="registryName1" --schema-name
 testschema --compatibility NONE --data-format AVRO --schema-definition "{\"type\":
 \"record\", \"name\": \"r1\", \"fields\": [ {\"name\": \"f1\", \"type\": \"int\"},
 {\"name\": \"f2\", \"type\": \"string\"} ]}"
```

```
aws glue create-schema --registry-id RegistryArn="arn:aws:glue:us-
east-2:901234567890:registry/registryName1" --schema-name testschema --compatibility
 NONE --data-format AVRO  --schema-definition "{\"type\": \"record\", \"name\": \"r1\",
 \"fields\": [ {\"name\": \"f1\", \"type\": \"int\"}, {\"name\": \"f2\", \"type\":
 \"string\"} ]}"
```

Per il formato dati JSON:

```
aws glue create-schema --registry-id RegistryName="registryName" --schema-name
 testSchemaJson --compatibility NONE --data-format JSON --schema-definition "{\"$schema
\": \"http://json-schema.org/draft-07/schema#\",\"type\":\"object\",\"properties\":
{\"f1\":{\"type\":\"string\"}}}"
```

```
aws glue create-schema --registry-id RegistryArn="arn:aws:glue:us-
east-2:901234567890:registry/registryName" --schema-name testSchemaJson --compatibility
 NONE --data-format JSON --schema-definition "{\"$schema\": \"http://json-schema.org/
draft-07/schema#\",\"type\":\"object\",\"properties\":{\"f1\":{\"type\":\"string\"}}}"
```

Per il formato dati Protobuf:

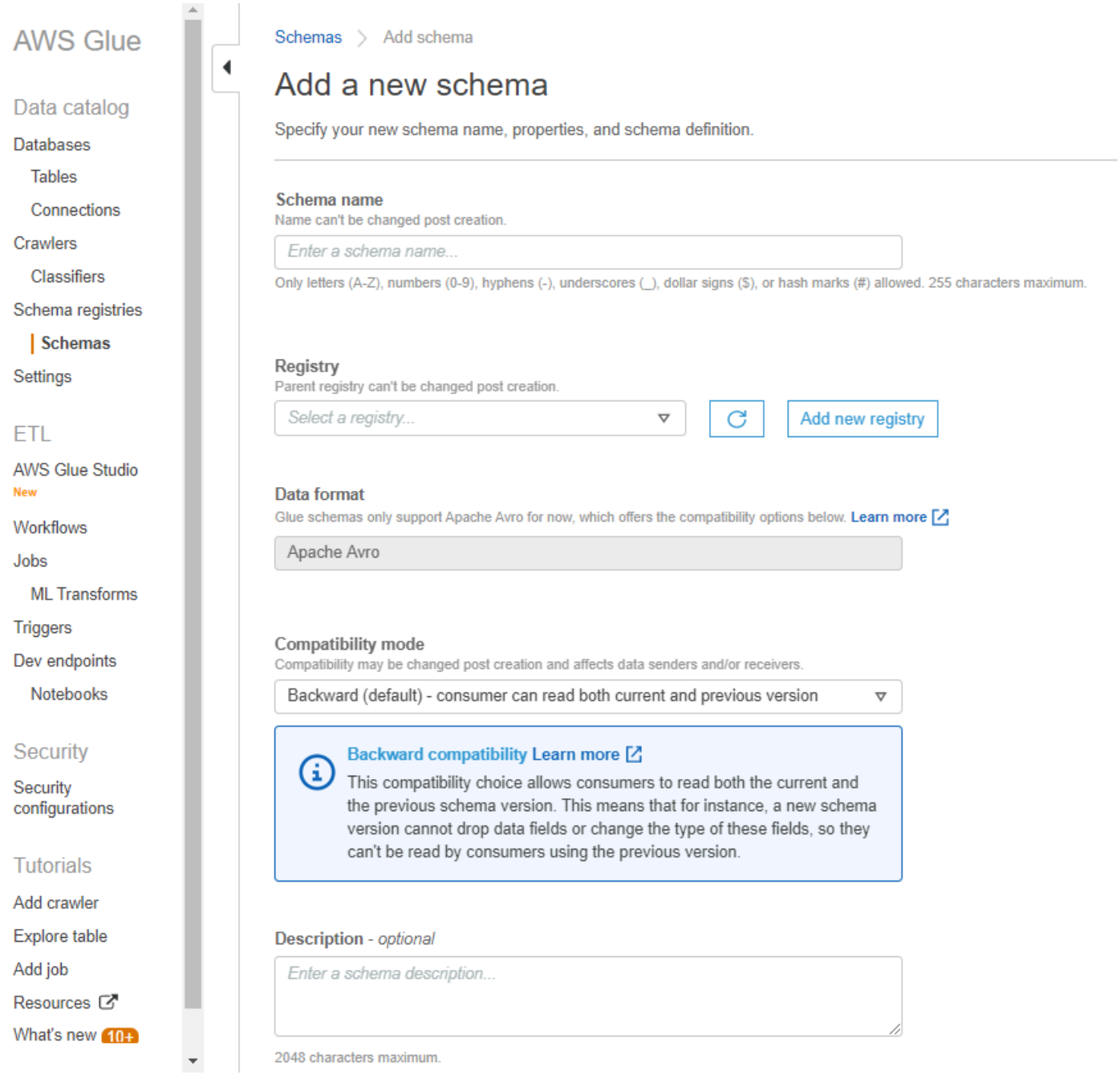
```
aws glue create-schema --registry-id RegistryName="registryName" --schema-name
 testSchemaProtobuf --compatibility NONE --data-format PROTOBUF --schema-definition
 "syntax = \"proto2\";package org.test;message Basic { optional int32 basic = 1;}"
```

```
aws glue create-schema --registry-id RegistryArn="arn:aws:glue:us-
east-2:901234567890:registry/registryName" --schema-name testSchemaProtobuf
 --compatibility NONE --data-format PROTOBUF --schema-definition "syntax =
 \"proto2\";package org.test;message Basic { optional int32 basic = 1;}"
```

Console AWS Glue

Per aggiungere un nuovo schema utilizzando la console AWS Glue:

1. Accedi alla console di AWS gestione e apri la AWS Glue console all'indirizzo https://console.aws.amazon.com/glue/.
2. Nel pannello di navigazione, in Data catalog (Catalogo dati), seleziona Schema (Schemi).
3. Seleziona Aggiungi schema (Aggiungi schema).
4. Inserisci uno Schema name (Nome schema) composto da lettere, numeri, trattini, trattini bassi, simboli di dollaro o cancelletti. Questo nome non può essere modificato.
5. Seleziona il registro in cui lo schema verrà archiviato dal menu a discesa. Il registro padre non può essere modificato dopo la creazione.
6. Lascia Data format (Formato dei dati) come Apache Avro o JSON. Questo formato si applica a tutte le versioni di questo schema.
7. Scegli una Compatibility mode (Modalità Compatibilità).
   - Backward (consigliato): il ricevitore può leggere sia la versione attuale che quella precedente.
   - Backward all: il ricevitore può leggere la versione attuale e tutte quelle precedenti.
   - Forward: il mittente può scrivere sia la versione attuale che quelle precedenti.
   - Forward All: il mittente può scrivere sia la versione attuale che tutte quelle precedenti.
   - Full: combinazione di Backward e Forward.
   - Full All: combinazione di Backward All e Forward All.
   - None: non vengono eseguiti controlli di compatibilità.
   - Disabled: impedisce il controllo delle versioni per questo schema.
8. Immetti un a Description (Descrizione) facoltativa per il registro con un massimo di 250 caratteri.

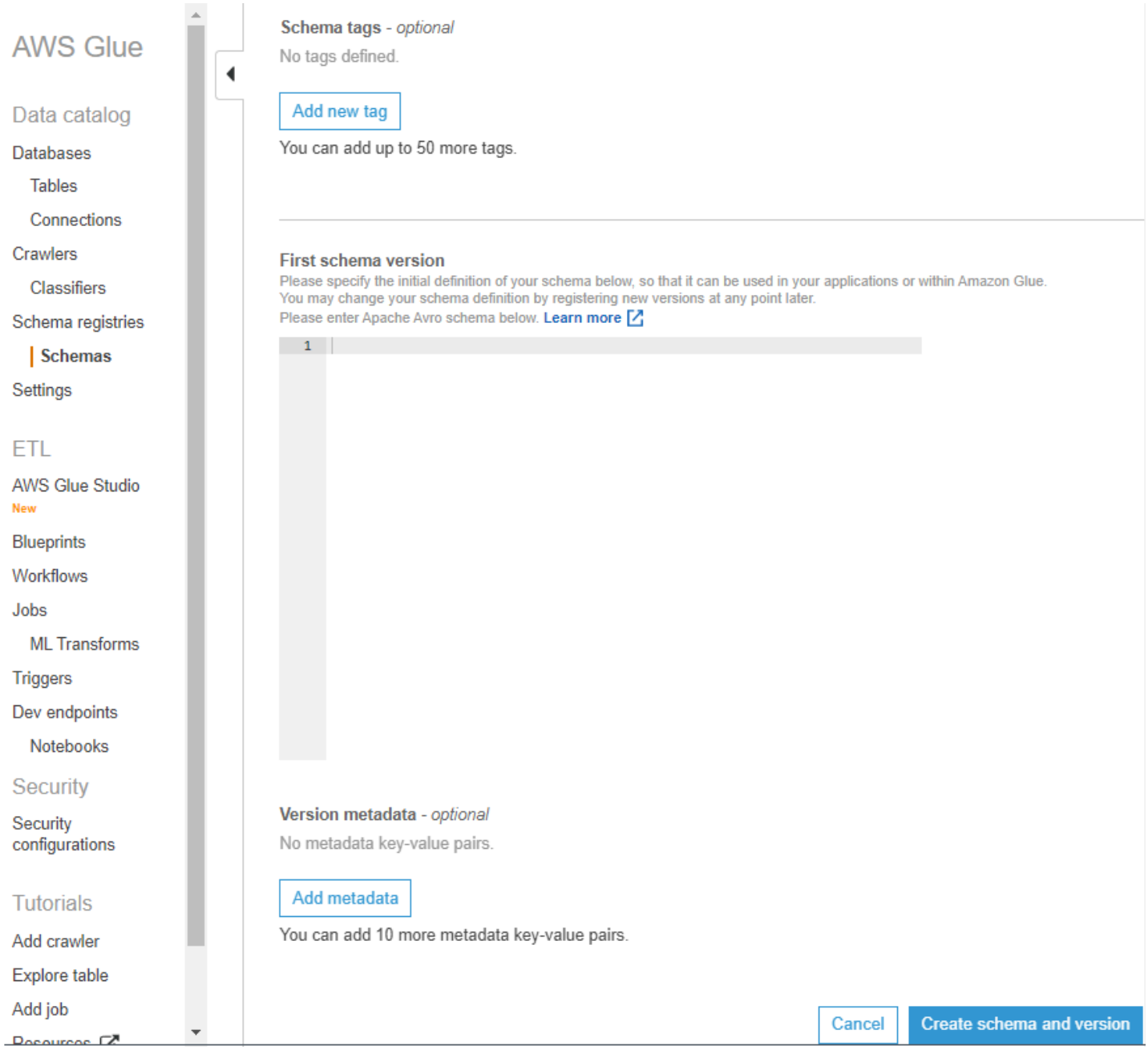



9. Facoltativamente, applica uno o più tag allo schema. Seleziona Add new tag (Aggiungi nuovo tag) e immetti una Tag key (Chiave di tag) e facoltativamente un Tag value (Valore di tag).
10. Immetti o incolla lo schema iniziale nella casella First schema version (Prima versione dello schema).

    Per il formato Avro, consulta Utilizzare il formato di dati Avro

    Per il formato JSON, consulta Utilizzare il formato di dati JSON

11. Facoltativamente, scegli Add metadata (Aggiungi metadata) per aggiungere metadati di versione per annotare o classificare la versione dello schema.
12. Scegli Create schema and version (Crea schema e versione).




Lo schema viene creato e viene visualizzato nell'elenco sotto Schemas (Schemi).

Utilizzare il formato di dati Avro

Avro fornisce servizi di serializzazione dei dati e scambio di dati. Avro memorizza la definizione dei dati in formato JSON semplificando la lettura e l'interpretazione. I dati stessi sono memorizzati in formato binario.

Per informazioni sulla definizione di uno schema Apache Avro, consulta la specifica di Apache Avro.

Utilizzare il formato di dati JSON

I dati possono essere serializzati con il formato JSON. Il formato di schemi JSON definisce lo standard per il formato di schemi JSON.

Aggiornamento di uno schema o di un registro

Una volta creati, è possibile modificare gli schemi, le versioni degli schemi o il registro.

Aggiornamento di un registro

È possibile aggiornare un registro utilizzando AWS Glue APIs o la AWS Glue console. Il nome di un registro esistente non può essere modificato. È possibile modificare la descrizione di un registro.

AWS Glue APIs

Per aggiornare un registro esistente, utilizza l'API UpdateRegistry azione (Python: update_registry).

Specifica una struttura `RegistryId` per indicare il registro da aggiornare. Passa una `Description` per modificare la descrizione di un registro.

```
aws glue update-registry --description updatedDescription --registry-id
 RegistryArn="arn:aws:glue:us-east-2:901234567890:registry/registryName1"
```

Console AWS Glue

Per aggiornare un registro utilizzando la console AWS Glue:

1. Accedi a Console di gestione AWS e apri la AWS Glue console all'indirizzo https://console.aws.amazon.com/glue/.
2. Nel pannello di navigazione, in Data catalog (Catalogo dati), seleziona Schema registries (Registri degli schemi).
3. Scegli un registro dall'elenco dei registri selezionando la relativa casella.
4. Dal menu Action (Operazioni), seleziona Edit registry (Modifica registro).

Aggiornamento di uno schema

È possibile aggiornare la descrizione o l'impostazione di compatibilità per uno schema.

Per aggiornare uno schema esistente, utilizza l'API UpdateSchema azione (Python: update_schema).

Specifica una struttura `SchemaId` per indicare lo schema da aggiornare. Deve essere fornito `VersionNumber` o `Compatibility`.

Esempio di codice 11:

```
aws glue update-schema --description testDescription --schema-id
 SchemaName="testSchema1",RegistryName="registryName1" --schema-version-number
 LatestVersion=true --compatibility NONE
```

```
aws glue update-schema --description testDescription --schema-id
 SchemaArn="arn:aws:glue:us-east-2:901234567890:schema/registryName1/testSchema1" --
schema-version-number LatestVersion=true --compatibility NONE
```

Aggiunta di una versione dello schema

Quando si aggiunge una versione dello schema, è necessario confrontare le versioni per assicurarsi che il nuovo schema venga accettato.

Per aggiungere una nuova versione a uno schema esistente, utilizza l'API RegisterSchemaVersion azione (Python: register_schema_version).

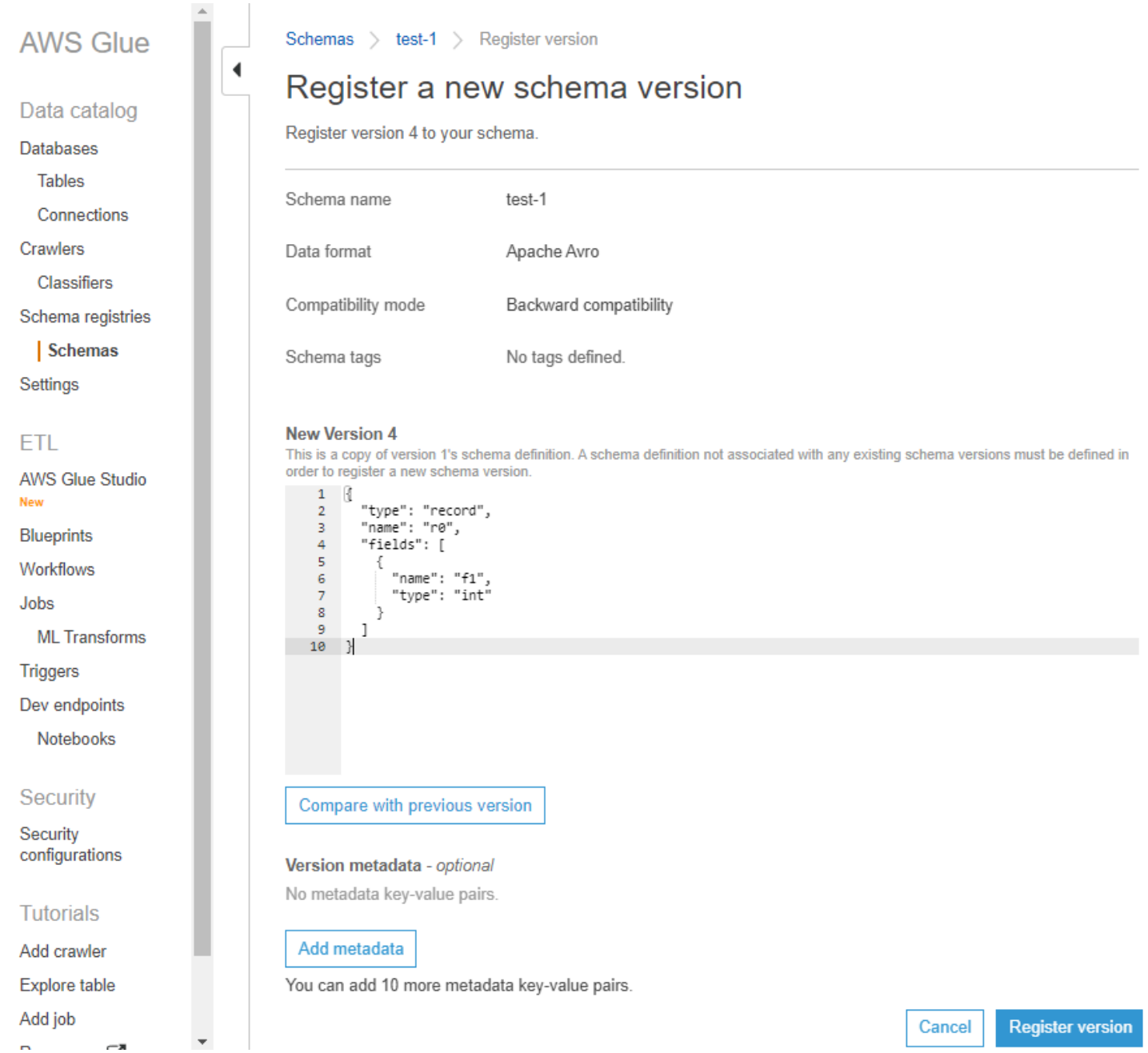
Specifica una struttura `SchemaId` per indicare lo schema per il quale si desidera aggiungere una versione e un valore di `SchemaDefinition` per definire lo schema.

Esempio di codice 12:

```
aws glue register-schema-version --schema-definition "{\"type\": \"record\", \"name\":
 \"r1\", \"fields\": [ {\"name\": \"f1\", \"type\": \"int\"}, {\"name\": \"f2\", \"type
\": \"string\"} ]}" --schema-id SchemaArn="arn:aws:glue:us-east-1:901234567890:schema/
registryName/testschema"
```

```
aws glue register-schema-version --schema-definition "{\"type\": \"record\", \"name\":
 \"r1\", \"fields\": [ {\"name\": \"f1\", \"type\": \"int\"}, {\"name\": \"f2\", \"type
\": \"string\"} ]}" --schema-id SchemaName="testschema",RegistryName="testregistry"
```

1. Accedi a Console di gestione AWS e apri la AWS Glue console all'indirizzo https://console.aws.amazon.com/glue/.
2. Nel pannello di navigazione, in Data catalog (Catalogo dati), seleziona Schema (Schemi).
3. Scegli lo schema dall'elenco degli schemi selezionando la relativa casella.
4. Seleziona uno o più schemi dall'elenco selezionando le caselle.
5. Nel menu Action (Operazioni), seleziona Register new version (Registra nuova versione).
6. Nella casella New version (Nuova versione), immetti o incolla il nuovo schema.
7. Seleziona Compare with previous version (Confronta con la versione precedente) per visualizzare le differenze con la versione precedente dello schema.
8. Facoltativamente, scegli Add metadata (Aggiungi metadata) per aggiungere metadati di versione per annotare o classificare la versione dello schema. Inserisci Key (Chiave) e facoltativamente Value (Valore).
9. Scegli Register version (Registra versione).



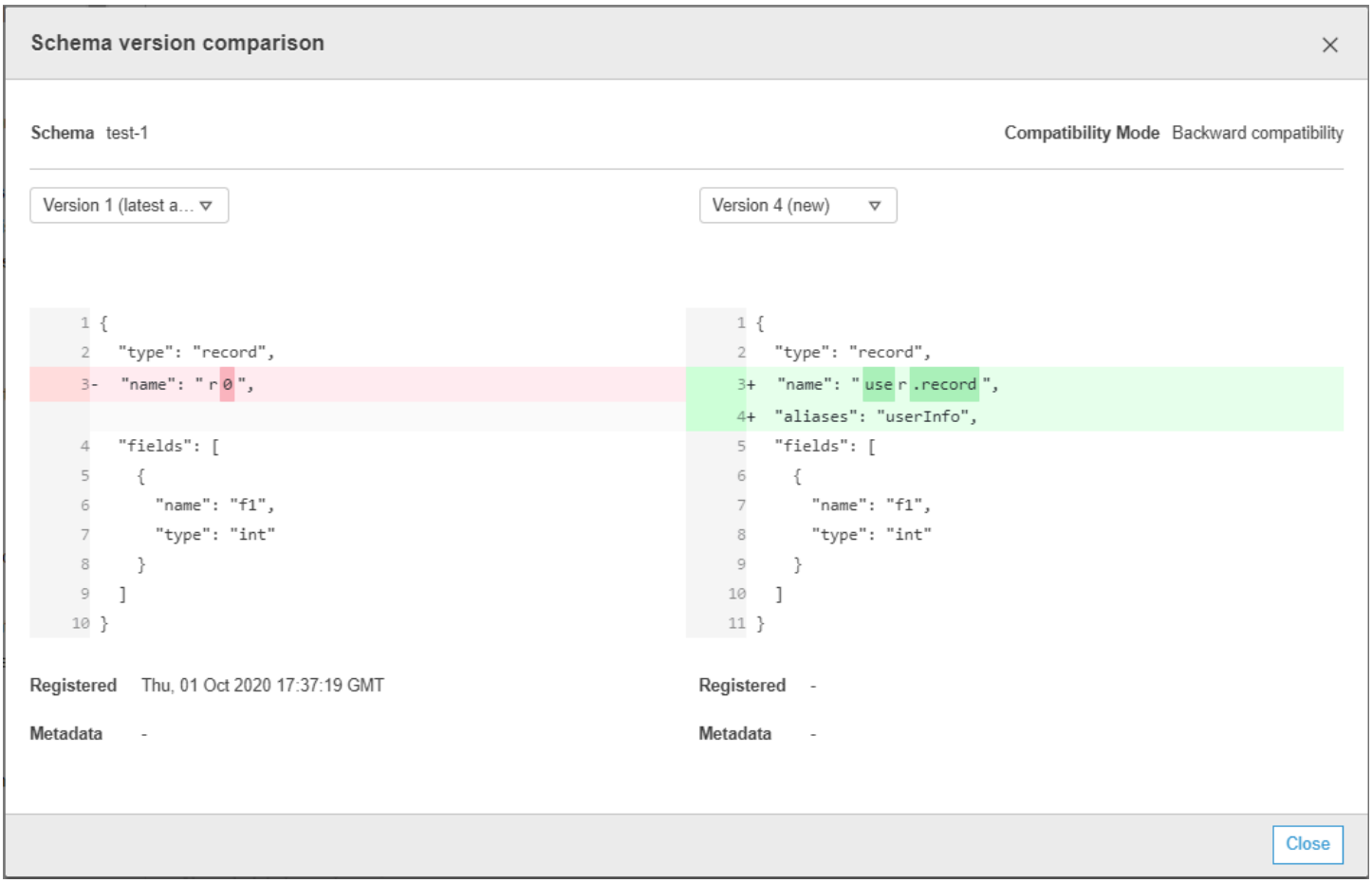

La versione degli schemi viene visualizzata nell'elenco delle versioni. Se la versione ha modificato la modalità di compatibilità, la versione verrà contrassegnata come checkpoint.

Esempio di confronto tra le versioni di uno schema

Selezionando Compare with previous version (Confronta con la versione precedente), le versioni precedenti e quelle nuove verranno mostrate insieme. Le informazioni modificate saranno evidenziate come segue:

- Giallo: indica le informazioni modificate.
- Verde: indica il contenuto aggiunto nella versione più recente.
- Rosso: indica il contenuto rimosso nella versione più recente.

È possibile eseguire il confronto anche con le versioni precedenti.




Eliminazione di uno schema o di un registro

L'eliminazione di uno schema, di una versione dello schema o di un registro è un'azione permanente che non può essere annullata.

Eliminazione di uno schema

Potresti voler eliminare uno schema quando non verrà più utilizzato all'interno di un registro, utilizzando o l'DeleteSchema azione (Python: delete_schema)API. Console di gestione AWS

L'eliminazione di uno o più schemi è un'azione permanente che non può essere annullata. Accertati che lo schema o gli schemi non siano più necessari.

Per eliminare uno schema dal registro, chiama l'API DeleteSchema azione (Python: delete_schema), specificando la struttura `SchemaId` per identificare lo schema.

Ad esempio:

```
aws glue delete-schema --schema-id SchemaArn="arn:aws:glue:us-
east-2:901234567890:schema/registryName1/schemaname"
```

```
aws glue delete-schema --schema-id SchemaName="TestSchema6-
deleteschemabyname",RegistryName="default-registry"
```

Console AWS Glue

Per eliminare uno schema dalla console AWS Glue:

1. Accedi a Console di gestione AWS e apri la AWS Glue console all'indirizzo https://console.aws.amazon.com/glue/.
2. Nel pannello di navigazione, in Data catalog (Catalogo dati), seleziona Schema registries (Registri degli schemi).
3. Scegli il registro che contiene lo schema dall'elenco dei registri.
4. Seleziona uno o più schemi dall'elenco selezionando le caselle.
5. Dal menu Action (Operazioni), scegli Delete schema (Elimina schema).
6. Inserisci il testo **`Delete`** nel campo per confermare l'eliminazione.
7. Seleziona Delete (Elimina).

Gli schemi specificati vengono eliminati dal registro.

Eliminazione di una versione dello schema

Man mano che gli schemi si accumulano nel registro, potresti voler eliminare le versioni indesiderate dello schema utilizzando Console di gestione AWS, o l'DeleteSchemaVersions azione (Python: delete_schema_versions)API. L'eliminazione di una o più versioni degli schemi è un'azione permanente che non può essere annullata. Accertati che le versioni degli schemi non siano più necessarie.

Durante l'eliminazione delle versioni degli schemi, tieni presente i seguenti vincoli:

- Non è possibile eliminare una versione con segno di spunta.

- L'intervallo di versioni contigue non può essere superiore a 25.
- La versione più recente non deve trovarsi nello stato in sospeso.

Specifica la struttura `SchemaId` per identificare lo schema e specifica `Versions` come intervallo di versioni da eliminare. Per ulteriori informazioni sulla specifica di una versione o un intervallo di versioni, consulta DeleteRegistry azione (Python: delete_registry). Le versioni degli schemi specificate vengono eliminate dal registro.

Chiamare l'API ListSchemaVersions azione (Python: list_schema_versions) dopo questa chiamata elencherà lo stato delle versioni eliminate.

Esempio:

```
aws glue delete-schema-versions --schema-id
 SchemaName="TestSchema6",RegistryName="default-registry" --versions "1-1"
```

```
aws glue delete-schema-versions --schema-id SchemaArn="arn:aws:glue:us-
east-2:901234567890:schema/default-registry/TestSchema6-NON-Existent" --versions "1-1"
```

1. Accedi a Console di gestione AWS e apri la AWS Glue console all'indirizzo. https://console.aws.amazon.com/glue/
2. Nel pannello di navigazione, in Data catalog (Catalogo dati), seleziona Schema registries (Registri degli schemi).
3. Scegli il registro che contiene lo schema dall'elenco dei registri.
4. Seleziona uno o più schemi dall'elenco selezionando le caselle.
5. Dal menu Action (Operazioni), scegli Delete schema (Elimina schema).
6. Inserisci il testo **`Delete`** nel campo per confermare l'eliminazione.
7. Seleziona Delete (Elimina).

Le versioni degli schemi specificate vengono eliminate dal registro.

Eliminazione di un registro

È possibile eliminare un registro quando gli schemi in esso contenuti non devono più essere organizzati in tale registro. Sarà necessario riassegnare tali schemi a un altro registro.

L'eliminazione di uno o più registri è un'azione permanente che non può essere annullata. Accertati che il registro o i registri non siano più necessari.

Il registro predefinito può essere eliminato utilizzando AWS CLI.

API AWS Glue

Per eliminare l'intero registro, inclusi gli schemi e tutte le relative versioni, chiama l'API DeleteRegistry azione (Python: delete_registry). Specifica una struttura `RegistryId` per identificare lo schema.

Ad esempio:

```
aws glue delete-registry --registry-id RegistryArn="arn:aws:glue:us-
east-2:901234567890:registry/registryName1"
```

```
aws glue delete-registry --registry-id RegistryName="TestRegistry-deletebyname"
```

Per ottenere lo stato dell'operazione di eliminazione, è possibile chiamare l'API `GetRegistry` dopo la chiamata asincrona.

Console AWS Glue

Per eliminare un registro dalla console AWS Glue:

1. Accedi a Console di gestione AWS e apri la AWS Glue console all'indirizzo https://console.aws.amazon.com/glue/.
2. Nel pannello di navigazione, in Data catalog (Catalogo dati), seleziona Schema registries (Registri degli schemi).
3. Scegli un registro dall'elenco selezionando una casella.
4. Dal menu Action (Operazioni), seleziona Delete registry (Elimina registro).
5. Inserisci il testo **`Delete`** nel campo per confermare l'eliminazione.
6. Seleziona Delete (Elimina).

I registri selezionati vengono eliminati da AWS Glue.

Esempi di IAM per i serializzatori

> **Note**
>
> AWS le politiche gestite concedono le autorizzazioni necessarie per i casi d'uso comuni. Per informazioni sull'utilizzo delle policy gestite per gestire il registro degli schemi, consulta AWS politiche gestite (predefinite) per AWS Glue.

Per i serializzatori, è necessario creare una policy minima simile a quella riportata di seguito per avere la possibilità di trovare lo `schemaVersionId` per una determinata definizione dello schema. Nota, per leggere gli schemi nel registro è necessario disporre delle autorizzazioni di lettura. È possibile limitare i registri che possono essere letti utilizzando la clausola `Resource`.

Esempio di codice 13:

```
{
    "Sid" : "GetSchemaByDefinition",
    "Effect" : "Allow",
    "Action" :
 [
        "glue:GetSchemaByDefinition"
    ],
        "Resource" : ["arn:aws:glue:us-east-2:012345678:registry/registryname-1",
                      "arn:aws:glue:us-east-2:012345678:schema/registryname-1/
schemaname-1",
                      "arn:aws:glue:us-east-2:012345678:schema/registryname-1/
schemaname-2"
                     ]
}
```

Inoltre, è possibile consentire ai produttori di creare nuovi schemi e versioni includendo i seguenti metodi aggiuntivi. Nota, dovresti essere in grado di ispezionare il registro per verificare gli schemi add/remove/evolve al suo interno. È possibile limitare i registri che possono essere ispezionati utilizzando la clausola `Resource`.

Esempio di codice 14:

```
{
    "Sid" : "RegisterSchemaWithMetadata",
```

```
    "Effect" : "Allow",
    "Action" :
 [
        "glue:GetSchemaByDefinition",
        "glue:CreateSchema",
        "glue:RegisterSchemaVersion",
        "glue:PutSchemaVersionMetadata",
    ],
    "Resource" : ["arn:aws:glue:aws-region:123456789012:registry/registryname-1",
                  "arn:aws:glue:aws-region:123456789012:schema/registryname-1/
schemaname-1",
                  "arn:aws:glue:aws-region:123456789012:schema/registryname-1/
schemaname-2"
                ]
}
```

Esempi di IAM per i deserializzatori

Per i deserializzatori (lato consumer), è necessario creare una policy simile a quella riportata di seguito per consentire al deserializzatore di recuperare lo schema dal registro degli schemi per la deserializzazione. Nota, devi essere in grado di ispezionare il registro per recuperare gli schemi al suo interno.

Esempio di codice 15:

```
{
    "Sid" : "GetSchemaVersion",
    "Effect" : "Allow",
    "Action" :
 [
        "glue:GetSchemaVersion"
    ],
    "Resource" : ["*"]
}
```

Connettività privata tramite AWS PrivateLink

Puoi utilizzarlo AWS PrivateLink per connettere il VPC del tuo produttore di dati AWS Glue definendo un'interfaccia per l'endpoint VPC. AWS Glue Quando utilizzi un endpoint VPC di interfaccia, la comunicazione tra il VPC e AWS Glue avviene completamente all'interno della rete AWS . Per ulteriori informazioni, consulta la pagina relativa all'utilizzo di AWS Glue con endpoint VPC.

Accesso ai CloudWatch parametri di Amazon

CloudWatch I parametri di Amazon sono disponibili come parte CloudWatch del piano gratuito. Puoi accedere a queste metriche nella CloudWatch console. Le metriche a livello di API includono CreateSchema (Successo e latenza), (Successo e latenza) GetSchemaByDefinition, (Successo e latenza), GetSchemaVersion (Successo e latenza), RegisterSchemaVersion (Successo e latenza). PutSchemaVersionMetadata Le metriche a livello di risorsa includono Registry. ThrottledByLimit, SchemaVersion. ThrottledByLimit, SchemaVersion .Taglia.

CloudFormation Modello di esempio per il registro degli schemi

Di seguito è riportato un modello di esempio per la creazione di risorse del registro degli schemi in CloudFormation. Per creare questo stack nell'account, copia il modello sopra in un file `SampleTemplate.yaml` ed esegui il comando seguente:

```
aws cloudformation create-stack --stack-name ABCSchemaRegistryStack --template-body
 "'cat SampleTemplate.yaml'"
```

Questo esempio usa `AWS::Glue::Registry` per creare un registro, `AWS::Glue::Schema` per creare uno schema, `AWS::Glue::SchemaVersion`per creare una versione dello schema e `AWS::Glue::SchemaVersionMetadata` per popolare i metadati della versione dello schema.

```
Description: "A sample CloudFormation template for creating Schema Registry resources."
Resources:
  ABCRegistry:
    Type: "AWS::Glue::Registry"
    Properties:
      Name: "ABCSchemaRegistry"
      Description: "ABC Corp. Schema Registry"
      Tags:
        Project: "Foo"
  ABCSchema:
    Type: "AWS::Glue::Schema"
    Properties:
      Registry:
        Arn: !Ref ABCRegistry
      Name: "TestSchema"
      Compatibility: "NONE"
      DataFormat: "AVRO"
      SchemaDefinition: >
        {"namespace":"foo.avro","type":"record","name":"user","fields":
[{"name":"name","type":"string"},{"name":"favorite_number","type":"int"}]}
```

```
      Tags:
        Project: "Foo"
  SecondSchemaVersion:
    Type: "AWS::Glue::SchemaVersion"
    Properties:
      Schema:
        SchemaArn: !Ref ABCSchema
      SchemaDefinition: >
        {"namespace":"foo.avro","type":"record","name":"user","fields":
[{"name":"status","type":"string", "default":"ON"}, {"name":"name","type":"string"},
{"name":"favorite_number","type":"int"}]}
  FirstSchemaVersionMetadata:
    Type: "AWS::Glue::SchemaVersionMetadata"
    Properties:
      SchemaVersionId: !GetAtt ABCSchema.InitialSchemaVersionId
      Key: "Application"
      Value: "Kinesis"
  SecondSchemaVersionMetadata:
    Type: "AWS::Glue::SchemaVersionMetadata"
    Properties:
      SchemaVersionId: !Ref SecondSchemaVersion
      Key: "Application"
      Value: "Kinesis"
```

## Integrazione con il registro degli schemi di AWS Glue

Queste sezioni descrivono le integrazioni con il registro degli schemi di AWS Glue. Gli esempi riportati in questa sezione mostrano uno schema con formato di dati AVRO. Per altri esempi, inclusi schemi con formato di dati JSON, consulta i test di integrazione e ReadMe le informazioni nel repository open source di Schema Registry. AWS Glue

### Argomenti

- Caso d'uso: connessione del registro degli schemi ad Amazon MSK o Apache Kafka
- Caso d'uso: integrazione del flusso di dati Amazon Kinesis con il registro degli schemi di AWS Glue
- Caso d'uso: Servizio gestito da Amazon per Apache Flink
- Caso d'uso: integrazione con AWS Lambda
- Caso d'uso: AWS Glue Data Catalog
- Caso d'uso: streaming AWS Glue
- Caso d'uso: flussi Apache Kafka

Caso d'uso: connessione del registro degli schemi ad Amazon MSK o Apache Kafka

Supponiamo che tu stia scrivendo dati su un argomento Apache Kafka e che possa seguire questi passaggi per iniziare.

1. Crea un cluster Amazon Streaming gestito per Apache Kafka (Amazon MSK) o Apache Kafka con almeno un argomento. Se si crea un cluster Amazon MSK, è possibile utilizzare il Console di gestione AWS. Segui queste istruzioni: Nozioni di base per l'uso di Amazon MSK nella Guida per gli sviluppatori di Amazon Managed Streaming for Apache Kafka.
2. Segui il passaggio Installazione delle SerDe librerie sopra.
3. Per creare registri dello schema, schemi o versioni, segui le istruzioni riportate nella sezione Guida introduttiva al registro degli schemi di questo documento.
4. Inizia i tuoi produttori e consumatori a utilizzare lo Schema Registry per scrivere e leggere record to/from sull'argomento Amazon MSK o Apache Kafka. È possibile trovare esempi di codice per produttori e consumatori nel ReadMe file delle librerie Serde. La libreria del registro degli schemi del produttore serializzerà automaticamente il record e aggiungerà un ID della versione dello schema al record.
5. Se lo schema di questo record è stato inserito o se la registrazione automatica è attivata, lo schema risulterà registrato nel registro degli schemi.
6. Il consumer che legge dall'argomento Amazon MSK o Apache Kafka, utilizzando la libreria del registro degli schemi di AWS Glue, cercherà automaticamente lo schema dal registro degli schemi.

Caso d'uso: integrazione del flusso di dati Amazon Kinesis con il registro degli schemi di AWS Glue

Per questa integrazione è necessario disporre di un flusso dei dati Amazon Kinesis. Per ulteriori informazioni, consulta Nozioni di base su Amazon Kinesis Data Streams nella Guida per gli sviluppatori di Amazon Kinesis Data Streams.

Esistono due modi per interagire con i dati in un flusso dei dati Kinesis.

- Tramite le librerie Kinesis Producer Library (KPL) e Kinesis Client Library (KCL) in Java. Il supporto multilingue non viene fornito.
- Tramite `PutRecords`Put`Record`, e `GetRecords` Kinesis APIs Data Streams disponibile in. AWS SDK per Java

Se attualmente utilizzi le KPL/KCL librerie, ti consigliamo di continuare a utilizzare quel metodo. Come mostrato negli esempi, esistono versioni KCL e KPL aggiornate con il registro degli schemi

integrato. Altrimenti, puoi usare il codice di esempio per sfruttare lo AWS Glue Schema Registry se usi direttamente il KDS. APIs

L'integrazione del registro degli schemi è disponibile solo con KPL v0.14.2 o versioni successive e con KCL v2.3 o versioni successive. L'integrazione del registro degli schemi con il formato di dati JSON è disponibile solo con KPL v0.14.8 o versioni successive e con KCL v2.3.6 o versioni successive.

Interazione con i dati utilizzando Kinesis SDK V2

Questa sezione descrive l'interazione con Kinesis utilizzando Kinesis SDK V2

```
// Example JSON Record, you can construct a AVRO record also
private static final JsonDataWithSchema record =
 JsonDataWithSchema.builder(schemaString, payloadString);
private static final DataFormat dataFormat = DataFormat.JSON;

//Configurations for Schema Registry
GlueSchemaRegistryConfiguration gsrConfig = new GlueSchemaRegistryConfiguration("us-
east-1");

GlueSchemaRegistrySerializer glueSchemaRegistrySerializer =
        new GlueSchemaRegistrySerializerImpl(awsCredentialsProvider, gsrConfig);
GlueSchemaRegistryDataFormatSerializer dataFormatSerializer =
        new GlueSchemaRegistrySerializerFactory().getInstance(dataFormat, gsrConfig);

Schema gsrSchema =
        new Schema(dataFormatSerializer.getSchemaDefinition(record), dataFormat.name(),
 "MySchema");

byte[] serializedBytes = dataFormatSerializer.serialize(record);

byte[] gsrEncodedBytes = glueSchemaRegistrySerializer.encode(streamName, gsrSchema,
 serializedBytes);

PutRecordRequest putRecordRequest = PutRecordRequest.builder()
        .streamName(streamName)
        .partitionKey("partitionKey")
        .data(SdkBytes.fromByteArray(gsrEncodedBytes))
        .build();
shardId = kinesisClient.putRecord(putRecordRequest)
        .get()
        .shardId();
```

```
GlueSchemaRegistryDeserializer glueSchemaRegistryDeserializer = new
 GlueSchemaRegistryDeserializerImpl(awsCredentialsProvider, gsrConfig);

GlueSchemaRegistryDataFormatDeserializer gsrDataFormatDeserializer =
        glueSchemaRegistryDeserializerFactory.getInstance(dataFormat, gsrConfig);

GetShardIteratorRequest getShardIteratorRequest = GetShardIteratorRequest.builder()
        .streamName(streamName)
        .shardId(shardId)
        .shardIteratorType(ShardIteratorType.TRIM_HORIZON)
        .build();

String shardIterator = kinesisClient.getShardIterator(getShardIteratorRequest)
        .get()
        .shardIterator();

GetRecordsRequest getRecordRequest = GetRecordsRequest.builder()
        .shardIterator(shardIterator)
        .build();
GetRecordsResponse recordsResponse = kinesisClient.getRecords(getRecordRequest)
        .get();

List<Object> consumerRecords = new ArrayList<>();
List<Record> recordsFromKinesis = recordsResponse.records();

for (int i = 0; i < recordsFromKinesis.size(); i++) {
    byte[] consumedBytes = recordsFromKinesis.get(i)
            .data()
            .asByteArray();

    Schema gsrSchema = glueSchemaRegistryDeserializer.getSchema(consumedBytes);
    Object decodedRecord =
 gsrDataFormatDeserializer.deserialize(ByteBuffer.wrap(consumedBytes),

 gsrSchema.getSchemaDefinition());
    consumerRecords.add(decodedRecord);
}
```

Interazione con i dati utilizzando le librerie KPL/KCL

Questa sezione descrive l'integrazione di Kinesis Data Streams con Schema Registry utilizzando le librerie. KPL/KCL Per ulteriori informazioni sull'utilizzo di KPL/KCL, consulta Sviluppo di produttori

utilizzando Amazon Kinesis Producer Library nella Guida per gli sviluppatori di Amazon Kinesis Data Streams.

Impostazione del registro degli schemi in KPL

1. Definisci la definizione dello schema per i dati, il formato dei dati e il nome dello schema creati nel registro degli schemi di AWS Glue.
2. Facoltativamente, puoi configurare l'oggetto `GlueSchemaRegistryConfiguration`.
3. Trasferisci l'oggetto dello schema a `addUserRecord API`.

```
private static final String SCHEMA_DEFINITION = "{"namespace": "example.avro",\n"
+ " "type": "record",\n"
+ " "name": "User",\n"
+ " "fields": [\n"
+ " {"name": "name", "type": "string"},\n"
+ " {"name": "favorite_number", "type": ["int", "null"]},\n"
+ " {"name": "favorite_color", "type": ["string", "null"]}\n"
+ " ]\n"
+ "}";

KinesisProducerConfiguration config = new KinesisProducerConfiguration();
config.setRegion("us-west-1")

//[Optional] configuration for Schema Registry.

GlueSchemaRegistryConfiguration schemaRegistryConfig =
new GlueSchemaRegistryConfiguration("us-west-1");

schemaRegistryConfig.setCompression(true);

config.setGlueSchemaRegistryConfiguration(schemaRegistryConfig);

///Optional configuration ends.

final KinesisProducer producer =
      new KinesisProducer(config);

final ByteBuffer data = getDataToSend();

com.amazonaws.services.schemaregistry.common.Schema gsrSchema =
    new Schema(SCHEMA_DEFINITION, DataFormat.AVRO.toString(), "demoSchema");
```

```
ListenableFuture<UserRecordResult> f = producer.addUserRecord(
config.getStreamName(), TIMESTAMP, Utils.randomExplicitHashKey(), data, gsrSchema);

private static ByteBuffer getDataToSend() {
      org.apache.avro.Schema avroSchema =
        new org.apache.avro.Schema.Parser().parse(SCHEMA_DEFINITION);

      GenericRecord user = new GenericData.Record(avroSchema);
      user.put("name", "Emily");
      user.put("favorite_number", 32);
      user.put("favorite_color", "green");

      ByteArrayOutputStream outBytes = new ByteArrayOutputStream();
      Encoder encoder = EncoderFactory.get().directBinaryEncoder(outBytes, null);
      new GenericDatumWriter<>(avroSchema).write(user, encoder);
      encoder.flush();
      return ByteBuffer.wrap(outBytes.toByteArray());
 }
```

Impostazione della libreria client di Kinesis

Svilupperai un consumer Kinesis Client Library in Java. Per ulteriori informazioni su, consulta Sviluppo di app Consumer Kinesis Client Library in Java nella Guida per gli sviluppatori di Amazon Kinesis Data Streams.

1. Crea un'istanza di `GlueSchemaRegistryDeserializer` passando un oggetto `GlueSchemaRegistryConfiguration`.
2. Passa `GlueSchemaRegistryDeserializer` a `retrievalConfig.glueSchemaRegistryDeserializer`.
3. Accedi allo schema dei messaggi in arrivo chiamando `kinesisClientRecord.getSchema()`.

```
GlueSchemaRegistryConfiguration schemaRegistryConfig =
    new GlueSchemaRegistryConfiguration(this.region.toString());

 GlueSchemaRegistryDeserializer glueSchemaRegistryDeserializer =
    new
 GlueSchemaRegistryDeserializerImpl(DefaultCredentialsProvider.builder().build(),
 schemaRegistryConfig);
```

```
RetrievalConfig retrievalConfig =
configsBuilder.retrievalConfig().retrievalSpecificConfig(new
PollingConfig(streamName, kinesisClient));
retrievalConfig.glueSchemaRegistryDeserializer(glueSchemaRegistryDeserializer);

 Scheduler scheduler = new Scheduler(
           configsBuilder.checkpointConfig(),
           configsBuilder.coordinatorConfig(),
           configsBuilder.leaseManagementConfig(),
           configsBuilder.lifecycleConfig(),
           configsBuilder.metricsConfig(),
           configsBuilder.processorConfig(),
           retrievalConfig
       );

public void processRecords(ProcessRecordsInput processRecordsInput) {
           MDC.put(SHARD_ID_MDC_KEY, shardId);
           try {
               log.info("Processing {} record(s)",
               processRecordsInput.records().size());
               processRecordsInput.records()
               .forEach(
                   r ->
                       log.info("Processed record pk: {} -- Seq: {} : data {} with
schema: {}",
                       r.partitionKey(),
r.sequenceNumber(), recordToAvroObj(r).toString(), r.getSchema()));
           } catch (Throwable t) {
               log.error("Caught throwable while processing records. Aborting.");
               Runtime.getRuntime().halt(1);
           } finally {
               MDC.remove(SHARD_ID_MDC_KEY);
           }
}

private GenericRecord recordToAvroObj(KinesisClientRecord r) {
   byte[] data = new byte[r.data().remaining()];
   r.data().get(data, 0, data.length);
   org.apache.avro.Schema schema = new
org.apache.avro.Schema.Parser().parse(r.schema().getSchemaDefinition());
   DatumReader datumReader = new GenericDatumReader<>(schema);

   BinaryDecoder binaryDecoder = DecoderFactory.get().binaryDecoder(data, 0,
data.length, null);
```

```
    return (GenericRecord) datumReader.read(null, binaryDecoder);
 }
```

Interazione con i dati utilizzando Kinesis Data Streams APIs

Questa sezione descrive l'integrazione di Kinesis Data Streams con Schema Registry utilizzando Kinesis Data Streams. APIs

1. Aggiorna queste dipendenze di Maven:

```
<dependencyManagement>
        <dependencies>
            <dependency>
                <groupId>com.amazonaws</groupId>
                <artifactId>aws-java-sdk-bom</artifactId>
                <version>1.11.884</version>
                <type>pom</type>
                <scope>import</scope>
            </dependency>
        </dependencies>
    </dependencyManagement>

    <dependencies>
        <dependency>
            <groupId>com.amazonaws</groupId>
            <artifactId>aws-java-sdk-kinesis</artifactId>
        </dependency>

        <dependency>
            <groupId>software.amazon.glue</groupId>
            <artifactId>schema-registry-serde</artifactId>
            <version>1.1.5</version>
        </dependency>

        <dependency>
            <groupId>com.fasterxml.jackson.dataformat</groupId>
            <artifactId>jackson-dataformat-cbor</artifactId>
            <version>2.11.3</version>
        </dependency>
    </dependencies>
```

2. Nel produttore, aggiungi le informazioni sull'intestazione dello schema utilizzando il l'API `PutRecords` o `PutRecord` in Kinesis Data Streams.

```
//The following lines add a Schema Header to the record
        com.amazonaws.services.schemaregistry.common.Schema awsSchema =
            new com.amazonaws.services.schemaregistry.common.Schema(schemaDefinition,
 DataFormat.AVRO.name(),
                schemaName);
        GlueSchemaRegistrySerializerImpl glueSchemaRegistrySerializer =
            new
 GlueSchemaRegistrySerializerImpl(DefaultCredentialsProvider.builder().build(), new
 GlueSchemaRegistryConfiguration(getConfigs()));
        byte[] recordWithSchemaHeader =
            glueSchemaRegistrySerializer.encode(streamName, awsSchema,
 recordAsBytes);
```

3. Nel produttore, utilizza l'API `PutRecords` o `PutRecord` per inserire il record nel flusso dei dati.
4. Nel consumer, rimuovi il record dello schema dall'intestazione e serializza un record dello schema Avro.

```
//The following lines remove Schema Header from record
        GlueSchemaRegistryDeserializerImpl glueSchemaRegistryDeserializer =
            new
 GlueSchemaRegistryDeserializerImpl(DefaultCredentialsProvider.builder().build(),
 getConfigs());
        byte[] recordWithSchemaHeaderBytes = new
 byte[recordWithSchemaHeader.remaining()];
        recordWithSchemaHeader.get(recordWithSchemaHeaderBytes, 0,
 recordWithSchemaHeaderBytes.length);
        com.amazonaws.services.schemaregistry.common.Schema awsSchema =
            glueSchemaRegistryDeserializer.getSchema(recordWithSchemaHeaderBytes);
        byte[] record =
 glueSchemaRegistryDeserializer.getData(recordWithSchemaHeaderBytes);

        //The following lines serialize an AVRO schema record
        if (DataFormat.AVRO.name().equals(awsSchema.getDataFormat())) {
            Schema avroSchema = new
 org.apache.avro.Schema.Parser().parse(awsSchema.getSchemaDefinition());
            Object genericRecord = convertBytesToRecord(avroSchema, record);
            System.out.println(genericRecord);
        }
```

Interazione con i dati utilizzando Kinesis Data Streams APIs

Di seguito è riportato un codice di esempio per l'utilizzo di and. `PutRecords` `GetRecords` APIs

```
//Full sample code
import
 com.amazonaws.services.schemaregistry.deserializers.GlueSchemaRegistryDeserializerImpl;
import
 com.amazonaws.services.schemaregistry.serializers.GlueSchemaRegistrySerializerImpl;
import com.amazonaws.services.schemaregistry.utils.AVROUtils;
import com.amazonaws.services.schemaregistry.utils.AWSSchemaRegistryConstants;
import org.apache.avro.Schema;
import org.apache.avro.generic.GenericData;
import org.apache.avro.generic.GenericDatumReader;
import org.apache.avro.generic.GenericDatumWriter;
import org.apache.avro.generic.GenericRecord;
import org.apache.avro.io.Decoder;
import org.apache.avro.io.DecoderFactory;
import org.apache.avro.io.Encoder;
import org.apache.avro.io.EncoderFactory;
import software.amazon.awssdk.auth.credentials.DefaultCredentialsProvider;
import software.amazon.awssdk.services.glue.model.DataFormat;

import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.IOException;
import java.nio.ByteBuffer;
import java.util.Collections;
import java.util.HashMap;
import java.util.Map;


public class PutAndGetExampleWithEncodedData {
    static final String regionName = "us-east-2";
    static final String streamName = "testStream1";
    static final String schemaName = "User-Topic";
    static final String AVRO_USER_SCHEMA_FILE = "src/main/resources/user.avsc";
    KinesisApi kinesisApi = new KinesisApi();

    void runSampleForPutRecord() throws IOException {
        Object testRecord = getTestRecord();
        byte[] recordAsBytes = convertRecordToBytes(testRecord);
        String schemaDefinition =
 AVROUtils.getInstance().getSchemaDefinition(testRecord);
```

```
       //The following lines add a Schema Header to a record
       com.amazonaws.services.schemaregistry.common.Schema awsSchema =
           new com.amazonaws.services.schemaregistry.common.Schema(schemaDefinition,
DataFormat.AVRO.name(),
               schemaName);
       GlueSchemaRegistrySerializerImpl glueSchemaRegistrySerializer =
           new
GlueSchemaRegistrySerializerImpl(DefaultCredentialsProvider.builder().build(), new
GlueSchemaRegistryConfiguration(regionName));
       byte[] recordWithSchemaHeader =
           glueSchemaRegistrySerializer.encode(streamName, awsSchema, recordAsBytes);

       //Use PutRecords api to pass a list of records
       kinesisApi.putRecords(Collections.singletonList(recordWithSchemaHeader),
streamName, regionName);

       //OR
       //Use PutRecord api to pass single record
       //kinesisApi.putRecord(recordWithSchemaHeader, streamName, regionName);
   }

   byte[] runSampleForGetRecord() throws IOException {
       ByteBuffer recordWithSchemaHeader = kinesisApi.getRecords(streamName,
regionName);

       //The following lines remove the schema registry header
       GlueSchemaRegistryDeserializerImpl glueSchemaRegistryDeserializer =
           new
GlueSchemaRegistryDeserializerImpl(DefaultCredentialsProvider.builder().build(), new
GlueSchemaRegistryConfiguration(regionName));
       byte[] recordWithSchemaHeaderBytes = new
byte[recordWithSchemaHeader.remaining()];
       recordWithSchemaHeader.get(recordWithSchemaHeaderBytes, 0,
recordWithSchemaHeaderBytes.length);

       com.amazonaws.services.schemaregistry.common.Schema awsSchema =
           glueSchemaRegistryDeserializer.getSchema(recordWithSchemaHeaderBytes);

       byte[] record =
glueSchemaRegistryDeserializer.getData(recordWithSchemaHeaderBytes);

       //The following lines serialize an AVRO schema record
       if (DataFormat.AVRO.name().equals(awsSchema.getDataFormat())) {
```

```
            Schema avroSchema = new
org.apache.avro.Schema.Parser().parse(awsSchema.getSchemaDefinition());
            Object genericRecord = convertBytesToRecord(avroSchema, record);
            System.out.println(genericRecord);
        }

        return record;
    }

    private byte[] convertRecordToBytes(final Object record) throws IOException {
        ByteArrayOutputStream recordAsBytes = new ByteArrayOutputStream();
        Encoder encoder = EncoderFactory.get().directBinaryEncoder(recordAsBytes,
null);
        GenericDatumWriter datumWriter = new
GenericDatumWriter<>(AVROUtils.getInstance().getSchema(record));
        datumWriter.write(record, encoder);
        encoder.flush();
        return recordAsBytes.toByteArray();
    }

    private GenericRecord convertBytesToRecord(Schema avroSchema, byte[] record) throws
IOException {
        final GenericDatumReader<GenericRecord> datumReader = new
GenericDatumReader<>(avroSchema);
        Decoder decoder = DecoderFactory.get().binaryDecoder(record, null);
        GenericRecord genericRecord = datumReader.read(null, decoder);
        return genericRecord;
    }

    private Map<String, String> getMetadata() {
        Map<String, String> metadata = new HashMap<>();
        metadata.put("event-source-1", "topic1");
        metadata.put("event-source-2", "topic2");
        metadata.put("event-source-3", "topic3");
        metadata.put("event-source-4", "topic4");
        metadata.put("event-source-5", "topic5");
        return metadata;
    }

    private GlueSchemaRegistryConfiguration getConfigs() {
        GlueSchemaRegistryConfiguration configs = new
GlueSchemaRegistryConfiguration(regionName);
        configs.setSchemaName(schemaName);
        configs.setAutoRegistration(true);
```

```
        configs.setMetadata(getMetadata());
        return configs;
    }

    private Object getTestRecord() throws IOException {
        GenericRecord genericRecord;
        Schema.Parser parser = new Schema.Parser();
        Schema avroSchema = parser.parse(new File(AVRO_USER_SCHEMA_FILE));

        genericRecord = new GenericData.Record(avroSchema);
        genericRecord.put("name", "testName");
        genericRecord.put("favorite_number", 99);
        genericRecord.put("favorite_color", "red");

        return genericRecord;
    }
}
```

Caso d'uso: Servizio gestito da Amazon per Apache Flink

Apache Flink è un diffuso framework open source e motore di elaborazione distribuito per calcoli con stato su flussi dei dati illimitati e limitati. Amazon Managed Service per Apache Flink è un AWS servizio completamente gestito che consente di creare e gestire applicazioni Apache Flink per elaborare dati di streaming.

Apache Flink open source fornisce una serie di origini e sink. Ad esempio, le origini dati predefinite includono la lettura da file, directory e socket e l'inserimento di dati da raccolte e iteratori.
I DataStream connettori Apache Flink forniscono codice per consentire ad Apache Flink di interfacciarsi con vari sistemi di terze parti, come Apache Kafka o Kinesis, come sorgenti. and/or

Per ulteriori informazioni, consulta la Guida per sviluppatori di Amazon Kinesis Data Analytics.

Connettore Apache Flink Kafka

Apache Flink fornisce un connettore per flusso dei dati Apache Kafka per la lettura e la scrittura di dati su argomenti Kafka con garanzie exactly-once. Il consumer Kafka di Flink, `FlinkKafkaConsumer`, fornisce l'accesso alla lettura da uno o più argomenti di Kafka. Il produttore Kafka di Apache Flink, `FlinkKafkaProducer`, consente di scrivere un flusso di record su uno o più argomenti Kafka. Per ulteriori informazioni, consulta Apache Kafka Connector.

### Connettore di flussi Apache Flink Kinesis

Il connettore del flusso dei dati Kinesis consente di accedere ai Amazon Kinesis Data Streams. `FlinkKinesisConsumer`Si tratta di una fonte di dati di streaming parallela che utilizza una sola volta per abbonarsi a più flussi Kinesis all'interno della stessa area di AWS servizio e può gestire in modo trasparente il ripartizionamento dei flussi mentre il processo è in esecuzione. Ogni sottoattività del consumer è responsabile del recupero dei record di dati da più partizioni Kinesis. Il numero di partizioni recuperate da ogni sottoattività cambierà man mano che le partizioni vengono chiuse e create da Kinesis. `FlinkKinesisProducer` utilizza Kinesis Producer Library (KPL) per inserire i dati da un flusso Apache Flink in un flusso Kinesis. Per ulteriori informazioni, consulta Amazon Kinesis Streams Connector.

Per ulteriori informazioni, consulta il AWS Gluerepository Github di schema .

### Integrazione con Apache Flink

La SerDes libreria fornita con Schema Registry si integra con Apache Flink. Per utilizzare Apache Flink, sarà necessario implementare le interfacce `SerializationSchema` e `DeserializationSchema` denominate `GlueSchemaRegistryAvroSerializationSchema` e `GlueSchemaRegistryAvroDeserializationSchema`, che possono essere collegate ai connettori Apache Flink.

#### Aggiunta di una dipendenza del registro degli schemi di AWS Glue nell'applicazione Apache Flink

Per impostare le dipendenze di integrazione sul registro degli schemi di AWS Glue nell'applicazione Apache Flink:

1. Aggiungi la dipendenza al file `pom.xml`.

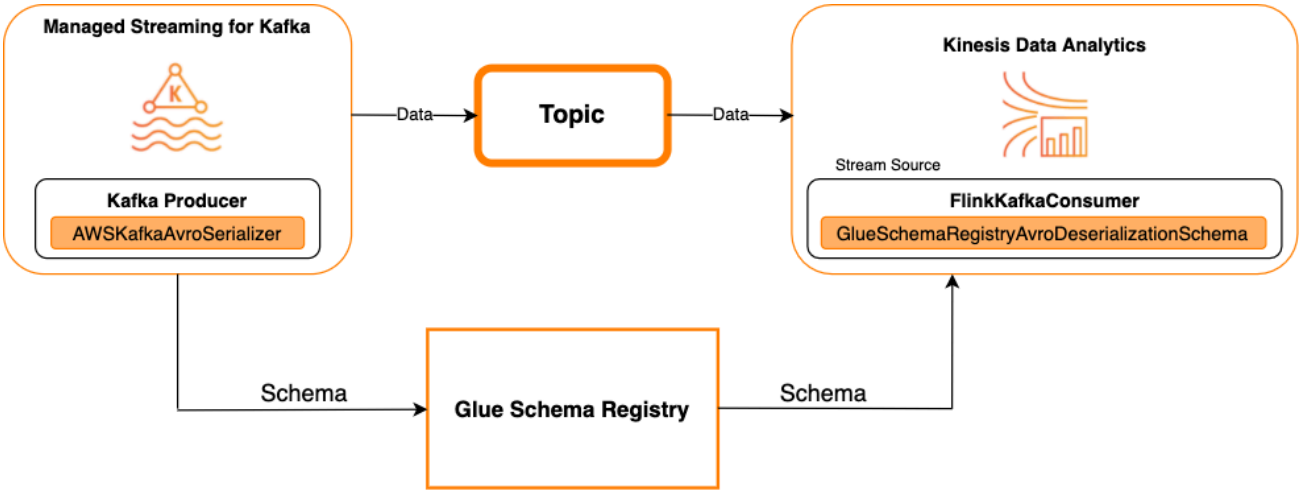
```
<dependency>
    <groupId>software.amazon.glue</groupId>
    <artifactId>schema-registry-flink-serde</artifactId>
    <version>1.0.0</version>
</dependency>
```

### Integrazione di Kafka o Amazon MSK con Apache Flink

È possibile usare Servizio gestito da Amazon per Apache Flink per Apache Flink con Kafka come origine o Kafka come sink.

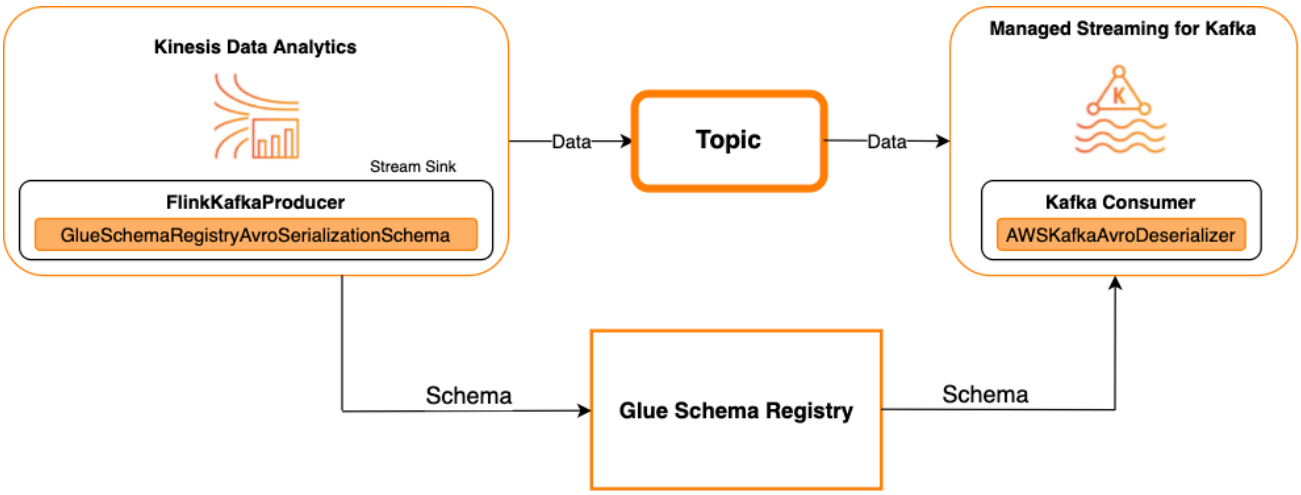
Kafka come fonte

Il diagramma seguente mostra l'integrazione di Flusso di dati Kinesis con Servizio gestito da Amazon per Apache Flink per Apache Flink, con Kafka come origine.




Kafka come sink

Il diagramma seguente mostra l'integrazione di Flusso di dati Kinesis con Servizio gestito da Amazon per Apache Flink per Apache Flink, con Kafka come sink.




Per integrare Kafka (o Amazon MSK) con Servizio gestito da Amazon per Apache Flink per Apache Flink, con Kafka come origine o Kafka come sink, apporta le modifiche al codice riportate di seguito. Aggiungi i blocchi di codice in grassetto al codice corrispondente nelle sezioni analoghe.

Se Kafka è la fonte, utilizza il codice deserializzatore (blocco 2). Se Kafka è il sink, utilizza il codice serializzatore (blocco 3).

```
StreamExecutionEnvironment env = StreamExecutionEnvironment.getExecutionEnvironment();
```

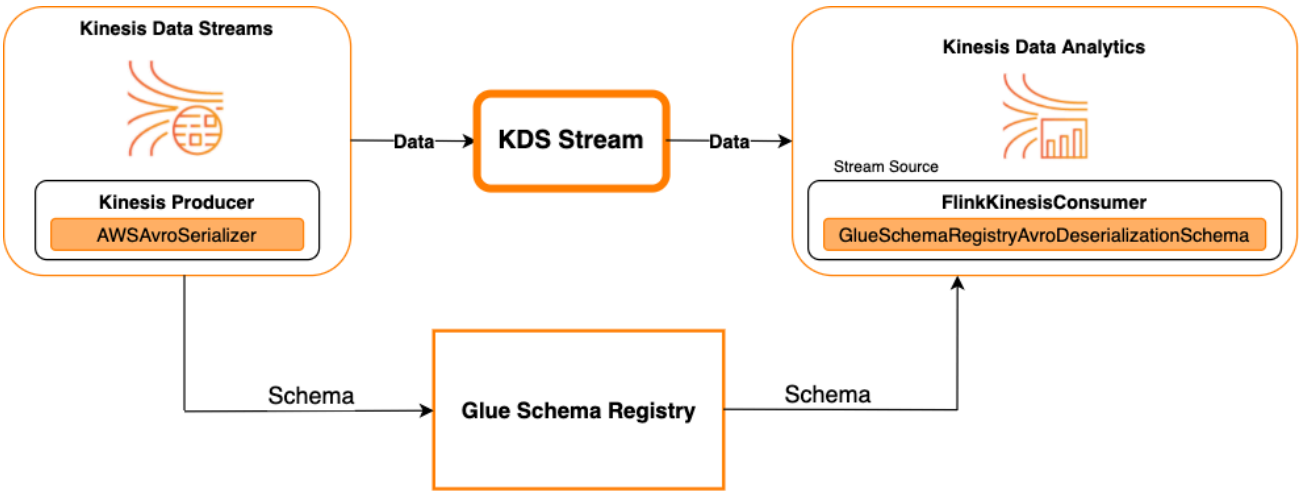
```
String topic = "topic";
Properties properties = new Properties();
properties.setProperty("bootstrap.servers", "localhost:9092");
properties.setProperty("group.id", "test");

// block 1
Map<String, Object> configs = new HashMap<>();
configs.put(AWSSchemaRegistryConstants.AWS_REGION, "aws-region");
configs.put(AWSSchemaRegistryConstants.SCHEMA_AUTO_REGISTRATION_SETTING, true);
configs.put(AWSSchemaRegistryConstants.AVRO_RECORD_TYPE,
 AvroRecordType.GENERIC_RECORD.getName());

FlinkKafkaConsumer<GenericRecord> consumer = new FlinkKafkaConsumer<>(
    topic,
    // block 2
    GlueSchemaRegistryAvroDeserializationSchema.forGeneric(schema, configs),
    properties);

FlinkKafkaProducer<GenericRecord> producer = new FlinkKafkaProducer<>(
    topic,
    // block 3
    GlueSchemaRegistryAvroSerializationSchema.forGeneric(schema, topic, configs),
    properties);

DataStream<GenericRecord> stream = env.addSource(consumer);
stream.addSink(producer);
env.execute();
```

### Integrazione di Kinesis Data Streams con Apache Flink

È possibile utilizzare Servizio gestito da Amazon per Apache Flink per Apache Flink con Flusso di dati Kinesis come origine o sink.

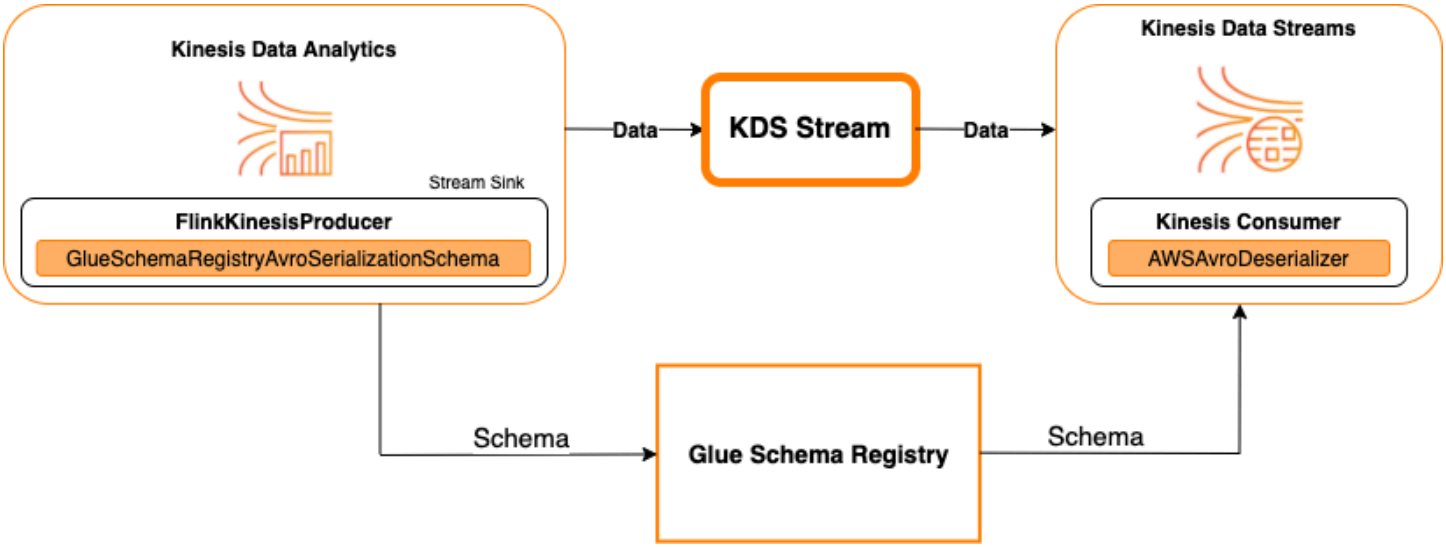
#### Kinesis Data Streams come fonte

Il diagramma seguente mostra l'integrazione di Flusso di dati Kinesis con Servizio gestito per Apache Flink per Apache Flink, con Flusso di dati Kinesis come origine.




Kinesis Data Streams come sink

Il diagramma seguente mostra l'integrazione di Flusso di dati Kinesis con Servizio gestito per Apache Flink per Apache Flink, con Flusso di dati Kinesis come sink.




Per integrare Flusso di dati Kinesis con Servizio gestito per Apache Flink per Apache Flink, con Flusso di dati Kinesis come origine o Flusso di dati Kinesis come sink, apporta le modifiche al codice riportate di seguito. Aggiungi i blocchi di codice in grassetto al codice corrispondente nelle sezioni analoghe.

Se Kinesis Data Streams è l'origine, utilizza il codice deserializzatore (blocco 2). Se Kinesis Data Streams il sink, utilizza il codice serializzatore (blocco 3).

```
StreamExecutionEnvironment env = StreamExecutionEnvironment.getExecutionEnvironment();
```

```
String streamName = "stream";
Properties consumerConfig = new Properties();
consumerConfig.put(AWSConfigConstants.AWS_REGION, "aws-region");
consumerConfig.put(AWSConfigConstants.AWS_ACCESS_KEY_ID, "aws_access_key_id");
consumerConfig.put(AWSConfigConstants.AWS_SECRET_ACCESS_KEY, "aws_secret_access_key");
consumerConfig.put(ConsumerConfigConstants.STREAM_INITIAL_POSITION, "LATEST");

// block 1
Map<String, Object> configs = new HashMap<>();
configs.put(AWSSchemaRegistryConstants.AWS_REGION, "aws-region");
configs.put(AWSSchemaRegistryConstants.SCHEMA_AUTO_REGISTRATION_SETTING, true);
configs.put(AWSSchemaRegistryConstants.AVRO_RECORD_TYPE,
 AvroRecordType.GENERIC_RECORD.getName());

FlinkKinesisConsumer<GenericRecord> consumer = new FlinkKinesisConsumer<>(
    streamName,
    // block 2
    GlueSchemaRegistryAvroDeserializationSchema.forGeneric(schema, configs),
    properties);

FlinkKinesisProducer<GenericRecord> producer = new FlinkKinesisProducer<>(
    // block 3
    GlueSchemaRegistryAvroSerializationSchema.forGeneric(schema, topic, configs),
    properties);
producer.setDefaultStream(streamName);
producer.setDefaultPartition("0");

DataStream<GenericRecord> stream = env.addSource(consumer);
stream.addSink(producer);
env.execute();
```

Caso d'uso: integrazione con AWS Lambda

Per utilizzare una AWS Lambda funzione come consumatore Apache Kafka/Amazon MSK e deserializzare i messaggi con codifica AVRO utilizzando AWS Glue Schema Registry, visita la pagina MSK Labs.

Caso d'uso: AWS Glue Data Catalog

Le tabelle AWS Glue supportano gli schemi che è possibile specificare manualmente o facendo riferimento al registro degli schemi di AWS Glue. Il registro degli schemi si integra con il catalogo dati per consentire l'utilizzo facoltativo degli schemi memorizzati nel registro degli schemi durante la creazione o l'aggiornamento di tabelle o partizioni AWS Glue nel catalogo dati. Per identificare

una definizione dello schema nel registro degli schemi, è necessario conoscere almeno l'ARN dello schema di cui fa parte. Una versione di uno schema, che contiene una definizione dello schema, può essere referenziata dal relativo UUID o numero di versione. C'è sempre una versione dello schema, la versione "più recente", che può essere cercata senza conoscere il suo numero di versione o UUID.

Chiamando le operazioni `CreateTable` o `UpdateTable`, passerai una struttura `TableInput` che contiene un `StorageDescriptor`, che può avere un `SchemaReference` a uno schema esistente nel registro degli schemi. Allo stesso modo, quando si chiama `GetTable` o `GetPartition` APIs, la risposta può contenere lo schema e il`SchemaReference`. Quando una tabella o una partizione è stata creata utilizzando riferimenti allo schema, il catalogo dati tenterà di recuperare lo schema per questo riferimento. Nel caso in cui non sia in grado di trovare lo schema nel registro degli schemi, restituisce uno schema vuoto nella risposta `GetTable`; in caso contrario, la risposta conterrà sia lo schema che il riferimento allo schema.

Puoi eseguire le seguenti operazioni dalla console AWS Glue.

Per eseguire queste operazioni e creare, aggiornare o visualizzare le informazioni sullo schema, è necessario assegnare un ruolo IAM all'utente chiamante che fornisce le autorizzazioni per l'API `GetSchemaVersion`.

Aggiunta di una tabella o aggiornamento dello schema di una tabella

L'aggiunta di una nuova tabella da uno schema esistente associa la tabella a una versione specifica dello schema. Una volta registrate le nuove versioni dello schema, è possibile aggiornare la definizione della tabella dalla pagina View table (Visualizza tabella) nella console AWS Glue o utilizzando l'API UpdateTable azione (Python: update_table).

Aggiunta di una tabella da uno schema esistente

Puoi creare una tabella AWS Glue da una versione dello schema nel registro utilizzando la console AWS Glue o l'API `CreateTable`.

API AWS Glue

Chiamando l'API `CreateTable`, passerai un `TableInput` che contiene una `StorageDescriptor`, che può avere un `SchemaReference` a uno schema esistente nel registro degli schemi.

Console AWS Glue

Per creare una tabella dalla consoleAWS Glue:

1. Accedi a Console di gestione AWS e apri la AWS Glue console all'indirizzo https://console.aws.amazon.com/glue/.
2. Nel pannello di navigazione, in Data catalog (Catalogo dati), seleziona Tables (Tabelle).
3. Nel menu Add Tables (Aggiungi tabelle), scegli Add table from existing schema (Aggiungi tabella da schema esistente).
4. Configura le proprietà della tabella e il datastore come indicato nella Guida per gli sviluppatori di AWS Glue.
5. Nella pagina Choose a Glue schema (Scegli uno schema di Glue), seleziona il Registry (Registro) in cui si trova lo schema.
6. Scegli Schema name (Nome schema) e seleziona la Version (Versione) dello schema da applicare.
7. Esamina l'anteprima dello schema e scegli Next (Successivo).
8. Rivedi e crea la tabella.

Lo schema e la versione applicati alla tabella vengono visualizzati nella colonna Glue schema (Schema di Glue) nell'elenco delle tabelle. È possibile visualizzare la tabella per vedere ulteriori dettagli.

### Aggiornamento dello schema per una tabella

Quando una nuova versione dello schema diventa disponibile, si consiglia di aggiornare lo schema di una tabella utilizzando l'API UpdateTable azione (Python: update_table) o la console AWS Glue.

> **Important**
>
> Aggiornando lo schema per una tabella esistente con uno schema AWS Glue specificato manualmente, il nuovo schema a cui si fa riferimento nel registro degli schemi potrebbe essere incompatibile. Questo potrebbe comportare la non riuscita dei processi.

#### API AWS Glue

Chiamando l'API `UpdateTable`, passerai un `TableInput` che contiene una `StorageDescriptor`, che può avere un `SchemaReference` a uno schema esistente nel registro degli schemi.

#### Console AWS Glue

Per aggiornare lo schema di una tabella dalla console AWS Glue:

1. Accedi a Console di gestione AWS e apri la AWS Glue console all'indirizzo https://console.aws.amazon.com/glue/.
2. Nel pannello di navigazione, in Data catalog (Catalogo dati), seleziona Tables (Tabelle).
3. Visualizza la tabella nell'elenco delle tabelle.
4. Fai clic su Update schema (Aggiorna schema) nella casella che ti informa di una nuova versione.
5. Esamina le differenze tra lo schema attuale e quello nuovo.
6. Scegli Show all schema differences (Mostra tutte le differenze di schema) per ulteriori dettagli.
7. Scegli Save table (Salva tabella) per accettare la nuova versione.

### Caso d'uso: streaming AWS Glue

Lo streaming AWS Glue consuma dati provenienti da fonti di streaming ed esegue operazioni ETL prima di scrivere su un sink di output. L'origine di streaming di input può essere specificata utilizzando una tabella dati o direttamente specificando la configurazione di origine.

Lo streaming AWS Glue supporta una tabella del catalogo dati per la sorgente di streaming creata con lo schema presente nel Registro degli schemi di AWS Glue. È possibile creare uno schema nel Registro degli schemi di AWS Glue e creare una tabella AWS Glue con una sorgente di streaming utilizzando questo schema. Questa tabella AWS Glue può essere utilizzata come input per un processo di streaming AWS Glue per la deserializzazione dei dati nel flusso di input.

Un punto da notare qui è quando lo schema nel Registro degli schemi AWS Glue cambia, è necessario riavviare il processo di streaming AWS Glue in modo da riflettere i cambiamenti nello schema.

### Caso d'uso: flussi Apache Kafka

L'API di Apache Kafka Streams è una libreria client per l'elaborazione e l'analisi dei dati memorizzati in Apache Kafka. Questa sezione descrive l'integrazione di Apache Kafka Streams con il registro degli schemi di AWS Glue, che consente di gestire e applicare gli schemi sulle applicazioni di streaming di dati. Per ulteriori informazioni su Apache Kafka Streams, consulta Apache Kafka Streams.

#### Integrazione con le librerie SerDes

Esiste una classe di `GlueSchemaRegistryKafkaStreamsSerde` con cui è possibile configurare un'applicazione Streams.

Codice di esempio di applicazione di flussi Kafka

Per utilizzare il registro degli schemi di AWS Glue all'interno di un'applicazione Apache Kafka Streams:

1. Configura l'applicazione Kafka Streams.

```
final Properties props = new Properties();
    props.put(StreamsConfig.APPLICATION_ID_CONFIG, "avro-streams");
    props.put(StreamsConfig.BOOTSTRAP_SERVERS_CONFIG, "localhost:9092");
    props.put(StreamsConfig.CACHE_MAX_BYTES_BUFFERING_CONFIG, 0);
    props.put(StreamsConfig.DEFAULT_KEY_SERDE_CLASS_CONFIG,
 Serdes.String().getClass().getName());
    props.put(StreamsConfig.DEFAULT_VALUE_SERDE_CLASS_CONFIG,
 AWSKafkaAvroSerDe.class.getName());
    props.put(ConsumerConfig.AUTO_OFFSET_RESET_CONFIG, "earliest");

    props.put(AWSSchemaRegistryConstants.AWS_REGION, "aws-region");
    props.put(AWSSchemaRegistryConstants.SCHEMA_AUTO_REGISTRATION_SETTING, true);
    props.put(AWSSchemaRegistryConstants.AVRO_RECORD_TYPE,
 AvroRecordType.GENERIC_RECORD.getName());
 props.put(AWSSchemaRegistryConstants.DATA_FORMAT, DataFormat.AVRO.name());
```

2. Crea un flusso dall'argomento avro-input.

```
StreamsBuilder builder = new StreamsBuilder();
final KStream<String, GenericRecord> source = builder.stream("avro-input");
```

3. Elabora i record di dati (nell'esempio vengono filtrati i record il cui valore favorite_color è rosa o in cui il valore amount è 15).

```
final KStream<String, GenericRecord> result = source
    .filter((key, value) -
> !"pink".equals(String.valueOf(value.get("favorite_color"))));
    .filter((key, value) -> !"15.0".equals(String.valueOf(value.get("amount"))));
```

4. Scrivi i risultati nell'argomento avro-output.

```
result.to("avro-output");
```

5. Avvia l'applicazione Apache Kafka Streams.

```
KafkaStreams streams = new KafkaStreams(builder.build(), props);
streams.start();
```

Risultati dell'implementazione

Questi risultati mostrano il processo di filtraggio dei record che sono stati filtrati nel passaggio 3 in base a favorite_color come "rosa" o valore come "15.0".

Record prima del filtraggio:

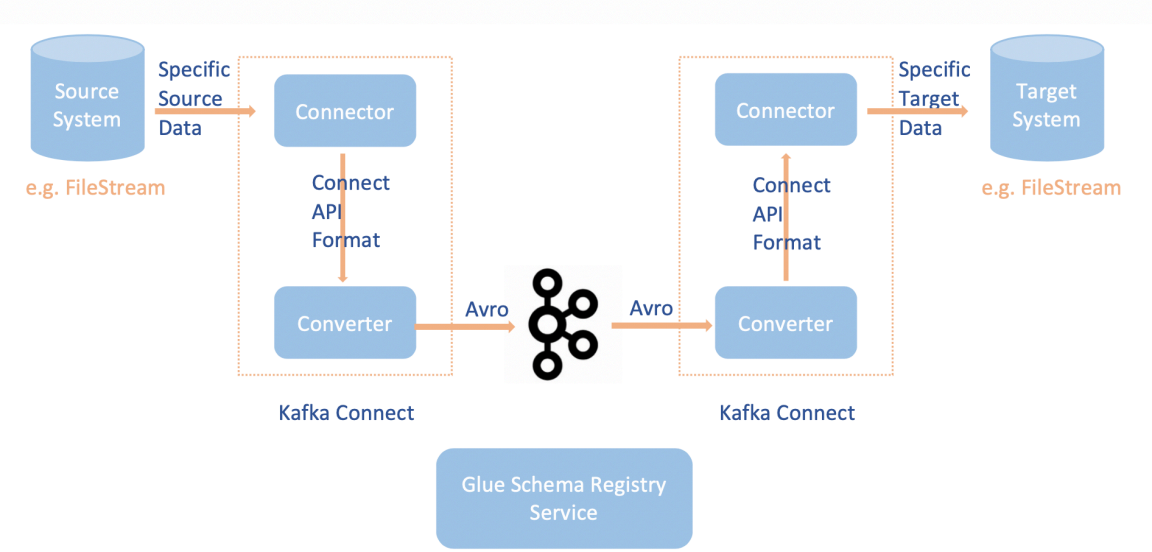
```
{"name": "Sansa", "favorite_number": 99, "favorite_color": "white"}
{"name": "Harry", "favorite_number": 10, "favorite_color": "black"}
{"name": "Hermione", "favorite_number": 1, "favorite_color": "red"}
{"name": "Ron", "favorite_number": 0, "favorite_color": "pink"}
{"name": "Jay", "favorite_number": 0, "favorite_color": "pink"}

{"id": "commute_1","amount": 3.5}
{"id": "grocery_1","amount": 25.5}
{"id": "entertainment_1","amount": 19.2}
{"id": "entertainment_2","amount": 105}
 {"id": "commute_1","amount": 15}
```

Record dopo il filtraggio:

```
{"name": "Sansa", "favorite_number": 99, "favorite_color": "white"}
{"name": "Harry", "favorite_number": 10, "favorite_color": "black"}
{"name": "Hermione", "favorite_number": 1, "favorite_color": "red"}
{"name": "Ron", "favorite_number": 0, "favorite_color": "pink"}

{"id": "commute_1","amount": 3.5}
{"id": "grocery_1","amount": 25.5}
{"id": "entertainment_1","amount": 19.2}
{"id": "entertainment_2","amount": 105}
```

## Caso d'uso: Apache Kafka Connect

L'integrazione di Apache Kafka Connect con il registro degli schemi di AWS Glue consente di ottenere informazioni sullo schema dai connettori. I convertitori Apache Kafka specificano il formato dei dati all'interno di Apache Kafka e come tradurli in dati Apache Kafka Connect. Ogni utente di Apache Kafka Connect dovrà configurare questi convertitori in base al formato desiderato dei dati quando vengono caricati o memorizzati in Apache Kafka. In questo modo, è possibile definire i propri convertitori per tradurre i dati di Apache Kafka Connect nel tipo utilizzato nel registro degli schemi di AWS Glue (ad esempio: Avro) e utilizzare il nostro serializzatore per registrare il suo schema ed eseguire la serializzazione. I convertitori sono anche in grado di utilizzare il nostro deserializzatore per deserializzare i dati ricevuti da Apache Kafka e convertirli nuovamente in dati Apache Kafka Connect. Di seguito è riportato un diagramma di flusso di lavoro di esempio.




1. Installare del progetto `aws-glue-schema-registry` con la clonazione del [repository Github per il registro degli schemi di AWS Glue](#).

```
git clone git@github.com:awslabs/aws-glue-schema-registry.git
cd aws-glue-schema-registry
mvn clean install
mvn dependency:copy-dependencies
```

2. Se hai intenzione di utilizzare Apache Kafka Connect in modalità standalone, aggiorna connect-standalone.properties seguendo le istruzioni di seguito per questo passaggio. Se prevedi di utilizzare Apache Kafka Connect in modalità distribuita, connect-avro-distributedaggiorna .properties seguendo le stesse istruzioni.

a. Aggiungi queste proprietà anche al file delle proprietà di Apache Kafka connect:

```
key.converter.region=aws-region
value.converter.region=aws-region
key.converter.schemaAutoRegistrationEnabled=true
value.converter.schemaAutoRegistrationEnabled=true
key.converter.avroRecordType=GENERIC_RECORD
value.converter.avroRecordType=GENERIC_RECORD
```

b. Aggiungi il comando seguente alla sezione Launch mode in .sh: kafka-run-class

```
-cp $CLASSPATH:"<your AWS GlueSchema Registry base directory>/target/dependency/*"
```

3. Aggiungi il comando seguente alla sezione Launch mode in .sh kafka-run-class

```
-cp $CLASSPATH:"<your AWS GlueSchema Registry base directory>/target/dependency/*"
```

L'URL dovrebbe essere simile a questo:

```
# Launch mode
if [ "x$DAEMON_MODE" = "xtrue" ]; then
  nohup "$JAVA" $KAFKA_HEAP_OPTS $KAFKA_JVM_PERFORMANCE_OPTS $KAFKA_GC_LOG_OPTS
 $KAFKA_JMX_OPTS $KAFKA_LOG4J_OPTS -cp $CLASSPATH:"/Users/johndoe/aws-glue-schema-
registry/target/dependency/*" $KAFKA_OPTS "$@" > "$CONSOLE_OUTPUT_FILE" 2>&1 < /dev/
null &
else
  exec "$JAVA" $KAFKA_HEAP_OPTS $KAFKA_JVM_PERFORMANCE_OPTS $KAFKA_GC_LOG_OPTS
 $KAFKA_JMX_OPTS $KAFKA_LOG4J_OPTS -cp $CLASSPATH:"/Users/johndoe/aws-glue-schema-
registry/target/dependency/*" $KAFKA_OPTS "$@"
fi
```

4. Se usi bash, esegui i seguenti comandi per configurare il tuo CLASSPATH nel bash_profile. Per qualsiasi altra shell, aggiorna l'ambiente di conseguenza.

```
echo 'export GSR_LIB_BASE_DIR=<>' >>~/.bash_profile
echo 'export GSR_LIB_VERSION=1.0.0' >>~/.bash_profile
echo 'export KAFKA_HOME=<your Apache Kafka installation directory>' >>~/.bash_profile
echo 'export CLASSPATH=$CLASSPATH:$GSR_LIB_BASE_DIR/avro-kafkaconnect-converter/
target/schema-registry-kafkaconnect-converter-$GSR_LIB_VERSION.jar:$GSR_LIB_BASE_DIR/
common/target/schema-registry-common-$GSR_LIB_VERSION.jar:$GSR_LIB_BASE_DIR/
```

```
avro-serializer-deserializer/target/schema-registry-serde-$GSR_LIB_VERSION.jar'
 >>~/.bash_profile
source ~/.bash_profile
```

5. (Facoltativo) Se vuoi eseguire il test con un'origine file semplice, clona il connettore dell'origine file.

```
git clone https://github.com/mmolimar/kafka-connect-fs.git
cd kafka-connect-fs/
```

   a. Sotto la configurazione del connettore di origine, modifica il formato dei dati in Avro, il lettore di file in `AvroFileReader` e aggiorna un oggetto Avro di esempio dal percorso del file da cui stai leggendo. Ad esempio:

```
vim config/kafka-connect-fs.properties
```

```
fs.uris=<path to a sample avro object>
policy.regexp=^.*\.avro$
file_reader.class=com.github.mmolimar.kafka.connect.fs.file.reader.AvroFileReader
```

   b. Installa il connettore di origine.

```
mvn clean package
echo "export CLASSPATH=\$CLASSPATH:\"\$(find target/ -type f -name '*.jar'| grep
 '\-package' | tr '\n' ':')\"" >>~/.bash_profile
source ~/.bash_profile
```

   c. Aggiorna le proprietà del sink in *<your Apache Kafka installation directory>*/`config/connect-file-sink.properties`, aggiorna il nome dell'argomento e il nome del file in uscita.

```
file=<output file full path>
topics=<my topic>
```

6. Avvia il connettore di origine (in questo esempio si tratta di un connettore dell'origine file).

```
$KAFKA_HOME/bin/connect-standalone.sh $KAFKA_HOME/config/connect-
standalone.properties config/kafka-connect-fs.properties
```

7. Esegui il connettore sink (in questo esempio si tratta di un connettore sink di file).

```
$KAFKA_HOME/bin/connect-standalone.sh $KAFKA_HOME/config/connect-
standalone.properties $KAFKA_HOME/config/connect-file-sink.properties
```

Per un esempio di utilizzo di Kafka Connect, guarda lo run-local-tests script.sh nella cartella integration-tests nel repository Github per lo Schema Registry. AWS Glue

## Migrazione da un registro degli schemi di terze parti al registro degli schemi di AWS Glue

La migrazione da un registro degli schemi di terze parti al registro degli schemi di AWS Glue ha una dipendenza sull'attuale registro degli schemi di terze parti esistente. Se in un argomento Apache Kafka sono presenti record che sono stati inviati utilizzando un registro degli schemi di terze parti, i consumer hanno bisogno di questo registro per deserializzare tali record. `AWSKafkaAvroDeserializer` offre la possibilità di specificare una classe di deserializzatore secondario che punta al deserializzatore di terze parti e viene utilizzato per deserializzare questi record.

Esistono due criteri per il ritiro di uno schema di terze parti. Innanzitutto, il ritiro può avvenire solo dopo che i record negli argomenti Apache Kafka che utilizzano il registro degli schemi di terze parti non sono più richiesti da e per tutti i consumer. In secondo luogo, il ritiro può avvenire quando gli argomenti di Apache Kafka diventano datati, a seconda del periodo di conservazione specificato per tali argomenti. Tieni presente che se disponi di argomenti con tempo di conservazione infinito, è comunque possibile eseguire la migrazione al registro degli schemi di AWS Glue, ma non sarà possibile ritirare il registro degli schemi di terze parti. Come soluzione alternativa, è possibile utilizzare un'applicazione o Mirror Maker 2 per leggere l'argomento corrente e generarlo in un nuovo argomento nel registro degli schemi di AWS Glue.

Per eseguire la migrazione da un registro degli schemi di terze parti al registro degli schemi di AWS Glue:

1. Crea un registro nel registro degli schemi di AWS Glue o utilizza il registro di default.
2. Arresta il consumer. Modificalo per includere il registro degli schemi di AWS Glue come deserializzatore primario e il registro degli schemi di terze parti come secondario.
   - Imposta le proprietà del consumer. In questo esempio, il secondary_deserializer è impostato su un deserializzatore diverso. Il comportamento è il seguente: il consumer recupera i record da Amazon MSK e tenta di utilizzare prima `AWSKafkaAvroDeserializer`. Se non è in grado di

leggere il magic byte che contiene l'ID dello schema Avro per il registro degli schemi di AWS Glue, `AWSKafkaAvroDeserializer` tenta quindi di utilizzare la classe di deserializzatore fornita nel secondary_deserializer. Le proprietà specifiche del deserializzatore secondario devono anche essere fornite nelle proprietà del consumer, come schema_registry_url_config e specific_avro_reader_config, come illustrato di seguito.

```
consumerProps.setProperty(ConsumerConfig.KEY_DESERIALIZER_CLASS_CONFIG,
 StringDeserializer.class.getName());
consumerProps.setProperty(ConsumerConfig.VALUE_DESERIALIZER_CLASS_CONFIG,
 AWSKafkaAvroDeserializer.class.getName());
consumerProps.setProperty(AWSSchemaRegistryConstants.AWS_REGION,
 KafkaClickstreamConsumer.gsrRegion);
consumerProps.setProperty(AWSSchemaRegistryConstants.SECONDARY_DESERIALIZER,
 KafkaAvroDeserializer.class.getName());
consumerProps.setProperty(KafkaAvroDeserializerConfig.SCHEMA_REGISTRY_URL_CONFIG,
 "URL for third-party schema registry");
consumerProps.setProperty(KafkaAvroDeserializerConfig.SPECIFIC_AVRO_READER_CONFIG,
 "true");
```

3. Riavvia il consumer.
4. Arresta il produttore e puntalo al registro degli schemi di AWS Glue.
   a. Imposta le proprietà del produttore. In questo esempio, il produttore utilizzerà il registro di default ed eseguirà la registrazione automatica delle versioni degli schemi.

```
producerProps.setProperty(ProducerConfig.KEY_SERIALIZER_CLASS_CONFIG,
 StringSerializer.class.getName());
producerProps.setProperty(ProducerConfig.VALUE_SERIALIZER_CLASS_CONFIG,
 AWSKafkaAvroSerializer.class.getName());
producerProps.setProperty(AWSSchemaRegistryConstants.AWS_REGION, "us-east-2");
producerProps.setProperty(AWSSchemaRegistryConstants.AVRO_RECORD_TYPE,
 AvroRecordType.SPECIFIC_RECORD.getName());
producerProps.setProperty(AWSSchemaRegistryConstants.SCHEMA_AUTO_REGISTRATION_SETTING,
 "true");
```

5. (Facoltativo) Sposta manualmente gli schemi e le versioni degli schemi esistenti dall'attuale registro degli schemi di terze parti al registro degli schemi di AWS Glue, nel registro di default nel registro degli schemi di AWS Glue o in un registro specifico non predefinito nel registro degli schemi di AWS Glue. Ciò può essere fatto esportando schemi dai registri di schemi di terze parti

in formato JSON e creando nuovi schemi nel AWS Glue registro degli schemi utilizzando o il. Console di gestione AWS AWS CLI

Questo passaggio può essere importante se è necessario abilitare i controlli di compatibilità con le versioni precedenti dello schema per le versioni dello schema appena create utilizzando AWS CLI and the Console di gestione AWS, o quando i produttori inviano messaggi con un nuovo schema con la registrazione automatica delle versioni dello schema attivata.

6. Avvia il produttore.

# Connessione ai dati

Una AWS Glue connessione è un oggetto Data Catalog che memorizza le credenziali di accesso, le stringhe URI, le informazioni sul cloud privato virtuale (VPC) e altro per un particolare archivio dati. AWS Glue i crawler, i job e gli endpoint di sviluppo utilizzano le connessioni per accedere a determinati tipi di archivi dati. È possibile utilizzare le connessioni sia per le origini che per le destinazioni e riutilizzare la stessa connessione su più crawler o più processi di estrazione, trasformazione e caricamento (ETL).

L'ultima versione dello schema delle AWS Glue connessioni offre un modo unificato per gestire le connessioni dati tra AWS servizi e applicazioni AWS Glue Amazon Athena, come Unified Studio. Amazon SageMaker AI

## Panoramica sull'utilizzo di connettori e connessioni

Una connessione contiene le proprietà necessarie per connettersi a un particolare datastore. Quando crei una connessione, questa viene archiviata in AWS Glue Data Catalog. Scegli un connettore e quindi crea una connessione basata su di esso.

Puoi sottoscrivere connettori per archivi dati non supportati in Marketplace AWS modo nativo e quindi utilizzarli durante la creazione di connessioni. Gli sviluppatori possono anche creare i propri connettori ed è possibile utilizzarli durante la creazione di connessioni.

> **Note**
>
> Le connessioni create utilizzando Marketplace AWS connettori personalizzati o in AWS Glue Studio vengono visualizzate nella AWS Glue console con il tipo impostato su. `UNKNOWN`

I passaggi seguenti descrivono il processo generale di utilizzo dei connettori in AWS Glue Studio:

1. Iscriviti a un connettore in Marketplace AWS oppure sviluppa il tuo connettore e caricalo su AWS Glue Studio. Per ulteriori informazioni, consulta Aggiunta di connettori a AWS Glue Studio.
2. Esamina le informazioni sull'utilizzo del connettore. Puoi trovare queste informazioni nella scheda Usage (Utilizzo) nella pagina prodotto del connettore. Ad esempio, se fai clic sulla scheda Utilizzo in questa pagina di prodotto, AWS Glue Connector for Google BigQuery, puoi vedere nella sezione Risorse aggiuntive un link a un blog sull'utilizzo di questo connettore.

3. Crea una connessione. Puoi scegliere quale connettore utilizzare e fornire informazioni aggiuntive per la connessione, ad esempio le credenziali di accesso, le stringhe URI e le informazioni sul cloud privato virtuale (VPC). Per ulteriori informazioni, consulta Creazione di connessioni per i connettori.
4. Creare un ruolo IAM per il processo. Il processo assume le autorizzazioni del ruolo IAM specificate al momento della creazione. Questo ruolo IAM deve avere le autorizzazioni necessarie per autenticare, estrarre e scrivere dati nei datastore.
5. Crea un processo ETL e configura le proprietà dell'origine dati per il processo ETL. Fornire le opzioni di connessione e le informazioni di autenticazione secondo le istruzioni fornite dal provider di connettori personalizzati. Per ulteriori informazioni, consulta Creazione di processi con connettori personalizzati.
6. Personalizza il processo ETL aggiungendo trasformazioni o datastore aggiuntivi, come descritto in Avvio di processi ETL visivi in AWS Glue Studio.
7. Se usi un connettore per la destinazione dati, configura le proprietà della destinazione dati per il processo ETL. Fornire le opzioni di connessione e le informazioni di autenticazione secondo le istruzioni fornite dal provider di connettori personalizzati. Per ulteriori informazioni, consulta the section called "Creazione di processi con connettori personalizzati".
8. Personalizza l'ambiente di esecuzione configurando le proprietà del processo, come descritto in Modificare le proprietà del processo.
9. Esegui il processo.

# Connessioni unificate

AWS recentemente ha introdotto una nuova funzionalità chiamata "SageMaker LakeHouse Connessioni» o "Connessioni AWS Glue unificate». Questa funzionalità consente di creare connessioni che possono essere utilizzate da più AWS servizi, come AWS Glue e Amazon Athena. Quando crei una fonte di dati in Amazon Athena, noterai una sezione che si riferisce agli input di AWS Glue connessione. In questo caso, Amazon Athena creerà automaticamente una AWS Glue connessione, includendo eventuali proprietà Amazon Athena specifiche nella sezione `AthenaProperties` della connessione.

D'altra parte, se crei una connessione direttamente in AWS Glue, ti verrà richiesto solo di inserire proprietà specifiche per AWS Glue e Apache Spark, che verranno memorizzate nelle sezioni `` e ConnectionProperties `SparkProperties` della connessione.

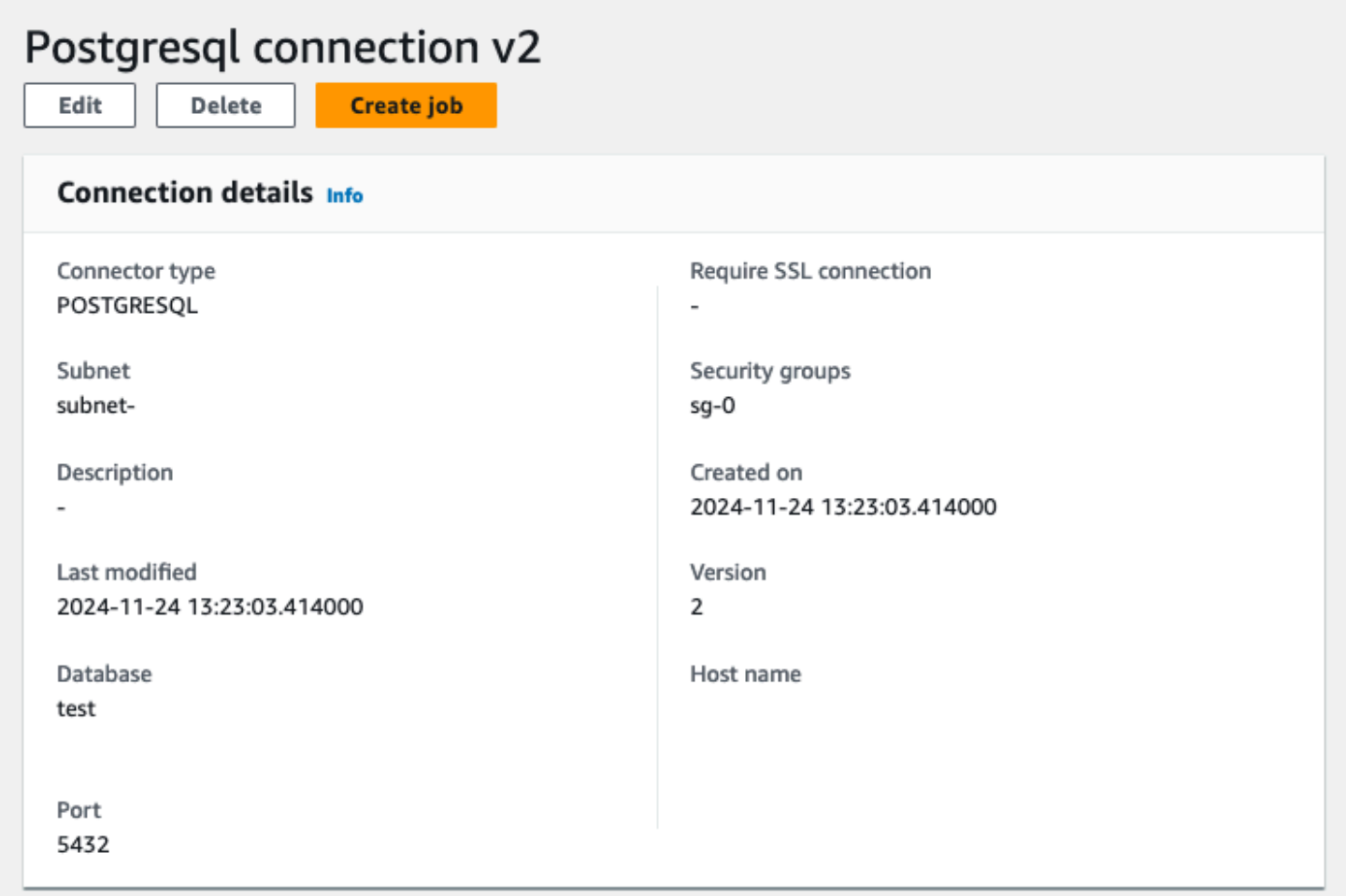
Entrambi questi scenari comportano la creazione di una «connessione unificata», ma le connessioni create in Amazon Athena sono configurate solo per l'uso interno Amazon Athena, mentre le connessioni create in AWS Glue sono configurate solo per l'uso interno. AWS Glue Tuttavia, è possibile aggiornare queste connessioni con le proprietà mancanti (una delle due Amazon Athena o le proprietà Spark) in modo che possano essere utilizzate da entrambi i servizi. Amazon SageMaker AI Unified Studio si occupa di questa operazione automaticamente inserendo tutte le proprietà necessarie (ConnectionProperties``, `AthenaProperties` e `SparkProperties`) sulla AWS Glue connessione, assicurando che la connessione possa essere utilizzata da entrambi AWS Glue e Amazon Athena.

È importante notare che, sebbene le chiamiamo «connessioni unificate», le connessioni create Amazon Athena internamente AWS Glue o singolarmente non sono realmente unificate a meno che non siano configurate correttamente per l'uso da parte di entrambi i servizi. Solo le connessioni create tramite SageMaker Unified Studio sono realmente unificate e utilizzabili da più servizi pronti all'uso.

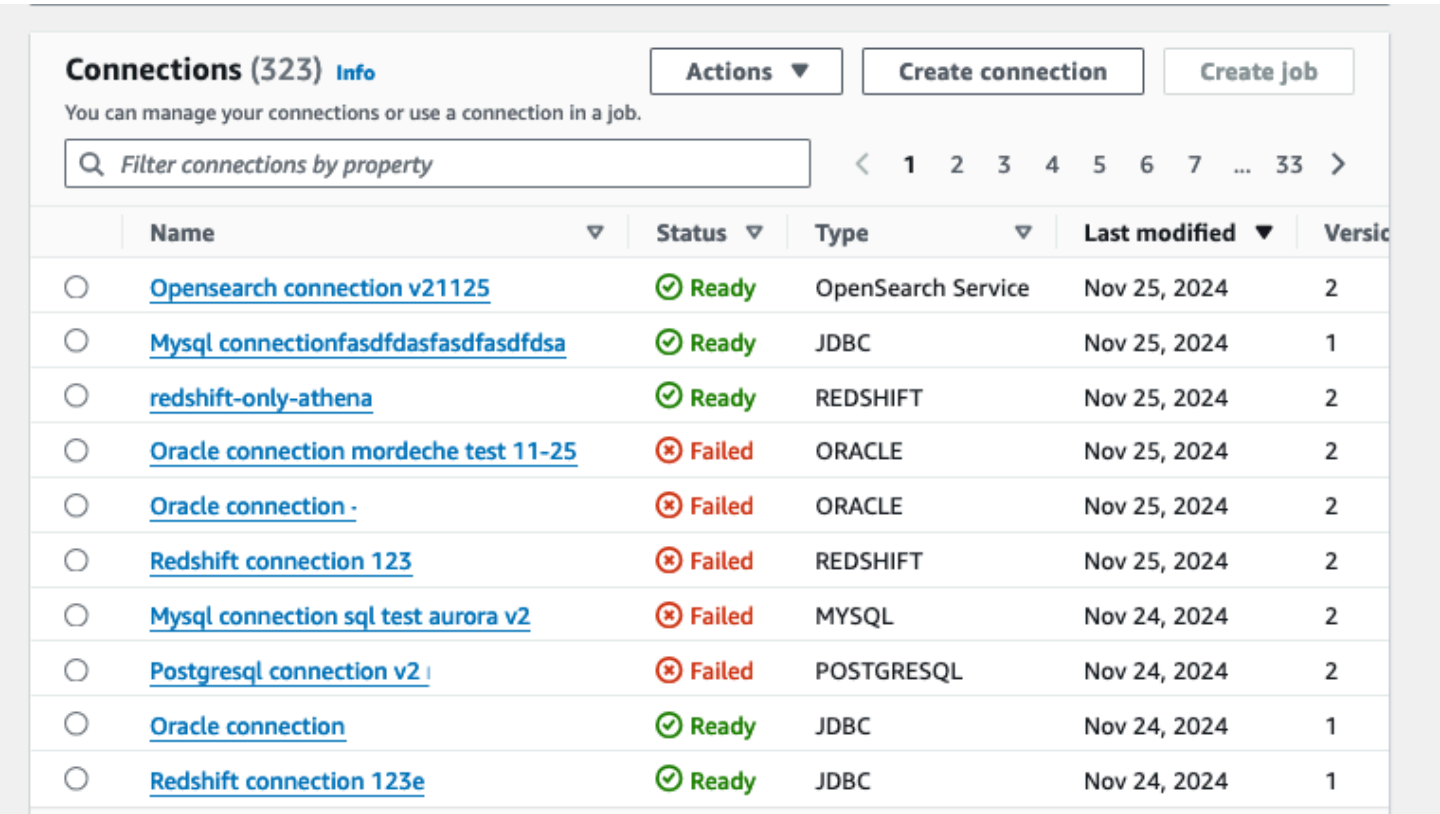
Inoltre, le connessioni create in non AWS Glue sono visibili in Amazon Athena perché Amazon Athena visualizza fonti di dati, che includono un riferimento a una AWS Glue connessione ma non sono la AWS Glue connessione stessa. Allo stesso modo, le connessioni create in non Amazon Athena sono visibili in AWS Glue Studio perché AWS Glue Studio filtra qualsiasi connessione che non è stata configurata con le impostazioni necessarie per AWS Glue.

AWS Glue Studio crea connessioni unificate per impostazione predefinita. Nella AWS Glue console, è possibile visualizzare la versione della connessione nella tabella delle connessioni nella pagina delle connessioni, nella pagina dei dettagli delle connessioni e nella tabella delle connessioni nella pagina dei dettagli del lavoro.

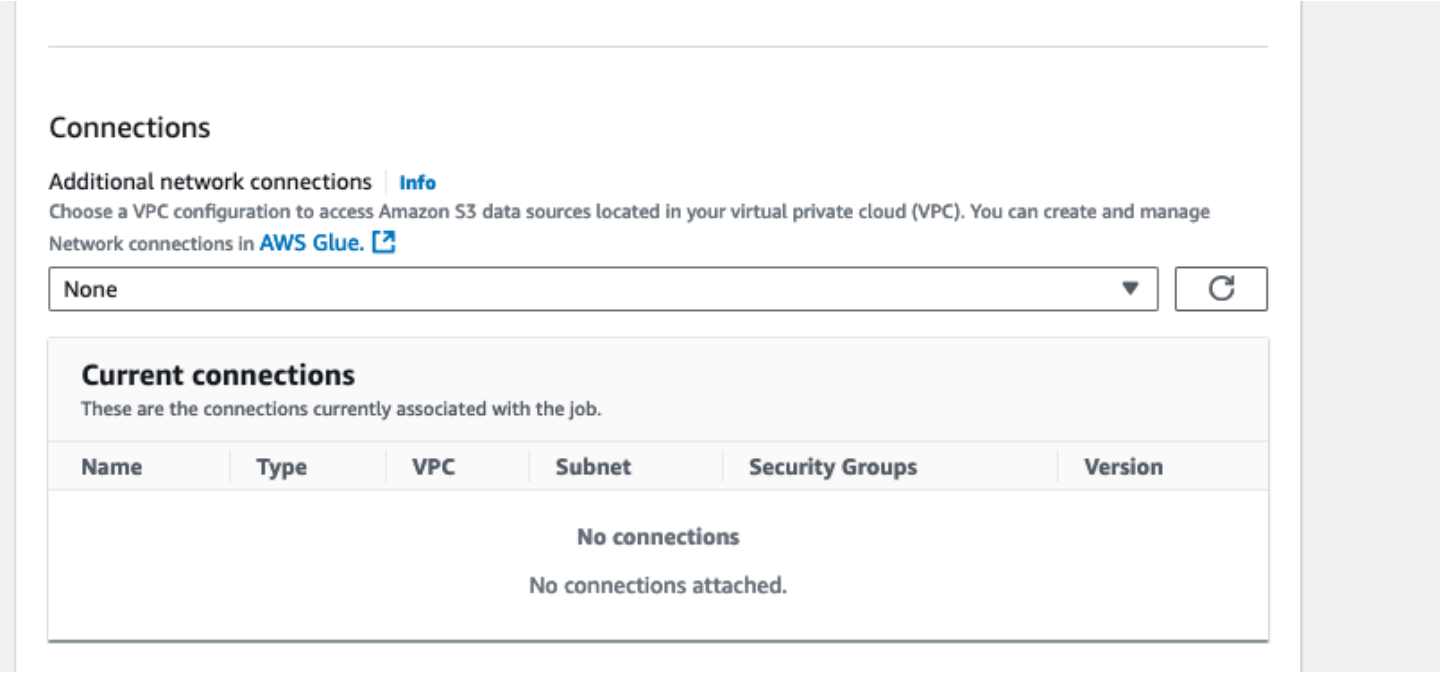
La versione della connessione è visibile nei dettagli della connessione:




La versione della connessione è visibile anche quando si visualizzano tutte le connessioni.




Infine, la versione della connessione è visibile nella scheda Dettagli del processo per un processo.




Con le connessioni di versione 2, sono disponibili le seguenti funzionalità estese di connettività dati:

- Individuazione del tipo di connessione: Supporto per la creazione di connessioni utilizzando modelli standardizzati. AWS Glue rileva automaticamente i tipi di connessione accessibili dall'utente e gli ingressi richiesti e opzionali per un determinato tipo di connessione.
- Riusabilità: definizioni di connessione riutilizzabili su motori e strumenti di elaborazione AWS dati come, e. AWS Glue Amazon Athena Amazon SageMaker AI Le connessioni ora contengono AthenaProperties, SparkProperties, PythonProperties che consentono di specificare proprietà di connessione environment/service specifiche per il calcolo oltre alle proprietà comuni memorizzate in. ConnectionProperties Athena ora crea connessioni AWS Glue specificando proprietà specifiche di Athena nella mappa delle proprietà. AthenaProperties
- Anteprima dei dati: possibilità di sfogliare i metadati e visualizzare in anteprima i dati da origini connesse.
- Metadati dei connettori: è possibile utilizzare connessioni riutilizzabili per scoprire i metadati delle tabelle.
- Segreti collegati al servizio: gli utenti possono fornire le credenziali di autenticazione necessarie OAuth, di base o personalizzate nella richiesta. `CreateConnection` L' CreateConnection API crea un Service Linked Secret nel tuo account e memorizza le credenziali per tuo conto.

## Tipi di autenticazione di supportati

Le connessioni unificate supportano i seguenti tipi di autenticazione:

- BASIC: la maggior parte dei tipi di connessione al database e dei tipi di AWS Glue connessione esistenti supporta l'autenticazione di base, che consiste in un nome utente e una password. In precedenza, la denominazione delle chiavi SecretsManager era specifica del connettore e, ad esempio, poteva essere user, username, UserName, opensearch.net.http.auth.user, ecc. È qui che le connessioni unificate standardizzavano i tipi di connessione di autenticazione di base sulle chiavi USERNAME e PASSWORD.
- OAUTH2— La maggior parte dei tipi di connessione SaaS lanciati di recente supporta il OAuth2 protocollo.
- PERSONALIZZATO: alcuni tipi di connessione dispongono di altri meccanismi di autenticazione come Google, BigQuery in cui gli utenti sono tenuti a fornire il JSON che ricevono da Google. BigQuery

## Considerazioni

Quando si crea una connessione unificata per le origini dati, considerare le seguenti differenze:

- Quando si crea una connessione unificata tramite AWS Glue Studio, le credenziali dell'utente vengono archiviate al Gestione dei segreti AWS posto della connessione stessa. Ciò significa che ora i processi devono accedere a Secrets Manager.
- Se i processi vengono eseguiti in un VPC, richiedono un endpoint VPC o un gateway NAT per accedere a Gestione dei segreti AWS e Secure Token Service (STS), il che comporta costi aggiuntivi.
- Per alcune fonti di dati (Redshift, SQL Server, MySQL, Oracle, PostgreSQL), la creazione di una connessione unificata tramite richiede l'accesso a e. AWS Glue Studio AWS STS Gestione dei segreti AWS Ciò è necessario per stabilire una connessione sicura e recuperare le credenziali necessarie per accedere a queste origini dati all'interno del cloud privato virtuale (VPC).
- La creazione di una connessione unificata tramite AWS Glue Studio richiede un ruolo IAM con autorizzazioni per accedere Gestione dei segreti AWS e gestire le risorse VPC (se si utilizza un VPC):
  - gestore dei segreti: GetSecretValue
  - gestore dei segreti: PutSecretValue
  - gestore dei segreti: DescribeSecret
  - ec2: CreateNetworkInterface
  - ec2: DeleteNetworkInterface
  - ec2: DescribeNetworkInterfaces
  - ec2: DescribeSubnets

# Connessioni disponibili

AWS Glue supporta i seguenti tipi di connessione:

- Adobe Analytics
- Adobe Marketo Engage
- Aurora (supportato se si usa il driver JDBC nativo; non tutte le funzionalità del driver sono utilizzabili)
- Amazon DocumentDB

- Amazon DynamoDB
- Amazon OpenSearch Service, da utilizzare con AWS Glue Spark.
- Amazon Redshift
- Asana
- Azure Cosmos, per l'uso di Azure Cosmos DB per NoSQL con processi ETL AWS Glue
- Azure SQL, da usare con per Spark. AWS Glue
- Blackbaud
- CircleCI
- Datadog
- Docusign Monitor
- Domo
- Dynatrace
- Annunci su Facebook
- Approfondimenti sulla pagina Facebook
- Freshdesk
- Freshsales
- Annunci Google
- Google Analytics 4
- Google BigQuery, da usare con AWS Glue Spark.
- Console di ricerca Google
- Fogli Google
- HubSpot
- Annunci Instagram
- Intercom
- JDBC
- Jira Cloud
- Kafka
- Kustomer
- LinkedIn
- Mailchimp

- Microsoft Dynamics 365 CRM
- Microsoft Teams
- Mixpanel
- Lunedì
- MongoDB
- Atlante MongoDB
- Okta
- Oracle NetSuite
- Paypal
- Pendo
- Pipedrive
- Productboard
- QuickBooks
- API REST (richiede la configurazione del tipo di connessione, vedi Connessioni API REST)
- Salesforce
- Salesforce Commerce Cloud
- Salesforce Marketing Cloud
- Salesforce Marketing Cloud Account Engagement (precedentemente Salesforce Pardot)
- SAP HANA, da utilizzare con AWS Glue Spark.
- SAP OData
- SendGrid
- ServiceNow
- Slack
- Smartsheet
- Annunci Snapchat
- Stripe
- Snowflake, da utilizzare con AWS Glue Spark.
- Teradata Vantage, se utilizzato per Spark. AWS Glue
- Twilio
- Vertica, da utilizzare con Spark. AWS Glue

- WooCommerce
- Zendesk
- Zoho CRM
- Riunioni Zoom
- Varie offerte Amazon Relational Database Service (Amazon RDS).
- Rete (designa una connessione a un'origine dei dati all'interno di un Amazon Virtual Private Cloud [Amazon VPC]).

Con AWS Glue Studio, puoi anche creare una connessione per un connettore. Un connettore è un pacchetto di codice opzionale che facilita l'accesso ai datastore in AWS Glue Studio. Per ulteriori informazioni, consulta Utilizzo di connettori e connessioni con AWS Glue Studio

Per informazioni su come connettersi ai database locali, vedi Come accedere e analizzare gli archivi dati locali utilizzando AWS Glue il sito Web AWS Big Data Blog.

Per creare una connessione con la configurazione VPC utilizzando un ruolo IAM personalizzato, deve disporre delle seguenti azioni di accesso VPC:

- gestore dei segreti: GetSecretValue
- gestore dei segreti: PutSecretValue
- gestore dei segreti: DescribeSecret
- ec2: CreateNetworkInterface
- ec2: DeleteNetworkInterface
- ec2: DescribeNetworkInterfaces
- ec2: DescribeSubnets

## Limitazioni

- Non puoi modificare le connessioni tramite la AWS Glue console se hai creato una connessione v2 utilizzando: AWS Glue APIs
  - Amazon DocumentDB
  - Amazon Aurora
  - MariaDB
  - Atlante MongoDB

- MongoDB

# Connessioni API REST

AWS Glue i connettori coprono un'ampia gamma di fonti di dati AWS sia esterne che esterne. Tuttavia, potrebbero esserci sistemi proprietari o piattaforme emergenti che i AWS Glue connettori nativi non supportano. Il connettore AWS Glue REST fornisce un meccanismo per configurare un connettore personalizzato per qualsiasi fonte di dati dotata di un'API basata su REST che funziona quindi allo stesso modo dei connettori supportati nativamente. AWS Glue Questa funzionalità riduce al minimo la necessità di creare librerie personalizzate o script ETL per accedere a queste fonti di dati.

Per abilitare il connettore API REST, AWS Glue ha introdotto una nuova AWS risorsa: Connection Type. Le seguenti operazioni possono essere utilizzate per gestire la connettività alle fonti di dati basate su API REST:

- RegisterConnectionType- Registra un tipo di connessione in AWS Glue base alla configurazione fornita. Per la struttura delle richieste, vedere RegisterConnectionType.
- DeleteConnectionType- Elimina una configurazione del tipo di connessione in AWS Glue. Vedi DeleteConnectionType.
- DescribeConnectionType- Restituisce tutti i dettagli delle opzioni supportate per un determinato tipo di connessione in AWS Glue. Vedi DescribeConnectionType.
- ListConnectionTypes- Restituisce un elenco di tipi di connessione con dettagli di alto livello su ciò che è supportato per ogni tipo di AWS Glue connessione. Per informazioni, consulta ListConnectionTypes.

## Creazione di un tipo di connessione REST in AWS Glue

Per creare un tipo di connessione REST, utilizza l' AWS CLI AWS SDK o l' AWS API per richiamare l' RegisterConnectionType API. I dettagli di configurazione vengono forniti in base all'API REST a cui ci si connette.

> **Note**
>
> Le connessioni REST non possono essere create utilizzando la AWS Glue console

## Prerequisito

Prima di tentare di configurare il connettore API AWS Glue REST, è necessario acquisire familiarità con il funzionamento interno dell'API REST. È utile disporre della documentazione dell'API REST per l'origine dei dati e di uno strumento per ispezionare la struttura di richiesta/risposta.

È necessario identificare le seguenti proprietà dell'API REST:

- Meccanismo di autenticazione utilizzato per le chiamate effettuate all'endpoint REST. AWS Glue I connettori API REST supportano: autenticazione di base, OAuth2 - ClientCredentials, OAuth2 - Codice di autorizzazione, OAuth2 - JWT e autenticazione personalizzata.
- Entità (oggetti dati) che è possibile trasferire tramite l'endpoint REST e i relativi metadati (come nomi di campo e tipi di dati).
- La struttura di richiesta e risposta HTTP. Questa struttura include il metodo HTTP, l'URL dell'endpoint, le intestazioni, i parametri di query e il formato del corpo della richiesta e del payload della risposta.
- Lo schema di impaginazione. AWS Glue I connettori API REST supportano due schemi di impaginazione: `cursor-based` e `offset-based` l'impaginazione.

## RegisterConnectionType

Registra un login ConnectionType in AWS Glue base alla configurazione fornita. La ConnectionType risorsa è una nuova AWS Glue risorsa che memorizza i dettagli su come le richieste e le risposte vengono interpretate dall'origine dati. Oggi è supportato solo il protocollo REST. Incapsula dettagli sulla fonte dell'API REST come:

- Proprietà di connessione: definisce BaseUrl la connessione all'API REST e qualsiasi parametro di richiesta aggiuntivo necessario per ricevere input durante. AWS Glue CreateConnection
- Configurazione dell'autenticazione: definisce come vengono autenticate le richieste all'API REST. Questa configurazione viene utilizzata quando si crea una AWS Glue connessione per il tipo di connessione.
- Configurazione REST: definisce la configurazione di richiesta e risposta HTTP per leggere i dati dalla fonte dell'API REST. Le seguenti proprietà devono essere configurate a livello globale o di entità:
  - Metodo di richiesta: definisce il metodo HTTP REST.

- Configurazione della risposta: definisce come deve essere interpretata la risposta dell'API (incluse le posizioni dei record o degli errori nel corpo).
- Configurazione della paginazione: definisce la strategia utilizzata dall'API REST per l'impaginazione dei record risultanti e il modo in cui le proprietà di impaginazione devono essere inviate/recuperate.
- Configurazione dell'endpoint di convalida: definisce l'endpoint da chiamare per convalidare una connessione. AWS Glue
- Configurazioni delle entità: definisce le entità REST supportate dal connettore e il relativo schema.

### RegisterConnectionType Sintassi della richiesta

Vedi RegisterConnectionType

Una volta registrato, il tipo di connessione API REST è pronto per essere utilizzato per creare connessioni all'origine dati configurata. Per istruzioni dettagliate, consulta Connessione a REST APIs.

## AWS Glue proprietà di connessione

Questo argomento include informazioni sulle proprietà delle AWS Glue connessioni.

Argomenti


- Proprietà di connessione richieste
- Connessione Amazon DocumentDB
- OpenSearch Connessione al servizio
- Connessione Amazon Redshift
- BigQuery Connessione Google
- Proprietà di connessione MySQL
- proprietà di connessione Oracle
- Proprietà di connessione PostgreSQL
- Connessione SAP HANA
- Connessione Snowflake
- proprietà di connessione di SQL Server

- Connessione Teradata Vantage
- Connessione Vertica
- Connessione Azure Cosmos
- Connessione Azure SQL
- Proprietà della connessione Salesforce
- AWS Glue Proprietà della connessione JDBC
- AWS Glue Proprietà di connessione MongoDB e MongoDB Atlas
- AWS Glue Proprietà della connessione SSL
- Proprietà della connessione Apache Kafka per l'autenticazione client

## Proprietà di connessione richieste

Quando si definisce una connessione sulla AWS Glue console, è necessario fornire valori per le seguenti proprietà:

Nome della connessione

Inserisci un nome univoco per la connessione.

Tipo di connessione

Scegliere JDBC o uno dei tipi di connessione specifici.

Per informazioni dettagliate sul tipo di connessione JDBC, consulta the section called "Proprietà della connessione JDBC"

Scegli Network (Rete) per la connessione a un'origine dati all'interno di un ambiente Amazon Virtual Private Cloud [Amazon VPC]).

A seconda del tipo scelto, la console AWS Glue visualizza altri campi obbligatori. Ad esempio, se selezioni Amazon RDS, devi scegliere il motore di database.

Require SSL connection (Connessione SSL necessaria)

Quando si seleziona questa opzione, è AWS Glue necessario verificare che la connessione al data store sia connessa tramite un Secure Sockets Layer (SSL) affidabile.

Per ulteriori informazioni, incluse le opzioni aggiuntive disponibili quando selezioni questa opzione, consulta the section called "Proprietà della connessione SSL".

Select MSK cluster (Amazon managed streaming for Apache Kafka (MSK) only) (Seleziona cluster MSK [solo Amazon Managed Streaming for Apache Kafka])

Speciifica un cluster MSK di un altro account. AWS

Server di bootstrap Kafka (solo Kafka) URLs

Specifica un elenco separato da virgole del server di bootstrap. URLs Includi il numero di porta. Ad esempio: b-1.vpc-test-2.o4q88o.c6.kafka.us-east-1.amazonaws.com:9094, b-2.vpc-test-2.o4q88o.c6.kafka.us-east-1.amazonaws.com:9094, b-3.vpc-test-2.o4q88o.c6.kafka.us-east-1.amazonaws.com:9094

## Connessione Amazon DocumentDB

Le seguenti proprietà vengono utilizzate per configurare una connessione Amazon DocumentDB.

HOST

(Obbligatorio, STRING) Il nome host dell'endpoint del cluster Amazon DocumentDB.

PORT

(Obbligatorio, STRING) Il numero di porta. Il valore deve essere un numero di porta valido.

ENFORCE_SSL

(Facoltativo, BOOLEAN) Se applicare SSL per la connessione. Il valore predefinito è `false`. Valori consentiti: `true` e `false`.

ROLE_ARN

(Facoltativo, STRING) L'ARN del ruolo IAM utilizzato per accedere Gestione dei segreti AWS e assegnare un indirizzo IP se viene specificato un Amazon VPC.

JDBC_PARAMS

(Facoltativo, STRING) Parametri di connessione aggiuntivi.

Autenticazione

La connessione Amazon DocumentDB supporta il seguente tipo di autenticazione:

- BASIC: si autentica con un nome utente e una password memorizzati in. Gestione dei segreti AWS

Utilizza le seguenti proprietà per configurare una connessione ad Amazon DocumentDB tramite Amazon VPC:

VPC

Scegli il nome del Virtual Private Cloud (VPC) che contiene l'archivio dati. La AWS Glue console elenca tutto VPCs per la regione corrente.

Sottorete

Scegli la sottorete all'interno della VPC che contiene l'archivio dati. La console AWS Glue elenca tutte le sottoreti per l'archivio dati nel VPC.

Gruppi di sicurezza

Scegli i gruppi di sicurezza associati al tuo data store. AWS Glue richiede uno o più gruppi di sicurezza con una regola di origine in entrata che AWS Glue consenta la connessione. La AWS Glue console elenca tutti i gruppi di sicurezza a cui è concesso l'accesso in entrata al tuo VPC. AWS Glue associa questi gruppi di sicurezza all'interfaccia elastica di rete collegata alla sottorete VPC.

## OpenSearch Connessione al servizio

Utilizzate le seguenti proprietà per configurare una connessione di OpenSearch servizio per i lavori AWS Glue ETL.

Endpoint di dominio

Un endpoint OpenSearch di dominio Amazon Service avrà il seguente modulo predefinito, https://search - *`domainName`* -*`unstructuredIdContent`*. *`region`*.es.amazonaws.com. Per ulteriori informazioni sull'identificazione dell'endpoint del tuo dominio, consulta Creazione e gestione dei domini Amazon OpenSearch Service nella documentazione di Amazon OpenSearch Service.

Porta

La porta aperta sull'endpoint.

AWS Segreto

Il nome segreto di un segreto in Gestione dei segreti AWS. AWS Glue si connetterà al OpenSearch Servizio utilizzando le chiavi del tuo segreto.

ROLE_ARN

(Facoltativo, STRING) L'ARN del ruolo IAM utilizzato per accedere Gestione dei segreti AWS e assegnare un indirizzo IP se viene specificato un Amazon VPC.

Utilizza le seguenti proprietà per configurare una connessione a un endpoint di OpenSearch servizio ospitato in Amazon VPC:

VPC

Scegli il nome del Virtual Private Cloud (VPC) che contiene l'archivio dati. La AWS Glue console elenca tutto VPCs per la regione corrente.

Sottorete

Scegli la sottorete all'interno della VPC che contiene l'archivio dati. La console AWS Glue elenca tutte le sottoreti per l'archivio dati nel VPC.

Gruppi di sicurezza

Scegli i gruppi di sicurezza associati al tuo data store. AWS Glue richiede uno o più gruppi di sicurezza con una regola di origine in entrata che AWS Glue consenta la connessione. La AWS Glue console elenca tutti i gruppi di sicurezza a cui è concesso l'accesso in entrata al tuo VPC. AWS Glue associa questi gruppi di sicurezza all'interfaccia elastica di rete collegata alla sottorete VPC.

## Connessione Amazon Redshift

Le seguenti proprietà vengono utilizzate per configurare una connessione Amazon Redshift.

HOST

(Obbligatorio, STRING) Il nome host dell'endpoint del cluster Amazon Redshift.

PORT

(Obbligatorio, STRING) Il numero di porta. Il valore deve essere un numero di porta valido.

DATABASE

(Obbligatorio, STRING) Il nome del database Amazon Redshift a cui connettersi.

ENFORCE_SSL

(Facoltativo, BOOLEAN) Se applicare SSL per la connessione. Il valore predefinito è `false`. Valori consentiti: `true` e `false`.

ROLE_ARN

(Facoltativo, STRING) L'ARN del ruolo IAM utilizzato per accedere Gestione dei segreti AWS e assegnare un indirizzo IP se viene specificato un Amazon VPC.

JDBC_PARAMS

(Facoltativo, STRING) Parametri JDBC aggiuntivi per la connessione.

Autenticazione

La connessione Amazon Redshift supporta i seguenti tipi di autenticazione:

- BASIC: si autentica con un nome utente e una password memorizzati in. Gestione dei segreti AWS
- IAM: si autentica con un accesso basato sui ruoli IAM.

Utilizza le seguenti proprietà per configurare una connessione a un endpoint Amazon Redshift ospitato in Amazon VPC:

VPC

Scegli il nome del Virtual Private Cloud (VPC) che contiene l'archivio dati. La AWS Glue console elenca tutto VPCs per la regione corrente.

Sottorete

Scegli la sottorete all'interno della VPC che contiene l'archivio dati. La console AWS Glue elenca tutte le sottoreti per l'archivio dati nel VPC.

Gruppi di sicurezza

Scegli i gruppi di sicurezza associati al tuo data store. AWS Glue richiede uno o più gruppi di sicurezza con una regola di origine in entrata che AWS Glue consenta la connessione. La AWS Glue console elenca tutti i gruppi di sicurezza a cui è concesso l'accesso in entrata al tuo VPC. AWS Glue associa questi gruppi di sicurezza all'interfaccia elastica di rete collegata alla sottorete VPC.

# BigQuery Connessione Google

Le seguenti proprietà vengono utilizzate per configurare una BigQuery connessione Google utilizzata nei lavori AWS Glue ETL. Per ulteriori informazioni, consulta the section called "BigQuery connessioni".

PROJECT_ID

(Obbligatorio, STRING) L'ID del progetto all'interno di Google Cloud Platform.

ROLE_ARN

(Facoltativo, STRING) L'ARN del ruolo IAM utilizzato per accedere Gestione dei segreti AWS e assegnare un indirizzo IP se viene specificato un Amazon VPC.

CATALOG_CASING_FILTER

(Facoltativo, STRING) Controlla come vengono elaborati i nomi delle entità e quali nomi vengono accettati dall'origine esterna. Questa proprietà si applica solo quando la connessione viene utilizzata con Data Catalog APIs e la federazione dei cataloghi. Non si applica ai lavori AWS Glue ETL o ad altri processi correlati alla connessione APIs. Nessun valore predefinito. Valori consentiti: `LOWERCASE_ONLY` e `UPPERCASE_ONLY`.

Autenticazione

La BigQuery connessione Google supporta il seguente tipo di autenticazione:

- PERSONALIZZATO: esegue l'autenticazione con le credenziali dell'account del servizio Google Cloud archiviate in. Gestione dei segreti AWS

Utilizza le seguenti proprietà quando configuri una BigQuery connessione Google tramite Amazon VPC:

VPC

Scegli il nome del Virtual Private Cloud (VPC) che contiene l'archivio dati. La AWS Glue console elenca tutto VPCs per la regione corrente.

Sottorete

Scegli la sottorete all'interno della VPC che contiene l'archivio dati. La console AWS Glue elenca tutte le sottoreti per l'archivio dati nel VPC.

Gruppi di sicurezza

Scegli i gruppi di sicurezza associati al tuo data store. AWS Glue richiede uno o più gruppi di sicurezza con una regola di origine in entrata che AWS Glue consenta la connessione. La AWS Glue console elenca tutti i gruppi di sicurezza a cui è concesso l'accesso in entrata al tuo VPC. AWS Glue associa questi gruppi di sicurezza all'interfaccia elastica di rete collegata alla sottorete VPC.

## Proprietà di connessione MySQL

Le seguenti proprietà vengono utilizzate per configurare una connessione unificata MySQL.

HOST

(Obbligatorio, STRING) Il nome host dell'endpoint del database MySQL.

PORT

(Obbligatorio, STRING) Il numero di porta. Il valore deve essere un numero di porta valido.

DATABASE

(Obbligatorio, STRING) Il nome del database a cui connettersi.

ENFORCE_SSL

(Facoltativo, BOOLEAN) Se applicare SSL per la connessione. Il valore predefinito è `false`. Valori consentiti: `true` e `false`.

ROLE_ARN

(Facoltativo, STRING) L'ARN del ruolo IAM utilizzato per accedere Gestione dei segreti AWS e assegnare un indirizzo IP se viene specificato un Amazon VPC.

JDBC_PARAMS

(Facoltativo, STRING) Parametri JDBC aggiuntivi per la connessione.

CATALOG_CASING_FILTER

(Facoltativo, STRING) Controlla come vengono elaborati i nomi delle entità e quali nomi vengono accettati dall'origine esterna. Questa proprietà si applica solo quando la connessione viene utilizzata con Data Catalog APIs e la federazione dei cataloghi. Non si applica ai lavori AWS Glue ETL o ad altri processi correlati alla connessione APIs. Nessun valore predefinito. Valori consentiti: `LOWERCASE_ONLY` e `UPPERCASE_ONLY`.

Autenticazione

La connessione MySQL supporta il seguente tipo di autenticazione:

- BASIC: si autentica con un nome utente e una password memorizzati in. Gestione dei segreti AWS

Utilizza le seguenti proprietà per configurare una connessione a un endpoint MySQL ospitato in Amazon VPC:

VPC

Scegli il nome del Virtual Private Cloud (VPC) che contiene l'archivio dati. La AWS Glue console elenca tutto VPCs per la regione corrente.

Sottorete

Scegli la sottorete all'interno della VPC che contiene l'archivio dati. La console AWS Glue elenca tutte le sottoreti per l'archivio dati nel VPC.

Gruppi di sicurezza

Scegli i gruppi di sicurezza associati al tuo data store. AWS Glue richiede uno o più gruppi di sicurezza con una regola di origine in entrata che AWS Glue consenta la connessione. La AWS Glue console elenca tutti i gruppi di sicurezza a cui è concesso l'accesso in entrata al tuo VPC. AWS Glue associa questi gruppi di sicurezza all'interfaccia elastica di rete collegata alla sottorete VPC.

## proprietà di connessione Oracle

Le seguenti proprietà vengono utilizzate per configurare una connessione unificata Oracle.

HOST

(Obbligatorio, STRING) Il nome host dell'endpoint del database Oracle.

PORT

(Obbligatorio, STRING) Il numero di porta. Il valore deve essere un numero di porta valido.

DATABASE

(Obbligatorio, STRING) Il nome del database a cui connettersi.

ENFORCE_SSL

(Facoltativo, BOOLEAN) Se applicare SSL per la connessione. Il valore predefinito è `false`. Valori consentiti: `true` e `false`.

ROLE_ARN

(Facoltativo, STRING) L'ARN del ruolo IAM utilizzato per accedere Gestione dei segreti AWS e assegnare un indirizzo IP se viene specificato un Amazon VPC.

JDBC_PARAMS

(Facoltativo, STRING) Parametri JDBC aggiuntivi per la connessione.

CATALOG_CASING_FILTER

(Facoltativo, STRING) Controlla come vengono elaborati i nomi delle entità e quali nomi vengono accettati dall'origine esterna. Questa proprietà si applica solo quando la connessione viene utilizzata con Data Catalog APIs e la federazione dei cataloghi. Non si applica ai lavori AWS Glue ETL o ad altri processi correlati alla connessione APIs. Il valore predefinito è `UPPERCASE_ONLY`. Valori consentiti: `LOWERCASE_ONLY` e `UPPERCASE_ONLY`.

Autenticazione

La connessione Oracle supporta il seguente tipo di autenticazione:

- BASIC: esegue l'autenticazione con un nome utente e una password memorizzati in. Gestione dei segreti AWS

Utilizza le seguenti proprietà per configurare una connessione a un endpoint Oracle ospitato in Amazon VPC:

VPC

Scegli il nome del Virtual Private Cloud (VPC) che contiene l'archivio dati. La AWS Glue console elenca tutto VPCs per la regione corrente.

Sottorete

Scegli la sottorete all'interno della VPC che contiene l'archivio dati. La console AWS Glue elenca tutte le sottoreti per l'archivio dati nel VPC.

Gruppi di sicurezza

Scegli i gruppi di sicurezza associati al tuo data store. AWS Glue richiede uno o più gruppi di sicurezza con una regola di origine in entrata che AWS Glue consenta la connessione. La AWS Glue console elenca tutti i gruppi di sicurezza a cui è concesso l'accesso in entrata al tuo VPC. AWS Glue associa questi gruppi di sicurezza all'interfaccia elastica di rete collegata alla sottorete VPC.

## Proprietà di connessione PostgreSQL

Le seguenti proprietà vengono utilizzate per configurare una connessione unificata PostgreSQL.

HOST

(Obbligatorio, STRING) Il nome host dell'endpoint del database PostgreSQL.

PORT

(Obbligatorio, STRING) Il numero di porta. Il valore deve essere un numero di porta valido.

DATABASE

(Obbligatorio, STRING) Il nome del database a cui connettersi.

ENFORCE_SSL

(Facoltativo, BOOLEAN) Se applicare SSL per la connessione. Il valore predefinito è `false`. Valori consentiti: `true` e `false`.

ROLE_ARN

(Facoltativo, STRING) L'ARN del ruolo IAM utilizzato per accedere Gestione dei segreti AWS e assegnare un indirizzo IP se viene specificato un Amazon VPC.

JDBC_PARAMS

(Facoltativo, STRING) Parametri JDBC aggiuntivi per la connessione.

CATALOG_CASING_FILTER

(Facoltativo, STRING) Controlla come vengono elaborati i nomi delle entità e quali nomi vengono accettati dall'origine esterna. Questa proprietà si applica solo quando la connessione viene utilizzata con Data Catalog APIs e la federazione dei cataloghi. Non si applica ai lavori AWS Glue

ETL o ad altri processi correlati alla connessione APIs. Il valore predefinito è `LOWERCASE_ONLY`. Valori consentiti: `LOWERCASE_ONLY` e `UPPERCASE_ONLY`.

Autenticazione

La connessione PostgreSQL supporta il seguente tipo di autenticazione:

- BASIC: esegue l'autenticazione con un nome utente e una password memorizzati in. Gestione dei segreti AWS

Utilizza le seguenti proprietà per configurare una connessione a un endpoint PostgreSQL ospitato in Amazon VPC:

VPC

Scegli il nome del Virtual Private Cloud (VPC) che contiene l'archivio dati. La AWS Glue console elenca tutto per la regione corrente. VPCs

Sottorete

Scegli la sottorete all'interno della VPC che contiene l'archivio dati. La console AWS Glue elenca tutte le sottoreti per l'archivio dati nel VPC.

Gruppi di sicurezza

Scegli i gruppi di sicurezza associati al tuo data store. AWS Glue richiede uno o più gruppi di sicurezza con una regola di origine in entrata che AWS Glue consenta la connessione. La AWS Glue console elenca tutti i gruppi di sicurezza a cui è concesso l'accesso in entrata al tuo VPC. AWS Glue associa questi gruppi di sicurezza all'interfaccia elastica di rete collegata alla sottorete VPC.

## Connessione SAP HANA

Le seguenti proprietà vengono utilizzate per configurare una connessione SAP HANA per i lavori ETL. AWS Glue

HOST

(Obbligatorio, STRING) Il nome host dell'endpoint del database SAP HANA.

PORT

(Obbligatorio, STRING) Il numero di porta. Il valore deve essere un numero di porta valido.

DATABASE

(Facoltativo, STRING) Il nome del database SAP HANA a cui connettersi.

ROLE_ARN

(Facoltativo, STRING) L'ARN del ruolo IAM utilizzato per accedere Gestione dei segreti AWS e assegnare un indirizzo IP se viene specificato un Amazon VPC.

JDBC_PARAMS

(Facoltativo, STRING) Parametri JDBC aggiuntivi per la connessione.

CATALOG_CASING_FILTER

(Facoltativo, STRING) Controlla come vengono elaborati i nomi delle entità e quali nomi vengono accettati dall'origine esterna. Questa proprietà si applica solo quando la connessione viene utilizzata con Data Catalog APIs e la federazione dei cataloghi. Non si applica ai lavori AWS Glue ETL o ad altri processi correlati alla connessione APIs. Il valore predefinito è `UPPERCASE_ONLY`. Valori consentiti: `LOWERCASE_ONLY` e `UPPERCASE_ONLY`.

Autenticazione

La connessione SAP HANA supporta il seguente tipo di autenticazione:

- BASIC: esegue l'autenticazione con un nome utente e una password memorizzati in. Gestione dei segreti AWS

Utilizza le seguenti proprietà per configurare una connessione a un endpoint SAP HANA ospitato in Amazon VPC:

VPC

Scegli il nome del Virtual Private Cloud (VPC) che contiene l'archivio dati. La AWS Glue console elenca tutto VPCs per la regione corrente.

Sottorete

Scegli la sottorete all'interno della VPC che contiene l'archivio dati. La console AWS Glue elenca tutte le sottoreti per l'archivio dati nel VPC.

Gruppi di sicurezza

Scegli i gruppi di sicurezza associati al tuo data store. AWS Glue richiede uno o più gruppi di sicurezza con una regola di origine in entrata che AWS Glue consenta la connessione. La AWS Glue console elenca tutti i gruppi di sicurezza a cui è concesso l'accesso in entrata al tuo VPC. AWS Glue associa questi gruppi di sicurezza all'interfaccia elastica di rete collegata alla sottorete VPC.

## Connessione Snowflake

Le seguenti proprietà vengono utilizzate per configurare una connessione Snowflake utilizzata nei job ETL. AWS Glue Quando esegui il crawling di Snowflake, utilizza una connessione JDBC.

HOST

(Obbligatorio, STRING) L'URL dell'host Snowflake. Il valore deve terminare con `.snowflakecomputing.com` o. `.snowflakecomputing.cn` Per ulteriori informazioni sull'endpoint Snowflake URLs, consulta Connessione agli account nella documentazione di Snowflake.

PORT

(Facoltativo, STRING) Il numero di porta per la connessione Snowflake.

DATABASE

(Facoltativo, STRING) Il nome del database Snowflake.

SCHEMA

(Facoltativo, STRING) Il nome dello schema Snowflake.

MAGAZZINO

(Facoltativo, STRING) Il nome del magazzino Snowflake.

JDBC_PARAMS

(Facoltativo, STRING) Parametri JDBC aggiuntivi per la connessione Snowflake.

CATALOG_CASING_FILTER

(Facoltativo, STRING) Controlla come vengono elaborati i nomi delle entità e quali nomi vengono accettati dall'origine esterna. Questa proprietà si applica solo quando la connessione viene utilizzata con Data Catalog APIs e la federazione dei cataloghi. Non si applica ai lavori AWS Glue

ETL o ad altri processi correlati alla connessione APIs. Questa proprietà si applica solo quando la connessione viene utilizzata con Data Catalog APIs e la federazione dei cataloghi. Non si applica ai lavori AWS Glue ETL o ad altri processi correlati alla connessione APIs. Il valore predefinito è `UPPERCASE_ONLY`. Valori consentiti: `LOWERCASE_ONLY` e `UPPERCASE_ONLY`.

ROLE_ARN

(Facoltativo, STRING) L'ARN del ruolo IAM utilizzato per accedere Gestione dei segreti AWS e assegnare un indirizzo IP se viene specificato un Amazon VPC.

Autenticazione

La connessione Snowflake supporta i seguenti tipi di autenticazione:

- BASIC: esegue l'autenticazione con un nome utente e una password memorizzati in. Gestione dei segreti AWS
- OAUTH2— Effettua l'autenticazione con il tipo di OAuth2 concessione del codice di autorizzazione.
- PERSONALIZZATO: esegue l'autenticazione con un nome utente e una chiave privata PEM archiviati in. Gestione dei segreti AWS

Utilizza le seguenti proprietà per configurare una connessione a un endpoint Snowflake ospitato in Amazon VPC utilizzando AWS PrivateLink.

VPC

Scegli il nome del Virtual Private Cloud (VPC) che contiene l'archivio dati. La AWS Glue console elenca tutto VPCs per la regione corrente.

Sottorete

Scegli la sottorete all'interno della VPC che contiene l'archivio dati. La console AWS Glue elenca tutte le sottoreti per l'archivio dati nel VPC.

Gruppi di sicurezza

Scegli i gruppi di sicurezza associati al tuo data store. AWS Glue richiede uno o più gruppi di sicurezza con una regola di origine in entrata che AWS Glue consenta la connessione. La AWS Glue console elenca tutti i gruppi di sicurezza a cui è concesso l'accesso in entrata al tuo VPC. AWS Glue associa questi gruppi di sicurezza all'interfaccia elastica di rete collegata alla sottorete VPC.

# proprietà di connessione di SQL Server

Le seguenti proprietà vengono utilizzate per configurare una connessione unificata di SQL Server.

HOST

(Obbligatorio, STRING) Il nome host dell'endpoint del database SQL Server.

PORT

(Obbligatorio, STRING) Il numero di porta. Il valore deve essere un numero di porta valido.

DATABASE

(Obbligatorio, STRING) Il nome del database a cui connettersi.

ENFORCE_SSL

(Facoltativo, BOOLEAN) Se applicare SSL per la connessione. Il valore predefinito è `false`. Valori consentiti: `true` e `false`.

ROLE_ARN

(Facoltativo, STRING) L'ARN del ruolo IAM utilizzato per accedere Gestione dei segreti AWS e assegnare un indirizzo IP se viene specificato un Amazon VPC.

JDBC_PARAMS

(Facoltativo, STRING) Parametri JDBC aggiuntivi per la connessione.

CATALOG_CASING_FILTER

(Facoltativo, STRING) Controlla come vengono elaborati i nomi delle entità e quali nomi vengono accettati dall'origine esterna. Questa proprietà si applica solo quando la connessione viene utilizzata con Data Catalog APIs e la federazione dei cataloghi. Non si applica ai lavori AWS Glue ETL o ad altri processi correlati alla connessione APIs. Nessun valore predefinito. Valori consentiti: `LOWERCASE_ONLY` e `UPPERCASE_ONLY`.

Autenticazione

La connessione SQL Server supporta il seguente tipo di autenticazione:

- BASIC: esegue l'autenticazione con un nome utente e una password memorizzati in. Gestione dei segreti AWS

Utilizza le seguenti proprietà per configurare una connessione a un endpoint SQL Server ospitato in Amazon VPC:

VPC

Scegli il nome del Virtual Private Cloud (VPC) che contiene l'archivio dati. La AWS Glue console elenca tutto VPCs per la regione corrente.

Sottorete

Scegli la sottorete all'interno della VPC che contiene l'archivio dati. La console AWS Glue elenca tutte le sottoreti per l'archivio dati nel VPC.

Gruppi di sicurezza

Scegli i gruppi di sicurezza associati al tuo data store. AWS Glue richiede uno o più gruppi di sicurezza con una regola di origine in entrata che AWS Glue consenta la connessione. La AWS Glue console elenca tutti i gruppi di sicurezza a cui è concesso l'accesso in entrata al tuo VPC. AWS Glue associa questi gruppi di sicurezza all'interfaccia elastica di rete collegata alla sottorete VPC.

## Connessione Teradata Vantage

Le seguenti proprietà vengono utilizzate per configurare una connessione Teradata Vantage per i lavori ETL. AWS Glue

HOST

(Obbligatorio, STRING) Il nome host dell'endpoint del database Teradata.

PORT

(Obbligatorio, STRING) Il numero di porta. Il valore deve essere un numero di porta valido.

DATABASE

(Obbligatorio, STRING) Il nome del database Teradata a cui connettersi.

ROLE_ARN

(Facoltativo, STRING) L'ARN del ruolo IAM utilizzato per accedere Gestione dei segreti AWS e assegnare un indirizzo IP se viene specificato un Amazon VPC.

JDBC_PARAMS

(Facoltativo, STRING) Parametri JDBC aggiuntivi per la connessione.

Autenticazione

La connessione Teradata Vantage supporta il seguente tipo di autenticazione:

- BASIC: si autentica con un nome utente e una password memorizzati in. Gestione dei segreti AWS

Utilizza le seguenti proprietà per configurare una connessione a un endpoint Teradata Vantage ospitato in Amazon VPC:

VPC

Scegli il nome del Virtual Private Cloud (VPC) che contiene l'archivio dati. La AWS Glue console elenca tutto VPCs per la regione corrente.

Sottorete

Scegli la sottorete all'interno della VPC che contiene l'archivio dati. La console AWS Glue elenca tutte le sottoreti per l'archivio dati nel VPC.

Gruppi di sicurezza

Scegli i gruppi di sicurezza associati al tuo data store. AWS Glue richiede uno o più gruppi di sicurezza con una regola di origine in entrata che AWS Glue consenta la connessione. La AWS Glue console elenca tutti i gruppi di sicurezza a cui è concesso l'accesso in entrata al tuo VPC. AWS Glue associa questi gruppi di sicurezza all'interfaccia elastica di rete collegata alla sottorete VPC.

## Connessione Vertica

Utilizzate le seguenti proprietà per configurare una connessione Vertica per i lavori ETL. AWS Glue

Host Vertica

Il nome host dell'installazione di Vertica.

Porta Vertica

La porta tramite cui è disponibile l'installazione di Vertica.

AWS Segreto

Il nome segreto di un segreto in Gestione dei segreti AWS. AWS Glue si connetterà a Vertica usando le chiavi del tuo segreto.

Utilizza le seguenti proprietà per configurare una connessione a un endpoint Vertica ospitato in Amazon VPC utilizzando.

VPC

Scegli il nome del Virtual Private Cloud (VPC) che contiene l'archivio dati. La AWS Glue console elenca tutto VPCs per la regione corrente.

Sottorete

Scegli la sottorete all'interno della VPC che contiene l'archivio dati. La console AWS Glue elenca tutte le sottoreti per l'archivio dati nel VPC.

Gruppi di sicurezza

Scegli i gruppi di sicurezza associati al tuo data store. AWS Glue richiede uno o più gruppi di sicurezza con una regola di origine in entrata che AWS Glue consenta la connessione. La AWS Glue console elenca tutti i gruppi di sicurezza a cui è concesso l'accesso in entrata al tuo VPC. AWS Glue associa questi gruppi di sicurezza all'interfaccia elastica di rete collegata alla sottorete VPC.

## Connessione Azure Cosmos

Usa le seguenti proprietà per configurare una connessione Azure Cosmos per i processi ETL. AWS Glue

URI dell'endpoint dell'account di Azure Cosmos DB

L'endpoint utilizzato per connettersi ad Azure Cosmos. Per ulteriori informazioni, consulta la documentazione relativa ad Azure.

AWS Segreto

Il nome segreto di un segreto in Gestione dei segreti AWS. AWS Glue si connetterà ad Azure Cosmos usando le chiavi del tuo segreto.

## Connessione Azure SQL

Usa le seguenti proprietà per configurare una connessione Azure SQL per AWS Glue i processi ETL.

URL Azure SQL

L'URL JDBC di un endpoint Azure SQL.

L'elenco deve essere nel seguente formato: `jdbc:sqlserver://databaseServerName:databasePort;databaseName=azuresqlDBname;`.

AWS Glue richiede le seguenti proprietà URL:

- `databaseName`: un database predefinito in Azure SQL a cui connettersi.

Per altre informazioni su JDBC URLs for Azure SQL Managed Instances, consulta la documentazione di Microsoft.

AWS Segreto

Il nome segreto di un segreto in Gestione dei segreti AWS. AWS Glue si connetterà ad Azure SQL usando le chiavi del tuo segreto.

## Proprietà della connessione Salesforce

Le seguenti proprietà vengono utilizzate per configurare una connessione Salesforce.

INSTANCE_URL

(Obbligatorio, STRING) L'URL dell'istanza Salesforce. Il valore deve essere un URL Salesforce valido (ad esempio,). `https://mycompany.my.salesforce.com`

SALESFORCE_ENVIRONMENT

(Obbligatorio, STRING) L'ambiente Salesforce a cui connettersi. Il valore predefinito è `Production`. Valori consentiti: `Production` e `Sandbox`.

ROLE_ARN

(Obbligatorio, STRING) L'ARN del ruolo IAM utilizzato per accedere Gestione dei segreti AWS e assegnare un indirizzo IP se viene specificato un Amazon VPC.

Autenticazione

La connessione Salesforce supporta il seguente tipo di autenticazione:

- OAUTH2— Effettua l'autenticazione con il codice di OAuth2 autorizzazione e i tipi di concessione JWT Bearer.

Utilizza le seguenti proprietà per configurare una connessione Salesforce tramite Amazon VPC:

VPC

Scegli il nome del Virtual Private Cloud (VPC) che contiene l'archivio dati. La AWS Glue console elenca tutto VPCs per la regione corrente.

Sottorete

Scegli la sottorete all'interno della VPC che contiene l'archivio dati. La console AWS Glue elenca tutte le sottoreti per l'archivio dati nel VPC.

Gruppi di sicurezza

Scegli i gruppi di sicurezza associati al tuo data store. AWS Glue richiede uno o più gruppi di sicurezza con una regola di origine in entrata che AWS Glue consenta la connessione. La AWS Glue console elenca tutti i gruppi di sicurezza a cui è concesso l'accesso in entrata al tuo VPC. AWS Glue associa questi gruppi di sicurezza all'interfaccia elastica di rete collegata alla sottorete VPC.

Le seguenti sono opzioni di connessione aggiuntive per Salesforce:

- `ENTITY_NAME`(String) - (Obbligatorio) Utilizzato per lettura/scrittura. Il nome del tuo oggetto in Salesforce.
- `API_VERSION`(String) - (Obbligatorio) Utilizzato per la lettura/scrittura. Versione dell'API Rest di Salesforce da usare.
- `SELECTED_FIELDS`(Elenco<String>) - Predefinito: vuoto (SELECT *). Utilizzato per la lettura. Colonne da selezionare per l'oggetto.
- `FILTER_PREDICATE`(String) - Valore predefinito: vuoto. Utilizzato per la lettura. Dev'essere in formato Spark SQL.
- `QUERY`(String) - Predefinito: vuoto. Utilizzato per la lettura. Query Spark SQL completa.
- `PARTITION_FIELD`(String) - Usato per la lettura. Campo da utilizzare per partizionare la query.
- `LOWER_BOUND`(String) - Usato per la lettura. Un valore limite inferiore inclusivo del campo di partizione scelto.

- `UPPER_BOUND`(String) - Usato per la lettura. Un valore limite superiore esclusivo del campo di partizione scelto.
- `NUM_PARTITIONS`(Numero intero) - Valore predefinito: 1. Utilizzato per la lettura. Numero di partizioni da leggere.
- `IMPORT_DELETED_RECORDS`(String) - Valore predefinito: FALSE. Utilizzato per la lettura. Per ottenere i record eliminati durante l'interrogazione.
- `WRITE_OPERATION`(String) - Valore predefinito: INSERT. Utilizzato per la scrittura. Il valore deve essere INSERT, UPDATE, UPSERT, DELETE.
- `ID_FIELD_NAMES`(String) - Valore predefinito: null. Utilizzato solo per UPSERT.

## AWS Glue Proprietà della connessione JDBC

AWS Glue Studio ora crea connessioni unificate per sorgenti dati MySQL, Oracle, PostgressSQL, Redshift e SQL Server, il che richiede passaggi aggiuntivi per l'accesso alle risorse Secrets Manager e VPC, che possono comportare costi aggiuntivi. È possibile accedere a queste connessioni scegliendo il nome della connessione corrispondente AWS Glue Studio .

Per ulteriori informazioni, consulta Considerazioni.

AWS Glue può connettersi ai seguenti archivi di dati tramite una connessione JDBC:

- Amazon Redshift
- Amazon Aurora
- Microsoft SQL Server
- MySQL
- Oracle
- PostgreSQL
- Snowflake, quando si usano i crawler. AWS Glue
- Aurora (supportato se si utilizza il driver JDBC nativo; non tutte le funzionalità del driver sono utilizzabili)
- Amazon RDS for MariaDB

> **Important**
>
> Al momento, un processo ETL può utilizzare solo una connessione sottorete. Se disponi di più archivi dati in un processo, devono essere nella stessa sottorete o essere accessibili dalla sottorete.
> Se scegli di importare le tue versioni dei driver JDBC per i crawler, AWS Glue i crawler consumeranno risorse nei job e in AWS Glue Amazon S3 per garantire che i driver forniti vengano eseguiti nel tuo ambiente. L'utilizzo aggiuntivo delle risorse si rifletterà nel tuo account. Inoltre, anche se fornisci il tuo driver JDBC, non significa che il crawler sarà in grado di sfruttare tutte le funzionalità del driver. I driver sono limitati alle proprietà descritte nella sezione Defining connections in the Data Catalog.

Di seguito sono riportate le proprietà aggiuntive per il tipo di connessione JDBC.

URL JDBC

Inserisci l'URL per l'archivio dati JDBC. Per la maggior parte dei motori di database, questo campo appare nel seguente formato. In questo formato, sostituisci*protocol*, *host* e con le tue informazioni. *port db_name*

`jdbc:`*`protocol`*`://`*`host`*`:`*`port`*`/`*`db_name`*

A seconda del motore di database, potrebbe essere necessario un altro formato di URL JDBC. Questo formato può avere un utilizzo leggermente diverso dei due punti (:) e della barra (/) o delle diverse parole chiave per specificare i database.

Affinché JDBC si connetta all'archivio dati, è necessario fornire un `db_name` nell'archivio dati. Il `db_name` viene utilizzato per stabilire una connessione di rete con lo `username` e la `password` forniti. Una volta connesso, AWS Glue può accedere ad altri database nell'archivio dati per eseguire un crawler o eseguire un processo ETL.

I seguenti esempi di URL JDBC mostrano la sintassi per diversi motori di database.

- Per la connessione a un archivio dati cluster Amazon Redshift con un database `dev`:

  ```
  jdbc:redshift://xxx.us-east-1.redshift.amazonaws.com:8192/dev
  ```

- Per la connessione a un archivio dati Amazon RDS for MySQL con un database `employee`:

```
jdbc:mysql://xxx-cluster.cluster-xxx.us-east-1.rds.amazonaws.com:3306/
employee
```

- Per la connessione a un archivio dati Amazon RDS for PostgreSQL con un database `employee`:

  ```
  jdbc:postgresql://xxx-cluster.cluster-xxx.us-
  east-1.rds.amazonaws.com:5432/employee
  ```

- Per connettersi a un archivio dati Amazon RDS for Oracle con un nome del servizio `employee`:

  ```
  jdbc:oracle:thin://@xxx-cluster.cluster-xxx.us-
  east-1.rds.amazonaws.com:1521/employee
  ```

  La sintassi per Amazon RDS for Oracle può seguire i seguenti modelli. In questi schemi, sostituisci*host*, *portservice_name*, e *SID* con le tue informazioni.

  - `jdbc:oracle:thin://@`*host*`:`*port*`/`*service_name*
  - `jdbc:oracle:thin://@`*host*`:`*port*`:`*SID*

- Per connettersi a un archivio dati Amazon RDS for Microsoft SQL Server con un database `employee`:

  ```
  jdbc:sqlserver://xxx-cluster.cluster-xxx.us-
  east-1.rds.amazonaws.com:1433;databaseName=employee
  ```

  La sintassi per Amazon RDS for SQL Server può seguire i seguenti modelli. In questi schemi *server_nameport*, sostituisci e *db_name* inserisci le tue informazioni.

  - `jdbc:sqlserver://`*server_name*`:`*port*`;database=`*db_name*
  - `jdbc:sqlserver://`*server_name*`:`*port*`;databaseName=`*db_name*

- Per connetterti a un' Amazon Aurora PostgreSQL istanza del `employee` database, specifica l'endpoint per l'istanza di database, la porta e il nome del database:

  ```
  jdbc:postgresql://employee_instance_1.xxxxxxxxxxxx.us-
  east-2.rds.amazonaws.com:5432/employee
  ```

- Per connetterti a un Amazon RDS for MariaDB data store con un `employee` database, specifica l'endpoint per l'istanza del database, la porta e il nome del database:

  ```
  jdbc:mysql://xxx-cluster.cluster-xxx.aws-
  region.rds.amazonaws.com:3306/employee
  ```

- > **Warning**
  >
  > Le connessioni JDBC Snowflake sono supportate solo dai crawler. AWS Glue Quando si utilizza il connettore Snowflake nei job, utilizzare il tipo di connessione Snowflake. AWS Glue

  Per la connessione a un'istanza Snowflake del database `sample`, specifica l'endpoint per l'istanza Snowflake, l'utente, il nome del database e il nome del ruolo. Inoltre, puoi aggiungere il parametro `warehouse`.

  ```
  jdbc:snowflake://account_name.snowflakecomputing.com/?
  user=user_name&db=sample&role=role_name&warehouse=warehouse_name
  ```

  > **Important**
  >
  > Per le connessioni Snowflake tramite JDBC, viene applicato l'ordine dei parametri nell'URL, che deve seguire l'ordine `user`, `db`, `role_name` e `warehouse`.

- Per connetterti a un'istanza Snowflake del `sample` database con link AWS privato, specifica l'URL JDBC snowflake come segue:

  ```
  jdbc:snowflake://account_name.region.privatelink.snowflakecomputing.com/?
  user=user_name&db=sample&role=role_name&warehouse=warehouse_name
  ```

Username

> **Note**
>
> Ti consigliamo di utilizzare un AWS segreto per memorizzare le credenziali di connessione invece di fornire direttamente il nome utente e la password. Per ulteriori informazioni, consulta Memorizzazione delle credenziali di connessione in Gestione dei segreti AWS.

Fornisci un nome utente che dispone dell'autorizzazione per accedere all'archivio dati JDBC.

Password

Inserisci la password per il nome utente che dispone dell'autorizzazione per accedere all'archivio dati JDBC.

Porta

Inserisci la porta usata nell'URL JDBC per la connessione a un'istanza Amazon RDS Oracle. Questo campo viene visualizzato solo quando l'opzione Require SSL connection (Richiedi connessione SSL) è selezionata per un'istanza Amazon RDS Oracle.

VPC

Scegli il nome del Virtual Private Cloud (VPC) che contiene l'archivio dati. La AWS Glue console elenca tutto VPCs per la regione corrente.

> **Important**
>
> Quando si utilizza una connessione JDBC ospitata all'esterno AWS, ad esempio con dati provenienti da Snowflake, il VPC deve disporre di un gateway NAT che suddivide il traffico in sottoreti pubbliche e private. La sottorete pubblica viene utilizzata per la connessione alla fonte esterna e la sottorete interna viene utilizzata per l'elaborazione da. AWS Glue Per informazioni sulla configurazione di Amazon VPC per le connessioni esterne, consulta le pagine Connect to the internet or other networks using NAT devices e Impostazione di Amazon VPC per le connessioni JDBC ai data store Amazon RDS su JDBC da AWS Glue.

Sottorete

Scegli la sottorete all'interno della VPC che contiene l'archivio dati. La console AWS Glue elenca tutte le sottoreti per l'archivio dati nel VPC.

Gruppi di sicurezza

Scegli i gruppi di sicurezza associati al tuo data store. AWS Glue richiede uno o più gruppi di sicurezza con una regola di origine in entrata che AWS Glue consenta la connessione. La AWS Glue console elenca tutti i gruppi di sicurezza a cui è concesso l'accesso in entrata al tuo VPC. AWS Glue associa questi gruppi di sicurezza all'interfaccia elastica di rete collegata alla sottorete VPC.

Nome della classe del driver JDBC: facoltativo

Fornisci il nome personalizzato della classe del driver JDBC:

- Postgres: org.postgresql.Driver
- MySQL: com.mysql.jdbc.Driver, com.mysql.cj.jdbc.Driver
- Redshift: com.amazon.redshift.jdbc.Driver, com.amazon.redshift.jdbc42.Driver
- Oracle — oracle.jdbc.driver. OracleDriver
- SQL Server: com.microsoft.sqlserver.jdbc. SQLServerAutista

Percorso S3 del driver JDBC: facoltativo

Fornisci la posizione Amazon S3 del driver JDBC personalizzato. Si tratta di un percorso assoluto verso un file .jar. Se desideri fornire dei driver JDBC per connetterti alle tue origini dati per i tuoi database supportati dai crawler, puoi specificare valori per i parametri `customJdbcDriverS3Path` e `customJdbcDriverClassName`.
L'utilizzo di un driver JDBC fornito da un cliente è limitato alle Proprietà di connessione richieste necessarie.

## AWS Glue Proprietà di connessione MongoDB e MongoDB Atlas

Di seguito sono riportate le proprietà aggiuntive per il tipo di connessione MongoDB o MongoDB Atlas.

URL MongoDB

Inserisci l'URL del tuo archivio dati MongoDB o MongoDB Atlas:

- Per MongoDB: mongodb://host:port/database. L'host può essere un nome host, un indirizzo IP o un socket di dominio UNIX. Se la stringa di connessione non specifica una porta, utilizza la porta MongoDB predefinita, 27017.
- Per MongoDB Atlas: mongodb+srv://server.example.com/database. L'host può essere un nome host che corrisponde a un record DNS SRV. Il formato SRV non richiede una porta e utilizzerà la porta MongoDB predefinita, 27017.

Username

> **Note**
>
> Ti consigliamo di utilizzare un AWS segreto per memorizzare le credenziali di connessione invece di fornire direttamente il nome utente e la password. Per ulteriori informazioni, consulta Memorizzazione delle credenziali di connessione in Gestione dei segreti AWS.

Fornisci un nome utente che dispone dell'autorizzazione per accedere all'archivio dati JDBC.

Password

Inserisci la password per il nome utente che dispone dell'autorizzazione per accedere all'archivio dati MongoDB o MongoDB Atlas.

## AWS Glue Proprietà della connessione SSL

Di seguito sono riportati i dettagli della proprietà Require SSL connection (Richiedi connessione SSL).

Se non è richiesta una connessione SSL, AWS Glue ignora gli errori quando utilizza SSL per crittografare una connessione all'archivio dati. Per istruzioni di configurazione, consulta la documentazione dell'archivio dati. Quando selezioni questa opzione, se AWS Glue non è in grado di connettersi, l'esecuzione del processo, del crawler o delle istruzioni ETL in un endpoint di sviluppo ha esito negativo.

> **Note**
>
> Snowflake supporta una connessione SSL per impostazione predefinita, quindi questa proprietà non è applicabile per Snowflake.

Questa opzione è convalidata sul lato client. AWS Glue Per le connessioni JDBC, si connette AWS Glue solo tramite SSL con convalida del certificato e del nome host. Il supporto per la connessione SSL è disponibile per:

- database Oracle
- Microsoft SQL Server

- PostgreSQL
- Amazon Redshift
- MySQL (solo istanze di Amazon RDS)
- Amazon Aurora MySQL (solo istanze di Amazon RDS)
- Amazon Aurora PostgreSQL (solo istanze Amazon RDS)
- Kafka, che include Amazon Streaming gestito per Apache Kafka
- MongoDB

> **Note**
>
> Per permettere a un archivio dati Amazon RDS Oracle di usare l'opzione Require SSL connection (Richiedi connessione SSL), devi creare e allegare un gruppo di opzioni all'istanza Oracle.
>
> 1. Accedi a Console di gestione AWS e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
> 2. Aggiungi un Option group (Gruppo di opzioni) all'istanza Amazon RDS Oracle. Per ulteriori informazioni su come aggiungere un gruppo di opzioni nella console Amazon RDS, consulta la sezione Creazione di un gruppo di opzioni
> 3. Aggiungere un'opzione al gruppo di opzioni per SSL. La porta specificata per SSL viene successivamente utilizzata quando crei un URL di connessione AWS Glue JDBC per l'istanza Amazon RDS Oracle. Per ulteriori informazioni su come aggiungere un'opzione nella console Amazon RDS, consulta Aggiunta di un'opzione a un gruppo di opzioni nella Guida per l'utente di Amazon RDS. Per ulteriori informazioni sull'opzione SSL di Oracle, consulta Oracle SSL nella Guida per l'utente di Amazon RDS.
> 4. Sulla AWS Glue console, crea una connessione all'istanza Amazon RDS Oracle. Nella definizione della connessione, seleziona Require SSL connection (Richiedi connessione SSL). Quando richiesto, inserisci la Port (Porta) utilizzata nell'opzione Amazon RDS Oracle SSL.

Le seguenti proprietà facoltative aggiuntive sono disponibili quando è selezionata l'opzione Require SSL connection (Richiedi connessione SSL) per una connessione:

Certificato JDBC personalizzato in S3

Se disponi di un certificato che stai attualmente utilizzando per la comunicazione SSL con i tuoi database locali o cloud, puoi utilizzare quel certificato per connessioni SSL a sorgenti o destinazioni di AWS Glue dati. Inserisci una posizione Amazon Simple Storage Service (Amazon S3) che contenga un certificato root personalizzato. AWS Glue utilizza questo certificato per stabilire una connessione SSL al database. AWS Glue gestisce solo certificati X.509. Il certificato deve essere codificato DER e fornito in formato PEM codificato Base64.

Se questo campo è lasciato vuoto, viene utilizzato il certificato predefinito.

Stringa di certificato JDBC personalizzata

Immetti le informazioni del certificato specifiche del database JDBC. Questa stringa viene utilizzata per la corrispondenza del dominio o la corrispondenza del nome distinto (DN). Per Oracle Database, questa stringa viene mappata al parametro `SSL_SERVER_CERT_DN` nella sezione di protezione del file `tnsnames.ora`. Per Microsoft SQL Server, questa stringa viene utilizzata come `hostNameInCertificate`.

Di seguito è riportato un esempio per il parametro `SSL_SERVER_CERT_DN` di Oracle Database.

```
cn=sales,cn=OracleContext,dc=us,dc=example,dc=com
```

Posizione del certificato emesso da una CA Kafka privata

Se disponi di un certificato che stai attualmente utilizzando per la comunicazione SSL con il tuo archivio dati Kafka, puoi utilizzare quel certificato con la tua connessione. AWS Glue Questa opzione è obbligatoria per gli archivi dati Kafka e facoltativa per gli archivi dati. Amazon Streaming gestito per Apache Kafka Inserisci una posizione Amazon Simple Storage Service (Amazon S3) che contenga un certificato root personalizzato. AWS Glue utilizza questo certificato per stabilire una connessione SSL all'archivio dati Kafka. AWS Glue gestisce solo certificati X.509. Il certificato deve essere codificato DER e fornito in formato PEM codificato Base64.

Ignora convalida certificato

Seleziona la casella di controllo Ignora la convalida del certificato per saltare la convalida del certificato personalizzato entro. AWS Glue Se scegli di convalidare, AWS Glue convalida l'algoritmo di firma e l'algoritmo a chiave pubblica dell'oggetto per il certificato. Se la convalida del certificato non va a buon fine, qualsiasi processo ETL o crawler che utilizza la connessione ha esito negativo.

Gli unici algoritmi di firma consentiti sono SHA256with RSA, RSA o SHA384with RSA. SHA512with Per l'algoritmo della chiave pubblica oggetto, la lunghezza della chiave deve essere almeno 2048.

Posizione keystore del client Kafka

La posizione Amazon S3 del file keystore del client per l'autenticazione lato client Kafka. Il percorso deve avere il formato s3://.jks. bucket/prefix/filename Deve terminare con il nome del file e l'estensione .jks.

Password del keystore del client Kafka (facoltativa)

La password per accedere al keystore fornito.

Password della chiave del client Kafka (facoltativa)

Un keystore può essere costituito da più chiavi, quindi questa è la password per accedere alla chiave client da utilizzare con la chiave lato server Kafka.

## Proprietà della connessione Apache Kafka per l'autenticazione client

AWS Glue supporta il framework Simple Authentication and Security Layer (SASL) per l'autenticazione quando si crea una connessione Apache Kafka. Il framework SASL supporta vari meccanismi di autenticazione e AWS Glue offre i protocolli SCRAM (nome utente e password), GSSAPI (protocollo Kerberos) e PLAIN.

Viene utilizzato AWS Glue Studio per configurare uno dei seguenti metodi di autenticazione client. Per ulteriori informazioni, vedere Creazione di connessioni per i connettori nella guida per l' AWS Glue Studio utente.

- Nessuno: nessuna autenticazione. Questo è utile se si crea una connessione a scopo di test.
- SASL/SCRAM-SHA-512: la scelta di questo metodo di autenticazione consentirà di specificare le credenziali di autenticazione. Sono disponibili due opzioni:
  - Usa AWS Secrets Manager (consigliato): se selezioni questa opzione, puoi memorizzare il nome utente e la password in AWS Secrets Manager e AWS Glue consentirne l'accesso quando necessario. Specifica il segreto che memorizza le credenziali di autenticazione SSL o SASL. Per ulteriori informazioni, consulta Memorizzazione delle credenziali di connessione in Gestione dei segreti AWS.
  - Inserisci direttamente un nome utente e una password.

- SASL/GSSAPI (Kerberos) - if you select this option, you can select the location of the keytab file, krb5.conf file and enter the Kerberos principal name and Kerberos service name. The locations for the keytab file and krb5.conf file must be in an Amazon S3 location. Since MSK does not yet support SASL/GSSAPI, questa opzione è disponibile solo per i cluster Apache Kafka gestiti dal cliente. Per ulteriori informazioni, consulta la Documentazione di MIT Kerberos: keytab.
- SASL/PLAIN: scegli questo metodo di autenticazione per specificare le credenziali di autenticazione. Sono disponibili due opzioni:
  - Usa AWS Secrets Manager (consigliato): se selezioni questa opzione, puoi memorizzare le tue credenziali in AWS Secrets Manager e consentire AWS Glue l'accesso alle informazioni quando necessario. Specifica il segreto che memorizza le credenziali di autenticazione SSL o SASL.
  - Fornisci direttamente nome utente e password.
- Autenticazione client SSL: selezionando questa opzione, puoi selezionare la posizione del keystore client Kafka navigando su Amazon S3. Facoltativamente, puoi inserire la password del keystore del client Kafka e la password della chiave del client Kafka.

# Memorizzazione delle credenziali di connessione in Gestione dei segreti AWS

Si consiglia di utilizzarle Gestione dei segreti AWS per fornire le credenziali di connessione per il data store. L'utilizzo di Secrets Manager in questo modo consente di AWS Glue accedere al segreto in fase di esecuzione per i lavori ETL e le esecuzioni dei crawler e aiuta a proteggere le credenziali.

Prerequisiti

Per utilizzare Secrets Manager con AWS Glue, devi concedere al tuo ruolo IAM l' AWS Glueautorizzazione a recuperare valori segreti. La policy AWS gestita `AWSGlueServiceRole` non include le Gestione dei segreti AWS autorizzazioni. Per le policy IAM di esempio, consulta Esempio: Autorizzazione per recuperare valori segreti nella Guida per l'utente di Gestione dei segreti AWS.

In base all'impostazione della rete, potrebbe essere necessario anche creare un endpoint VPC per stabilire una connessione privata tra il VPC e Secrets Manager. Per ulteriori informazioni, consulta Utilizzo di un endpoint VPC Gestione dei segreti AWS.

Per creare un segreto per AWS Glue

1. Segui le istruzioni in Creazione e gestione di segreti nella Guida per l'utente di Gestione dei segreti AWS . L'esempio JSON seguente mostra come specificare le credenziali nella scheda Plaintext quando crei un segreto per AWS Glue.

```
{
  "username": "EXAMPLE-USERNAME",
  "password": "EXAMPLE-PASSWORD"
}
```

2. Associa il tuo segreto a una connessione utilizzando l' AWS Glue Studio interfaccia. Per ulteriori informazioni, consulta la pagina Creating connections for connectors nella Guida per l'utente di AWS Glue Studio .

# Aggiungere una AWS Glue connessione

Puoi connetterti alle fonti di dati in AWS Glue for Spark a livello di codice. Per ulteriori informazioni, consulta Tipi e opzioni di connessione per ETL in AWS Glue per Spark

Puoi anche usare la AWS Glue console per aggiungere, modificare, eliminare e testare le connessioni. Per informazioni sulle AWS Glue connessioni, consultaConnessione ai dati.

Argomenti

- Connessione ad Adobe Analytics
- Connessione ad Adobe Marketo Engage
- Connessione ad Amazon Redshift in AWS Glue Studio
- Connessione ad Asana
- Connessione ad Azure Cosmos DB in AWS Glue Studio
- Connessione ad Azure SQL in AWS Glue Studio
- Connessione a Edge NXT di Blackbaud Raiser
- Connessione a CircleCI
- Connessione a Datadog
- Connessione a Docusign Monitor
- Connessione a Domo
- Connessione a Dynatrace

- Connessione ad Annunci su Facebook
- Connessione a Facebook Page Insights
- Connessione a Freshdesk
- Connessione a Freshsales
- Connessione a Google Ads
- Connessione a Google Analytics 4
- Connessione a Google BigQuery in AWS Glue Studio
- Connessione a Google Search Console
- Connessione a Google Sheets
- Connessione a HubSpot
- Connessione a Instagram Ads
- Connessione a Intercom in AWS Glue Studio
- Connessione a Jira Cloud
- Connessione a Kustomer
- Connessione a LinkedIn
- Connessione a Mailchimp
- Connessione a Microsoft Dynamics 365 CRM
- Connessione a Microsoft Teams
- Connessione a Mixpanel
- Connessione al Monday
- Connessione a MongoDB in AWS Glue Studio
- Connessione a Oracle NetSuite
- Connessione al OpenSearch servizio in AWS Glue Studio
- Connessione a Okta
- Connessione a PayPal
- Connessione a Pendo
- Connessione a Pipedrive
- Connessione a Productboard
- Connessione a QuickBooks
- Connessione a un'API REST

- Connessione a Salesforce
- Connessione a Salesforce Marketing Cloud
- Connessione a Salesforce Commerce Cloud
- Connessione a Salesforce Marketing Cloud Account Engagement
- Connessione a SAP HANA in AWS Glue Studio
- Connessione a SAP OData
- Connessione a SendGrid
- Connessione a ServiceNow
- Connessione a Slack in AWS Glue Studio
- Connessione a Smartsheet
- Connessione a Snapchat Ads in AWS Glue Studio
- Connessione a Snowflake in AWS Glue Studio
- Connessione a Stripe in AWS Glue Studio
- Connessione a Teradata Vantage in AWS Glue Studio
- Connessione a Twilio
- Connessione a Vertica in AWS Glue Studio
- Connessione a WooCommerce
- Connessione a Zendesk
- Connessione a Zoho CRM
- Connessione a Zoom Meetings
- Aggiunta di una connessione JDBC utilizzando i propri driver JDBC

## Connessione ad Adobe Analytics

Adobe Analytics è una piattaforma di analisi dati affidabile che raccoglie dati da esperienze digitali multicanale che supportano il percorso del cliente e fornisce strumenti per l'analisi dei dati. Si tratta di una piattaforma comunemente usata dai professionisti del marketing e dagli analisti aziendali per scopi di analisi aziendale. Se sei un utente di Adobe Analytics, puoi connetterti AWS Glue al tuo account Adobe Analytics. Quindi, puoi usare Adobe Analytics come origine dati nei tuoi processi ETL. Esegui questi processi per trasferire dati tra Adobe Analytics e AWS servizi o altre applicazioni supportate.

Argomenti

- AWS Glue supporto per Adobe Analytics
- Policy contenenti le operazioni API per creare e utilizzare le connessioni
- Configurazione di Adobe Analytics
- Configurazione delle connessioni Adobe Analytics
- Lettura da entità di Adobe Analytics
- Opzioni di connessione di Adobe Analytics
- Creazione di un account Adobe Analytics
- Limitazioni


## AWS Glue supporto per Adobe Analytics

AWS Glue supporta Adobe Analytics come segue:

Supportato come origine?

Sì. Puoi utilizzare i lavori AWS Glue ETL per interrogare i dati da Adobe Analytics.

Supportato come destinazione?

No.

Versioni dell'API Adobe Analytics supportate

v2.0

## Policy contenenti le operazioni API per creare e utilizzare le connessioni

La seguente politica di esempio descrive le AWS autorizzazioni richieste per la creazione e l'utilizzo delle connessioni. Se si sta creando un nuovo ruolo, creare una policy che contenga quanto segue:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
```

```
        "glue:ListConnectionTypes",
        "glue:DescribeConnectionType",
        "glue:RefreshOAuth2Tokens",
        "glue:ListEntities",
        "glue:DescribeEntity"
      ],
      "Resource": "*"
    }
  ]
}
```

In alternativa, se non desideri utilizzare il metodo precedente, utilizza le seguenti policy IAM:

- AWSGlueServiceRole — Garantisce l'accesso alle risorse che diversi AWS Glue processi richiedono per l'esecuzione per conto dell'utente. Queste risorse includono Amazon S3 AWS Glue, IAM, CloudWatch Logs e Amazon EC2. Se segui la convenzione di denominazione delle risorse specificata in questa politica, AWS Glue i processi dispongono delle autorizzazioni necessarie. Questa policy è in genere collegata ai ruoli specificati quando si definiscono crawler, processi ed endpoint di sviluppo.
- AWSGlueConsoleFullAccess — Concede l'accesso completo alle AWS Glue risorse quando un'identità a cui è allegata la politica utilizza la AWS console di gestione. Se si segue la convenzione per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questa policy è in genere associata agli utenti della AWS Glue console.

## Configurazione di Adobe Analytics

Prima di poter AWS Glue utilizzare il trasferimento da Adobe Analytics, devi soddisfare i seguenti requisiti:

Requisiti minimi

- Avere un account Adobe Analytics con e-mail e password. Per ulteriori informazioni sulla creazione di account, consulta Creating a Adobe Analytics account.
- L'account Adobe Analytics è abilitato per l'accesso alle API. L'accesso all'API è abilitato per impostazione predefinita per le edizioni Select, Prime e Ultimate.

Se soddisfi questi requisiti, sei pronto per connetterti AWS Glue al tuo account Adobe Analytics. Per le connessioni tipiche, non è necessario fare nient'altro in Adobe Analytics.

## Configurazione delle connessioni Adobe Analytics

Adobe Analytics supporta il tipo di concessione `AUTHORIZATION_CODE` per `OAuth2`.

Questo tipo di concessione è considerato `OAuth` "a tre gambe" in quanto si basa sul reindirizzamento degli utenti al server di autorizzazione di terzi per l'autenticazione. Gli utenti possono scegliere di creare la propria app connessa in Adobe Analytics e fornire il proprio ID client e il segreto del client durante la creazione di connessioni tramite la AWS Glue console. In questo scenario, verranno comunque reindirizzati ad Adobe Analytics per accedere e autorizzare AWS Glue ad accedere alle risorse.

Questo tipo di concessione genera un token di aggiornamento e un token di accesso. Il token di accesso dura poco tempo e può essere aggiornato automaticamente senza l'interazione dell'utente utilizzando il token di aggiornamento.

Per la documentazione pubblica di Adobe Analytics sulla creazione di un'app connessa per il OAuth flusso AUTHORIZATION_CODE, consulta Adobe Analytics. APIs

Per configurare una connessione Adobe Analytics:

1. In AWS Secrets Manager, crea un segreto con i seguenti dettagli:

   Per le app connesse gestite dal cliente: il segreto deve contenere l'app connessa Consumer Secret con `USER_MANAGED_CLIENT_APPLICATION_CLIENT_SECRET` come chiave.

   > **Note**
   >
   > È necessario creare un segreto per ogni connessione in AWS Glue.

2. In AWS Glue Studio, crea una connessione in Connessioni dati seguendo i passaggi seguenti:
   a. In Tipo di connessione, seleziona Adobe Analytics.
   b. Fornisci `x_api_key, instanceUrl` di Adobe Analytics a cui desideri connetterti.
   c. Seleziona il ruolo IAM per il quale AWS Glue può assumere e dispone delle autorizzazioni per le seguenti azioni:

      JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
```

```
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:DescribeSecret",
        "secretsmanager:GetSecretValue",
        "secretsmanager:PutSecretValue",
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DeleteNetworkInterface"
      ],
      "Resource": "*"
    }
  ]
}
```

d. Seleziona quello `secretName` che desideri utilizzare per questa connessione per AWS Glue inserire i token.

e. Selezionare le opzioni di rete se si desidera utilizzare la propria rete.

3. Concedi il permesso di lettura `secretName` al ruolo IAM associato al tuo AWS Glue lavoro.

## Lettura da entità di Adobe Analytics

Prerequisiti

Un oggetto di Adobe Analytics da cui desideri leggere. Consulta la tabella delle entità supportate qui di seguito per verificare quelle disponibili.

Entità supportate

| Entità | Può essere Filtrato | Supporta Limite | Supporta Ordina per | Supporta Seleziona * | Supporta Partizionamento |
|---|---|---|---|---|---|
| Annotazione | Sì | Sì | Sì | Sì | No |
| Metriche calcolate | Sì | Sì | Sì | Sì | No |
| Funzione delle | Sì | No | No | Sì | No |

| Entità | Può essere Filtrato | Supporta Limite | Supporta Ordina per | Supporta Seleziona * | Supporta Partizionamento |
|---|---|---|---|---|---|
| metriche calcolate | | | | | |
| Condivisioni di metadati dei componenti | Sì | Sì | No | Sì | No |
| Intervalli di date | Sì | Sì | No | Sì | No |
| Dimensioni | Sì | No | No | Sì | No |
| Metriche | Sì | No | No | Sì | No |
| Progetti | Sì | No | No | Sì | No |
| Segnala l'elemento principale | Sì | Sì | No | Sì | No |
| Segmenti | Sì | Sì | Sì | Sì | No |
| Log di utilizzo | Sì | Sì | No | Sì | No |

Esempio

```
adobeAnalytics_read = glueContext.create_dynamic_frame.from_options(
     connection_type="adobeanalytics",
     connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "annotation/ex*****",
        "API_VERSION": "v2.0"
 })
```

Dettagli sull'entità e sul campo di Adobe Analytics

- Annotazioni
- Metriche calcolate
- Metadati dei componenti
- Intervalli di date
- Dimensioni
- Parametri
- Progetti
- Report
- Segmenti
- Utenti
- Log di utilizzo

## Opzioni di connessione di Adobe Analytics

Di seguito sono riportate le opzioni di connessione di Adobe Analytics:

- `ENTITY_NAME` (stringa): (obbligatorio) utilizzato per lettura/scrittura. Il nome dell'oggetto in Adobe Analytics.
- `API_VERSION` (stringa): (obbligatorio) utilizzato per lettura/scrittura. La versione dell'API Rest di Adobe Analytics da utilizzare. Ad esempio: v2.0.
- `X_API_KEY` (stringa): (obbligatorio) utilizzato per lettura/scrittura. È necessario autenticare lo sviluppatore o l'applicazione che effettua richieste all'API.
- `SELECTED_FIELDS` (List<String>): predefinito: vuoto (SELECT *). Utilizzato per la lettura. Colonne da selezionare per l'oggetto.
- `FILTER_PREDICATE`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Dev'essere in formato Spark SQL.
- `QUERY`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Query Spark SQL completa.

## Creazione di un account Adobe Analytics

1. Effettuare la registrazione al programma partner Exchange tramite l'accesso al programma Adobe Partner.
2. Scegliere Iscriviti al programma Exchange.

3. Effettuare la registrazione o creare un account utilizzando l'indirizzo e-mail aziendale.
4. Dalla finestra dei suggerimenti, selezionare l'azienda adeguata che dispone di un abbonamento al prodotto Adobe Analytics.
5. Assicurarsi che l'account venga registrato all'interno di un'organizzazione valida (dall'elenco disponibile) che dispone di un abbonamento attivo ad Adobe Analytics.
6. Dopo l'approvazione da parte dell'amministrazione dell'azienda, attivare l'account cliccando sul link presente nell'e-mail di approvazione.

Verificare se l'account creato ha accesso al servizio Adobe Analytics.

1. Accedere alla Adobe Admin Console.
2. Verificare il nome dell'organizzazione nell'angolo in alto a destra della pagina per assicurarsi di aver effettuato l'accesso all'azienda corretta.
3. Selezionare Prodotti e verificare se è disponibile Adobe Analytics.

> **Note**
>
> Se nessuna organizzazione è disponibile o il prodotto Adobe Analytics è disattivato o non è disponibile, è probabile che il tuo account non sia associato a un'organizzazione che non and/or ha un abbonamento attivo ad Adobe Analytics. Contattare l'amministratore di sistema per richiedere l'accesso a questo servizio per l'account.

Creazione di un progetto credenziali **`OAuth2.0`**

1. Accedi all'account Adobe Analytics in cui desideri creare l'app OAuth 2.0.
2. Selezionare Progetto, poi Crea un nuovo progetto.
3. Per aggiungere un progetto, selezionare Aggiungi al progetto, poi API.
4. Selezionare l'API Adobe Analytics.
5. Selezionare OAUTH come autenticazione dell'utente.
6. Selezionare Web come 0AUTH e indicare l'URI di reindirizzamento.

   Per l'URI di reindirizzamento e il relativo pattern, fare riferimento a quanto segue:

- URI di reindirizzamento `OAuth 2.0` predefinito: un URI di reindirizzamento predefinito è l'URL della pagina a cui Adobe accederà durante il processo di autenticazione. Ad esempio, `https://ap-southeast-2.console.aws.amazon.com/appflow/oauth`
- OAuth Schema URI di reindirizzamento 2.0: un pattern URI di reindirizzamento è un percorso URI (o elenco di percorsi separati da virgole) a cui Adobe può reindirizzare (se richiesto) quando il flusso di accesso è completo. Ad esempio, `https://ap-southeast-2\\.console\\.aws\\.amazon\\.com`

7. Aggiungere i seguenti ambiti:
   - `openid`
   - `read_organizations`
   - `additional_info.projectedProductContext`
   - `additional_info.job_function`
8. Scegliere Salva credenziali.
9. Dopo aver creato l'app, copiare i valori `Client ID` e `Client Secret` in un file di testo.

## Limitazioni

Di seguito sono riportate le limitazioni per il connettore Adobe Analytics:

- Adobe Analytics non supporta il partizionamento basato su campi e su record. Il partizionamento basato su campi non è supportato poiché non è possibile eseguire query sui campi partizionati. Il partizionamento basato su record non è supportato poiché non è previsto l'utilizzo di un valore di "offset" per l'impaginazione.
- Nell'entità `Report Top Item`, i parametri di query `startDate` e `endDate` non funzionano come previsto. La risposta non viene filtrata in base a questi parametri, con conseguenti problemi di filtraggio e flusso incrementale per quest'entità.
- Per le entità `Annotation`, `Calculated Metrics`, `Calculated Metrics Function`, `Date Ranges`, `Dimension`, `Metric`, `Project`, `Report Top Items` e `Segment`, il parametro di query `locale` specifica quale linguaggio deve essere utilizzato per le sezioni localizzate delle risposte e non filtra i record. Ad esempio, `locale="ja_JP"` mostrerà i dati in giapponese.
- Entità `Report Top Item`: attualmente, il filtro per i campi `dateRange` e `lookupNoneValues` non funziona.

- Entità `Segment`: con il valore del filtro `includeType="templates"`, i filtri per gli altri campi non funzionano.
- Entità `Date Range`: il filtro per il campo `curatedRsid` non funziona.
- Entità `Metric entity`: il filtro per il campo segmentabile con valore "false" restituisce risultati per i valori true e false.

## Connessione ad Adobe Marketo Engage

Adobe Marketo Engage è una piattaforma di automazione del marketing che consente agli esperti di marketing di gestire programmi e campagne multicanale personalizzati per clienti potenziali ed esistenti.

Argomenti


- AWS Glue supporto per Adobe Marketo Engage
- Policy contenenti le operazioni API per creare e utilizzare le connessioni
- Configurazione di Adobe Marketo Engage
- Configurazione delle connessioni Adobe Marketo Engage
- Lettura dalle entità Adobe Marketo Engage
- Scrivere a entità Adobe Marketo Engage
- Opzioni di connessione Adobe Marketo Engage
- Limitazioni e note per il connettore Adobe Marketo Engage


### AWS Glue supporto per Adobe Marketo Engage

AWS Glue supporta Adobe Marketo Engage come segue:

Supportato come origine?

Sì. Puoi utilizzare i lavori AWS Glue ETL per interrogare i dati da Adobe Marketo Engage.

Supportato come destinazione?

Sì. Puoi utilizzare i lavori AWS Glue ETL per scrivere dati su Adobe Marketo.

Versioni dell'API Adobe Marketo Engage supportate

Sono supportate le seguenti versioni dell'API Adobe Marketo Engage:

- v1

Per il supporto delle entità per le versioni specifiche, consulta Supported entities for Source.

### Policy contenenti le operazioni API per creare e utilizzare le connessioni

La seguente policy di esempio descrive le autorizzazioni AWS IAM richieste per la creazione e l'utilizzo delle connessioni. Se si sta creando un nuovo ruolo, creare una policy che contenga quanto segue:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:ListConnectionTypes",
        "glue:DescribeConnectionType",
        "glue:RefreshOAuth2Tokens",
        "glue:ListEntities",
        "glue:DescribeEntity"
      ],
      "Resource": "*"
    }
  ]
}
```

In alternativa, se non desideri utilizzare il metodo precedente, utilizza le seguenti policy IAM gestite:

- AWSGlueServiceRole— Garantisce l'accesso alle risorse che AWS Glue i vari processi richiedono per l'esecuzione per conto dell'utente. Queste risorse includono AWS Glue Amazon S3, IAM, CloudWatch Logs e Amazon. EC2 Se segui la convenzione di denominazione delle risorse specificata in questa politica, AWS Glue i processi dispongono delle autorizzazioni richieste. Questa policy è in genere collegata ai ruoli specificati quando si definiscono crawler, processi ed endpoint di sviluppo.
- AWSGlueConsoleFullAccess— Concede l'accesso completo alle AWS Glue risorse quando un'identità a cui è allegata la policy utilizza la AWS console di gestione. Se si segue la convenzione

per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questa policy è in genere associata agli utenti della AWS Glue console.

## Configurazione di Adobe Marketo Engage

Prima di poter AWS Glue utilizzare il trasferimento di dati da Adobe Market a Engage, devi soddisfare questi requisiti:

Requisiti minimi

I requisiti minimi sono i seguenti:

- Un account Adobe Marketo Engage con le credenziali del cliente.
- Il tuo account Adobe Marketo Engage dispone dell'accesso all'API con una licenza valida.

Se soddisfi questi requisiti, sei pronto per connetterti al tuo account Adobe AWS Glue Marketo Engage. Per le connessioni tipiche, non è necessario fare nient'altro in Adobe Marketo Engage.

Ottenere le credenziali 2.0 OAuth

Per ottenere le credenziali API ed effettuare chiamate autenticate alla tua istanza, consulta Adobe Marketo Rest API

## Configurazione delle connessioni Adobe Marketo Engage

Adobe Marketo Engage supporta il tipo di concessione CLIENT CREDENTIALS per. OAuth2

- Questo tipo di concessione è considerato OAuth 2.0 a 2 gambe in quanto viene utilizzato dai client per ottenere un token di accesso al di fuori del contesto di un utente. AWS Glue è in grado di utilizzare l'ID client e il client secret per autenticare Adobe Marketo Engage APIs , forniti dai servizi personalizzati definiti dall'utente.
- Ogni servizio personalizzato è di proprietà di un utente che utilizza solo API e dispone di una serie di ruoli e autorizzazioni che autorizzano il servizio a eseguire azioni specifiche. Un token di accesso è associato a un singolo servizio personalizzato.
- Questo tipo di concessione determina un token di accesso di breve durata e che può essere rinnovato chiamando un endpoint dell'identità.
- Per la documentazione pubblica di Adobe Marketo Engage per OAuth 2.0 con credenziali client, consulta Authentication nella Adobe Marketo Engage Developer Guide.

Per configurare una connessione Adobe Marketo Engage:

1. In AWS Secrets Manager, crea un segreto con i seguenti dettagli:
   a. Per le app connesse gestite dal cliente: il segreto deve contenere l'app connessa Consumer Secret con `USER_MANAGED_CLIENT_APPLICATION_CLIENT_SECRET` come chiave.
   b. Nota: è necessario creare un segreto per la connessione in AWS Glue.
2. In AWS Glue Glue Studio, crea una connessione in Connessioni dati seguendo i passaggi seguenti:
   a. In Tipo di connessione, seleziona Adobe Marketo Engage.
   b. Fornisci l'istanza `INSTANCE_URL` di Adobe Marketo Engage a cui desideri connetterti.
   c. Seleziona il ruolo AWS IAM che AWS Glue può assumere e dispone delle autorizzazioni per le seguenti azioni:

      JSON

      ```
      {
        "Version":"2012-10-17",
        "Statement": [
          {
            "Effect": "Allow",
            "Action": [
              "secretsmanager:DescribeSecret",
              "secretsmanager:GetSecretValue",
              "secretsmanager:PutSecretValue",
              "ec2:CreateNetworkInterface",
              "ec2:DescribeNetworkInterfaces",
              "ec2:DeleteNetworkInterface"
            ],
            "Resource": "*"
          }
        ]
      }
      ```

   d. Seleziona quello `secretName` che desideri utilizzare per questa connessione per AWS Glue inserire i token.
   e. Selezionare le opzioni di rete se si desidera utilizzare la propria rete.
3. Concedi il permesso di lettura `secretName` al ruolo IAM associato al tuo AWS Glue lavoro.

## Lettura dalle entità Adobe Marketo Engage

Prerequisito

Un oggetto di Adobe Marketo Engage da cui desideri leggere. Avrai bisogno del nome dell'oggetto, come lead, attività oppure oggetti personalizzati. Le tabelle seguenti mostrano le entità supportate.

Entità supportate per l'origine (sincrone):

| Entità | Può essere filtrato | Supporta limite | Supporta Ordina per | Supporta Seleziona * | Supporta Partizionamento |
|---|---|---|---|---|---|
| leads | Sì | Sì | No | Sì | No |
| attività | Sì | Sì | No | Sì | No |
| customobjects | Sì | Sì | No | Sì | No |

Entità supportate per l'origine (asincrone):

| Entità | Può essere filtrato | Supporta limite | Supporta Ordina per | Supporta Seleziona * | Supporta Partizionamento |
|---|---|---|---|---|---|
| leads | Sì | No | No | Sì | Sì |
| attività | Sì | No | No | Sì | No |
| customobjects | Sì | No | No | Sì | Sì |

Esempio:

```
adobe-marketo-engage_read = glueContext.create_dynamic_frame.from_options(
    connection_type="adobe-marketo-engage",
```

```
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "leads",
        "API_VERSION": "v2",
        "INSTANCE_URL": "https://539-t**-6**.mktorest.com"
    }
```

Dettagli dell'entità e dei campi di Adobe Marketo Engage:

Entità con metadati statici:

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| attività | sinceDatetime (supportato solo in modalità sincrona) | DateTime | >= (solo per la modalità sincrona) |
| | createdAt (supporta to solo in modalità asincrona) | DateTime | between (solo per la modalità asincrona) |
| | activitiesTypeId | Numero intero | = |
| | adobe-marketo-enga geGUID | Long | = (solo per la modalità sincrona) |
| | leadId | Long | N/D |
| | activityDate | DateTime | N/D |
| | campaignId | Long | N/D |
| | primaryAttributeVa lueId | Numero intero | N/D |
| | primaryAttributeValue | Stringa | N/A |
| | attributes | Stringa | N/A |

Entità con metadati dinamici:

Per le seguenti entità, Adobe Marketo Engage fornisce endpoint per recuperare i metadati in modo dinamico, cosicché il supporto dell'operatore venga acquisito a livello di tipo di dati per ciascuna entità.

| Entità | Tipo di dati | Operatori supportati |
|---|---|---|
| leads | Numero intero | = (solo per la modalità sincrona) |
| | DateTime | between (solo per la modalità asincrona) |
| | Stringa | = (solo per la modalità sincrona) |
| | Long | N/D |
| | Booleano | N/D |
| | Data | N/D |
| | Float | N/D |
| customobjects | Numero intero | N/D |
| | DateTime | between (solo per la modalità asincrona) |
| | Stringa | = (solo per la modalità sincrona) |
| | Data | N/D |
| | Long | N/D |
| | Booleano | N/D |
| | Float | N/D |

Query di partizionamento

Puoi fornire le opzioni Spark aggiuntive PARTITION_FIELD, LOWER_BOUND, UPPER_BOUND e NUM_PARTITIONS se desideri utilizzare la concorrenza in Spark. Con questi parametri, la query originale viene suddivisa in NUM_PARTITIONS query secondarie che possono essere eseguite contemporaneamente dalle attività Spark.

- PARTITION_FIELD: il nome del campo da utilizzare per partizionare la query.
- LOWER_BOUND: un valore limite inferiore inclusivo del campo di partizione scelto.

  Per il DateTime campo, accettiamo il valore in formato ISO.

  Esempio di valore valido:

  ```
  "2024-07-01T00:00:00.000Z"
  ```

- UPPER_BOUND: un valore limite superiore esclusivo del campo di partizione scelto.
- NUM_PARTITIONS: il numero di partizioni.

Nella seguente tabella sono descritti i dettagli del campo di supporto del partizionamento delle entità:

| Nome dell'entità | Campi di partizionamento | Tipo di dati |
|---|---|---|
| leads | createdAt | DateTime |
| | updateAt | DateTime |
| customobjects | updatedAt | DateTime |

Esempio:

```
adobe-marketo-engage_read = glueContext.create_dynamic_frame.from_options(
    connection_type="adobe-marketo-engage",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "leads",
        "API_VERSION": "v1",
        "PARTITION_FIELD": "createdAt"
        "LOWER_BOUND": "2024-07-01T00:00:00.000Z"
```

```
        "UPPER_BOUND": "2024-07-02T00:00:00.000Z"
        "NUM_PARTITIONS": "10"
    }
```

## Scrivere a entità Adobe Marketo Engage

Prerequisiti

- Un oggetto di Adobe Marketo a cui si desidera scrivere. È necessario il nome dell'oggetto, come `leads` o `customobjects`.
- Il connettore Adobe Marketo supporta tre operazioni di scrittura:
  - INSERT
  - UPSERT
  - UPDATE
- Per le operazioni di scrittura `UPSERT` e `UPDATE`, è necessario indicare l'opzione `ID_FIELD_NAMES` per specificare il campo ID per i record. Quando si lavora con l'entità `leads`, utilizzare `email` come `ID_FIELD_NAMES` per le operazioni `UPSERT` e `id` per le operazioni `UPDATE`. Per l'entità `customobjects`, utilizzare `marketoGUID` e `ID_FIELD_NAMES` per le operazioni `UPDATE` e `UPSERT`.

Entità supportate per la destinazione (sincrone)

| Nome dell'entità | Sarà supportato come connettore di destinazione | Può essere inserito | Può essere Updated | Può essere alterato |
|---|---|---|---|---|
| lead | Sì | Sì (in blocco) | Sì (in blocco) | Sì (in blocco) |
| customobjects | Sì | Sì (in blocco) | Sì (in blocco) | Sì (in blocco) |

Esempio:

Operazione INSERT:

```
marketo_write = glueContext.write_dynamic_frame.from_options(
    frame=frameToWrite,
    connection_type="marketo",
```

```
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "leads",
        "API_VERSION": "v1",
        "WRITE_OPERATION": "INSERT"
    }
```

Operazione UPDATE:

```
marketo_write = glueContext.write_dynamic_frame.from_options(
    frame=frameToWrite,
    connection_type="marketo",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "leads",
        "API_VERSION": "v1",
        "WRITE_OPERATION": "UPDATE",
        "ID_FIELD_NAMES": "id"
    }
```

> **Note**
>
> Per le entità `leads` e `customobjects`, Adobe Marketo fornisce endpoint per recuperare i metadati in modo dinamico, affinché i campi scrivibili vengano identificati dalla risposta dell'API Marketo.

## Opzioni di connessione Adobe Marketo Engage

Di seguito sono riportate le opzioni di connessione per Adobe Marketo Engage:

- `ENTITY_NAME`(String): (obbligatorio) utilizzato per la lettura. Il nome dell'oggetto in Adobe Marketo Engage.
- `API_VERSION`(String): (obbligatorio) utilizzato per la lettura. La versione di Rest API di Adobe Marketo Engage che desideri utilizzare. Ad esempio: v1.
- `SELECTED_FIELDS`(List<String>): predefinito: vuoto (SELECT *). Utilizzato per la lettura. Colonne da selezionare per l'oggetto.
- `FILTER_PREDICATE`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Dev'essere in formato Spark SQL.

- `QUERY`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Query Spark SQL completa.
- `PARTITION_FIELD`(Stringa): utilizzato per la lettura. Campo da utilizzare per partizionare la query.
- `LOWER_BOUND`(Stringa): utilizzato per la lettura. Un valore limite inferiore inclusivo del campo di partizione scelto.
- `UPPER_BOUND`(Stringa): utilizzato per la lettura. Un valore limite superiore esclusivo del campo di partizione scelto.
- `NUM_PARTITIONS`(Numero intero): predefinito: 1. Utilizzato per la lettura. Numero di partizioni da leggere.
- `TRANSFER_MODE`(Stringa): predefinito: SYNC. Utilizzato per la lettura asincrona.
- `WRITE_OPERATION`(Stringa): predefinito: INSERT. Utilizzato per la scrittura. Il valore deve essere INSERT, UPDATE, UPSERT.
- `ID_FIELD_NAMES`(Stringa): predefinito: nullo. Obbligatorio per UPDATE e UPSERT.

## Limitazioni e note per il connettore Adobe Marketo Engage

Di seguito sono riportate le limitazioni o le note per il connettore Adobe Marketo Engage:

- 'sinceDateTime' e 'activityTypeId' sono parametri di filtro obbligatori per l'entità Sync Activities.
- Agli abbonamenti vengono allocate 50.000 chiamate API al giorno (che vengono ripristinate ogni giorno alle 00:00 CST). È possibile acquistare una capacità giornaliera aggiuntiva come parte di un abbonamento Adobe Marketo Engage. Per riferimento, consulta Adobe Marketo Rest API.
- Il periodo di tempo massimo per il filtro dell'intervallo di date (`createdAt` o `updatedAt`) è 31 giorni. Per riferimento, consulta Bulk Extract - Marketo Developers.
- Agli abbonamenti viene assegnato un massimo di 10 processi di estrazione in blocco in coda in un dato momento. Per riferimento, consulta Bulk Extract - Marketo Developers.
- Per impostazione predefinita, i processi di estrazione sono limitati a 500 MB al giorno (che vengono ripristinati ogni giorno alle 00:00 CST). È possibile acquistare una capacità giornaliera aggiuntiva come parte di un abbonamento Adobe Marketo Engage. Per riferimento, consulta Bulk Extract - Marketo Developers.
- Il numero massimo di processi di esportazione simultanei è 2. Per riferimento, consulta Bulk Extract - Marketo Developers.
- Il numero massimo di processi di esportazione in coda (inclusi i processi attualmente in corso) è 10. Per riferimento, consulta Bulk Extract - Marketo Developers.
- La dimensione massima consentita del file è di 1 GB per l'estrazione da un processo in blocco.

- Una volta creato un processo asincrono, il periodo di conservazione dei file sarà di 7 giorni prima della scadenza. Per riferimento, consulta Bulk Extract - Marketo Developers.
- `createdAt` o `updatedAt` sono parametri di filtro obbligatori per l'entità Lead asincroni.
- `createdAt` è un parametro di filtro obbligatorio per l'entità Attività asincrone.
- `updatedAt` è un parametro di filtro obbligatorio per l'entità Oggetto personalizzato asincrono.
- Quando si utilizzano i connettori AWS Glue SaaS, gli utenti non sono in grado di identificare quali record specifici hanno avuto esito negativo durante un'operazione di scrittura su piattaforme SaaS di destinazione in caso di guasti parziali.
- Quando si utilizzano le operazioni di scrittura sincrone, tutti i campi con valori nulli nel file di input verranno automaticamente eliminati e non inviati alla piattaforma SaaS.
- Puoi creare o aggiornare fino a 300 record in un batch per la scrittura sincrona.

Per ulteriori informazioni, consulta Adobe Marketo Engage Integration Best Practices e Bulk Extract.

## Connessione ad Amazon Redshift in AWS Glue Studio

> **Note**
>
> Puoi usare Spark AWS Glue per leggere e scrivere su tabelle in Amazon Redshift database esterni a. AWS Glue Studio Per configurare i AWS Glue job Amazon Redshift in modo programmatico, vedi. Connessioni Redshift

AWS Gluefornisce supporto integrato per Amazon Redshift. AWS Glue Studiofornisce un'interfaccia visiva a cui connettersi Amazon Redshift, creare lavori di integrazione dei dati ed eseguirli sul runtime Spark AWS Glue Studio senza server.

Argomenti

- Creare una Amazon Redshift connessione
- Creazione di un nodo Amazon Redshift sorgente
- Creazione di un nodo Amazon Redshift di destinazione
- Opzioni avanzate

## Creare una Amazon Redshift connessione

### Autorizzazioni necessarie

Sono necessarie autorizzazioni aggiuntive per utilizzare Amazon Redshift cluster e ambienti Amazon Redshift serverless. Per ulteriori informazioni su come aggiungere autorizzazioni ai processi ETL, consulta la pagina Review IAM permissions needed for ETL jobs.

- redshift: DescribeClusters
- redshift senza server: ListWorkgroups
- redshift senza server: ListNamespaces

### Panoramica di

Quando si aggiunge una Amazon Redshift connessione, è possibile scegliere una Amazon Redshift connessione esistente o creare una nuova connessione quando si aggiunge un nodo Data source - Redshift in. AWS Glue Studio

AWS Gluesupporta sia i Amazon Redshift cluster che gli ambienti Amazon Redshift serverless. Quando si crea una connessione, gli ambienti Amazon Redshift serverless visualizzano l'etichetta serverless accanto all'opzione di connessione.

Per ulteriori informazioni su come creare una Amazon Redshift connessione, consulta Spostamento di dati da e verso. Amazon Redshift

## Creazione di un nodo Amazon Redshift sorgente

### Autorizzazioni necessarie

AWS Glue Studio i lavori che utilizzano fonti di Amazon Redshift dati richiedono autorizzazioni aggiuntive. Per ulteriori informazioni su come aggiungere autorizzazioni ai processi ETL, consulta la pagina Review IAM permissions needed for ETL jobs.

Per utilizzare una connessione sono necessarie le seguenti autorizzazioni. Amazon Redshift

- redshift-data: ListSchemas
- dati redshift: ListTables
- dati redshift: DescribeTable
- dati redshift: ExecuteStatement

- dati redshift: DescribeStatement
- dati redshift: GetStatementResult

Aggiungere una fonte di dati Amazon Redshift

Per aggiungere un nodo Origine dati: Amazon Redshift:

1. Scegli il tipo di Amazon Redshift accesso:

   - Connessione dati diretta (consigliata): scegli questa opzione se desideri accedere direttamente ai tuoi dati Amazon Redshift . Questa è l'opzione consigliata nonché quella predefinita.
   - Data Catalog tables: scegli questa opzione se hai delle tabelle di Catalogo dati che desideri utilizzare.

2. Se scegli Connessione dati diretta, scegli la connessione per la tua fonte di Amazon Redshift dati. Ciò presuppone che la connessione esista già e che sia possibile effettuare una selezione tra le connessioni esistenti. Se devi creare una connessione, scegli Crea connessione Redshift. Per ulteriori informazioni, consulta la pagina Overview of using connectors and connections.

   Dopo aver scelto una connessione, puoi visualizzare le proprietà della connessione facendo clic su Visualizza proprietà. Le informazioni sulla connessione sono visibili, tra cui URL, gruppi di sicurezza, sottorete, zona di disponibilità, descrizione, nonché timestamp di creazione (UTC) e ultimo aggiornamento (UTC).

3. Scegli un'opzione Amazon Redshift di origine:

   - Scegli una singola tabella: questa è la tabella che contiene i dati a cui desideri accedere da un'unica Amazon Redshift tabella.
   - Inserisci una query personalizzata: ti consente di accedere a un set di dati da più tabelle Amazon Redshift in base alla tua query personalizzata.

4. Se hai scelto una singola tabella, scegli lo Amazon Redshift schema. L'elenco degli schemi disponibili tra cui scegliere è determinato dalla tabella selezionata.

   In alternativa, scegli Inserisci query personalizzata. Scegli questa opzione per accedere a un set di dati personalizzato da più tabelle Amazon Redshift . Quando scegli questa opzione, inserisci la Amazon Redshift query.

   Quando ti connetti a un ambiente Amazon Redshift senza server, aggiungi la seguente autorizzazione alla query personalizzata:

```
GRANT SELECT ON ALL TABLES IN <schema> TO PUBLIC
```

Puoi scegliere Acquisisci schema per leggere lo schema in base alla query che hai inserito. Puoi anche scegliere Open Redshift Query Editor per inserire una Amazon Redshift query. Per ulteriori informazioni, consulta la pagina Querying a database using the query editor.

5. In Prestazioni e sicurezza, scegli la directory di gestione temporanea di Amazon S3 e il ruolo IAM.
   - Directory di gestione temporanea di Amazon S3: scegli la posizione Amazon S3 per la gestione temporanea dei dati.
   - Ruolo IAM: scegli il ruolo IAM che può scrivere nella posizione Amazon S3 che hai selezionato.
6. In Parametri Redshift personalizzati - facoltativo, inserisci il parametro e il valore.

## Creazione di un nodo Amazon Redshift di destinazione

### Autorizzazioni necessarie

AWS Glue Studio i lavori che utilizzano Amazon Redshift data target richiedono autorizzazioni aggiuntive. Per ulteriori informazioni su come aggiungere autorizzazioni ai processi ETL, consulta la pagina Review IAM permissions needed for ETL jobs.

Per utilizzare una connessione sono necessarie le seguenti autorizzazioni. Amazon Redshift

- redshift-data: ListSchemas
- dati redshift: ListTables

### Aggiungere un nodo di destinazione Amazon Redshift

Per creare un nodo Amazon Redshift di destinazione:

1. Scegliete una Amazon Redshift tabella esistente come destinazione o inserite un nuovo nome per la tabella.
2. Quando utilizzi il nodo di destinazione Destinazione dati - Redshift, puoi scegliere tra le seguenti opzioni:

- AGGIUNGI: se esiste già una tabella, scarica tutti i nuovi dati nella tabella come inserto. Se la tabella non esiste, procedi alla sua creazione e quindi inserisci tutti i nuovi dati.

  Inoltre, seleziona la casella se desideri aggiornare (UPSERT) i record esistenti nella tabella di destinazione. La tabella deve già esistere, altrimenti l'operazione avrà esito negativo.
- MERGE: AWS Glue aggiorna o aggiunge i dati alla tabella di destinazione in base alle condizioni specificate.

  > **Note**
  >
  > Per utilizzare l'azione di unione inAWS Glue, è necessario abilitare la funzionalità di Amazon Redshift unione. Per istruzioni su come abilitare l'unione per la tua Amazon Redshift istanza, vedi MERGE (anteprima).

  Scegli le opzioni:

  - Scegli chiavi e operazioni semplici: scegli le colonne da utilizzare come chiavi di corrispondenza tra i dati di origine e il set di dati di destinazione.

    Specifica le seguenti opzioni in caso di corrispondenza:

    - Aggiorna il record nel set di dati di destinazione con i dati dell'origine.
    - Elimina il record nel set di dati di destinazione.

    Specifica le seguenti opzioni in caso di mancata corrispondenza:

    - Inserisci i dati di origine come nuova riga nel set di dati di destinazione.
    - Non fare nulla.
  - Inserisci un'istruzione MERGE personalizzata: puoi quindi scegliere Convalida l'istruzione MERGE per verificare che l'istruzione sia valida o non valida.
- TRUNCATE: se esiste già una tabella, tronca i dati della tabella cancellando prima il contenuto della tabella di destinazione. Se il troncamento ha esito positivo, inserisci tutti i dati. Se la tabella non esiste, procedi alla sua creazione e quindi inserisci tutti i dati. Se il troncamento non va a buon fine, l'operazione non andrà a buon fine.
- DROP: se esiste già una tabella, elimina i metadati e i dati della tabella. Se l'eliminazione ha esito positivo, inserisci tutti i dati. Se la tabella non esiste, procedi alla sua creazione e quindi inserisci tutti i dati. Se l'eliminazione non va a buon fine, l'operazione non andrà a buon fine.

- CREATE: crea una nuova tabella con il nome predefinito. Se il nome della tabella esiste già, crea una nuova tabella aggiungendo il suffisso nel formato `job_datetime` al nome per renderlo unico. Questo inserirà tutti i dati nella nuova tabella. Se la tabella esiste già, al nome finale della tabella verrà aggiunto il suffisso. Se la tabella non esiste, verrà creata una tabella. In entrambi i casi, verrà creata una nuova tabella.

### Opzioni avanzate

Vedi Uso del connettore Amazon Redshift Spark acceso. AWS Glue

## Connessione ad Asana

Asana è una soluzione di collaborazione in team basata su cloud che aiuta i team a organizzare, pianificare e completare attività e progetti. Se sei un utente di Asana, il tuo account contiene dati relativi ai tuoi spazi di lavoro, ai tuoi progetti, alle tue attività, ai tuoi team e altro ancora. Puoi trasferire dati da Asana a determinati servizi AWS o ad altre applicazioni supportate.

Argomenti


- AWS Glue supporto per Asana
- Policy contenenti le operazioni API per creare e utilizzare le connessioni
- Configurazione di Asana
- Configurazione delle connessioni Asana
- Lettura dalle entità di Asana
- Opzioni di connessione di Asana
- Creazione di un account Asana
- Limitazioni


### AWS Glue supporto per Asana

AWS Glue supporta Asana come segue:

Supportato come origine?

Sì. Puoi utilizzare i job AWS Glue ETL per interrogare i dati di Asana.

Supportato come destinazione?

No.

Versioni dell'API Asana supportate

1.0

## Policy contenenti le operazioni API per creare e utilizzare le connessioni

La seguente politica di esempio descrive le AWS autorizzazioni richieste per la creazione e l'utilizzo delle connessioni. Se si sta creando un nuovo ruolo, creare una policy che contenga quanto segue:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:ListConnectionTypes",
        "glue:DescribeConnectionType",
        "glue:RefreshOAuth2Tokens",
        "glue:ListEntities",
        "glue:DescribeEntity"
      ],
      "Resource": "*"
    }
  ]
}
```

In alternativa, se non desideri utilizzare il metodo precedente, utilizza le seguenti policy IAM:

- AWSGlueServiceRole — Garantisce l'accesso alle risorse che diversi AWS Glue processi richiedono per l'esecuzione per conto dell'utente. Queste risorse includono Amazon S3 AWS Glue, IAM, CloudWatch Logs e Amazon EC2. Se segui la convenzione di denominazione delle risorse specificata in questa politica, AWS Glue i processi dispongono delle autorizzazioni necessarie. Questa policy è in genere collegata ai ruoli specificati quando si definiscono crawler, processi ed endpoint di sviluppo.
- AWSGlueConsoleFullAccess — Concede l'accesso completo alle AWS Glue risorse quando un'identità a cui è allegata la politica utilizza la AWS console di gestione. Se si segue la

convenzione per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questa policy è in genere associata agli utenti della AWS Glue console.

## Configurazione di Asana

Prima di poter AWS Glue utilizzare il trasferimento da Asana, devi soddisfare i seguenti requisiti:

Requisiti minimi

- Avere un account Asana con e-mail e password. Per ulteriori informazioni sulla creazione di account, consulta Creating a Asana account.
- Devi avere un AWS account creato con il servizio di accesso a AWS Glue.
- Verifica di aver creato una delle seguenti risorse nell'account Asana:
  - Un'app per sviluppatori che supporta l'autenticazione `OAuth 2.0`. Per ulteriori istruzioni, consulta la documentazione per gli OAuthsviluppatori di Asana. In alternativa, consulta the section called "Creazione di un account Asana".
  - Un token di accesso personale. Per ulteriori informazioni, consulta il token di accesso personale https://developers.asana.com/docs/personal-access-token nella documentazione per gli sviluppatori di Asana.

Se soddisfi questi requisiti, sei pronto per connetterti AWS Glue al tuo account Adobe Analytics. Per le connessioni tipiche, non è necessario fare nient'altro in Adobe Analytics.

## Configurazione delle connessioni Asana

Asana supporta il tipo di concessione `AUTHORIZATION_CODE` per `OAuth2`.

Questo tipo di concessione è considerato `OAuth` "a tre gambe" in quanto si basa sul reindirizzamento degli utenti al server di autorizzazione di terzi per l'autenticazione. Gli utenti possono scegliere di creare la propria app connessa su Asana e fornire il proprio ID cliente e il segreto del client quando creano connessioni tramite la AWS Glue console. In questo scenario, verranno comunque reindirizzati ad Asana per effettuare il login e autorizzare l'accesso AWS Glue alle proprie risorse.

Questo tipo di concessione genera un token di aggiornamento e un token di accesso. Il token di accesso dura poco tempo e può essere aggiornato automaticamente senza l'interazione dell'utente utilizzando il token di aggiornamento.

Per la documentazione pubblica di Asana sulla creazione di un'app connessa per `AUTHORIZATION_CODE OAuth` Flow, consulta Asana. APIs

Per configurare una connessione Asana:

1. In AWS Secrets Manager, crea un segreto con i seguenti dettagli:
   - Per le app connesse gestite dal cliente: il segreto deve contenere l'app connessa Consumer Secret con `USER_MANAGED_CLIENT_APPLICATION_CLIENT_SECRET` come chiave.
   - 
     > **Note**
     >
     > È necessario creare un segreto per la connessione in AWS Glue.
2. In AWS Glue Studio, crea una connessione in Connessioni dati seguendo i passaggi seguenti:
   a. In Tipo di connessione, seleziona Asana.
   b. Fornisci l'ambiente Asana.
   c. Seleziona il ruolo IAM per il quale AWS Glue può assumere e dispone delle autorizzazioni per le seguenti azioni:

      JSON

      ```
      {
        "Version":"2012-10-17",
        "Statement": [
          {
            "Effect": "Allow",
            "Action": [
              "secretsmanager:DescribeSecret",
              "secretsmanager:GetSecretValue",
              "secretsmanager:PutSecretValue",
              "ec2:CreateNetworkInterface",
              "ec2:DescribeNetworkInterfaces",
              "ec2:DeleteNetworkInterface"
            ],
            "Resource": "*"
          }
        ]
      }
      ```

   d. Seleziona quello `secretName` che desideri utilizzare per questa connessione per AWS Glue inserire i token.

e. Selezionare le opzioni di rete se si desidera utilizzare la propria rete.

3. Concedi il permesso di lettura `secretName` al ruolo IAM associato al tuo AWS Glue lavoro.

## Lettura dalle entità di Asana

### Prerequisiti

Un oggetto di Asana da cui desideri leggere. Consulta la tabella delle entità supportate qui di seguito per verificare quelle disponibili.

### Entità supportate come origine

| Entità | Può essere Filtrato | Supporta Limite | Supporta Ordina per | Supporta Seleziona * | Supporta Partizionamento |
|---|---|---|---|---|---|
| Workspace | No | Sì | No | Sì | No |
| Tag | No | Sì | No | Sì | No |
| Utente | No | Sì | No | Sì | No |
| Portfolio | No | Sì | No | Sì | No |
| Team | No | Sì | No | Sì | No |
| Progetto | Sì | Sì | No | Sì | No |
| Sezione | No | Sì | No | Sì | No |
| Operazione | Sì | No | No | Sì | Sì |
| Obiettivo | Sì | Sì | No | Sì | No |
| AuditLogEvent | Sì | Sì | No | Sì | No |
| Aggiornamento di stato | Sì | Sì | No | Sì | No |

| Entità | Può essere Filtrato | Supporta Limite | Supporta Ordina per | Supporta Seleziona * | Supporta Partizionamento |
|---|---|---|---|---|---|
| Campo personalizzato | No | Sì | No | Sì | No |
| Breve descrizione del progetto | Sì | No | No | Sì | Sì |

Esempio

```
read_read = glueContext.create_dynamic_frame.from_options(
    connection_type="Asana",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "task/workspace:xxxx",
        "API_VERSION": "1.0",
        "PARTITION_FIELD": "created_at",
        "LOWER_BOUND": "2024-02-05T14:09:30.115Z",
        "UPPER_BOUND": "2024-06-07T13:30:00.134Z",
        "NUM_PARTITIONS": "3"
    }
```

Dettagli dell'entità e del campo di Asana

- Workspace
- Tag
- Utente
- Portfolio
- Team
- Progetto
- Sezione
- Attività
- Obiettivo

- AuditLogEvent
- Aggiornamento di stato
- Campo personalizzato
- Breve descrizione del progetto

Query di partizionamento

Se si desidera utilizzare la simultaneità in Spark, possono essere fornite opzioni Spark `PARTITION_FIELD`, `LOWER_BOUND`, `UPPER_BOUND`, `NUM_PARTITIONS` aggiuntive. Con questi parametri, la query originale viene suddivisa in `NUM_PARTITIONS` query secondarie che possono essere eseguite contemporaneamente dalle attività Spark.

- `PARTITION_FIELD`: il nome del campo da utilizzare per partizionare la query.
- `LOWER_BOUND`: un valore limite inferiore inclusivo del campo di partizione scelto.

  Per la data, accettiamo il formato di data di Spark utilizzato nelle query di Spark SQL. Esempio di valori validi: `2024-06-07T13:30:00.134Z`.
- `UPPER_BOUND`: un valore limite superiore esclusivo del campo di partizione scelto.
- `NUM_PARTITIONS`: numero di partizioni.

I dettagli di supporto del campo di partizionamento per entità sono riportati nella tabella seguente.

| Nome entità | Campo di partizionamento | Tipo di dati |
|---|---|---|
| Operazione | created_at | DateTime |
| Operazione | modified_at | DateTime |

Esempio

```
read_read = glueContext.create_dynamic_frame.from_options(
    connection_type="Asana",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "task/workspace:xxxx",
        "API_VERSION": "1.0",
```

```
        "PARTITION_FIELD": "created_at",
        "LOWER_BOUND": "2024-02-05T14:09:30.115Z",
        "UPPER_BOUND": "2024-06-07T13:30:00.134Z",
        "NUM_PARTITIONS": "3"
    }
```

## Opzioni di connessione di Asana

Di seguito sono riportate le opzioni di connessione per Asana:

- `ENTITY_NAME`(Stringa): (obbligatorio) utilizzato per la lettura/scrittura. Il nome del tuo oggetto in Asana.
- `API_VERSION`(Stringa): (obbligatorio) utilizzato per la lettura/scrittura. Versione di Rest API di Asana da utilizzare. Ad esempio: 1.0.
- `SELECTED_FIELDS`(Elenco<Stringa>): predefinito: vuoto (SELECT *). Utilizzato per la lettura. Colonne da selezionare per l'oggetto.
- `FILTER_PREDICATE`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Dev'essere in formato Spark SQL.
- `QUERY`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Query Spark SQL completa.
- `PARTITION_FIELD`(Stringa): utilizzato per la lettura. Campo da utilizzare per partizionare la query.
- `LOWER_BOUND`(Stringa): utilizzato per la lettura. Un valore limite inferiore inclusivo del campo di partizione scelto.
- `UPPER_BOUND`(Stringa): utilizzato per la lettura. Un valore limite superiore esclusivo del campo di partizione scelto.
- `NUM_PARTITIONS`(Numero intero): predefinito: 1. Utilizzato per la lettura. Numero di partizioni da leggere.

## Creazione di un account Asana

1. Crea un account Asana e scegli Registrati.
2. Dopo aver effettuato l'accesso, si apre la pagina Configurazione dell'account. Completa questa procedura:
    - Rivedi il modulo di configurazione dell'account.
    - Inserisci tutti i dettagli pertinenti per creare il tuo account Asana.

- Ricontrolla se le informazioni sono corrette.

3. Scegli Crea account o Invia (il testo esatto del pulsante può variare) per finalizzare la configurazione dell'account.

Creazione dell'app in Asana per **`OAuth2.0`**

1. Accedi all'account Asana utilizzando le tue credenziali cliente di Asana.
2. Scegli l'icona del tuo profilo utente nell'angolo in alto a destra e seleziona Le mie impostazioni dal menu a discesa.
3. Seleziona la scheda App, quindi seleziona Gestisci app per sviluppatori.
4. Seleziona Crea nuova app e inserisci i dettagli pertinenti.
5. Scegli Crea app.
6. Nella pagina Le mie app:

   a. Seleziona OAuthe nella sezione Credenziali dell'app, prendi nota del tuo ID cliente e del segreto del cliente.

   b. Nella URLs sezione Reindirizzamento, aggiungi gli URL di reindirizzamento necessari.

> **Note**
>
> Inserisci l'URI di reindirizzamento utilizzando questo formato: `https://{aws-region-code}.console.aws.amazon.com/gluestudio/oauth`. Esempio: negli Stati Uniti orientali (Virginia settentrionale), utilizza: `https://us-east-1.console.aws.amazon.com/gluestudio/oauth`

Creazione dell'app in Asana per il token **`PAT`**

1. Accedi all'account Asana utilizzando le tue credenziali cliente di Asana.
2. Scegli l'icona del tuo profilo utente nell'angolo in alto a destra e seleziona Impostazioni del mio profilo dal menu a discesa.
3. Seleziona la scheda App, quindi Account del servizio.
4. Seleziona Crea nuova app e inserisci i dettagli pertinenti.
5. Scegli Aggiungi account del servizio.
6. La pagina successiva mostra il token, copialo e conservalo in modo sicuro.

> **Important**
>
> Questo token verrà visualizzato solo una volta. Assicurati di copiarlo e conservarlo in modo sicuro.

## Limitazioni

Di seguito sono riportate le limitazioni per il connettore di Asana:

- Gli account del servizio nei domini aziendali possono accedere solo agli endpoint dell'API dei log di audit. Per accedere a questi endpoint è necessaria l'autenticazione con il token di accesso personale di un account del servizio.
- È possibile accedere all'entità Obiettivo solo per gli account utente con piano Premium o superiore.
- `Audit Log Event Entity`: nel connettore, i campi `start_at` e `end_at` sono combinati in un unico campo "start_end_at" per supportare il filtraggio e il trasferimento incrementale.
- Il partizionamento non può essere supportato per il `Date` campo, anche se supporta gli operatori greater-than-or-equal -to e less-than-or-equal -to. Scenario: creato un processo con `partitionField` come `due_on` (datatype: data), `lowerBound` come `2019-09-14`, `upperBound` come `2019-09-16` e `numPartition` come 2. La parte di filtraggio dell'URL dell'endpoint viene creata come segue:
  - partition1: due_on.before=2019-09-14&due_on.after=2019-09-14
  - partition2: due_on.before=2019-09-15&due_on.after=2019-09-15 Risultato:
  - In partition1, otteniamo dati con due_date 2019-09-14 e 2019-09-15
  - In partition2, otteniamo gli stessi dati con due_date 2019-09-15 (che era in partition1) insieme ad altri dati, causando la duplicazione dei dati.
- Il filtraggio e il partizionamento non possono essere supportati sullo stesso campo poiché viene generato un errore di richiesta errata dal lato SaaS.
- L'entità Attività richiede almeno 1 campo nei criteri di filtro. Esiste una limitazione con Asana in cui l'impaginazione non viene identificata senza ordinare i record in base a un campo incentrato sul tempo. Pertanto, il campo Created_at viene utilizzato insieme all'impaginazione per distinguere il set di record successivo. Il campo Created_at è contrassegnato come obbligatorio nel filtro, con un valore predefinito di 2000-01-01T 00:00:00 Z se non fornito. Per ulteriori informazioni sulla paginazione, consulta Tasks in a workspace

# Connessione ad Azure Cosmos DB in AWS Glue Studio

AWS Glue fornisce il supporto integrato per Azure Cosmos DB. AWS Glue Studio fornisce un'interfaccia visiva per connettersi ad Azure Cosmos DB per NoSQL, creare processi di integrazione dei dati ed eseguirli sul runtime Spark serverless AWS Glue Studio.

Argomenti


- Creazione di una connessione Azure Cosmos DB
- Creazione di un nodo sorgente di Azure Cosmos DB
- Creazione di un nodo destinazione di Azure Cosmos DB
- Opzioni avanzate


## Creazione di una connessione Azure Cosmos DB

Prerequisiti:

- In Azure, dovrai identificare o generare una chiave di Azure Cosmos DB da usare da,. AWS Glue`cosmosKey` Per altre informazioni, consulta Accesso sicuro ai dati in Azure Cosmos DB nella documentazione di Azure.

Per configurare una connessione ad Azure Cosmos DB:

1. Nel Gestione dei segreti AWS, crea un segreto usando la tua chiave Azure Cosmos DB. Per creare un segreto in Secrets Manager, segui il tutorial disponibile in Crea un Gestione dei segreti AWS segreto nella Gestione dei segreti AWS documentazione. Dopo aver creato il segreto, mantieni il nome del segreto *`secretName`* per il passaggio successivo.
   - Quando selezionate le coppie chiave/valore, create una coppia per la chiave `spark.cosmos.accountKey` con il valore. *`cosmosKey`*
2. Nella AWS Glue console, crea una connessione seguendo la procedura riportata di seguito. the section called "Aggiungere una AWS Glue connessione" Dopo aver creato la connessione, conserva il nome della connessione*`connectionName`*, per utilizzi futuri in AWS Glue.
   - In Tipo di connessione, seleziona Azure Cosmos DB.
   - Quando selezioni un AWS segreto, fornisci*`secretName`*.

## Creazione di un nodo sorgente di Azure Cosmos DB

### Prerequisiti necessari

- Una connessione AWS Glue Azure Cosmos DB, configurata con un Gestione dei segreti AWS segreto, come descritto nella sezione precedente,. the section called "Creazione di una connessione Azure Cosmos DB"
- Autorizzazioni appropriate sul processo per leggere il segreto utilizzato dalla connessione.
- Un container Azure Cosmos DB per NoSQL da cui desideri leggere. Avrai bisogno delle informazioni di identificazione per il container.

  Un container Azure Cosmos per NoSQL è identificato dal database e dal container. È necessario fornire i nomi del database e del contenitore quando ci si connette all'API di Azure Cosmos for NoSQL. `cosmosDBName` `cosmosContainerName`

### Aggiungere un'origine dati di Azure Cosmos DB

Per aggiungere un nodo di origine dati: Azure Cosmos DB:

1. Scegli la connessione per la tua origine dati Azure Cosmos DB. Dato che l'hai creato, dovrebbe essere disponibile nel menu a discesa. Se devi creare una connessione, scegli Crea una connessione Azure Cosmos DB. Per ulteriori informazioni, consulta la sezione the section called "Creazione di una connessione Azure Cosmos DB" precedente.

   Dopo aver scelto una connessione, puoi visualizzare le proprietà della connessione facendo clic su Visualizza proprietà.
2. Scegli il nome del database Cosmos DB: fornisci il nome del database da cui vuoi leggere,. `cosmosDBName`
3. Scegli Azure Cosmos DB Container: fornisci il nome del contenitore da cui vuoi leggere,. `cosmosContainerName`
4. Facoltativamente, scegli Query personalizzata per Azure Cosmos DB: fornisci una query SQL SELECT per recuperare informazioni specifiche da Azure Cosmos DB.
5. In Proprietà personalizzate di Azure Cosmos, inserisci i parametri e i valori necessari.

## Creazione di un nodo destinazione di Azure Cosmos DB

Prerequisiti necessari

- Una connessione AWS Glue Azure Cosmos DB, configurata con un Gestione dei segreti AWS segreto, come descritto nella sezione precedente,. the section called "Creazione di una connessione Azure Cosmos DB"
- Autorizzazioni appropriate sul processo per leggere il segreto utilizzato dalla connessione.
- Una tabella di Azure Cosmos DB su cui scrivere. Avrai bisogno delle informazioni di identificazione per il container. È necessario creare il container prima di chiamare il metodo di connessione.

  Un container Azure Cosmos per NoSQL è identificato dal database e dal container. È necessario fornire i nomi del database e del contenitore quando ci si connette all'API di Azure Cosmos for NoSQL. *cosmosDBName* *cosmosContainerName*

Aggiungere una destinazione dati di Azure Cosmos DB

Per aggiungere un nodo di destinazione dati: Azure Cosmos DB:

1. Scegli la connessione per la tua origine dati Azure Cosmos DB. Dato che l'hai creato, dovrebbe essere disponibile nel menu a discesa. Se devi creare una connessione, scegli Crea una connessione Azure Cosmos DB. Per ulteriori informazioni, consulta la sezione the section called "Creazione di una connessione Azure Cosmos DB" precedente.

   Dopo aver scelto una connessione, puoi visualizzare le proprietà della connessione facendo clic su Visualizza proprietà.
2. Scegli il nome del database Cosmos DB: fornisci il nome del database da cui vuoi leggere,. *cosmosDBName*
3. Scegli Azure Cosmos DB Container: fornisci il nome del contenitore da cui vuoi leggere,. *cosmosContainerName*
4. In Proprietà personalizzate di Azure Cosmos, inserisci i parametri e i valori necessari.

## Opzioni avanzate

È possibile fornire opzioni avanzate durante la creazione di un nodo Azure Cosmos DB. Queste opzioni sono le stesse disponibili durante la programmazione AWS Glue per gli script Spark.

Per informazioni, consulta the section called "Connessioni Azure Cosmos DB".

# Connessione ad Azure SQL in AWS Glue Studio

AWS Glue fornisce il supporto integrato per Azure SQL. AWS Glue Studio fornisce un'interfaccia visiva per connettersi ad Azure SQL, creare processi di integrazione dei dati ed eseguirli sul runtime Spark serverless AWS Glue Studio.

Argomenti


- Creazione di una connessione Azure SQL
- Creazione di un nodo sorgente di Azure SQL
- Creazione di un nodo destinazione di Azure SQL
- Opzioni avanzate


## Creazione di una connessione Azure SQL

Per connetterti ad Azure SQL da AWS Glue, dovrai creare e archiviare le tue credenziali SQL di Azure in un Gestione dei segreti AWS segreto, quindi associare quel segreto a una connessione SQL di Azure. AWS Glue

Per configurare una connessione ad Azure SQL:

1. In Gestione dei segreti AWS, crea un segreto usando le tue credenziali SQL di Azure. Per creare un segreto in Secrets Manager, segui il tutorial disponibile in Crea un Gestione dei segreti AWS segreto nella Gestione dei segreti AWS documentazione. Dopo aver creato il segreto, mantieni il nome del segreto *`secretName`* per il passaggio successivo.
   - Quando selezionate le coppie chiave/valore, create una coppia per la chiave `user` con il valore. *`azuresqlUsername`*
   - Quando selezionate le coppie chiave/valore, create una coppia per la chiave `password` con il valore. *`azuresqlPassword`*
2. Nella AWS Glue console, crea una connessione seguendo i passaggi riportati di seguito. the section called "Aggiungere una AWS Glue connessione" Dopo aver creato la connessione, conserva il nome della connessione*`connectionName`*, per utilizzi futuri in AWS Glue.
   - In Tipo di connessione, seleziona Azure SQL.
   - Quando fornisci l'URL SQL di Azure, fornisci un URL di endpoint JDBC.

L'elenco deve essere nel seguente formato:
`jdbc:sqlserver://databaseServerName:databasePort;databaseName=azuresqlDBnam`

AWS Glue richiede le seguenti proprietà URL:

- `databaseName`: un database predefinito in Azure SQL a cui connettersi.

Per altre informazioni su JDBC URLs for Azure SQL Managed Instances, consulta la documentazione di Microsoft.

- Quando selezioni un AWS segreto, fornisci. *`secretName`*

## Creazione di un nodo sorgente di Azure SQL

Prerequisiti necessari

- Una connessione AWS Glue Azure SQL, configurata con un Gestione dei segreti AWS segreto, come descritto nella sezione precedente,the section called "Creazione di una connessione Azure SQL".
- Autorizzazioni appropriate sul processo per leggere il segreto utilizzato dalla connessione.
- Una tabella SQL di Azure da cui desideri leggere,. *`tableName`*

  Una tabella SQL di Azure è identificata dal database, dallo schema e dal nome. È necessario fornire il nome del database e della tabella durante la connessione ad Azure SQL. È inoltre necessario fornire lo schema se diverso da quello predefinito, "pubblico". Il database viene fornito tramite una proprietà URL in*`connectionName`*, lo schema e il nome della `dbtable` tabella tramite.

Aggiungere un'origine dati di Azure SQL

Per aggiungere un nodo di origine dati: Azure SQL:

1. Scegli la connessione per la tua origine dati Azure SQL. Dato che l'hai creato, dovrebbe essere disponibile nel menu a discesa. Se devi creare una connessione, scegli Crea una connessione Azure SQL. Per ulteriori informazioni, consulta la sezione the section called "Creazione di una connessione Azure SQL" precedente.

   Dopo aver scelto una connessione, puoi visualizzare le proprietà della connessione facendo clic su Visualizza proprietà.

2. Scegli un'opzione Origine Azure SQL:

   - Scegli una singola tabella: accedi a tutti i dati da un'unica tabella.
   - Inserisci una query personalizzata: accedi a un set di dati da più tabelle in base alla tua query personalizzata.

3. Se hai scelto una singola tabella, inserisci*tableName*.

   Se hai scelto Inserisci una query personalizzata, inserisci una query TransactSQL SELECT.

4. In Proprietà personalizzate di Azure SQL, inserisci i parametri e i valori necessari.

## Creazione di un nodo destinazione di Azure SQL

Prerequisiti necessari

- Una connessione AWS Glue Azure SQL, configurata con un Gestione dei segreti AWS segreto, come descritto nella sezione precedente,the section called "Creazione di una connessione Azure SQL".
- Autorizzazioni appropriate sul processo per leggere il segreto utilizzato dalla connessione.
- Una tabella SQL di Azure su cui scrivere,. *tableName*

  Una tabella SQL di Azure è identificata dal database, dallo schema e dal nome. È necessario fornire il nome del database e della tabella durante la connessione ad Azure SQL. È inoltre necessario fornire lo schema se diverso da quello predefinito, "pubblico". Il database viene fornito tramite una proprietà URL in*connectionName*, lo schema e il nome della `dbtable` tabella tramite.

Aggiungere una destinazione dati di Azure SQL

Per aggiungere un nodo di destinazione dati: Azure SQL:

1. Scegli la connessione per la tua origine dati Azure SQL. Dato che l'hai creato, dovrebbe essere disponibile nel menu a discesa. Se devi creare una connessione, scegli Crea una connessione Azure SQL. Per ulteriori informazioni, consulta la sezione the section called "Creazione di una connessione Azure SQL" precedente.

   Dopo aver scelto una connessione, puoi visualizzare le proprietà della connessione facendo clic su Visualizza proprietà.

2. Configura il nome della tabella fornendo*tableName*.
3. In Proprietà personalizzate di Azure SQL, inserisci i parametri e i valori necessari.

### Opzioni avanzate

È possibile fornire opzioni avanzate durante la creazione di un nodo Azure SQL. Queste opzioni sono le stesse disponibili durante la programmazione AWS Glue per gli script Spark.

Per informazioni, consulta the section called "Connessioni Azure SQL".

## Connessione a Edge NXT di Blackbaud Raiser

Edge NXT di Blackbaud Raiser è una soluzione software completa basata su cloud per la raccolta fondi e la gestione dei donatori, creata appositamente per le organizzazioni no-profit e per tutta la comunità impegnata nel sociale. Questo connettore si basa sull'API SKY di Blackbaud Raiser Edge NXT e fornisce operazioni che aiutano a gestire le entità presenti all'interno di Edge NXT di Raiser.

Argomenti

- AWS Glue supporto per Blackbaud Raiser's Edge NXT
- Policy contenenti le operazioni API per creare e utilizzare le connessioni
- Configurazione di Edge NXT di Blackbaud Raiser
- Configurazione delle connessioni Edge NXT di Blackbaud Raiser
- Lettura dalle entità Edge NXT di Blackbaud Raiser
- Opzioni di connessione Edge NXT di Blackbaud Raiser
- Limitazioni di Edge NXT di Blackbaud Raiser

### AWS Glue supporto per Blackbaud Raiser's Edge NXT

AWS Glue supporta Edge NXT di Blackbaud Raiser come segue:

Supportato come origine?

Sì. È possibile utilizzare i job AWS Glue ETL per interrogare i dati da Edge NXT di Blackbaud Raiser.

Supportato come destinazione?

No.

Versioni supportate dell'API Edge NXT di Blackbaud Raiser

Sono supportate le seguenti versioni dell'API Edge NXT di Blackbaud Raiser:

- v1

## Policy contenenti le operazioni API per creare e utilizzare le connessioni

La seguente policy di esempio descrive le autorizzazioni AWS IAM richieste per la creazione e l'utilizzo delle connessioni. Se si sta creando un nuovo ruolo, creare una policy che contenga quanto segue:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:ListConnectionTypes",
        "glue:DescribeConnectionType",
        "glue:RefreshOAuth2Tokens",
        "glue:ListEntities",
        "glue:DescribeEntity"
      ],
      "Resource": "*"
    }
  ]
}
```

In alternativa, se non desideri utilizzare il metodo precedente, utilizza le seguenti policy IAM gestite:

- AWSGlueServiceRole— Garantisce l'accesso alle risorse che AWS Glue i vari processi richiedono per l'esecuzione per conto dell'utente. Queste risorse includono AWS Glue Amazon S3, IAM, CloudWatch Logs e Amazon. EC2 Se segui la convenzione di denominazione delle risorse specificata in questa politica, AWS Glue i processi dispongono delle autorizzazioni richieste. Questa policy è in genere collegata ai ruoli specificati quando si definiscono crawler, processi ed endpoint di sviluppo.

- AWSGlueConsoleFullAccess— Concede l'accesso completo alle AWS Glue risorse quando un'identità a cui è allegata la policy utilizza la AWS console di gestione. Se si segue la convenzione per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questa policy è in genere associata agli utenti della AWS Glue console.

## Configurazione di Edge NXT di Blackbaud Raiser

Prima di poter utilizzare il trasferimento AWS Glue di dati da Edge NXT di Blackbaud Raiser, devi soddisfare questi requisiti:

Requisiti minimi

I requisiti minimi sono i seguenti:

- Avere un account Edge NXT di Blackbaud Raiser.
- Hai generato un token di accesso nel tuo account Edge NXT di Blackbaud Raiser con l'ambito appropriato assegnato per accedere a. read/write APIs Per ulteriori informazioni, consultare Autorizzazione.

Se soddisfi questi requisiti, sei pronto per connetterti all'account Edge NXT AWS Glue di Blackbaud Raiser.

## Configurazione delle connessioni Edge NXT di Blackbaud Raiser

Edge NXT di Blackbaud Raiser supporta il tipo di concessione AUTHORIZATION_CODE per. OAuth2

- Questo tipo di concessione è considerato «a tre gambe» in OAuth quanto si basa sul reindirizzamento degli utenti a un server di autorizzazione di terze parti per autenticare l'utente. Viene utilizzato durante la creazione di connessioni tramite la console. AWS Glue La AWS Glue console reindirizzerà l'utente a Edge NXT di Blackbaud Raiser, dove l'utente deve effettuare il login e consentire AWS Glue le autorizzazioni richieste per accedere alla propria istanza Edge NXT di Blackbaud Raiser.
- Gli utenti possono scegliere di creare la propria app connessa in Edge NXT di Blackbaud Raiser e fornire il proprio ID client, chiave di sottoscrizione e URL di istanza durante la creazione di connessioni tramite la console. AWS Glue In questo scenario, verranno comunque reindirizzati a Edge NXT di Blackbaud Raiser per accedere e autorizzare l'accesso alle proprie risorse. AWS Glue

- Questo tipo di concessione genera un token di aggiornamento e un token di accesso. Il token di accesso dura poco tempo e può essere aggiornato automaticamente senza l'interazione dell'utente utilizzando il token di aggiornamento.
- Per la documentazione pubblica di Edge NXT di Blackbaud Raiser sulla creazione di un'app connessa per il flusso del codice di autorizzazione, vedi Autorizzazione. OAuth

Per configurare una connessione Edge NXT di Blackbaud Raiser:

1. In AWS Secrets Manager, crea un segreto con i seguenti dettagli:
   a. Per le app connesse gestite dal cliente, il segreto deve contenere la chiave API dell'app connessa con *USER_MANAGED_CLIENT_APPLICATION_CLIENT_SECRET* come chiave.
   b. Nota: devi creare un segreto per le tue connessioni in AWS Glue.

1. In AWS Glue Glue Studio, crea una connessione in Connessioni dati seguendo i passaggi seguenti:
   a. Quando selezioni un'origine dati, scegli Edge NXT di Blackbaud Raiser.
   b. Fornisci il `INSTANCE_URL` dell'account Edge NXT di Blackbaud Raiser a cui desideri connetterti.
   c. Fornisci il `clientId` dell'applicazione client gestita dall'utente.
   d. Fornisci la chiave di sottoscrizione associata al tuo account.
   e. Seleziona il ruolo AWS IAM che AWS Glue può assumere e dispone delle autorizzazioni per le seguenti azioni:

      JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:DescribeSecret",
        "secretsmanager:GetSecretValue",
        "secretsmanager:PutSecretValue",
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DeleteNetworkInterface"
```

```
      ],
      "Resource": "*"
    }
  ]
}
```

f. Seleziona quello `secretName` che desideri utilizzare per questa connessione per AWS Glue inserire i token.

g. Selezionare le opzioni di rete se si desidera utilizzare la propria rete.

2. Concedi il permesso di lettura `secretName` al ruolo IAM associato al tuo AWS Glue lavoro.

## Lettura dalle entità Edge NXT di Blackbaud Raiser

Prerequisito

Un oggetto Edge NXT di Blackbaud Raiser da cui si desidera leggere. È necessario il nome dell'oggetto.

Entità supportate come origine:

| Entità | Può essere filtrato | Supporta limite | Supporta Ordina per | Supporta Seleziona * | Supporta partizionamento |
|---|---|---|---|---|---|
| Indirizzo del costituente | Sì | Sì | No | Sì | Sì |
| Istruzione del costituente | Sì | Sì | No | Sì | Sì |
| Indirizzo e-mail del costituente | Sì | Sì | No | Sì | Sì |
| Numero di telefono del costituente | Sì | Sì | No | Sì | Sì |

| Entità | Può essere filtrato | Supporta limite | Supporta Ordina per | Supporta Seleziona * | Supporta partizionamento |
|---|---|---|---|---|---|
| Nota del costituente | Sì | Sì | No | Sì | Sì |
| Relazione del costituente | Sì | Sì | No | Sì | Sì |
| Presenza online del costituente | Sì | Sì | No | Sì | Sì |
| Opportunità | Sì | Sì | No | Sì | Sì |
| Contestazione | Sì | Sì | No | Sì | Sì |
| Campagna | Sì | Sì | No | Sì | Sì |
| Fondi | Sì | Sì | No | Sì | Sì |
| Pacchetto | Sì | Sì | No | Sì | Sì |
| Batch Gift | Sì | Sì | No | Sì | No |
| Partecipante dell'evento | Sì | Sì | Sì | Sì | Sì |
| Entità che si occupa della raccolta fondi del costituente | No | No | No | Sì | No |
| Gift | Sì | Sì | Sì | Sì | Sì |
| Appartenenza | Sì | Sì | No | Sì | Sì |

| Entità | Può essere filtrato | Supporta limite | Supporta Ordina per | Supporta Seleziona * | Supporta partizionamento |
|---|---|---|---|---|---|
| Azione | Sì | Sì | No | Sì | No |
| Costituente | Sì | Sì | Sì | Sì | Sì |
| Beni del costituente | Sì | Sì | No | Sì | Sì |
| Event | Sì | Sì | Sì | Sì | Sì |
| Campo personalizzato Gift | Sì | Sì | No | Sì | Sì |

Esempio:

```
blackbaud_read = glueContext.create_dynamic_frame.from_options(
    connection_type="BLACKBAUD",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "entityName",
        "API_VERSION": "v1",
        "SUBSCRIPTION_KEY": <Subscription key associated with one's developer account>
    }
```

Dettagli dell'entità e del campo Edge NXT di Blackbaud Raiser

Per ulteriori informazioni sui dettagli dell'entità e del campo, consultare:

- Azione
- Costituente
- Indirizzo del costituente
- Iscrizione del costituente
- Entità che si occupa della raccolta fondi del costituente
- Istruzione del costituente

- Indirizzo e-mail del costituente
- Numero di telefono del costituente
- Nota del costituente
- Presenza online del costituente
- Relazione del costituente
- Evento
- Partecipante dell'evento
- Contestazione
- Campagna
- Fondi
- Pacchetto
- Gift
- Campo personalizzato Gift
- Batch Gift
- Opportunità
- Codici del costituente

> **Note**
>
> I tipi di dati Struct e List vengono convertiti in tipo di dati String e il tipo di DateTime dati viene convertito in Timestamp nella risposta dei connettori.

Query di partizionamento

Partizionamento basato sul campo:

Edge NXT di Blackbaud Raiser non supporta il partizionamento basato sul campo o basato su record.

Partizionamento basato su record:

Puoi fornire l'opzione Spark aggiuntiva `NUM_PARTITIONS` se desideri utilizzare la concorrenza in Spark. Con questo parametro, la query originale viene suddivisa in `NUM_PARTITIONS` query secondarie che possono essere eseguite contemporaneamente dalle attività Spark.

Nel partizionamento basato su record, l'API Edge NXT di Blackbaud Raiser esegue query sul numero totale di record presenti e lo divide per il numero NUM_PARTITIONS fornito. Il numero di record risultante viene quindi recuperato contemporaneamente da ciascuna sotto-query.

- NUM_PARTITIONS: il numero di partizioni.

Esempio:

```
blackbaud_read = glueContext.create_dynamic_frame.from_options(
    connection_type="BLACKBAUD",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "entityName",
        "API_VERSION": "v1",
        "NUM_PARTITIONS": "2",
        "SUBSCRIPTION_KEY": <Subscription key associated with one's developer account>
    }
```

## Opzioni di connessione Edge NXT di Blackbaud Raiser

Di seguito sono riportate le opzioni di connessione per Edge NXT di Blackbaud Raiser:

- ENTITY_NAME(String): (obbligatorio) utilizzato per la lettura. Il nome del tuo oggetto in Edge NXT di Blackbaud Raiser.
- API_VERSION(String): (obbligatorio) utilizzato per la lettura. La versione dell'API Rest Edge NXT di Blackbaud Raiser che si desidera utilizzare.
- SELECTED_FIELDS(List<String>): predefinito: vuoto (SELECT *). Utilizzato per la lettura. Colonne da selezionare per l'oggetto.
- FILTER_PREDICATE(Stringa): predefinito: vuoto. Utilizzato per la lettura. Dev'essere in formato Spark SQL.
- QUERY(Stringa): predefinito: vuoto. Utilizzato per la lettura. Query Spark SQL completa.
- NUM_PARTITIONS(Integer): predefinito: 1. Utilizzato per la lettura. Numero di partizioni da leggere. Valore di esempio: 10.
- SUBSCRIPTION_KEY(String): (obbligatorio) predefinito, vuoto. Utilizzato per la lettura. Chiave di sottoscrizione associata al proprio account sviluppatore.

## Limitazioni di Edge NXT di Blackbaud Raiser

Di seguito sono riportate le limitazioni o le note relative a Edge NXT di Blackbaud Raiser:

- Il SaaS supporta solo l'operatore `EQUAL_TO`, che restituisce i risultati creati o modificati nella o dopo la data specificata. Inoltre, il campo "id" è un tipo di dati String. Non è possibile identificare i campi che non ammettono valori null. Pertanto, il partizionamento basato sul campo non è supportato.
- Il pull incrementale è supportato solo dall'entità `Event` con frequenze giornaliere, mensili e settimanali.
- L'entità che si occupa della raccolta fondi di un costituente restituisce un massimo di 20 record.
- Partizionamento basato su record:
  - Non supportato dalle entità `Action`, `Constituent Fundraiser Assignment` o `Gift Batch`.
  - Il partizionamento basato sui record con il predicato del filtro è supportato solo dalle entità `Event` e `Event Participant`. Se il predicato di un filtro viene utilizzato con qualsiasi altra entità supportata dai record, verrà generata un'eccezione.
- Nell'entità `Gift Custom Field`, il campo 'value' deve essere usato insieme al campo 'category', altrimenti si ottiene una risposta non filtrata. Pertanto, per obbligare l'utente a inserire il campo 'category' durante il filtraggio con il campo 'value', verrà generata un'eccezione in caso di non osservanza del requisito.
- I campi `date_added` e `last_modified` per tutte le entità applicabili non supportano gli operatori comparativi. Supportano solo l'operatore equal to. Inoltre, non esiste alcun campo che possa essere abbinato a quelli sopra menzionati per fornire una serie di record. Pertanto, questi campi possono essere solo soggetti a query e non possono supportare il trasferimento incrementale.
- Il campo `added_by` dell'entità `Gift Batch` non sarà considerato filtrabile in quanto potrebbe non emettere i risultati corretti.
- È presente una latenza di circa 30 minuti per il recupero dei record tramite l'endpoint `/GET Gift List` al momento dell'inserimento dei dati nell'entità `Gift`.
- Il supporto per il trasferimento incrementale è stato interrotto per l'entità Gift a causa delle limitazioni dell'origine dati.
- È presente una latenza di 10 minuti per il campo di stato nell'entità Opportunity.
- L'entità `Fundraiser Assignment` ha `Constituent` come entità dipendente. Il connettore ne carica al massimo 5.000 tra cui IDs scegliere, per evitare che la dimensione della risposta superi la dimensione massima consentita del carico utile.

# Connessione a CircleCI

CircleCI è una piattaforma di integrazione continua e distribuzione continua. Il tuo account CircleCI contiene dati relativi ai tuoi progetti, alle tue pipeline, ai tuoi flussi di lavoro e altro ancora. Se sei un utente CircleCI, puoi connetterti AWS Glue al tuo account CircleCI. Quindi, puoi usare CircleCI come origine dati nei tuoi processi ETL. Esegui questi processi per trasferire dati tra CircleCI AWS e servizi o altre applicazioni supportate.

Argomenti


- AWS Glue supporto per CircleCI
- Policy contenenti le operazioni API per creare e utilizzare le connessioni
- Configurazione di CircleCI
- Configurazione delle connessioni CircleCI
- Lettura da entità CircleCI
- Opzioni di connessione CircleCI
- Limitazioni di CircleCI


## AWS Glue supporto per CircleCI

AWS Glue supporta CircleCI come segue:

Supportato come origine?

Sì. È possibile utilizzare i job AWS Glue ETL per interrogare i dati da CircleCI.

Supportato come destinazione?

No.

Versioni API CircleCI supportate

Sono supportate le seguenti versioni API CircleCI:

- v2

## Policy contenenti le operazioni API per creare e utilizzare le connessioni

La seguente policy di esempio descrive le autorizzazioni AWS IAM richieste per la creazione e l'utilizzo delle connessioni. Se si sta creando un nuovo ruolo, creare una policy che contenga quanto segue:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:ListConnectionTypes",
        "glue:DescribeConnectionType",
        "glue:RefreshOAuth2Tokens",
        "glue:ListEntities",
        "glue:DescribeEntity"
      ],
      "Resource": "*"
    }
  ]
}
```

In alternativa, se non desideri utilizzare il metodo precedente, utilizza le seguenti policy IAM gestite:

- AWSGlueServiceRole— Garantisce l'accesso alle risorse che AWS Glue i vari processi richiedono per l'esecuzione per conto dell'utente. Queste risorse includono AWS Glue Amazon S3, IAM, CloudWatch Logs e Amazon. EC2 Se segui la convenzione di denominazione delle risorse specificata in questa politica, AWS Glue i processi dispongono delle autorizzazioni richieste. Questa policy è in genere collegata ai ruoli specificati quando si definiscono crawler, processi ed endpoint di sviluppo.
- AWSGlueConsoleFullAccess— Concede l'accesso completo alle AWS Glue risorse quando un'identità a cui è allegata la policy utilizza la AWS console di gestione. Se si segue la convenzione per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questa policy è in genere associata agli utenti della AWS Glue console.

## Configurazione di CircleCI

Prima di poter AWS Glue utilizzare il trasferimento di dati da CircleCI, è necessario soddisfare questi requisiti:

Requisiti minimi

I requisiti minimi sono i seguenti:

- Avere un account con CircleCI che contiene i dati che si desidera trasferire.
- Nelle impostazioni utente del tuo account, aver creato un token API personale. Per ulteriori informazioni, consultare Creazione di un token API personale.
- Fornisci il token API personale a AWS Glue durante la creazione della connessione.

Se soddisfi questi requisiti, sei pronto per connetterti AWS Glue al tuo account CircleCI.

## Configurazione delle connessioni CircleCI

CircleCI supporta l'autenticazione personalizzata.

Per configurare una connessione CircleCI:

1. In AWS Secrets Manager, crea un segreto con i seguenti dettagli:
   a. Per le app connesse gestite dal cliente, il segreto deve contenere la chiave API dell'app connessa con `Circle-Token` come chiave.
   b. Nota: devi creare un segreto per le tue connessioni in AWS Glue.

1. In AWS Glue Glue Studio, crea una connessione in Connessioni dati seguendo i passaggi seguenti:
   a. Quando si seleziona una Origine dati, selezionare CircleCI.
   b. Seleziona il ruolo AWS IAM che AWS Glue può assumere e dispone delle autorizzazioni per le seguenti azioni:

      JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
```

```
      "Effect": "Allow",
      "Action": [
        "secretsmanager:DescribeSecret",
        "secretsmanager:GetSecretValue",
        "secretsmanager:PutSecretValue",
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DeleteNetworkInterface"
      ],
      "Resource": "*"
    }
  ]
}
```

c. Seleziona quello `secretName` che desideri utilizzare per questa connessione per AWS Glue inserire i token.

d. Selezionare le opzioni di rete se si desidera utilizzare la propria rete.

2. Concedi il permesso di lettura `secretName` al ruolo IAM associato al tuo AWS Glue lavoro.

## Lettura da entità CircleCI

Prerequisito

Un oggetto di CircleCI da cui si desidera leggere. È necessario il nome dell'oggetto.

Entità supportate come origine:

| Entità | Può essere filtrato | Supporta limite | Supporta Ordina per | Supporta Seleziona * | Supporta il partizionamento |
|---|---|---|---|---|---|
| Contesto | Sì | No | No | Sì | No |
| Parametri di riepilogo dell'organizzazione | Sì | No | No | Sì | No |
| Gasdotto | No | No | No | Sì | No |

| Entità | Può essere filtrato | Supporta limite | Supporta Ordina per | Supporta Seleziona * | Supporta il partizionamento |
|---|---|---|---|---|---|
| Flusso di lavoro di un'attività di pipeline | Sì | No | No | Sì | No |
| Ramificazione del progetto | Sì | No | No | Sì | No |
| Test flaky del progetto | No | No | No | Sì | No |
| Parametri di riepilogo del progetto | Sì | No | No | Sì | No |
| Schedule | No | No | No | Sì | No |
| Serie temporali del lavoro del flusso | Sì | No | No | Sì | No |
| Parametro e tendenza del flusso di lavoro | Sì | No | No | Sì | No |
| Esecuzione recente del flusso di lavoro | Sì | No | No | Sì | No |

| Entità | Può essere filtrato | Supporta limite | Supporta Ordina per | Supporta Seleziona * | Supporta il partizionamento |
|---|---|---|---|---|---|
| Parametro di riepilogo del flusso di lavoro | Sì | No | No | Sì | No |
| Parametro di test del flusso di lavoro | Sì | No | No | Sì | No |

Esempio:

```
circleci_read = glueContext.create_dynamic_frame.from_options(
    connection_type="circleci",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "context/e7ea2945-dccb-4205-b673-8391fe1b3a4c",
        "API_VERSION": "v2"
    }
```

Dettagli dell'entità e del campo di CircleCI

Per ulteriori informazioni sui dettagli dell'entità e del campo, consulta:

- Contesti
- Parametri di riepilogo del progetto
- Serie temporali del lavoro del flusso
- Parametri di riepilogo dell'organizzazione
- Ramificazione del progetto
- Test flaky del progetto
- Esecuzione recente del flusso di lavoro
- Parametri di riepilogo del flusso di lavoro
- Parametri e tendenze del flusso di lavoro

- Parametri di test del flusso di lavoro
- Pipeline
- Flusso di lavoro della pipeline
- Orari

Entità con metadati statici:

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| Contesto | Creato alle | Stringa | |
| | ID | Stringa | |
| | Name | Stringa | |
| | Tipo di proprietario | Stringa | EQUAL_TO |
| Parametro di riepilogo dell'organizzazione | Tutti i progetti | List | |
| | Dati dell'orga nizzazione | Struct | |
| | Dati del progetto dell'organizzazione | List | |
| | Nomi del progetto | Stringa | EQUAL_TO |
| | Finestra di report | Stringa | EQUAL_TO |
| Gasdotto | Ramo | Stringa | EQUAL_TO |
| | Creato alle | Stringa | |
| | Errori | List | |
| | ID | Stringa | |
| | Numero | Numero intero | |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | Nome comune del progetto | Stringa | |
| | Stato | Stringa | |
| | Trigger | Struct | |
| | Parametri trigger | Struct | |
| | Aggiornato alle | Stringa | |
| | VCS | Struct | |
| Flusso di lavoro di un'attività di pipeline | Annullato da | Stringa | |
| | Creato alle | Stringa | |
| | Errore di | Stringa | |
| | ID | Stringa | |
| | Name | Stringa | |
| | ID pipeline | Stringa | |
| | Numero della pipeline | Numero intero | |
| | Nome comune del progetto | Stringa | |
| | Iniziato da | Stringa | |
| | Status | Stringa | |
| | Fermato alle | Stringa | |
| | Tag | Stringa | |
| Ramificazione del progetto | Rami | List | |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | ID dell'organizzazione | Stringa | |
| | ID progetto | Stringa | |
| | Workflow Name (Nome flusso di lavoro) | Stringa | EQUAL_TO |
| Test flaky del progetto | ClassName | Stringa | |
| | File | Stringa | |
| | Nome processo | Stringa | |
| | Numero del lavoro | Numero intero | |
| | Numero della pipeline | Numero intero | |
| | Origine | Stringa | |
| | Nome del test | Stringa | |
| | Tempo impiegato | Numero intero | |
| | Numero di flaky | Numero intero | |
| | Flusso di lavoro creato alle | Stringa | |
| | Workflow ID (ID flusso di lavoro) | Stringa | |
| | Workflow Name (Nome flusso di lavoro) | Stringa | |
| Parametri di riepilogo del progetto | Tutte le ramificazioni | List | |
| | Tutti i flussi di lavoro | List | |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | Rami | Stringa | EQUAL_TO |
| | ID organizzazione | Stringa | |
| | Dati di progetto | Struct | |
| | ID progetto | Stringa | |
| | Dati della ramificazione del flusso di lavoro del progetto | List | |
| | Dati del flusso di lavoro del progetto | List | |
| | Finestra di report | Stringa | EQUAL_TO |
| | Nomi del flusso di lavoro | Stringa | EQUAL_TO |
| Schedule | Attore | Struct | |
| | Creato alle | Stringa | |
| | Description | Stringa | |
| | ID | Stringa | |
| | Name | Stringa | |
| | Parameters | Struct | |
| | Nome comune del progetto | Stringa | |
| | Orario | Struct | |
| | Aggiornato alle | Stringa | |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| Serie temporali del lavoro del flusso | Ramo | Stringa | EQUAL_TO |
| | Granularity (Granularità) | Stringa | EQUAL_TO |
| | Terminato al massimo alle | Stringa | |
| | Metriche | Struct | |
| | Iniziato almeno alle | Stringa | |
| | Name | Stringa | |
| | Data di inizio e fine | DateTime | EQUAL_TO, BETWEEN |
| | Time stamp | Stringa | |
| Parametro e tendenza del flusso di lavoro | Tutte le ramificazioni | Booleano | EQUAL_TO |
| | Rami | Stringa | EQUAL_TO |
| | Metriche | Struct | |
| | Tendenze | Struct | |
| | Nomi del flusso di lavoro | List | |
| Esecuzione recente del flusso di lavoro | Tutte le ramificazioni | Booleano | EQUAL_TO |
| | Ramo | Stringa | EQUAL_TO |
| | Creato alle | Stringa | |
| | Crediti utilizzati | Numero intero | |
| | Durata | Numero intero | |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | ID | Stringa | |
| | Approvazione | Booleano | |
| | Data di inizio e fine | DateTime | EQUAL_TO, BETWEEN |
| | Status | Stringa | |
| | Fermato alle | Stringa | |
| Parametro di riepilogo del flusso di lavoro | Tutte le ramificazioni | Booleano | EQUAL_TO |
| | Ramo | Stringa | EQUAL_TO |
| | Metriche | Struct | |
| | Name | Stringa | |
| | ID progetto | Stringa | |
| | Finestra di report | Stringa | EQUAL_TO |
| | Fine finestra | Stringa | |
| | Inizio finestra | Stringa | |
| Parametro di test del flusso di lavoro | Conteggio medio dei test | Numero intero | |
| | Ramo | Stringa | EQUAL_TO |
| | Test più falliti | List | |
| | Test più falliti extra | Numero intero | |
| | Test più lenti | List | |
| | Test più lenti extra | Numero intero | |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | Esecuzioni di test | List | |
| | Esecuzioni di test totali | Numero intero | |

> **Note**
>
> I tipi di dati Struct e List vengono convertiti in tipo di dati String nella risposta del connettore.

Query di partizionamento

CircleCI non supporta il partizionamento basato su campi o su record.

## Opzioni di connessione CircleCI

Di seguito sono riportate le opzioni di connessione per CircleCI:

- `ENTITY_NAME`(Stringa): (obbligatorio) utilizzato per la lettura. Il nome dell'oggetto in CircleCI.
- `API_VERSION`(Stringa): (obbligatorio) utilizzato per la lettura. Versione dell'API Rest di CircleCI da utilizzare.
- `SELECTED_FIELDS`(List<String>): predefinito: vuoto (SELECT *). Utilizzato per la lettura. Colonne da selezionare per l'oggetto.
- `FILTER_PREDICATE`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Dev'essere in formato Spark SQL.
- `QUERY`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Query Spark SQL completa.

## Limitazioni di CircleCI

Di seguito sono riportate le limitazioni o le note di CircleCI:

- CircleCI non supporta il partizionamento basato su campi o record.
- I campi di filtro contenenti “-” (trattino) funzionano solo se sono racchiusi all'interno di caratteri di apice. Ad esempio: `workflow-name` = "abc"

- Il tipo GitLab VCS non può essere supportato in quanto non esiste un modo programmatico per recuperare l' «ID progetto» richiesto per il percorso dell'entità VCS. GitLab

# Connessione a Datadog

Datadog è una piattaforma di monitoraggio e analisi per applicazioni su scala cloud, come infrastrutture, applicazioni, servizi e strumenti.

Argomenti


- AWS Glue supporto per Datadog
- Policy contenenti le operazioni API per creare e utilizzare le connessioni
- Configurazione di Datadog
- Configurazione delle connessioni Datadog
- Leggere dalle entità Datadog
- Opzioni di connessione a Datadog
- Creazione di un account Datadog
- Limitazioni


## AWS Glue supporto per Datadog

AWS Glue supporta Datadog come segue:

Supportato come origine?

Sì. È possibile utilizzare i job AWS Glue ETL per interrogare i dati da Datadog.

Supportato come destinazione?

No.

Versioni dell'API Datadog supportate

- v1
- v2

## Policy contenenti le operazioni API per creare e utilizzare le connessioni

La seguente politica di esempio descrive le AWS autorizzazioni richieste per la creazione e l'utilizzo delle connessioni. Se si sta creando un nuovo ruolo, creare una policy che contenga quanto segue:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:ListConnectionTypes",
        "glue:DescribeConnectionType",
        "glue:RefreshOAuth2Tokens",
        "glue:ListEntities",
        "glue:DescribeEntity"
      ],
      "Resource": "*"
    }
  ]
}
```

In alternativa, se non desideri utilizzare il metodo precedente, utilizza le seguenti policy IAM:

- AWSGlueServiceRole — Garantisce l'accesso alle risorse che diversi AWS Glue processi richiedono per l'esecuzione per conto dell'utente. Queste risorse includono Amazon S3 AWS Glue, IAM, CloudWatch Logs e Amazon EC2. Se segui la convenzione di denominazione delle risorse specificata in questa politica, AWS Glue i processi dispongono delle autorizzazioni necessarie. Questa policy è in genere collegata ai ruoli specificati quando si definiscono crawler, processi ed endpoint di sviluppo.
- AWSGlueConsoleFullAccess — Concede l'accesso completo alle AWS Glue risorse quando un'identità a cui è allegata la politica utilizza la AWS console di gestione. Se si segue la convenzione per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questa policy è in genere associata agli utenti della AWS Glue console.

## Configurazione di Datadog

Prima di poter AWS Glue utilizzare il trasferimento da Datadog, è necessario soddisfare i seguenti requisiti:

Requisiti minimi

- Hai un account Datadog con e. DD-API-KEY DD-APPLICATION-KEY Per ulteriori informazioni sulla creazione di account, consulta Creazione di un account Datadog.
- L'account Datadog dispone dell'accesso all'API con una licenza valida.

Datadog supporta i seguenti sei. URLs Tutti i client API Datadog sono configurati di default per utilizzare il sito Datadog. US1 APIs Se ti trovi sul sito Datadog EU, devi selezionare l'URL https://api.datadoghq.eu con la fine del `DD-API-KEY` sito Datadog EU per `DD-APPLICATION-KEY` accedere a. APIs Allo stesso modo, per altri siti, è necessario selezionare il rispettivo sito URLs con il `DD-API-KEY and DD-APPLICATION-KEY` rispettivo sito.

- US1 URL API: https://api.datadoghq.com https://api.datadoghq.com
- URL API UE: https://api.datadoghq.eu
- US3 URL DELL'API — https://api.us3.datadoghq.com
- US5 URL DELL'API — https://api.us5.datadoghq.com
- URL DELL'API S1-FED — https://api.ddog-gov.com
- URL dell'API per il Giappone — https://api.ap1.datadoghq.com

Se soddisfi questi requisiti, sei pronto per connetterti AWS Glue al tuo account Datadog.

## Configurazione delle connessioni Datadog

Datadog supporta l'autenticazione personalizzata. Di seguito sono riportati i passaggi per configurare la connessione a Datadog:

Per configurare una connessione Datadog:

1. In AWS Secrets Manager, crea un segreto con i seguenti dettagli:

   Per le app connesse gestite dal cliente: il segreto deve contenere l'app connessa Consumer Secret con **`API_KEY`** e **`APPLICATION_KEY`** come chiavi.

> **Note**
>
> È necessario creare un segreto per ogni connessione in AWS Glue.

2. In AWS Glue Studio, crea una connessione in Connessioni dati seguendo i passaggi seguenti:
   a. In Tipo di connessione, seleziona Datadog.
   b. Fornisci `Instance_Url` di Datadog a cui desideri connetterti.
   c. Seleziona il ruolo IAM per il quale AWS Glue può assumere e dispone delle autorizzazioni per le seguenti azioni:

      JSON

      ```
      {
        "Version":"2012-10-17",
        "Statement": [
          {
            "Effect": "Allow",
            "Action": [
              "secretsmanager:DescribeSecret",
              "secretsmanager:GetSecretValue",
              "secretsmanager:PutSecretValue",
              "ec2:CreateNetworkInterface",
              "ec2:DescribeNetworkInterfaces",
              "ec2:DeleteNetworkInterface"
            ],
            "Resource": "*"
          }
        ]
      }
      ```

   d. Seleziona quello `secretName` che desideri utilizzare per questa connessione per AWS Glue inserire i token.
   e. Selezionare le opzioni di rete se si desidera utilizzare la propria rete.
3. Concedi il permesso di lettura `secretName` al ruolo IAM associato al tuo AWS Glue lavoro.

## Leggere dalle entità Datadog

### Prerequisiti

Un oggetto Datadog da cui si desidera leggere. Consulta la tabella delle entità supportate qui di seguito per verificare quelle disponibili.

Entità supportate

| Entità | Può essere Filtrato | Supporta Limite | Supporta Ordina per | Supporta Seleziona * | Supporta il partizionamento |
|---|---|---|---|---|---|
| Serie temporali metriche | Sì | No | No | Sì | No |
| Registrare query | Sì | Sì | Sì | Sì | No |

Esempio

```
Datadog_read = glueContext.create_dynamic_frame.from_options(
    connection_type="datadog",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "log-queries",
        "API_VERSION": "v2",
        "INSTANCE_URL": "https://api.datadoghq.com",
        "FILTER_PREDICATE": "from = `2023-10-03T09:00:26Z`"
    }
```

Dettagli dell'entità e del campo di Datadog

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| Serie temporali metriche | error | Stringa | N/A |
| | aggr | Stringa | N/A |
| | attributes | Struct | N/A |
| | display_name | Stringa | N/A |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | end | DateTime | N/A |
| | expression | Stringa | N/A |
| | intervallo | Numero intero | N/A |
| | length | Numero intero | N/A |
| | parametro | Stringa | N/A |
| | pointlist | List | N/A |
| | query_index | Numero intero | N/A |
| | scope | Stringa | N/A |
| | rapida | DateTime | N/A |
| | tag_set | List | N/A |
| | unità | Struct | N/A |
| | from_to_date | DateTime | BETWEEN |
| | query | Stringa | EQUAL_TO |
| | status | Stringa | N/A |
| | tipo | Stringa | N/A |
| | host | Stringa | N/A |
| Registrare query | id | Stringa | N/A |
| | attributes | Struct | N/A |
| | timestamp | DateTime | N/A |
| | tipo | Stringa | N/A |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | from | DateTime | BETWEEN,EQUAL_TO |
| | indici | List | EQUAL_TO |
| | storage_tier | Stringa | EQUAL_TO |
| | query | Stringa | EQUAL_TO |

## Opzioni di connessione a Datadog

Di seguito sono riportate le opzioni di connessione per Datadog:

- `ENTITY_NAME`(Stringa): (obbligatorio) utilizzato per lettura/scrittura. Il nome del tuo oggetto in Datadog.
- `API_VERSION`(Stringa): (obbligatorio) utilizzato per lettura/scrittura. La versione Datadog Rest API che desideri utilizzare. La versione `v1` supporta l'entità `metrics-timeseries`, mentre la versione `v2` supporta l'entità `log-queries`.
- `INSTANCE_URL`(Stringa): (obbligatorio) utilizzato per la lettura. URL dell'istanza Datadog. L'URL dell'istanza Datadog varia in base alla regione.
- `SELECTED_FIELDS`(List<String>): predefinito: vuoto (SELECT *). Utilizzato per la lettura. Colonne da selezionare per l'oggetto.
- `FILTER_PREDICATE`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Dev'essere in formato Spark SQL.
- `QUERY`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Query Spark SQL completa.

## Creazione di un account Datadog

1. Vai a https://www.datadoghq.com/.
2. Scegli INIZIA GRATUITAMENTE.
3. Inserisci le informazioni richieste e registrati.
4. Installa il programma di installazione dell'agente di Datadog come suggerito.

5. Assicurarsi che l'account venga registrato all'interno di un'organizzazione valida (dall'elenco disponibile) che dispone di un abbonamento attivo a Datadog.
6. Dopo aver effettuato l'accesso al tuo account Datadog, passa con il mouse sul tuo nome utente nell'angolo in alto a destra per visualizzare i dettagli delle chiavi:
   a. Per ottenere la tua chiave API, seleziona Chiavi API.
   b. Per ottenere la chiave dell'applicazione, seleziona Chiavi dell'applicazione.

### Limitazioni

Di seguito sono riportate le limitazioni per il connettore di Datadog:

- Datadog non supporta il partizionamento basato su campi o record.
- `from` è un parametro di filtro obbligatorio per l'entità `Log Queries`.
- `from_to_date` e `query` sono parametri di filtro obbligatori per l'entità `Metrics Timeseries`.

## Connessione a Docusign Monitor

Docusign Monitor aiuta le organizzazioni a proteggere i propri accordi con il monitoraggio delle attività. round-the-clock L'API Monitor trasmette queste informazioni di monitoraggio delle attività direttamente agli stack di sicurezza o agli strumenti di visualizzazione dei dati esistenti, consentendo ai team di rilevare attività non autorizzate, indagare sugli incidenti e rispondere rapidamente alle minacce verificate. Offre inoltre la flessibilità necessaria ai team di sicurezza per personalizzare pannelli di controllo e avvisi in modo da soddisfare specifiche esigenze aziendali.

Argomenti

- AWS Glue supporto per Docusign Monitor
- Policy contenenti le operazioni API per creare e utilizzare le connessioni
- Configurazione di Docusign Monitor
- Configurazione delle connessioni Docusign Monitor
- Lettura dalle entità Docusign Monitor
- Opzioni di connessione per Docusign Monitor
- Limitazioni di Docusign Monitor

## AWS Glue supporto per Docusign Monitor

AWS Glue supporta Docusign Monitor come segue:

Supportato come origine?

Sì. È possibile utilizzare i job AWS Glue ETL per interrogare i dati da Docusign Monitor.

Supportato come destinazione?

No.

Versioni delle API Docusign Monitor supportate

Sono supportate le seguenti versioni delle API Docusign Monitor:

- v2.0

## Policy contenenti le operazioni API per creare e utilizzare le connessioni

La seguente policy di esempio descrive le autorizzazioni AWS IAM richieste per la creazione e l'utilizzo delle connessioni. Se si sta creando un nuovo ruolo, creare una policy che contenga quanto segue:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:ListConnectionTypes",
        "glue:DescribeConnectionType",
        "glue:RefreshOAuth2Tokens",
        "glue:ListEntities",
        "glue:DescribeEntity"
      ],
      "Resource": "*"
    }
  ]
```

```
}
```

In alternativa, se non desideri utilizzare il metodo precedente, utilizza le seguenti policy IAM gestite:

- AWSGlueServiceRole— Garantisce l'accesso alle risorse che AWS Glue i vari processi richiedono per l'esecuzione per conto dell'utente. Queste risorse includono AWS Glue Amazon S3, IAM, CloudWatch Logs e Amazon. EC2 Se segui la convenzione di denominazione delle risorse specificata in questa politica, AWS Glue i processi dispongono delle autorizzazioni richieste. Questa policy è in genere collegata ai ruoli specificati quando si definiscono crawler, processi ed endpoint di sviluppo.
- AWSGlueConsoleFullAccess— Concede l'accesso completo alle AWS Glue risorse quando un'identità a cui è allegata la policy utilizza la AWS console di gestione. Se si segue la convenzione per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questa policy è in genere associata agli utenti della AWS Glue console.

## Configurazione di Docusign Monitor

Prima di poter AWS Glue utilizzare il trasferimento di dati da Docusign Monitor alle destinazioni supportate, è necessario soddisfare questi requisiti:

Requisiti minimi

I requisiti minimi sono i seguenti:

- Avere un account Docusign che consente di utilizzare il prodotto software Docusign in Docusign Monitor.
- Nella console per sviluppatori del tuo account Docusign, hai creato un'app di integrazione OAuth 2.0 per. AWS Glue

  Questa app fornisce le credenziali del client che AWS Glue utilizza per accedere ai tuoi dati in modo sicuro quando effettua chiamate autenticate al tuo account. Per ulteriori informazioni, vedere OAuth 2.0 nella documentazione di Docusign Monitor.

Se soddisfi questi requisiti, sei pronto per connetterti AWS Glue al tuo account Docusign Monitor.

## Configurazione delle connessioni Docusign Monitor

Docusign Monitor supporta il tipo di concessione AUTHORIZATION_CODE.

- Questo tipo di concessione è considerato a tre gambe in OAuth quanto si basa sul reindirizzamento degli utenti al server di autorizzazione di terze parti per autenticare l'utente. Viene utilizzato durante la creazione di connessioni tramite la console. AWS Glue
- Gli utenti possono scegliere di creare la propria app connessa in Docusign Monitor e fornire il proprio ID client e il segreto del client durante la creazione di connessioni tramite la AWS Glue console. In questo scenario, verranno comunque reindirizzati a Docusign Monitor per accedere e AWS Glue autorizzare l'accesso alle proprie risorse.
- Questo tipo di concessione genera un token di aggiornamento e un token di accesso. Il token di accesso dura poco tempo e può essere aggiornato automaticamente senza l'interazione dell'utente utilizzando il token di aggiornamento.
- Per la documentazione pubblica di Docusign Monitor sulla creazione di un'app connessa per il OAuth flusso del codice di autorizzazione, consulta Docusign OAuth Connect.

Per configurare una connessione Docusign Monitor:

1. In AWS Secrets Manager, crea un segreto con i seguenti dettagli:
   a. Per le app connesse gestite dal cliente, il segreto deve contenere la chiave API dell'app connessa con `USER_MANAGED_CLIENT_APPLICATION_CLIENT_SECRET` come chiave.
   b. Nota: devi creare un segreto per le tue connessioni in AWS Glue.

1. In AWS Glue Glue Studio, crea una connessione in Connessioni dati seguendo i passaggi seguenti:
   a. In Connessioni, scegli Crea connessione.
   b. Quando scegli un'Origine dati, seleziona Docusign Monitor.
   c. Seleziona il ruolo AWS IAM che AWS Glue può assumere e dispone delle autorizzazioni per le seguenti azioni:

      JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:DescribeSecret",
```

```
        "secretsmanager:GetSecretValue",
        "secretsmanager:PutSecretValue",
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DeleteNetworkInterface"
      ],
      "Resource": "*"
    }
  ]
}
```

d. Fornisci l'applicazione client gestita dagli utenti ClientId dell'app Docusign Monitor.

e. Seleziona quello `secretName` che desideri utilizzare per questa connessione per AWS Glue inserire i token.

f. Selezionare le opzioni di rete se si desidera utilizzare la propria rete.

2. Concedi il permesso di lettura `secretName` al ruolo IAM associato al tuo AWS Glue lavoro.

## Lettura dalle entità Docusign Monitor

Prerequisito

Un oggetto di Docusign Monitor da cui desideri leggere.

Entità supportate come origine:

| Entità | Può essere filtrato | Supporta limite | Supporta Ordina per | Supporta Seleziona * | Supporta il partizionamento |
|---|---|---|---|---|---|
| Monitoraggio dei dati | Sì | Sì | No | Sì | No |

Esempio:

```
docusignmonitor_read = glueContext.create_dynamic_frame.from_options(
    connection_type="docusign_monitor",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "monitoring-data",
```

```
        "API_VERSION": "v2.0"
    }
```

Dettagli dell'entità e del campo di Docusign Monitor

Entità con metadati statici:

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| Monitoraggio dei dati | timestamp | DateTime | N/D |
| | eventId | Stringa | N/A |
| | applicazione | Stringa | N/A |
| | ambiente | Stringa | N/A |
| | site | Stringa | N/A |
| | traceToken | Stringa | N/A |
| | organizationId | Stringa | N/A |
| | accountId | Stringa | N/A |
| | userId | Stringa | N/A |
| | oggetto | Stringa | N/A |
| | operazione | Stringa | N/A |
| | property | Stringa | N/A |
| | field | Stringa | N/A |
| | result | Stringa | N/A |
| | IntegratorKey | Stringa | N/A |
| | customerVisible | Stringa | N/A |
| | version | Stringa | N/A |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | userAgent | Stringa | N/A |
| | userAgentClientInformazioni | Struct | N/D |
| | ipAddress | Stringa | N/A |
| | ipAddressLocation | Struct | N/D |
| | data | Stringa | N/A |
| | origine | Stringa | N/A |
| | latitudine | Double | N/D |
| | longitudine | Double | N/D |
| | città | Stringa | N/A |
| | stato | Stringa | N/A |
| | paese | Stringa | N/A |
| | usUserMemberOfDomain | Booleano | N/D |
| | affectedUserIsMemberOfDomain | Booleano | N/D |
| | proxyStatus | Stringa | N/A |
| | proxyType | Stringa | N/A |
| | proxyLevel | Stringa | N/A |
| | referencedUserId | Stringa | N/A |
| | device | Stringa | N/A |
| | browser | Stringa | N/A |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | cursor | DateTime | EQUAL_TO |

Query di partizionamento

Docusign Monitor non supporta il partizionamento basato su campi o record.

## Opzioni di connessione per Docusign Monitor

Di seguito sono riportate le opzioni di connessione per Docusign Monitor:

- `ENTITY_NAME`(String): (obbligatorio) utilizzato per la lettura. Il nome del tuo oggetto in Docusign Monitor.
- `API_VERSION`(String): (obbligatorio) utilizzato per la lettura. Versione dell'API Rest di Docusign Monitor da utilizzare.
- `SELECTED_FIELDS`(List<String>): predefinito: vuoto (SELECT *). Utilizzato per la lettura. Colonne da selezionare per l'oggetto.
- `QUERY`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Query Spark SQL completa.
- `FILTER_PREDICATE`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Dev'essere in formato Spark SQL.

## Limitazioni di Docusign Monitor

Di seguito sono riportate le limitazioni o le note relative a Docusign Monitor:

- Quando si applica un filtro utilizzando il campo `cursor`, l'API recupera i record relativi ai sette giorni successivi a partire dalla data specificata.
- Se non viene fornito alcun filtro, l'API recupera i record relativi ai sette giorni precedenti rispetto alla data corrente della richiesta API.
- Docusign Monitor non supporta il partizionamento basato su campi o record.
- Docusign Monitor non supporta la funzionalità Ordina per.

# Connessione a Domo

Domo è uno strumento che funziona come un pannello di controllo basato sul cloud. La piattaforma applicativa aziendale di Domo pone le basi necessarie per estendere lo strumento, in modo da poter creare soluzioni personalizzate più velocemente.

Argomenti


- AWS Glue supporto per Domo
- Policy contenenti le operazioni API per creare e utilizzare le connessioni
- Configurazione di Domo
- Configurazione delle connessioni Domo
- Lettura dalle entità di Domo
- Opzioni di connessione di Domo
- Limitazioni di Domo


## AWS Glue supporto per Domo

AWS Glue supporta Domo come segue:

Supportato come origine?

Sì. È possibile utilizzare i job AWS Glue ETL per interrogare i dati da Domo.

Supportato come destinazione?

No.

Versioni delle API Domo supportate

Sono supportate le seguenti versioni delle API Domo:

- v1

## Policy contenenti le operazioni API per creare e utilizzare le connessioni

La seguente policy di esempio descrive le autorizzazioni AWS IAM richieste per la creazione e l'utilizzo delle connessioni. Se si sta creando un nuovo ruolo, creare una policy che contenga quanto segue:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:ListConnectionTypes",
        "glue:DescribeConnectionType",
        "glue:RefreshOAuth2Tokens",
        "glue:ListEntities",
        "glue:DescribeEntity"
      ],
      "Resource": "*"
    }
  ]
}
```

In alternativa, se non desideri utilizzare il metodo precedente, utilizza le seguenti policy IAM gestite:

- AWSGlueServiceRole— Garantisce l'accesso alle risorse che AWS Glue i vari processi richiedono per l'esecuzione per conto dell'utente. Queste risorse includono AWS Glue Amazon S3, IAM, CloudWatch Logs e Amazon. EC2 Se segui la convenzione di denominazione delle risorse specificata in questa politica, AWS Glue i processi dispongono delle autorizzazioni richieste. Questa policy è in genere collegata ai ruoli specificati quando si definiscono crawler, processi ed endpoint di sviluppo.
- AWSGlueConsoleFullAccess— Concede l'accesso completo alle AWS Glue risorse quando un'identità a cui è allegata la policy utilizza la AWS console di gestione. Se si segue la convenzione per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questa policy è in genere associata agli utenti della AWS Glue console.

## Configurazione di Domo

Prima di poterli utilizzare AWS Glue per trasferire dati da Domo alle destinazioni supportate, devi soddisfare questi requisiti:

Requisiti minimi

I requisiti minimi sono i seguenti:

- Hai un account Domo abilitato per l'accesso alle API.
- Nel tuo account sviluppatore Domo hai un'app che fornisce le credenziali del client che AWS Glue utilizza per accedere ai tuoi dati in modo sicuro quando effettua chiamate autenticate al tuo account. Per ulteriori informazioni, consulta Creazione di un'app per sviluppatori Domo.

Se soddisfi questi requisiti, sei pronto per AWS Glue connetterti al tuo account Domo.

Creazione di un'app per sviluppatori Domo

Per ottenere l'ID client e il segreto client devi creare un account sviluppatore.

1. Vai alla pagina di accesso per gli sviluppatori di Domo.
2. Selezionare Login (Accesso).
3. Fornisci il nome dominio e fai clic su Continua.
4. Passa con il mouse su Il mio account e seleziona Nuovo client.
5. Fornisci il nome e la descrizione, seleziona l'ambito ("dati") e scegli Crea.
6. Recupera l'ID client e il Segreto client generati dal nuovo client creato.

## Configurazione delle connessioni Domo

Domo supporta il tipo di concessione CLIENT_CREDENTIALS per. OAuth2

- Questo tipo di concessione è considerato a due vie in OAuth quanto solo l'applicazione client si autentica sul server, senza alcun coinvolgimento dell'utente.
- Gli utenti possono scegliere di creare la propria app connessa in Domo e fornire il proprio ID client e il segreto del client durante la creazione di connessioni tramite la console. AWS Glue
- Per la documentazione pubblica di Domo sulla creazione di un'app connessa per il OAuth flusso del codice di autorizzazione, vedi OAuth Autenticazione.

Per configurare una connessione Domo:

1. In AWS Secrets Manager, crea un segreto con i seguenti dettagli:

a. Per l'app connessa gestita dal cliente, il segreto deve contenere il token di accesso all'app connessa, `client_secret`, e `client_id`.

b. Nota: devi creare un segreto per le tue connessioni in AWS Glue.

1. In AWS Glue Glue Studio, crea una connessione in Connessioni dati seguendo i passaggi seguenti:

   a. In Connessioni, scegli Crea connessione.

   b. Quando si seleziona una Origine dati, selezionare Domo.

   c. Seleziona il ruolo AWS IAM che AWS Glue può assumere e dispone delle autorizzazioni per le seguenti azioni:

      JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:DescribeSecret",
        "secretsmanager:GetSecretValue",
        "secretsmanager:PutSecretValue",
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DeleteNetworkInterface"
      ],
      "Resource": "*"
    }
  ]
}
```

   d. Seleziona quello `secretName` che desideri utilizzare per questa connessione per AWS Glue inserire i token.

   e. Selezionare le opzioni di rete se si desidera utilizzare la propria rete.

2. Concedi il permesso di lettura `secretName` al ruolo IAM associato al tuo AWS Glue lavoro.

## Lettura dalle entità di Domo

Prerequisito

Un oggetto di Domo da cui si desidera leggere. Sarà necessario il nome dell'oggetto, ad esempio Set di dati o Policy relative alle autorizzazioni sui dati. Nella tabella seguente sono illustrate le entità supportate.

Entità supportate come origine:

| Entità | Può essere filtrato | Supporta limite | Supporta Ordina per | Supporta Seleziona * | Supporta il partizionamento |
|---|---|---|---|---|---|
| Set di dati | Sì | Sì | Sì | Sì | Sì |
| Policy relative alle autorizzazioni sui dati | No | No | No | Sì | No |

Esempio:

```
Domo_read = glueContext.create_dynamic_frame.from_options(
    connection_type="domo",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "dataset",
        "API_VERSION": "v1"
    }
```

Dettagli dell'entità e del campo di Domo

Entità con metadati statici:

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| Policy relative alle autorizzazioni sui dati | id | Long | N/D |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | tipo | Stringa | N/A |
| | nome | Stringa | N/A |
| | filtri | List | N/D |
| | utenti | List | N/D |
| | virtualUsers | List | N/D |
| | gruppi | List | N/D |

Per la seguente entità, Domo fornisce endpoint per recuperare i metadati in modo dinamico, cosicché il supporto dell'operatore venga acquisito a livello di tipo di dati per l'entità.

| Entità | Tipo di dati | Operatori supportati |
|---|---|---|
| Set di dati | Numero intero | =, !=, <, >, >=, <= |
| | Long | =, !=, <, >, >=, <= |
| | Stringa | =, !=, CONTAINS |
| | Data | =, >, >=, <, <=, BETWEEN |
| | DateTime | =, <, <=, >, >=, BETWEEN |
| | Booleano | =, != |
| | Double | =, !=, <, >, >=, <= |
| | List | N/D |
| | Struct | N/D |

Query di partizionamento

Partizionamento basato sul campo

Puoi fornire le opzioni Spark aggiuntive PARTITION_FIELD, LOWER_BOUND, UPPER_BOUND e NUM_PARTITIONS se desideri utilizzare la concorrenza in Spark. Con questi parametri, la query originale viene suddivisa in NUM_PARTITIONS query secondarie che possono essere eseguite contemporaneamente dalle attività Spark.

- PARTITION_FIELD: il nome del campo da utilizzare per partizionare la query.
- LOWER_BOUND: un valore limite inferiore inclusivo del campo di partizione scelto.

  Per il DateTime campo, accettiamo il valore in formato ISO.

  Esempio di valore valido:

  ```
  "2023-01-15T11:18:39.205Z"
  ```

  Per il campo Data, accettiamo il valore in formato ISO.

  Esempio di valore valido:

  ```
  "2023-01-15"
  ```

- UPPER_BOUND: un valore limite superiore esclusivo del campo di partizione scelto.

  Esempio di valore valido:

  ```
  "2023-02-15T11:18:39.205Z"
  ```

- NUM_PARTITIONS: il numero di partizioni.

I dettagli di supporto del campo di partizionamento per entità sono riportati nella tabella seguente:

| Nome dell'entità | Campi di partizionamento | Tipo di dati |
|---|---|---|
| Set di dati | Qualsiasi campo Date/Time basato [metadati dinamici] | DateTime |
| | Qualsiasi campo basato su Data [metadati dinamici] | Data |

Esempio:

```
Domo_read = glueContext.create_dynamic_frame.from_options(
    connection_type="domo",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "dataset",
        "API_VERSION": "v1",
        "PARTITION_FIELD": "permissionTime"
        "LOWER_BOUND": "2023-01-15T11:18:39.205Z"
        "UPPER_BOUND": "2023-02-15T11:18:39.205Z"
        "NUM_PARTITIONS": "2"
    }
```

Partizionamento basato su record

È possibile fornire l'opzione Spark aggiuntiva `NUM_PARTITIONS` se si desidera utilizzare la simultaneità in Spark. Con questo parametro, la query originale viene suddivisa in `NUM_PARTITIONS` query secondarie che possono essere eseguite contemporaneamente dalle attività Spark.

Nel partizionamento basato su record, Domo esegue query sul numero totale di record presenti e lo divide per il numero `NUM_PARTITIONS` fornito. Il numero di record risultante viene quindi recuperato contemporaneamente da ciascuna sotto-query.

Esempio:

```
Domo_read = glueContext.create_dynamic_frame.from_options(
    connection_type="domo",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "dataset",
        "API_VERSION": "v1",
        "NUM_PARTITIONS": "2"
    }
```

## Opzioni di connessione di Domo

Di seguito sono riportate le opzioni di connessione per Domo:

- `ENTITY_NAME`(String): (obbligatorio) utilizzato per la lettura. Il nome del tuo oggetto in Domo.
- `API_VERSION`(String): (obbligatorio) utilizzato per la lettura. Versione dell'API Rest di Domo da utilizzare.

- `SELECTED_FIELDS`(List<String>): predefinito: vuoto (SELECT *). Utilizzato per la lettura. Colonne da selezionare per l'oggetto.
- `FILTER_PREDICATE`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Dev'essere in formato Spark SQL.
- `QUERY`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Query Spark SQL completa.
- `PARTITION_FIELD`(Stringa): utilizzato per la lettura. Campo da utilizzare per partizionare la query.
- `LOWER_BOUND`(Stringa): utilizzato per la lettura. Un valore limite inferiore inclusivo del campo di partizione scelto.
- `UPPER_BOUND`(Stringa): utilizzato per la lettura. Un valore limite superiore esclusivo del campo di partizione scelto.
- `NUM_PARTITIONS`(Numero intero): predefinito: 1. Utilizzato per la lettura. Numero di partizioni da leggere.

### Limitazioni di Domo

Di seguito sono riportate le limitazioni o le note relative a Domo:

- A causa di una limitazione dell'SDK, il filtro non funziona come previsto per i campi interrogabili che iniziano con il simbolo '_' (ad esempio: _BATCH_ID).
- A causa di una limitazione dell'API, il filtro funziona sulla data precedente alla data fornita. Ciò influisce anche sul pull incrementale. Per superare questa limitazione e ottenere i dati relativi alla data richiesta, seleziona una data in base al tuo fuso orario rispetto all'UTC.

## Connessione a Dynatrace

Dynatrace è una piattaforma che offre analisi e automazione per un'osservabilità e una sicurezza complete. È specializzata nel monitoraggio e nell'ottimizzazione delle prestazioni delle applicazioni, dell'infrastruttura e dell'esperienza utente.

Argomenti


- AWS Glue supporto per Dynatrace
- Policy contenenti le operazioni API per creare e utilizzare le connessioni
- Configurazione di Dynatrace
- Configurazione delle connessioni a Dynatrace

- Lettura da entità Dynatrace
- Opzioni di connessione per Dynatrace
- Limitazioni di Dynatrace

## AWS Glue supporto per Dynatrace

AWS Glue supporta Dynatrace come segue:

Supportato come origine?

Sì. È possibile utilizzare i job AWS Glue ETL per interrogare i dati provenienti da Dynatrace.

Supportato come destinazione?

No.

Versioni API Dynatrace supportate

Sono supportate le seguenti versioni API di Dynatrace:

- v2

## Policy contenenti le operazioni API per creare e utilizzare le connessioni

La seguente policy di esempio descrive le autorizzazioni AWS IAM richieste per la creazione e l'utilizzo delle connessioni. Se si sta creando un nuovo ruolo, creare una policy che contenga quanto segue:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:ListConnectionTypes",
        "glue:DescribeConnectionType",
        "glue:RefreshOAuth2Tokens",
        "glue:ListEntities",
```

```
        "glue:DescribeEntity"
      ],
      "Resource": "*"
    }
  ]
}
```

In alternativa, se non desideri utilizzare il metodo precedente, utilizza le seguenti policy IAM gestite:

- AWSGlueServiceRole— Garantisce l'accesso alle risorse che AWS Glue i vari processi richiedono per l'esecuzione per conto dell'utente. Queste risorse includono AWS Glue Amazon S3, IAM, CloudWatch Logs e Amazon. EC2 Se segui la convenzione di denominazione delle risorse specificata in questa politica, AWS Glue i processi dispongono delle autorizzazioni richieste. Questa policy è in genere collegata ai ruoli specificati quando si definiscono crawler, processi ed endpoint di sviluppo.
- AWSGlueConsoleFullAccess— Concede l'accesso completo alle AWS Glue risorse quando un'identità a cui è allegata la policy utilizza la AWS console di gestione. Se si segue la convenzione per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questa policy è in genere associata agli utenti della AWS Glue console.

## Configurazione di Dynatrace

Prima di poter utilizzare il trasferimento AWS Glue di dati da Dynatrace, è necessario soddisfare i seguenti requisiti:

Requisiti minimi

I requisiti minimi sono i seguenti:

- Avere un account Dynatrace.
- Hai generato un token di accesso nel tuo account Dynatrace con l' read/write ambito appropriato assegnato per accedere a. APIs Per ulteriori informazioni, vedi Generare un token.

Se soddisfi questi requisiti, sei pronto per connetterti AWS Glue al tuo account Dynatrace.

## Configurazione delle connessioni a Dynatrace

Dynatrace supporta l'autenticazione personalizzata.

Per configurare una connessione Dynatrace:

1. In AWS Secrets Manager, crea un segreto con i seguenti dettagli:
   a. Per le app connesse gestite dal cliente, il segreto deve contenere la chiave API dell'app connessa con *apiToken* come chiave.
   b. Nota: devi creare un segreto per le tue connessioni in AWS Glue.

1. In AWS Glue Glue Studio, crea una connessione in Connessioni dati seguendo i passaggi seguenti:
   a. Quando selezioni una Origine dati, scegli Dynatrace.
   b. Fornisci `INSTANCE_URL` dell'account Dynatrace a cui intendi connetterti.
   c. Seleziona il ruolo AWS IAM che AWS Glue può assumere e dispone delle autorizzazioni per le seguenti azioni:

      JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:DescribeSecret",
        "secretsmanager:GetSecretValue",
        "secretsmanager:PutSecretValue",
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DeleteNetworkInterface"
      ],
      "Resource": "*"
    }
  ]
}
```

   d. Seleziona quello `secretName` che desideri utilizzare per questa connessione per AWS Glue inserire i token.
   e. Selezionare le opzioni di rete se si desidera utilizzare la propria rete.
2. Concedi il permesso di lettura `secretName` al ruolo IAM associato al tuo AWS Glue lavoro.

## Lettura da entità Dynatrace

Prerequisito

Un oggetto di Dynatrace da cui si desidera leggere. È necessario il nome dell'oggetto, ad esempio "problema".

Entità supportate come origine:

| Entità | Può essere filtrato | Supporta limite | Supporta Ordina per | Supporta Seleziona * | Supporta il partizion amento |
|---|---|---|---|---|---|
| Problema | Sì | Sì | Sì | Sì | No |

Esempio:

```
Dynatrace_read = glueContext.create_dynamic_frame.from_options(
    connection_type="Dynatrace",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "problem",
        "API_VERSION": "v2",
        "INSTANCE_URL": "https://[instanceName].live.dynatrace.com"
    }
```

Dettagli dell'entità e del campo di Dynatrace:

Dynatrace fornisce endpoint per recuperare i metadati in modo dinamico per le entità supportate. Di conseguenza, il supporto dell'operatore viene acquisito a livello di tipo di dati.

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| Problema | affectedEntities | List | EQUAL_TO |
| | displayId | Stringa | EQUAL_TO |
| | endTime | DateTime | |
| | entityTags | List | |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | evidenceDetails | Struct | |
| | impactAnalysis | Struct | |
| | impactLevel | Stringa | EQUAL_TO |
| | impactedEntities | List | EQUAL_TO |
| | linkedProblemInfo | Struct | |
| | managementZones | List | EQUAL_TO |
| | problemFilters | List | |
| | recentComments | Struct | |
| | rootCauseEntity | Struct | EQUAL_TO |
| | problemId | Stringa | EQUAL_TO |
| | severityLevel | Stringa | EQUAL_TO |
| | startTime | DateTime | BETWEEN |
| | status | Stringa | EQUAL_TO |
| | titolo | Stringa | |
| | from | DateTime | EQUAL_TO, BETWEEN |
| | problemFilterIds | Stringa | EQUAL_TO |
| | problemFilterNames | Stringa | EQUAL_TO |
| | managementZoneIds | Stringa | EQUAL_TO |
| | testo | Stringa | EQUAL_TO |
| | underMaintenance | Booleano | EQUAL_TO |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | message | Stringa | |

Query di partizionamento

Dynatrace non supporta il partizionamento basato su campi o record.

## Opzioni di connessione per Dynatrace

Di seguito sono riportate le opzioni di connessione per Dynatrace:

- `ENTITY_NAME`(String): (obbligatorio) utilizzato per la lettura. Il nome del tuo oggetto in Dynatrace.
- `API_VERSION`(String): (obbligatorio) utilizzato per la lettura. Versione dell'API Rest di Dynatrace da utilizzare.
- `INSTANCE_URL`(Stringa): utilizzato per la lettura. Un URL valido dell'istanza Dynatrace.
- `SELECTED_FIELDS`(List<String>): predefinito: vuoto (SELECT *). Utilizzato per la lettura. Colonne da selezionare per l'oggetto.
- `FILTER_PREDICATE`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Dev'essere in formato Spark SQL.
- `QUERY`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Query Spark SQL completa.

## Limitazioni di Dynatrace

Di seguito sono riportate le limitazioni o le note relative a Dynatrace:

- Dynatrace non supporta il partizionamento basato su campi o record.
- Per la funzionalità Seleziona tutto, se si specifica il "campo" nel filtro, non sarà possibile visualizzare più di 10 record per pagina.
- La dimensione massima della pagina supportata è 500. Se selezioni uno dei campi [`evidenceDetails, impactAnalysis, recentComments`] durante la creazione del flusso, il numero di record per pagina sarà impostato di default su 10.

# Connessione ad Annunci su Facebook

Annunci su Facebook è una potente piattaforma pubblicitaria digitale usata da aziende di tutte le dimensioni per raggiungere il proprio pubblico di riferimento e realizzare diversi obiettivi di marketing. Questa piattaforma permette agli inserzionisti di creare annunci personalizzati che possono essere visualizzati su tutta la famiglia di app e servizi di Facebook, tra cui Facebook e Messenger. Grazie alle sue funzionalità di targeting avanzate, Annunci su Facebook consente alle aziende di accedere a dati demografici e relativi a interessi, comportamenti e località specifici.

Argomenti


- AWS Glue supporto per Facebook Ads
- Policy contenenti le operazioni API per creare e utilizzare le connessioni
- Configurazione di Facebook Ads
- Configurazione delle connessioni su Facebook Ads
- Lettura dalle entità di Facebook Ads
- Opzioni di connessione a Facebook Ads
- Limitazioni e note per il connettore di Facebook Ads


## AWS Glue supporto per Facebook Ads

AWS Glue supporta Facebook Ads come segue:

Supportato come origine?

Sì. Puoi utilizzare i lavori AWS Glue ETL per interrogare i dati di Facebook Ads.

Supportato come destinazione?

No.

Versioni supportate dell'API Facebook Ads

Le versioni dell'API di Facebook Ads supportate sono le seguenti:

- v17.0
- v18.0

- v19.0
- v20.0

## Policy contenenti le operazioni API per creare e utilizzare le connessioni

La seguente policy di esempio descrive le autorizzazioni AWS IAM richieste per la creazione e l'utilizzo delle connessioni. Se si sta creando un nuovo ruolo, creare una policy che contenga quanto segue:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:ListConnectionTypes",
        "glue:DescribeConnectionType",
        "glue:RefreshOAuth2Tokens",
        "glue:ListEntities",
        "glue:DescribeEntity"
      ],
      "Resource": "*"
    }
  ]
}
```

In alternativa, se non desideri utilizzare il metodo precedente, utilizza le seguenti policy IAM gestite:

- AWSGlueServiceRole— Garantisce l'accesso alle risorse che AWS Glue i vari processi richiedono per l'esecuzione per conto dell'utente. Queste risorse includono AWS Glue Amazon S3, IAM, CloudWatch Logs e Amazon. EC2 Se segui la convenzione di denominazione delle risorse specificata in questa politica, AWS Glue i processi dispongono delle autorizzazioni richieste. Questa policy è in genere collegata ai ruoli specificati quando si definiscono crawler, processi ed endpoint di sviluppo.
- AWSGlueConsoleFullAccess— Concede l'accesso completo alle AWS Glue risorse quando un'identità a cui è allegata la policy utilizza la AWS console di gestione. Se si segue la convenzione

per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questa policy è in genere associata agli utenti della AWS Glue console.

## Configurazione di Facebook Ads

Prima di poter AWS Glue utilizzare il trasferimento di dati da Facebook Ads, devi soddisfare questi requisiti:

Requisiti minimi

I requisiti minimi sono i seguenti:

- Gli account Facebook Standard sono accessibili direttamente tramite Facebook.
- L'autenticazione dell'utente è necessaria per generare il token di accesso.
- Il connettore Facebook Ads SDK implementerà il OAuth flusso User Access Token.
- Stiamo utilizzando OAuth2 .0 per autenticare le nostre richieste API su Facebook Ads. Questa autenticazione basata sul Web rientra nell'architettura dell'autenticazione a più fattori (MFA), che è un superset di 2FA.
- L'utente deve concedere le autorizzazioni per accedere agli endpoint. Per accedere ai dati dell'utente, l'autorizzazione degli endpoint viene gestita tramite autorizzazioni e funzionalità.

Ottenere le credenziali 2.0 OAuth

Per ottenere le credenziali API così da effettuare chiamate autenticate alla tua istanza, consulta REST API nella Guida per gli sviluppatori di Facebook Ads.

## Configurazione delle connessioni su Facebook Ads

Facebook Ads supporta il tipo di concessione AUTHORIZATION_CODE per. OAuth2

- Questo tipo di concessione è considerato a tre vie in OAuth quanto si basa sul reindirizzamento degli utenti al server di autorizzazione di terze parti per autenticare l'utente. Viene utilizzato durante la creazione di connessioni tramite la console. AWS Glue
- Gli utenti possono comunque scegliere di creare la propria app connessa in Facebook Ads e fornire il proprio ID cliente e il segreto del client quando creano connessioni tramite la AWS Glue console. In questo scenario, verranno comunque reindirizzati a Facebook Ads per accedere e autorizzare l'accesso AWS Glue alle proprie risorse.

- Questo tipo di concessione genera un token di accesso. Un token utente di sistema in scadenza è valido per 60 giorni dalla data di generazione o aggiornamento. Per creare continuità, lo sviluppatore deve aggiornare il token di accesso entro 60 giorni. In caso contrario, il token di accesso verrà perso e lo sviluppatore dovrà ottenerne un altro per poter accedere di nuovo all'API. Consulta Refresh Access Token.
- Per la documentazione pubblica di Facebook Ads sulla creazione di un'app connessa per il OAuth flusso del codice di autorizzazione, consulta Using OAuth 2.0 to Access Google APIs nella guida Google per sviluppatori.

Per configurare una connessione a Facebook Ads:

1. In AWS Glue Glue Studio, crea una connessione in Connessioni dati seguendo i passaggi seguenti:
   a. Quando selezioni un tipo di connessione, seleziona Facebook Ads.
   b. Fornire `INSTANCE_URL` dell'istanza Facebook Ads a cui connettersi.
   c. Seleziona il ruolo AWS IAM che AWS Glue può assumere e dispone delle autorizzazioni per le seguenti azioni:

      JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:DescribeSecret",
        "secretsmanager:GetSecretValue",
        "secretsmanager:PutSecretValue",
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DeleteNetworkInterface"
      ],
      "Resource": "*"
    }
  ]
}
```

d. Seleziona quello `secretName` che desideri utilizzare per questa connessione per AWS Glue inserire i token.

e. Selezionare le opzioni di rete se si desidera utilizzare la propria rete.

2. Concedi il permesso di lettura `secretName` al ruolo IAM associato al tuo AWS Glue lavoro.

## Lettura dalle entità di Facebook Ads

Prerequisito

Un oggetto di Facebook Ads da cui desideri leggere. È necessario il nome dell'oggetto. Le tabelle seguenti mostrano le entità supportate.

Entità supportate come origine:

| Entità | Può essere filtrato | Supporta limite | Supporta Ordina per | Supporta Seleziona * | Supporta il partizionamento |
|---|---|---|---|---|---|
| Campagna | Sì | Sì | No | Sì | Sì |
| Set di annunci | Sì | Sì | No | Sì | Sì |
| Inserzioni | Sì | Sì | No | Sì | Sì |
| Creative Ad | No | Sì | No | Sì | No |
| Informazioni approfondite: conto | No | Sì | No | Sì | No |
| Account pubblicitari | Sì | Sì | No | Sì | No |
| Informazioni approfondite: inserzione | Sì | Sì | No | Sì | Sì |

| Entità | Può essere filtrato | Supporta limite | Supporta Ordina per | Supporta Seleziona * | Supporta il partizionamento |
|---|---|---|---|---|---|
| Approfondimenti - AdSet | Sì | Sì | No | Sì | Sì |
| Informazioni approfondite: campagna | Sì | Sì | No | Sì | Sì |

Esempio:

```
FacebookAds_read = glueContext.create_dynamic_frame.from_options(
    connection_type="FacebookAds",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "entityName",
        "API_VERSION": "v20.0"
    }
```

Dettagli sull'entità e sul campo di Facebook Ads

Per ulteriori informazioni sui dettagli dell'entità e del campo, consultare:

- Account pubblicitario
- Campagna
- Set di annunci
- Inserzione
- Creative Ad
- Informazioni approfondite sull'account pubblicitario
- Informazioni approfondite sulle inserzioni
- Approfondimenti AdSets
- Informazioni approfondite sulle campagne

Per ulteriori informazioni, consulta l'API Marketing.

> **Note**
>
> I tipi di dati Struct e List vengono convertiti in tipo di dati String nella risposta dei connettori.

Query di partizionamento

Puoi fornire le opzioni Spark aggiuntive `PARTITION_FIELD`, `LOWER_BOUND`, `UPPER_BOUND` e `NUM_PARTITIONS` se desideri utilizzare la concorrenza in Spark. Con questi parametri, la query originale viene suddivisa in `NUM_PARTITIONS` query secondarie che possono essere eseguite contemporaneamente dalle attività Spark.

- `PARTITION_FIELD`: il nome del campo da utilizzare per partizionare la query.
- `LOWER_BOUND`: un valore limite inferiore inclusivo del campo di partizione scelto.

  Per il DateTime campo, accettiamo il formato di timestamp Spark utilizzato nelle query SQL di Spark.

  Esempio di valore valido:

  ```
  "2022-01-01"
  ```

- `UPPER_BOUND`: un valore limite superiore esclusivo del campo di partizione scelto.
- `NUM_PARTITIONS`: il numero di partizioni.

Esempio:

```
FacebookADs_read = glueContext.create_dynamic_frame.from_options(
    connection_type="FacebookAds",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "entityName",
        "API_VERSION": "v20.0",
        "PARTITION_FIELD": "created_time"
        "LOWER_BOUND": "2022-01-01"
        "UPPER_BOUND": "2024-01-02"
        "NUM_PARTITIONS": "10"
    }
```

## Opzioni di connessione a Facebook Ads

Le seguenti sono le opzioni di connessione per Facebook Ads:

- `ENTITY_NAME`(Stringa): (obbligatorio) utilizzato per la lettura. Il nome del tuo oggetto in Facebook Ads.
- `API_VERSION`(Stringa): (obbligatorio) utilizzato per la lettura. La versione dell'API Rest di Facebook Ads che desideri utilizzare. Ad esempio: v1.
- `SELECTED_FIELDS`(List<String>): predefinito: vuoto (SELECT *). Utilizzato per la lettura. Colonne da selezionare per l'oggetto.
- `FILTER_PREDICATE`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Dev'essere in formato Spark SQL.
- `QUERY`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Query Spark SQL completa.
- `PARTITION_FIELD`(Stringa): utilizzato per la lettura. Campo da utilizzare per partizionare la query.
- `LOWER_BOUND`(Stringa): utilizzato per la lettura. Un valore limite inferiore inclusivo del campo di partizione scelto.
- `UPPER_BOUND`(Stringa): utilizzato per la lettura. Un valore limite superiore esclusivo del campo di partizione scelto.
- `NUM_PARTITIONS`(Numero intero): predefinito: 1. Utilizzato per la lettura. Numero di partizioni da leggere.
- `TRANSFER_MODE`(Stringa): predefinito: SYNC. Utilizzato per la lettura asincrona.

## Limitazioni e note per il connettore di Facebook Ads

Di seguito sono riportate le limitazioni o le note per il connettore di Facebook Ads:

- Poiché Facebook Ads supporta i metadati dinamici, è possibile interrogare tutti i campi. Tutti i campi supportano il filtraggio e i record vengono recuperati se i dati sono disponibili, altrimenti Facebook restituisce un errore di richiesta non valida (400) con un relativo messaggio di errore.
- Il conteggio delle chiamate di un'app è il numero di chiamate che un utente può effettuare durante una finestra continua di un'ora moltiplicando 200 per il numero di utenti. Per i dettagli sui limiti di velocità, consulta Limiti della frequenza e Limiti della frequenza dei casi d'uso business.

# Connessione a Facebook Page Insights

Le pagine Facebook permettono alle aziende e ad altri gruppi di interesse di creare pagine per il social network Facebook.com. Le aziende usano queste pagine per condividere gli orari di apertura, fare annunci e interagire con i clienti online. Se sei un utente di Facebook Page Insights, puoi connetterti AWS Glue al tuo account Facebook Page Insights. Puoi usare Facebook Page Insights come origine dati per i tuoi processi ETL. Esegui questi processi per trasferire dati da Facebook Page Insights ai AWS servizi o ad altre applicazioni supportate.

Argomenti


- AWS Glue supporto per Facebook Page Insights
- Policy contenenti le operazioni API per creare e utilizzare le connessioni
- Configurazione degli Approfondimenti sulla pagina Facebook
- Configurazione delle connessioni degli Approfondimenti sulla pagina Facebook
- Lettura da entità degli Approfondimenti sulla pagina Facebook
- Opzioni di connessione degli Approfondimenti sulla pagina Facebook
- Limitazioni e note per il connettore degli Approfondimenti sulla pagina Facebook


## AWS Glue supporto per Facebook Page Insights

AWS Glue supporta Facebook Page Insights come segue:

Supportato come origine?

Sì. Puoi utilizzare i job AWS Glue ETL per interrogare i dati da Facebook Page Insights.

Supportato come destinazione?

No.

Versioni supportate dell'API degli Approfondimenti sulla pagina Facebook

Le versioni dell'API degli Approfondimenti sulla pagina Facebook supportate sono le seguenti:

- v17
- v18
- v19

- v20
- v21

## Policy contenenti le operazioni API per creare e utilizzare le connessioni

La seguente policy di esempio descrive le autorizzazioni AWS IAM richieste per la creazione e l'utilizzo delle connessioni. Se si sta creando un nuovo ruolo, creare una policy che contenga quanto segue:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:ListConnectionTypes",
        "glue:DescribeConnectionType",
        "glue:RefreshOAuth2Tokens",
        "glue:ListEntities",
        "glue:DescribeEntity"
      ],
      "Resource": "*"
    }
  ]
}
```

In alternativa, se non desideri utilizzare il metodo precedente, utilizza le seguenti policy IAM gestite:

- AWSGlueServiceRole— Garantisce l'accesso alle risorse che AWS Glue i vari processi richiedono per l'esecuzione per conto dell'utente. Queste risorse includono AWS Glue Amazon S3, IAM, CloudWatch Logs e Amazon. EC2 Se segui la convenzione di denominazione delle risorse specificata in questa politica, AWS Glue i processi dispongono delle autorizzazioni richieste. Questa policy è in genere collegata ai ruoli specificati quando si definiscono crawler, processi ed endpoint di sviluppo.
- AWSGlueConsoleFullAccess— Concede l'accesso completo alle AWS Glue risorse quando un'identità a cui è allegata la policy utilizza la AWS console di gestione. Se si segue la convenzione

per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questa policy è in genere associata agli utenti della AWS Glue console.

## Configurazione degli Approfondimenti sulla pagina Facebook

Prima di poter AWS Glue utilizzare il trasferimento di dati da Facebook Page Insights, devi soddisfare questi requisiti:

Requisiti minimi

I requisiti minimi sono i seguenti:

- Gli account Facebook Standard sono accessibili direttamente tramite Facebook.
- L'autenticazione dell'utente è necessaria per generare il token di accesso.
- Il connettore Facebook Page Insights implementa il OAuth flusso User Access Token.
- Il connettore utilizza OAuth2 .0 per autenticare le nostre richieste API su Facebook Page Insights. Questo rientra nell'architettura dell'autenticazione a più fattori (MFA), che è un superset di 2FA. È un'autenticazione basata sul web.
- L'utente deve concedere le autorizzazioni per accedere agli endpoint. Per accedere ai dati dell'utente, l'autorizzazione degli endpoint viene gestita tramite autorizzazioni e funzionalità.

## Configurazione delle connessioni degli Approfondimenti sulla pagina Facebook

Per configurare una connessione agli Approfondimenti sulla pagina Facebook:

1. In AWS Glue Glue Studio, crea una connessione in Connessioni dati seguendo i passaggi seguenti:
   a. Quando selezioni un tipo di connessione, seleziona gli Approfondimenti sulla pagina Facebook.
   b. Seleziona il ruolo AWS IAM che AWS Glue può assumere e dispone delle autorizzazioni per le seguenti azioni:

      JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
```

```
      "Action": [
        "secretsmanager:DescribeSecret",
        "secretsmanager:GetSecretValue",
        "secretsmanager:PutSecretValue",
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DeleteNetworkInterface"
      ],
      "Resource": "*"
    }
  ]
}
```

c. Seleziona l'URL del codice di autorizzazione.

d. Seleziona quello `secretName` che desideri utilizzare per questa connessione per AWS Glue inserire i token.

e. Selezionare le opzioni di rete se si desidera utilizzare la propria rete.

2. Concedi il permesso di lettura `secretName` al ruolo IAM associato al tuo AWS Glue lavoro.

## Lettura da entità degli Approfondimenti sulla pagina Facebook

Prerequisito

Un oggetto degli Approfondimenti sulla pagina Facebook da cui desideri leggere. È necessario il nome dell'oggetto.

Entità supportate come origine:

| Entità | Può essere filtrato | Supporta limite | Supporta Ordina per | Supporta Seleziona * | Supporta il partizionamento |
|---|---|---|---|---|---|
| Contenuto della pagina | Sì | No | Sì | Sì | Sì |
| Clic sul CTA della pagina | Sì | No | No | Sì | Sì |

| Entità | Può essere filtrato | Supporta limite | Supporta Ordina per | Supporta Seleziona * | Supporta il partizionamento |
|---|---|---|---|---|---|
| Coinvolgimento di pagina | Sì | No | No | Sì | Sì |
| Impression della pagina | Sì | No | No | Sì | Sì |
| Post della pagina | Sì | No | No | Sì | Sì |
| Coinvolgimento dei post della pagina | No | No | No | Sì | No |
| Reazioni ai post della pagina | No | No | No | Sì | No |
| Reazioni della pagina | Sì | No | No | Sì | Sì |
| Storie | Sì | No | No | Sì | Sì |
| Dati demografici degli utenti della pagina | Sì | No | No | Sì | Sì |
| Visualizzazioni dei video della pagina | Sì | No | No | Sì | Sì |

| Entità | Può essere filtrato | Supporta limite | Supporta Ordina per | Supporta Seleziona * | Supporta il partizionamento |
|---|---|---|---|---|---|
| Visualizzazioni della pagina | Sì | No | No | Sì | Sì |
| Post dei video della pagina | Sì | No | No | Sì | Sì |
| Pagine | No | Sì | No | Sì | No |
| Feed | Sì | Sì | No | Sì | Sì |

Esempio:

```
facebookPageInsights_read = glueContext.create_dynamic_frame. from options(
    connection_type="facebookpageinsights",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "entityName",
        "API_VERSION": "v21"
   }
```

Dettagli del campo degli Approfondimenti sulla pagina Facebook:

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| Contenuto della pagina | Name | Stringa | N/A |
| | Periodo | Periodo | EQUAL_TO |
| | Since | DateTime | EQUAL_TO |
| | Valori | List | N/D |
| | Titolo | Stringa | N/A |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | Description | Stringa | N/A |
| | descrizione_da_api_doc | Stringa | N/A |
| | Id | Stringa | N/A |
| Clic sul CTA della pagina | Name | Stringa | N/A |
| | Periodo | Periodo | EQUAL_TO |
| | Since | DateTime | EQUAL_TO |
| | Valori | List | N/D |
| | Titolo | Stringa | N/A |
| | Description | Stringa | N/A |
| | descrizione_da_api_doc | Stringa | N/A |
| | Id | Stringa | N/A |
| Coinvolgimento di pagina | Name | Stringa | N/A |
| | Periodo | Periodo | EQUAL_TO |
| | Since | DateTime | EQUAL_TO |
| | Valori | List | N/D |
| | Titolo | Stringa | N/A |
| | Description | Stringa | N/A |
| | descrizione_da_api_doc | Stringa | N/A |
| | Id | Stringa | N/A |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| Impression della pagina | Name | Stringa | N/A |
| | Periodo | Periodo | EQUAL_TO |
| | Since | DateTime | EQUAL_TO |
| | Valori | List | N/D |
| | Titolo | Stringa | N/A |
| | Description | Stringa | N/A |
| | descrizione_da_api_doc | Stringa | N/A |
| | Id | Stringa | N/A |
| Post della pagina | Name | Stringa | N/A |
| | Periodo | Periodo | EQUAL_TO |
| | Since | DateTime | EQUAL_TO |
| | Valori | List | N/D |
| | Titolo | Stringa | N/A |
| | Description | Stringa | N/A |
| | descrizione_da_api_doc | Stringa | N/A |
| | Id | Stringa | N/A |
| Coinvolgimento dei post della pagina | Name | Stringa | N/A |
| | Periodo | Periodo | EQUAL_TO |
| | Valori | List | N/D |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | Titolo | Stringa | N/A |
| | Description | Stringa | N/A |
| | descrizione_da_api_doc | Stringa | N/A |
| | Id | Stringa | N/A |
| Reazioni ai post della pagina | Name | Stringa | N/A |
| | Periodo | Periodo | EQUAL_TO |
| | Valori | List | N/D |
| | Titolo | Stringa | N/A |
| | Description | Stringa | N/A |
| | descrizione_da_api_doc | Stringa | N/A |
| | Id | Stringa | N/A |
| Dati demografici degli utenti della pagina | Name | Stringa | N/A |
| | Periodo | Periodo | EQUAL_TO |
| | Since | DateTime | EQUAL_TO |
| | Valori | List | N/D |
| | Titolo | Stringa | N/A |
| | Description | Stringa | N/A |
| | descrizione_da_api_doc | Stringa | N/A |
| | Id | Stringa | N/A |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| Visualizzazioni dei video della pagina | Name | Stringa | N/A |
| | Periodo | Periodo | EQUAL_TO |
| | Since | DateTime | EQUAL_TO |
| | Valori | List | N/D |
| | Titolo | Stringa | N/A |
| | Description | Stringa | N/A |
| | descrizione_da_api_doc | Stringa | N/A |
| | Id | Stringa | N/A |
| Visualizzazioni della pagina | Name | Stringa | N/A |
| | Periodo | Periodo | EQUAL_TO |
| | Since | DateTime | EQUAL_TO |
| | Valori | List | N/D |
| | Titolo | Stringa | N/A |
| | Description | Stringa | N/A |
| | descrizione_da_api_doc | Stringa | N/A |
| | Id | Stringa | N/A |
| Post dei video della pagina | Name | Stringa | N/A |
| | Periodo | Periodo | EQUAL_TO |
| | Since | DateTime | EQUAL_TO |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | Valori | List | N/D |
| | Titolo | Stringa | N/A |
| | Description | Stringa | N/A |
| | descrizione_da_api_doc | Stringa | N/A |
| | Id | Stringa | N/A |
| Pagine | Name | Stringa | N/A |
| | Informazioni | Stringa | N/A |
| | access_token | Stringa | N/A |
| | ad_campaign | Stringa | N/A |
| | Affiliazione | Stringa | N/A |
| | app_id | Stringa | N/A |
| | artists_we_like | Stringa | N/A |
| | Abbigliamento | Stringa | N/A |
| | Premi | Stringa | N/A |
| | band_interests | Stringa | N/A |
| | band_members | Stringa | N/A |
| | best_page | Stringa | N/A |
| | Bio | Stringa | N/A |
| | Compleanno | Stringa | N/A |
| | booking_agent | Stringa | N/A |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | Costruito | Stringa | N/A |
| | can_checkin | Stringa | N/A |
| | can_post | Stringa | N/A |
| | Categoria | Stringa | N/A |
| | elenco_categoria | List | N/D |
| | Controlli | Numero intero | N/D |
| | company_overview | Stringa | N/A |
| | connected_instagram_account | Stringa | N/A |
| | contact_address | Stringa | N/A |
| | country_page_likes | Numero intero | N/D |
| | Copertura | Struct | N/D |
| | culinary_team | Stringa | N/A |
| | current_location | Stringa | N/A |
| | delivery_and_pickup_option_info | List | N/D |
| | Description | Stringa | N/A |
| | description_html | Stringa | N/A |
| | differently_open_offerings | List | N/D |
| | directed_by | Stringa | N/A |
| | display_subtext | Stringa | N/A |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | displayed_message_response_time | Stringa | N/A |
| | E-mail | Stringa | N/A |
| | Impegno | Stringa | N/A |
| | fan_count | Numero intero | N/D |
| | featured_video | Stringa | N/A |
| | Funzionalità | Stringa | N/A |
| | followers_count | Numero intero | N/D |
| | food_styles | List | N/D |
| | Fondato | Stringa | N/A |
| | general_info | Stringa | N/A |
| | general_manager | Stringa | N/A |
| | Genere | Stringa | N/A |
| | global_brand_page_name | Stringa | N/A |
| | global_brand_root_id | Stringa | N/A |
| | has_added_app | Booleano | N/D |
| | has_transitioned_to_new_page_experience | Booleano | N/D |
| | has_whatsapp_business_number | Booleano | N/D |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | has_whatsapp_number | Booleano | N/D |
| | Città natale | Stringa | N/A |
| | Ore | Struct | N/D |
| | Impressum | Stringa | N/A |
| | Influenze | Stringa | N/A |
| | instagram_business_account | Stringa | N/A |
| | is_always_open | Booleano | N/D |
| | is_chain | Booleano | N/D |
| | is_community_page | Booleano | N/D |
| | is_eligible_for_branded_content | Booleano | N/D |
| | is_messenger_bot_get_started_enabled | Booleano | N/D |
| | is_messenger_platform_bot | Booleano | N/D |
| | is_owned | Booleano | N/D |
| | is_permanently_closed | Booleano | N/D |
| | is_published | Booleano | N/D |
| | Name | Stringa | N/A |
| | Processi | List | N/D |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | is_unclaimed | Booleano | N/D |
| | is_webhooks_subscribed | Booleano | N/D |
| | leadgen_tos_acceptance_time | DateTime | N/D |
| | leadgen_tos_accepted | Booleano | N/D |
| | leadgen_tos_accepting_user | Stringa | N/A |
| | leadgen_tos_accepting_user | Struct | N/D |
| | Link | Link | N/D |
| | Location (Ubicazione) | Struct | N/D |
| | Membri | Stringa | N/A |
| | merchant_review_status | Stringa | N/A |
| | messenger_ads_default_icebreakers | List | N/D |
| | messenger_ads_default_page_welcome_message | Struct | N/D |
| | messenger_ads_default_quick_replies | List | N/D |
| | messenger_ads_quick_replies_type | Stringa | N/A |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | Mission | Stringa | N/A |
| | Mpg | Stringa | N/A |
| | name_with_location_descriptor | Stringa | N/A |
| | Rete | Stringa | N/A |
| | new_like_count | Numero intero | N/D |
| | offer_eligible | Booleano | N/D |
| | overall_star_rating | Float | N/D |
| | page_token | Stringa | N/A |
| | parent_page | Stringa | N/A |
| | Parking | Stringa | N/A |
| | payment_options | Struct | N/D |
| | personal_info | Stringa | N/A |
| | personal_interests | Stringa | N/A |
| | pharma_safety_info | Stringa | N/A |
| | Telefono | Stringa | N/A |
| | pickup_options | List | N/D |
| | place_type | Stringa | N/A |
| | plot_outline | Stringa | N/A |
| | press_contact | Stringa | N/A |
| | price_range | Stringa | N/A |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | privacy_info_url | Stringa | N/A |
| | produced_by | Stringa | N/A |
| | Prodotti | Stringa | N/A |
| | promotion_eligible | Booleano | N/D |
| | promotion_ineligible_reason | Stringa | N/A |
| | public_transit | Stringa | N/A |
| | rating_count | Numero intero | N/D |
| | record_label | Stringa | N/A |
| | release_date | Stringa | N/A |
| | restaurant_services | Struct | N/D |
| | restaurant_specialties | Struct | N/D |
| | Schedule | Stringa | N/A |
| | screenplay_by | Stringa | N/A |
| | Stagione | Stringa | N/A |
| | single_line_address | Stringa | N/A |
| | Con protagonista | Stringa | N/A |
| | start_info | Struct | N/D |
| | store_code | Stringa | N/A |
| | store_location_descriptor | Stringa | N/A |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | store_number | Numero intero | N/D |
| | Studio | Stringa | N/A |
| | supports_donate_button_in_live_video | Booleano | N/D |
| | talking_about_count | Numero intero | N/D |
| | temporary_status | Stringa | N/A |
| | unread_message_count | Numero intero | N/D |
| | unread_notif_count | Numero intero | N/D |
| | unseen_message_count | Numero intero | N/D |
| | Username | Stringa | N/A |
| | verification_status | Stringa | N/A |
| | voip_info | Struct | N/D |
| | Website | Stringa | N/A |
| | were_here_count | Numero intero | N/D |
| | whatsapp_number | Stringa | N/A |
| written_by | Stringa | N/A | |
| Feed | Id | Stringa | N/A |
| | Azioni | List | N/D |
| | admin_creator | Oggetto | N/D |
| | Applicazione | Oggetto | N/D |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | Allegati | Oggetti | N/D |
| | backdated_time | DateTime | N/D |
| | call_to_action | Oggetto | N/D |
| | can_reply_privately | Booleano | N/D |
| | child_attachments | List | N/D |
| | Coordinates | Struct | N/D |
| | created_time | DateTime | N/D |
| | Event | Struct | N/D |
| | expanded_height | Numero intero | N/D |
| | expanded_width | Numero intero | N/D |
| | feed_targeting | Oggetto | N/D |
| | Da | Oggetto | N/D |
| | full_picture | Stringa | N/A |
| | Altezza | Numero intero | N/D |
| | Icon | Stringa | N/A |
| | instagram_eligibility | Stringa | N/A |
| | is_eligible_for_promotion | Booleano | N/D |
| | is_expired | Booleano | N/D |
| | is_hidden | Booleano | N/D |
| | is_inline_created | Booleano | N/D |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | is_instagram_eligible | Booleano | N/D |
| | is_popular | Booleano | N/D |
| | is_published | Booleano | N/D |
| | is_spherical | Booleano | N/D |
| | Messaggio | Stringa | N/A |
| | message_tags | List | N/D |
| | multi_share_end_card | Booleano | N/D |
| | multi_share_optimized | Booleano | N/D |
| | parent_id | Stringa | N/A |
| | permalink_url | Stringa | N/A |
| | Place | Stringa | N/A |
| | Privacy | Oggetto | N/D |
| | promotable_id | Stringa | N/A |
| | promotion_status | Stringa | N/A |
| | Properties | List | N/D |
| | scheduled_publish_time | Float | N/D |
| | Azioni | Oggetto | N/D |
| | status_type | Stringa | N/A |
| | Storia | Stringa | N/A |
| | story_tags | List | N/D |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | Sottoscritto | Booleano | N/D |
| | Target | Struct | N/D |
| | Definizione del target | Oggetto | N/D |
| | Per | Oggetto | N/D |
| | timeline_visibility | Stringa | N/A |
| | updated_time | DateTime | N/D |
| | Via | Struct | N/D |
| | video_buying_eligibility | List | N/D |
| | Larghezza | Numero intero | N/D |
| | Since | DateTime | EQUAL_TO |

Query di partizionamento

Partizionamento basato su filtri:

È possibile fornire le opzioni Spark aggiuntive `PARTITION_FIELD`, `LOWER_BOUND`, `UPPER_BOUND` e `NUM_PARTITIONS` se si desidera utilizzare la simultaneità in Spark. Con questi parametri, la query originale viene suddivisa in `NUM_PARTITIONS` query secondarie che possono essere eseguite contemporaneamente dalle attività Spark.

- `PARTITION_FIELD`: il nome del campo da utilizzare per partizionare la query.
- `LOWER_BOUND`: un valore limite inferiore inclusivo del campo di partizione scelto.

  Per il campo Datetime accettiamo il formato di timestamp Spark utilizzato nelle query SQL di Spark.

  Esempi di valori validi:

```
"2024-09-30T01:01:01.000Z"
```

- UPPER_BOUND: un valore limite superiore esclusivo del campo di partizione scelto.
- NUM_PARTITIONS: il numero di partizioni.

Esempio:

```
facebookPageInsights_read = glueContext.create_dynamic_frame.from_options(
     connection_type="facebookpageinsights",
     connection_options={
         "connectionName": "connectionName",
         "ENTITY_NAME": "entityName",
         "API_VERSION": "v21",
         "PARTITION_FIELD": "created_Time"
         "LOWER_BOUND": "2024-10-27T07:00:00+0000"
         "UPPER_BOUND": "2024-10-27T07:00:00+0000"
         "NUM_PARTITIONS": "10"
     }
```

## Opzioni di connessione degli Approfondimenti sulla pagina Facebook

Le seguenti sono le opzioni di connessione per gli Approfondimenti sulla pagina Facebook:

- ENTITY_NAME(String): (obbligatorio) utilizzato per la lettura. Il nome del tuo oggetto negli Approfondimenti sulla pagina Facebook.
- API_VERSION(String): (obbligatorio) utilizzato per la lettura. Versione dell'API Rest degli Approfondimenti sulla pagina Facebook che desideri utilizzare.
- SELECTED_FIELDS(List<String>): predefinito: vuoto (SELECT *). Utilizzato per la lettura. Colonne da selezionare per l'oggetto.
- FILTER_PREDICATE(Stringa): predefinito: vuoto. Utilizzato per la lettura. Dev'essere in formato Spark SQL.
- QUERY(Stringa): predefinito: vuoto. Utilizzato per la lettura. Query Spark SQL completa.
- PARTITION_FIELD(Stringa): utilizzato per la lettura. Campo da utilizzare per partizionare la query.
- LOWER_BOUND(Stringa): utilizzato per la lettura. Un valore limite inferiore inclusivo del campo di partizione scelto.
- UPPER_BOUND(Stringa): utilizzato per la lettura. Un valore limite superiore esclusivo del campo di partizione scelto.
- NUM_PARTITIONS(Numero intero): predefinito: 1. Utilizzato per la lettura. Numero di partizioni da leggere.

- `INSTANCE_URL`(Stringa): (obbligatorio) utilizzato per la lettura. Un URL di istanza valido di Facebook Page Insights.

### Limitazioni e note per il connettore degli Approfondimenti sulla pagina Facebook

Di seguito sono riportate le limitazioni o le note per il connettore degli Approfondimenti sulla pagina Facebook:

- La maggior parte delle metriche viene aggiornata una volta ogni 24 ore.
- Sono disponibili solo i dati degli ultimi due anni di approfondimenti.
- È possibile visualizzare solo 90 giorni di approfondimenti alla volta utilizzando i parametri `since` e `until`.

## Connessione a Freshdesk

Freshdesk è un software di assistenza clienti basato su cloud ricco di funzionalità e facile da utilizzare. Grazie ai diversi canali di assistenza disponibili, tra cui chat dal vivo, e-mail, telefono e social media, permette di aiutare i clienti tramite il loro metodo di comunicazione preferito. Se sei un utente Freshdesk, puoi connetterti AWS Glue al tuo account Freshdesk. Puoi usare Freshdesk come origine dati nei tuoi processi ETL. Esegui questi job per trasferire dati da Freshdesk ai AWS servizi o ad altre applicazioni supportate.

Argomenti


- AWS Glue supporto per Freshdesk
- Policy contenenti le operazioni API per creare e utilizzare le connessioni
- Configurazione di Freshdesk
- Configurazione delle connessioni Freshdesk
- Lettura da entità Freshdesk
- Opzioni di connessione Freshdesk
- Limitazioni e note per il connettore Freshdesk


### AWS Glue supporto per Freshdesk

AWS Glue supporta Freshdesk come segue:

Supportato come origine?

Sì, sincrona e asincrona. È possibile utilizzare i job AWS Glue ETL per interrogare i dati da Freshdesk.

Supportato come destinazione?

No.

Versioni API Freshdesk supportate

Sono supportate le seguenti versioni API Freshdesk:

- v2

## Policy contenenti le operazioni API per creare e utilizzare le connessioni

La seguente policy di esempio descrive le autorizzazioni AWS IAM richieste per la creazione e l'utilizzo delle connessioni. Se si sta creando un nuovo ruolo, creare una policy che contenga quanto segue:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:ListConnectionTypes",
        "glue:DescribeConnectionType",
        "glue:RefreshOAuth2Tokens",
        "glue:ListEntities",
        "glue:DescribeEntity"
      ],
      "Resource": "*"
    }
  ]
}
```

In alternativa, se non desideri utilizzare il metodo precedente, utilizza le seguenti policy IAM gestite:

- AWSGlueServiceRole— Garantisce l'accesso alle risorse che AWS Glue i vari processi richiedono per l'esecuzione per conto dell'utente. Queste risorse includono AWS Glue Amazon S3, IAM, CloudWatch Logs e Amazon. EC2 Se segui la convenzione di denominazione delle risorse specificata in questa politica, AWS Glue i processi dispongono delle autorizzazioni richieste. Questa policy è in genere collegata ai ruoli specificati quando si definiscono crawler, processi ed endpoint di sviluppo.
- AWSGlueConsoleFullAccess— Concede l'accesso completo alle AWS Glue risorse quando un'identità a cui è allegata la policy utilizza la AWS console di gestione. Se si segue la convenzione per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questa policy è in genere associata agli utenti della AWS Glue console.

## Configurazione di Freshdesk

Prima di poter AWS Glue trasferire dati da Freshdesk, devi soddisfare questi requisiti:

Requisiti minimi

I requisiti minimi sono i seguenti:

- Un account Freshdesk. Puoi scegliere tra le edizioni Free, Growth, Pro o Enterprise.
- Una chiave API per un utente Freshdesk.

## Configurazione delle connessioni Freshdesk

Freshdesk supporta l'autenticazione personalizzata.

Per la documentazione pubblica di Freshdesk sulla generazione delle chiavi API richieste per l'autorizzazione personalizzata, consulta l'autenticazione diFreshdesk.

Di seguito sono riportati i passaggi per configurare la connessione a Freshdesk:

- In AWS Secrets Manager, crea un segreto con i seguenti dettagli:
  - Per le app connesse gestite dal cliente, il segreto deve contenere la chiave API dell'app connessa con `apiKey` come chiave. Nota: è necessario creare un segreto per la connessione in AWS Glue.
- In AWS Glue Studio, crea una connessione in Connessioni dati seguendo i passaggi seguenti:
  - Quando si seleziona un'Origine dati, selezionare Freshdesk.

- Fornire `INSTANCE_URL` dell'istanza Freshdesk a cui connettersi.
- Seleziona il ruolo AWS IAM che AWS Glue può assumere e dispone delle autorizzazioni per le seguenti azioni:

  JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:DescribeSecret",
        "secretsmanager:GetSecretValue",
        "secretsmanager:PutSecretValue",
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DeleteNetworkInterface"
      ],
      "Resource": "*"
    }
  ]
}
```

  - Seleziona quello `secretName` che desideri utilizzare per questa connessione per AWS Glue inserire i token.
  - Selezionare le opzioni di rete se si desidera utilizzare la propria rete.
- Concedi il permesso di lettura `secretName` al ruolo IAM associato al tuo AWS Glue lavoro.
- Nella configurazione del AWS Glue lavoro, fornisci `connectionName` una connessione di rete aggiuntiva.

## Lettura da entità Freshdesk

Prerequisito

Un oggetto Freshdesk da cui si desidera leggere. È necessario il nome dell'oggetto.

Entità supportate come origine Sync:

| Entità | Può essere filtrato | Supporta limite | Supporta Ordina per | Supporta Seleziona * | Supporta il partizionamento |
|---|---|---|---|---|---|
| Agents (Agenti) | Sì | Sì | No | Sì | Sì |
| Orario lavorativo | No | Sì | No | Sì | Sì |
| Azienda | Sì | Sì | No | Sì | Sì |
| Contatti | Sì | Sì | No | Sì | Sì |
| Conversazioni | No | Sì | No | Sì | No |
| Configurazioni e-mail | No | Sì | No | Sì | No |
| Caselle di posta | Sì | Sì | Sì | Sì | No |
| Categorie del forum | No | Sì | No | Sì | No |
| Forum | No | Sì | No | Sì | No |
| Gruppi | No | Sì | No | Sì | No |
| Prodotti | No | Sì | No | Sì | No |
| Roles | No | Sì | No | Sì | No |
| Indici di soddisfazione | Sì | Sì | No | Sì | No |
| Competenze | No | Sì | No | Sì | No |
| Soluzioni | Sì | Sì | No | Sì | No |

| Entità | Può essere filtrato | Supporta limite | Supporta Ordina per | Supporta Seleziona * | Supporta il partizionamento |
|---|---|---|---|---|---|
| Sondaggi | No | Sì | No | Sì | No |
| Ticket | Sì | Sì | Sì | Sì | Sì |
| Voci temporali | Sì | Sì | No | Sì | No |
| Argomenti | No | Sì | No | Sì | No |
| Commenti sull'argomento | No | Sì | No | Sì | No |

Entità supportate come origine Async:

| Entità | Versione API | Può essere filtrato | Supporta limite | Supporta Ordina per | Supporta Seleziona * | Supporta il partizionamento |
|---|---|---|---|---|---|---|
| Aziende | v2 | No | No | No | No | No |
| Contatti | v2 | No | No | No | No | No |

Esempio:

```
freshdesk_read = glueContext.create_dynamic_frame.from_options(
    connection_type="freshdesk",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "entityName",
        "API_VERSION": "v2"
    }
```

Informazioni sull'entità e sul campo di Freshdesk:

| Entità | Campo |
| --- | --- |
| Agents (Agenti) | https://developers.freshdesk.com/api/#list_all_agents |
| Ore lavorative | https://developers.freshdesk.com/api/#list_all_business_hours |
| Commenti | https://developers.freshdesk.com/api/#comment_attributess |
| Azienda | https://developers.freshdesk.com/api/#companies |
| Contatti | https://developers.freshdesk.com/api/#list_all_contacts |
| Conversazioni | https://developers.freshdesk.com/api/#list_all_ticket_notes |
| Configurazioni E-mail | https://developers.freshdesk.com/api/#list_all_email_configs |
| Caselle di posta | https://developers.freshdesk.com/api/#list_all_email_mailboxes |
| Categories del forum | https://developers.freshdesk.com/api/#category_attributes |
| Forum | https://developers.freshdesk.com/api/#forum_attributes |
| Gruppi | https://developers.freshdesk.com/api/#list_all_groups |
| Prodotti | https://developers.freshdesk.com/api/#list_all_products |

| Entità | Campo |
| --- | --- |
| Roles | https://developers.freshdesk.com/api/#list_all_roles |
| Indici di soddisfazione | https://developers.freshdesk.com/api/#view_all_satisfaction_ratingss |
| Competenze | https://developers.freshdesk.com/api/#list_all_skills |
| Soluzioni | https://developers.freshdesk.com/api/#solution_content |
| Sondaggi | https://developers.freshdesk.com/api/#list_all_survey |
| Ticket | https://developers.freshdesk.com/api/#list_all_tickets |
| Voci temporali | https://developers.freshdesk.com/api/#list_all_time_entries |
| Argomenti | https://developers.freshdesk.com/api/#topic_attributes |

Query di partizionamento

Partizionamento basato su filtri:

È possibile fornire le opzioni Spark aggiuntive `PARTITION_FIELD`, `LOWER_BOUND`, `UPPER_BOUND` e `NUM_PARTITIONS` se si desidera utilizzare la simultaneità in Spark. Con questi parametri, la query originale viene suddivisa in `NUM_PARTITIONS` query secondarie che possono essere eseguite contemporaneamente dalle attività Spark.

- `PARTITION_FIELD`: il nome del campo da utilizzare per partizionare la query.
- `LOWER_BOUND`: un valore limite inferiore inclusivo del campo di partizione scelto.

  Per il campo Datetime accettiamo il formato di timestamp Spark utilizzato nelle query SQL di Spark.

Esempi di valori validi:

```
"2024-09-30T01:01:01.000Z"
```

- UPPER_BOUND: un valore limite superiore esclusivo del campo di partizione scelto.
- NUM_PARTITIONS: il numero di partizioni.

Esempio:

```
freshDesk_read = glueContext.create_dynamic_frame.from_options(
     connection_type="freshdesk",
     connection_options={
         "connectionName": "connectionName",
         "ENTITY_NAME": "entityName",
         "API_VERSION": "v2",
         "PARTITION_FIELD": "Created_Time"
         "LOWER_BOUND": " 2024-10-27T23:16:08Z“
         "UPPER_BOUND": " 2024-10-27T23:16:08Z"
         "NUM_PARTITIONS": "10"
     }
```

## Opzioni di connessione Freshdesk

Di seguito sono riportate le opzioni di connessione per Freshdesk:

- ENTITY_NAME(String): (obbligatorio) utilizzato per la lettura. Il nome del tuo oggetto in Freshdesk.
- API_VERSION(String): (obbligatorio) utilizzato per la lettura. Versione dell'API Rest di Freshdesk da utilizzare.
- SELECTED_FIELDS(List<String>): predefinito: vuoto (SELECT *). Utilizzato per la lettura. Colonne da selezionare per l'oggetto.
- FILTER_PREDICATE(Stringa): predefinito: vuoto. Utilizzato per la lettura. Dev'essere in formato Spark SQL.
- QUERY(Stringa): predefinito: vuoto. Utilizzato per la lettura. Query Spark SQL completa.
- PARTITION_FIELD(Stringa): utilizzato per la lettura. Campo da utilizzare per partizionare la query.
- LOWER_BOUND(Stringa): utilizzato per la lettura. Un valore limite inferiore inclusivo del campo di partizione scelto.

- UPPER_BOUND(Stringa): utilizzato per la lettura. Un valore limite superiore esclusivo del campo di partizione scelto.
- NUM_PARTITIONS(Numero intero): predefinito: 1. Utilizzato per la lettura. Numero di partizioni da leggere.
- INSTANCE_URL(Stringa): (obbligatorio) utilizzato per la lettura. Un URL di istanza Freshdesk valido.
- TRANSFER_MODE(Stringa) - Utilizzato per indicare se il tipo di elaborazione è uguale SYNC o ASYNC impostato su SYNC per impostazione predefinita. (Facoltativo)

## Limitazioni e note per il connettore Freshdesk

Di seguito sono riportate le limitazioni o le note per il connettore Freshdesk:

- Le entità Company, Contacts e Tickets con filtrazione hanno limitazioni di impaginazione. Restituiscono solo 30 record per pagina e il valore della pagina può essere impostato su un massimo di 10 (recuperando un massimo di 300 record).
- L'Ticketsentità non recupera record più vecchi di 30 giorni.
- Le entità Company, Contacts e Tickets supportano il tipo di dati "Date" nel filtraggio. È necessario selezionare le frequenze di attivazione successive "Daily" per queste tre entità. La selezione di "Minutes" o "Hourly" può portare a dati duplicati. Inoltre, durante la selezione di questi campi per il filtraggio, deve essere selezionato solo il valore della data, poiché prenderà in considerazione solo la parte relativa alla data del timestamp selezionato.
- Il numero di chiamate API al minuto dipende dal tuo piano. Questo limite viene applicato a livello di account indipendentemente da fattori come il numero di agenti o gli indirizzi IP utilizzati per effettuare le chiamate. Per tutti gli utenti della versione di prova, esiste un limite API predefinito di 50 chiamate/minuto. Per ulteriori informazioni, consulta Freshdesk
- Per qualsiasi entità, viene elaborato un solo Export/Async Job alla volta. Un nuovo lavoro verrà elaborato solo dopo che il lavoro esistente è stato completato con successo o non è riuscito. Per ulteriori informazioni, consulta Freshdesk
- I seguenti campi sono supportati per le chiamate all'API Sync, ma non supported/allowed devono essere passati nel corpo della richiesta dell'API Async.
  - id
  - created_at
  - updated_at

- updated_since
- attiva
- company_id
- other_companies
- avatar
- view_all_tickets
- deleted (eliminato)
- other_emails
- stato
- tag
- tag

## Connessione a Freshsales

Freshsales è un CRM intuitivo che aiuta i rappresentanti commerciali a eliminare le congetture dal processo di vendita. Grazie al telefono e alla posta elettronica integrati, alle attività, agli appuntamenti e alle note, i rappresentanti commerciali non devono più passare da una scheda all'altra per seguire i potenziali clienti. Permette di gestire meglio le trattative con la visualizzazione della pipeline e di portare a termine un maggior numero di trattative. Se sei un utente Freshsales, puoi connetterti AWS Glue al tuo account Freshsales. Puoi usare Freshsales come origine dati nei tuoi processi ETL. Esegui questi lavori per trasferire i dati da Freshsales ai AWS servizi o ad altre applicazioni supportate.

Argomenti


- AWS Glue supporto per Freshsales
- Policy contenenti le operazioni API per creare e utilizzare le connessioni
- Configurazione di Freshsales
- Configurazione delle connessioni Freshsales
- Lettura da entità Freshsales
- Opzioni di connessione Freshsales
- Limitazioni di Freshsales

## AWS Glue supporto per Freshsales

AWS Glue supporta Freshsales come segue:

Supportato come origine?

Sì. È possibile utilizzare i job AWS Glue ETL per interrogare i dati di Freshsales.

Supportato come destinazione?

No.

Versioni API Freshsales supportate

Sono supportate le seguenti versioni API Freshsales:

- v1.0

## Policy contenenti le operazioni API per creare e utilizzare le connessioni

La seguente policy di esempio descrive le autorizzazioni AWS IAM richieste per la creazione e l'utilizzo delle connessioni. Se si sta creando un nuovo ruolo, creare una policy che contenga quanto segue:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:ListConnectionTypes",
        "glue:DescribeConnectionType",
        "glue:RefreshOAuth2Tokens",
        "glue:ListEntities",
        "glue:DescribeEntity"
      ],
      "Resource": "*"
    }
  ]
}
```

In alternativa, se non desideri utilizzare il metodo precedente, utilizza le seguenti policy IAM gestite:

- AWSGlueServiceRole— Garantisce l'accesso alle risorse che AWS Glue i vari processi richiedono per l'esecuzione per conto dell'utente. Queste risorse includono AWS Glue Amazon S3, IAM, CloudWatch Logs e Amazon. EC2 Se segui la convenzione di denominazione delle risorse specificata in questa politica, AWS Glue i processi dispongono delle autorizzazioni richieste. Questa policy è in genere collegata ai ruoli specificati quando si definiscono crawler, processi ed endpoint di sviluppo.
- AWSGlueConsoleFullAccess— Concede l'accesso completo alle AWS Glue risorse quando un'identità a cui è allegata la policy utilizza la AWS console di gestione. Se si segue la convenzione per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questa policy è in genere associata agli utenti della AWS Glue console.

## Configurazione di Freshsales

Prima di poter utilizzare AWS Glue per trasferire dati da Freshsales, devi soddisfare questi requisiti:

Requisiti minimi

I requisiti minimi sono i seguenti:

- Hai un account Freshsales.
- Hai una chiave API utente.

Se soddisfi questi requisiti, sei pronto per connetterti AWS Glue al tuo account Freshsales. Per le connessioni tipiche, non è necessario fare nient'altro in Freshsales.

## Configurazione delle connessioni Freshsales

Freshsales supporta l'autenticazione personalizzata.

Per la documentazione pubblica di Freshsales sulla generazione delle chiavi API richieste per l'autenticazione personalizzata, consulta l'Autenticazione.

Per configurare una connessione Freshsales:

1. In AWS Secrets Manager, crea un segreto con i seguenti dettagli:

a. Per le app connesse gestite dal cliente, il segreto deve contenere la chiave API dell'app connessa con *apiSecretKey* come chiave. Il segreto deve contenere anche un'altra coppia chiave-valore con *apiKey* come chiave e *token* come valore.

b. Nota: devi creare un segreto per le tue connessioni in AWS Glue.

1. In AWS Glue Glue Studio, crea una connessione in Connessioni dati seguendo i passaggi seguenti:

   a. Quando si seleziona una Origine dati, selezionare Freshsales.

   b. Fornire `INSTANCE_URL` dell'account Freshsales a cui connettersi.

   c. Seleziona il ruolo AWS IAM che AWS Glue può assumere e dispone delle autorizzazioni per le seguenti azioni:

      JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:DescribeSecret",
        "secretsmanager:GetSecretValue",
        "secretsmanager:PutSecretValue",
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DeleteNetworkInterface"
      ],
      "Resource": "*"
    }
  ]
}
```

   d. Seleziona quello `secretName` che desideri utilizzare per questa connessione per AWS Glue inserire i token.

   e. Selezionare le opzioni di rete se si desidera utilizzare la propria rete.

2. Concedi il permesso di lettura `secretName` al ruolo IAM associato al tuo AWS Glue lavoro.

## Lettura da entità Freshsales

Prerequisito

Un oggetto Freshsales da cui si desidera leggere. È necessario il nome dell'oggetto.

Entità supportate come origine:

| Entità | Può essere filtrato | Supporta limite | Supporta Ordina per | Supporta Seleziona * | Supporta il partizionamento |
|---|---|---|---|---|---|
| Account | Sì | Sì | Sì | Sì | Sì |
| Contatti | Sì | Sì | Sì | Sì | Sì |

Esempio:

```
freshSales_read = glueContext.create_dynamic_frame.from_options(
     connection_type="freshsales",
     connection_options={
         "connectionName": "connectionName",
         "ENTITY_NAME": "entityName",
         "API_VERSION": "v1.0"
     }
```

Dettagli dell'entità e del campo di Freshsales:

Freshsales fornisce endpoint per recuperare i metadati in modo dinamico per le entità supportate. Di conseguenza, il supporto dell'operatore viene acquisito a livello di tipo di dati.

| Entità | Tipo di dati | Operatori supportati |
|---|---|---|
| Entità Freshsale (tutte) | Numero intero | !=,=,<,<=,>,>=,BETWEEN |
| | Stringa | Like, =, != |
| | BigInteger | !=,=,<,<=,>,>=,BETWEEN |
| | Booleano | = |

| Entità | Tipo di dati | Operatori supportati |
|---|---|---|
| | Double | !=,=,<,<=,>,>=,BETWEEN |
| | BigDecimal | !=,=,<,<=,>,>=,BETWEEN |
| | Data | !=,=,<,<=,>,>=,BETWEEN |
| | DateTime | !=,=,<,<=,>,>=,BETWEEN |
| | Struct | N/D |
| | List | N/D |

Query di partizionamento

Partizionamento basato su filtri:

È possibile fornire le opzioni Spark aggiuntive `PARTITION_FIELD`, `LOWER_BOUND`, `UPPER_BOUND` e `NUM_PARTITIONS` se si desidera utilizzare la simultaneità in Spark. Con questi parametri, la query originale viene suddivisa in `NUM_PARTITIONS` query secondarie che possono essere eseguite contemporaneamente dalle attività Spark.

- `PARTITION_FIELD`: il nome del campo da utilizzare per partizionare la query.
- `LOWER_BOUND`: un valore limite inferiore inclusivo del campo di partizione scelto.

  Per il campo Datetime, accettiamo il valore in formato ISO.

  Esempi di valori validi:

  ```
  "2024-09-30T01:01:01.000Z"
  ```

- `UPPER_BOUND`: un valore limite superiore esclusivo del campo di partizione scelto.
- `NUM_PARTITIONS`: il numero di partizioni.

Esempio:

```
freshSales_read = glueContext.create_dynamic_frame.from_options(
     connection_type="freshsales",
```

```
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "entityName",
        "API_VERSION": "v1",
        "PARTITION_FIELD": "Created_Time"
        "LOWER_BOUND": " 2024-10-15T21:16:25Z"
        "UPPER_BOUND": " 2024-10-20T21:25:50Z"
        "NUM_PARTITIONS": "10"
    }
```

## Opzioni di connessione Freshsales

Di seguito sono riportate le opzioni di connessione per Freshsales:

- `ENTITY_NAME`(String): (obbligatorio) utilizzato per la lettura. Il nome del tuo oggetto in Freshsales.
- `API_VERSION`(String): (obbligatorio) utilizzato per la lettura. Versione dell'API Rest di Freshsales da usare.
- `SELECTED_FIELDS`(List<String>): predefinito: vuoto (SELECT *). Utilizzato per la lettura. Colonne da selezionare per l'oggetto.
- `FILTER_PREDICATE`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Dev'essere in formato Spark SQL.
- `QUERY`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Query Spark SQL completa.
- `PARTITION_FIELD`(Stringa): utilizzato per la lettura. Campo da utilizzare per partizionare la query.
- `LOWER_BOUND`(Stringa): utilizzato per la lettura. Un valore limite inferiore inclusivo del campo di partizione scelto.
- `UPPER_BOUND`(Stringa): utilizzato per la lettura. Un valore limite superiore esclusivo del campo di partizione scelto.
- `NUM_PARTITIONS`(Numero intero): predefinito: 1. Utilizzato per la lettura. Numero di partizioni da leggere.
- `INSTANCE_URL`(String): utilizzato per la lettura. Un URL di istanza Freshsales valido.

## Limitazioni di Freshsales

Di seguito sono riportate le limitazioni o le note per Freshsales:

- In Freshsales, il limite di frequenza API è di 1000 richieste API all'ora per account (vedi Errori). Ma questo limite è estensibile con il piano di abbonamento Enterprise (vedi il confronto dei piani).

# Connessione a Google Ads

L'API Google Ads è l'interfaccia programmatica di Google Ads, usata per gestire campagne e account Google Ads complessi o di grandi dimensioni. Se sei un utente Google Ads, puoi connetterti AWS Glue al tuo account Google Ads. Quindi, puoi usare Google Ads come origine dati nei tuoi processi ETL. Esegui questi processi per trasferire dati tra Google Ads e AWS i servizi o altre applicazioni supportate.

Argomenti


- AWS Glue supporto per Google Ads
- Policy contenenti le operazioni API per creare e utilizzare le connessioni
- Configurazione di Google Ads
- Configurazione delle connessioni Google Ads
- Lettura da entità di Google Ads
- Opzioni di connessione Google Ads
- Creazione di un account Google Ads
- Limitazioni


## AWS Glue supporto per Google Ads

AWS Glue supporta Google Ads come segue:

Supportato come origine?

Sì. Puoi utilizzare i lavori AWS Glue ETL per interrogare i dati di Google Ads.

Supportato come destinazione?

No.

Versioni dell'API Google Ads supportate

v18

## Policy contenenti le operazioni API per creare e utilizzare le connessioni

La seguente politica di esempio descrive le AWS autorizzazioni richieste per la creazione e l'utilizzo delle connessioni. Se si sta creando un nuovo ruolo, creare una policy che contenga quanto segue:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:ListConnectionTypes",
        "glue:DescribeConnectionType",
        "glue:RefreshOAuth2Tokens",
        "glue:ListEntities",
        "glue:DescribeEntity"
      ],
      "Resource": "*"
    }
  ]
}
```

È anche possibile utilizzare le seguenti policy IAM per autorizzare l'accesso:

- AWSGlueServiceRole— Garantisce l'accesso alle risorse che diversi AWS Glue processi richiedono per l'esecuzione per conto dell'utente. Queste risorse includono Amazon S3 AWS Glue, IAM, CloudWatch Logs e Amazon EC2. Se segui la convenzione di denominazione delle risorse specificata in questa politica, AWS Glue i processi dispongono delle autorizzazioni necessarie. Questa policy è in genere collegata ai ruoli specificati quando si definiscono crawler, processi ed endpoint di sviluppo.
- AWSGlueConsoleFullAccess— Concede l'accesso completo alle AWS Glue risorse quando un'identità a cui è allegata la policy utilizza la AWS console di gestione. Se si segue la convenzione per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questa policy è in genere associata agli utenti della AWS Glue console.

## Configurazione di Google Ads

Prima di poter AWS Glue utilizzare il trasferimento da Google Ads, devi soddisfare questi requisiti:

Requisiti minimi

- Hai un account Google Ads con e-mail e password. Per ulteriori informazioni sulla creazione di account, consulta Creating a Google Ads account.
- Il tuo account di Google Ads è abilitato all'accesso tramite API. Tutti gli utilizzi dell'API di Google Ads sono disponibili senza costi aggiuntivi.
- Il tuo account Google Ads ti consente di installare app connesse. Se non hai accesso a questa funzionalità, contatta l'amministratore di Google Ads.

Se soddisfi questi requisiti, sei pronto per connetterti AWS Glue al tuo account Google Ads.

## Configurazione delle connessioni Google Ads

Google Ads supporta il tipo di `AUTHORIZATION_CODE` sovvenzione per OAuth2.

Questo tipo di concessione è considerato «a tre gambe» in OAuth quanto si basa sul reindirizzamento degli utenti al server di autorizzazione di terze parti per autenticare l'utente. Viene utilizzato durante la creazione di connessioni tramite la console. AWS Glue La AWS Glue Console reindirizzerà l'utente a Google Ads, dove l'utente deve effettuare il login e concedere AWS Glue le autorizzazioni richieste per accedere alla propria istanza Google Ads.

Gli utenti possono scegliere di creare la propria app connessa in Google Ads e fornire il proprio ID cliente e il segreto del client quando creano connessioni tramite la AWS Glue Console. In questo scenario, verranno comunque reindirizzati a Google Ads per accedere e autorizzare l'accesso AWS Glue alle proprie risorse.

Questo tipo di concessione genera un token di aggiornamento e un token di accesso. Il token di accesso dura poco tempo e può essere aggiornato automaticamente senza l'interazione dell'utente utilizzando il token di aggiornamento.

Per ulteriori informazioni, consulta la documentazione pubblica di Google Ads sulla creazione di un'app connessa per il flusso del codice di autorizzazione. OAuth

Per configurare una connessione a Google Ads:

1. In AWS Secrets Manager, crea un segreto con i seguenti dettagli. È necessario creare un segreto per ogni connessione in AWS Glue.
   a. Per il tipo di AuthorizationCode sovvenzione:

- Per le app connesse gestite dal cliente: il segreto deve contenere l'app connessa Consumer Secret con `USER_MANAGED_CLIENT_APPLICATION_CLIENT_SECRET` come chiave.

2. In AWS Glue Glue Studio, crea una connessione in Connessioni dati seguendo i passaggi seguenti:
   a. Quando selezioni un tipo di connessione, seleziona Google Ads.
   b. Fornisci `developer token` di Google Ads a cui desideri connetterti.
   c. Fornisci i dati `MANAGER ID` di Google Ads se desideri accedere come gestore.
   d. Seleziona il ruolo IAM che AWS Glue può assumere e dispone delle autorizzazioni per le seguenti azioni:

      JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:DescribeSecret",
        "secretsmanager:GetSecretValue",
        "secretsmanager:PutSecretValue",
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DeleteNetworkInterface"
      ],
      "Resource": "*"
    }
  ]
}
```

   e. Seleziona quello `secretName` che desideri utilizzare per questa connessione per AWS Glue inserire i token.
   f. Selezionare le opzioni di rete se si desidera utilizzare la propria rete.
3. Concedi il permesso di lettura `secretName` al ruolo IAM associato al tuo AWS Glue lavoro.

## Lettura da entità di Google Ads

### Prerequisiti

- Un oggetto di Google Ads da cui si desidera leggere. Consulta la tabella delle entità supportate qui di seguito per verificare le entità disponibili.

Entità supportate

| Entità | Può essere Filtrato | Supporta Limite | Supporta Ordina per | Supporta Seleziona * | Supporta il partizionamento |
|---|---|---|---|---|---|
| Annuncio gruppo di annunci | Sì | Sì | Sì | No | Sì |
| Gruppo di annunci | Sì | Sì | Sì | No | Sì |
| Budget campagna | Sì | Sì | Sì | Sì | Sì |
| Budget account | Sì | No | Sì | Sì | No |
| Campagna | Sì | Sì | Sì | Sì | Sì |
| Account | Sì | No | Sì | No | No |

Esempio

```
googleAds_read = glueContext.create_dynamic_frame.from_options(
    connection_type="googleads",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "campaign-3467***",
        "API_VERSION": "v16"
    }
```

Dettagli sull'entità e sul campo di Google Ads

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| Account | resourceName | Stringa | !=, = |
| Account | callReportingEnabled | Booleano | !=, = |
| Account | callConversionReportingAbilitato | Booleano | !=, = |
| Account | callConversionAction | Stringa | !=, = |
| Account | conversionTrackingId | BigInteger | BETWEEN, =, !=, <, >, <=, >= |
| Account | crossAccountConversionTrackingId | BigInteger | BETWEEN, =, !=, <, >, <=, >= |
| Account | payPerConversionEligibilityFailureReasons | List | |
| Account | id | BigInteger | BETWEEN, =, !=, <, >, <=, >= |
| Account | currencyCode | Stringa | !=, =, LIKE |
| Account | timezone | Stringa | !=, =, LIKE |
| Account | autoTaggingEnabled | Booleano | !=, = |
| Account | hasPartnersBadge | Booleano | !=, = |
| Account | manager | Booleano | !=, = |
| Account | testAccount | Booleano | !=, = |
| Account | data | Data | BETWEEN, =, <, >, <=, >= |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| Account | costMicros | BigInteger | BETWEEN, =, !=, <, >, <=, >= |
| Account | acceptedCustomerDataTermini | Booleano | |
| Account | conversionTrackingStatus | Stringa | !=, =, LIKE |
| Account | enhancedConversionsForLeadsEnabled | Booleano | |
| Account | googleAdsConversionCliente | Stringa | |
| Account | status | Stringa | !=, = |
| Account | allConversionsByConversionDate | Double | !=, =, <, > |
| Account | allConversionsValueByConversionDate | Double | !=, =, <, > |
| Account | conversionsByConversionData | Double | !=, =, <, > |
| Account | conversionsValueByConversionDate | Double | !=, =, <, > |
| Account | valuePerAllConversionsByConversionDate | Double | !=, =, <, > |
| Account | videoViews | BigInteger | BETWEEN, =, !=, <, >, <=, >= |
| Account | clic | BigInteger | BETWEEN, =, !=, <, >, <=, >= |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| Account | invalidClicks | BigInteger | BETWEEN, =, !=, <, >, <=, >= |
| Account | costPerAllConversioni | Double | !=, =, <, > |
| Account | costPerConversion | Double | !=, =, <, > |
| Account | conversions | Double | !=, =, <, > |
| Account | absoluteTopImpressionPercentuale | Double | !=, =, <, > |
| Account | impression | BigInteger | BETWEEN, =, !=, <, >, <=, >= |
| Account | topImpressionPercentage | Double | !=, =, <, > |
| Account | averageCpc | Double | !=, =, <, > |
| Account | activeViewMeasurableCostMicros | BigInteger | BETWEEN, =, !=, <, >, <=, >= |
| Account | averageCost | Double | !=, =, <, > |
| Account | ctr | Double | !=, =, <, > |
| Account | activeViewCtr | Double | !=, =, <, > |
| Account | searchImpressionShare | Double | !=, =, <, > |
| Account | conversionAction | Stringa | !=, = |
| Account | conversionActionCategory | Stringa | !=, = |
| Account | conversionActionName | Stringa | !=, =, LIKE |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| Budget account | resourceName | Stringa | !=, = |
| Budget account | status | Stringa | !=, = |
| Budget account | proposedEndTimeTipo | Stringa | !=, = |
| Budget account | approvedEndTimeTipo | Stringa | !=, = |
| Budget account | id | BigInteger | BETWEEN, =, !=, <, >, <=, >= |
| Budget account | billingSetup | Stringa | !=, = |
| Budget account | nome | Stringa | !=, =, LIKE |
| Budget account | approvedStartDateOra | DateTime | BETWEEN, =, <, <=, >, >= |
| Budget account | proposedSpendingLimitMicros | BigInteger | BETWEEN, =, !=, <, >, <=, >= |
| Budget account | approvedSpendingLimitMicros | BigInteger | BETWEEN, =, !=, <, >, <=, >= |
| Budget account | adjustedSpendingLimitMicros | BigInteger | BETWEEN, =, !=, <, >, <=, >= |
| Budget account | amountServedMicros | BigInteger | BETWEEN, =, !=, <, >, <=, >= |
| Gruppo di annunci | resourceName | Stringa | !=, =, LIKE |
| Gruppo di annunci | status | Stringa | !=, =, LIKE |
| Gruppo di annunci | tipo | Stringa | !=, =, LIKE |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| Gruppo di annunci | id | BigInteger | BETWEEN, =, !=, <, >, <=, >= |
| Gruppo di annunci | nome | Stringa | !=, =, LIKE |
| Gruppo di annunci | campaign | Stringa | !=, = |
| Gruppo di annunci | cpcBidMicros | BigInteger | BETWEEN, =, !=, <, >, <=, >= |
| Gruppo di annunci | targetCpaMicros | BigInteger | BETWEEN, =, !=, <, >, <=, >= |
| Gruppo di annunci | cpmBidMicros | BigInteger | BETWEEN, =, !=, <, >, <=, >= |
| Gruppo di annunci | cpvBidMicros | BigInteger | BETWEEN, =, !=, <, >, <=, >= |
| Gruppo di annunci | targetCpmMicros | BigInteger | BETWEEN, =, !=, <, >, <=, >= |
| Gruppo di annunci | effectiveTargetCpaMicros | BigInteger | BETWEEN, =, !=, <, >, <=, >= |
| Gruppo di annunci | data | Data | BETWEEN, =, <, >, <=, >= |
| Gruppo di annunci | costMicros | BigInteger | BETWEEN, =, !=, <, >, <=, >= |
| Gruppo di annunci | useAudienceGrouped | Booleano | !=, = |
| Gruppo di annunci | effectiveCpcBidMicros | BigInteger | BETWEEN, =, !=, <, >, <=, >= |
| Gruppo di annunci | allConversionsByConversionDate | Double | !=, =, <, > |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| Gruppo di annunci | allConversionsValueByConversionDate | Double | !=, =, <, > |
| Gruppo di annunci | conversionsByConversionData | Double | !=, =, <, > |
| Gruppo di annunci | conversionsValueByConversionDate | Double | !=, =, <, > |
| Gruppo di annunci | valuePerAllConversionsByConversionDate | Double | !=, =, <, > |
| Gruppo di annunci | valuePerConversionsByConversionDate | Double | !=, =, <, > |
| Gruppo di annunci | averageCost | Double | !=, =, <, > |
| Gruppo di annunci | costPerAllConversioni | Double | !=, =, <, > |
| Gruppo di annunci | costPerConversion | Double | !=, =, <, > |
| Gruppo di annunci | averagePageViews | Double | !=, =, <, > |
| Gruppo di annunci | videoViews | BigInteger | BETWEEN, =, !=, <, >, <=, >= |
| Gruppo di annunci | clic | BigInteger | BETWEEN, =, !=, <, >, <=, >= |
| Gruppo di annunci | allConversions | Double | !=, =, <, > |
| Gruppo di annunci | averageCpc | Double | !=, =, <, > |
| Gruppo di annunci | absoluteTopImpressionPercentuale | Double | !=, =, <, > |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| Gruppo di annunci | impression | BigInteger | BETWEEN, =, !=, <, >, <=, >= |
| Gruppo di annunci | topImpressionPercentage | Double | !=, =, <, > |
| Gruppo di annunci | activeViewCtr | Double | !=, =, <, > |
| Gruppo di annunci | ctr | Double | !=, =, <, > |
| Gruppo di annunci | searchTopImpressionCondividere | Double | !=, =, <, > |
| Gruppo di annunci | searchImpressionShare | Double | !=, =, <, > |
| Gruppo di annunci | searchAbsoluteTopImpressionShare | Double | !=, =, <, > |
| Gruppo di annunci | relativeCtr | Double | !=, =, <, > |
| Gruppo di annunci | conversionAction | Stringa | !=, = |
| Gruppo di annunci | conversionActionCategory | Stringa | !=, = |
| Gruppo di annunci | conversionActionName | Stringa | !=, =, LIKE |
| Gruppo di annunci | updateMask | Stringa | |
| Gruppo di annunci | creazione | Struct | |
| Gruppo di annunci | aggiorna | Struct | |
| Gruppo di annunci | primaryStatus | Stringa | !=, = |
| Gruppo di annunci | primaryStatusReasons | List | |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| Annuncio gruppo di annunci | resourceName | Stringa | !=, = |
| Annuncio gruppo di annunci | id | BigInteger | BETWEEN, =, !=, <, >, <=, >= |
| Annuncio gruppo di annunci | status | Stringa | !=, = |
| Annuncio gruppo di annunci | labels | List | |
| Annuncio gruppo di annunci | adGroup | Stringa | !=, = |
| Annuncio gruppo di annunci | costMicros | BigInteger | BETWEEN, =, !=, <, >, <=, >= |
| Annuncio gruppo di annunci | approvalStatus | Stringa | !=, = |
| Annuncio gruppo di annunci | reviewStatus | Stringa | !=, = |
| Annuncio gruppo di annunci | adStrength | Stringa | !=, = |
| Annuncio gruppo di annunci | tipo | Stringa | !=, = |
| Annuncio gruppo di annunci | businessName | Stringa | !=, =, LIKE |
| Annuncio gruppo di annunci | data | Data | BETWEEN, =, <, >, <=, >= |
| Annuncio gruppo di annunci | allConversionsByConversionDate | Double | !=, =, <, > |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| Annuncio gruppo di annunci | allConversionsValueByConversionDate | Double | !=, =, <, > |
| Annuncio gruppo di annunci | conversionsByConversionData | Double | !=, =, <, > |
| Annuncio gruppo di annunci | conversionsValueByConversionDate | Double | !=, =, <, > |
| Annuncio gruppo di annunci | valuePerAllConversionsByConversionDate | Double | !=, =, <, > |
| Annuncio gruppo di annunci | valuePerConversionsByConversionDate | Double | !=, =, <, > |
| Annuncio gruppo di annunci | activeViewMeasurableCostMicros | BigInteger | BETWEEN, =, !=, <, >, <=, >= |
| Annuncio gruppo di annunci | averageCost | Double | !=, =, <, > |
| Annuncio gruppo di annunci | costPerAllConversioni | Double | !=, =, <, > |
| Annuncio gruppo di annunci | costPerConversion | Double | !=, =, <, > |
| Annuncio gruppo di annunci | clic | BigInteger | BETWEEN, =, !=, <, >, <=, >= |
| Annuncio gruppo di annunci | averagePageViews | Double | !=, =, <, > |
| Annuncio gruppo di annunci | videoViews | BigInteger | BETWEEN, =, !=, <, >, <=, >= |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| Annuncio gruppo di annunci | allConversions | Double | !=, =, <, > |
| Annuncio gruppo di annunci | averageCpc | Double | !=, =, <, > |
| Annuncio gruppo di annunci | topImpressionPercentage | Double | !=, =, <, > |
| Annuncio gruppo di annunci | impression | BigInteger | BETWEEN, =, !=, <, >, <=, >= |
| Annuncio gruppo di annunci | absoluteTopImpressionPercentuale | Double | !=, =, <, > |
| Annuncio gruppo di annunci | activeViewCtr | Double | !=, =, <, > |
| Annuncio gruppo di annunci | ctr | Double | !=, =, <, > |
| Annuncio gruppo di annunci | conversionAction | Stringa | !=, = |
| Annuncio gruppo di annunci | conversionActionCategory | Stringa | !=, = |
| Annuncio gruppo di annunci | conversionActionName | Stringa | !=, =, LIKE |
| Annuncio gruppo di annunci | updateMask | Stringa | |
| Annuncio gruppo di annunci | creazione | Struct | |
| Annuncio gruppo di annunci | aggiorna | Struct | |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| Annuncio gruppo di annunci | policyValidationParameter | Struct | |
| Annuncio gruppo di annunci | primaryStatus | Stringa | !=, = |
| Annuncio gruppo di annunci | primaryStatusReasons | List | |
| Campagna | resourceName | Stringa | !=, = |
| Campagna | status | Stringa | !=, = |
| Campagna | baseCampaign | Stringa | !=, = |
| Campagna | nome | Stringa | !=, =, LIKE |
| Campagna | id | BigInteger | BETWEEN, =, !=, <, >, <=, >= |
| Campagna | campaignBudget | Stringa | !=, =, LIKE |
| Campagna | startDate | Data | BETWEEN, =, <, >, <=, >= |
| Campagna | endDate | Data | BETWEEN, =, <, >, <=, >= |
| Campagna | adServingOptimizationStato | Stringa | !=, = |
| Campagna | advertisingChannelType | Stringa | !=, = |
| Campagna | advertisingChannelSubTipo | Stringa | !=, = |
| Campagna | experimentType | Stringa | !=, = |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| Campagna | servingStatus | Stringa | !=, = |
| Campagna | biddingStrategyType | Stringa | !=, = |
| Campagna | domainName | Stringa | !=, =, LIKE |
| Campagna | languageCode | Stringa | !=, =, LIKE |
| Campagna | useSuppliedUrlsSolo | Booleano | !=, = |
| Campagna | positiveGeoTargetTipo | Stringa | !=, = |
| Campagna | negativeGeoTargetTipo | Stringa | !=, = |
| Campagna | paymentMode | Stringa | !=, = |
| Campagna | optimizationGoalTypes | List | |
| Campagna | data | Data | BETWEEN, =, <, >, <=, >= |
| Campagna | averageCost | Double | |
| Campagna | clic | BigInteger | BETWEEN, =, !=, <, >, <=, >= |
| Campagna | costMicros | BigInteger | BETWEEN, =, !=, <, >, <=, >= |
| Campagna | impression | BigInteger | BETWEEN, =, !=, <, >, <=, >= |
| Campagna | useAudienceGrouped | Booleano | !=, = |
| Campagna | activeViewMeasurableCostMicros | BigInteger | BETWEEN, =, !=, <, >, <=, >= |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| Campagna | costPerAllConversioni | Double | !=, =, <, > |
| Campagna | costPerConversion | Double | !=, =, <, > |
| Campagna | invalidClicks | BigInteger | BETWEEN, =, !=, <, >, <=, >= |
| Campagna | publisherPurchasedClicks | BigInteger | BETWEEN, =, !=, <, >, <=, >= |
| Campagna | averagePageViews | Double | !=, =, <, > |
| Campagna | videoViews | BigInteger | BETWEEN, =, !=, <, >, <=, >= |
| Campagna | allConversionsByConversionDate | Double | !=, =, <, > |
| Campagna | allConversionsValueByConversionDate | Double | !=, =, <, > |
| Campagna | conversionsByConversionData | Double | !=, =, <, > |
| Campagna | conversionsValueByConversionDate | Double | !=, =, <, > |
| Campagna | valuePerAllConversionsByConversionDate | Double | !=, =, <, > |
| Campagna | valuePerConversionsByConversionDate | Double | !=, =, <, > |
| Campagna | allConversions | Double | !=, =, <, > |
| Campagna | absoluteTopImpressionPercentuale | Double | !=, =, <, > |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| Campagna | searchAbsoluteTopImpressionShare | Double | !=, =, <, > |
| Campagna | averageCpc | Double | !=, =, <, > |
| Campagna | searchImpressionShare | Double | !=, =, <, > |
| Campagna | searchTopImpressionCondividere | Double | !=, =, <, > |
| Campagna | activeViewCtr | Double | !=, =, <, > |
| Campagna | ctr | Double | !=, =, <, > |
| Campagna | relativeCtr | Double | !=, =, <, > |
| Campagna | updateMask | Stringa | |
| Campagna | creazione | Struct | |
| Campagna | aggiorna | Struct | |
| Budget campagna | resourceName | Stringa | !=, = |
| Budget campagna | id | BigInteger | BETWEEN, =, !=, <, >, <=, >= |
| Budget campagna | status | Stringa | !=, = |
| Budget campagna | deliveryMethod | Stringa | !=, = |
| Budget campagna | punto | Stringa | !=, = |
| Budget campagna | tipo | Stringa | !=, = |
| Budget campagna | nome | Stringa | !=, =, LIKE |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| Budget campagna | amountMicros | BigInteger | BETWEEN, =, !=, <, >, <=, >= |
| Budget campagna | explicitlyShared | Booleano | !=, = |
| Budget campagna | referenceCount | BigInteger | BETWEEN, =, !=, <, >, <=, >= |
| Budget campagna | hasRecommendedBudget | Booleano | !=, = |
| Budget campagna | data | Data | BETWEEN, =, <, >, <=, >= |
| Budget campagna | costMicros | BigInteger | BETWEEN, =, !=, <, >, <=, >= |
| Budget campagna | startDate | Data | BETWEEN, =, <, >, <=, >= |
| Budget campagna | endDate | Data | BETWEEN, =, <, >, <=, >= |
| Budget campagna | maximizeConversionValueTargetRoas | Double | !=, =, <, > |
| Budget campagna | maximizeConversionsTargetCpaMicros | BigInteger | BETWEEN, =, !=, <, >, <=, >= |
| Budget campagna | selectiveOptimizationConversionAzioni | Stringa | |
| Budget campagna | averageCost | Double | !=, =, <, > |
| Budget campagna | costPerAllConversioni | Double | !=, =, <, > |
| Budget campagna | costPerConversion | Double | !=, =, <, > |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| Budget campagna | videoViews | BigInteger | BETWEEN, =, !=, <, >, <=, >= |
| Budget campagna | clicks | BigInteger | BETWEEN, =, !=, <, >, <=, >= |
| Budget campagna | allConversions | Double | !=, =, <, > |
| Budget campagna | valuePerAllConversioni | Double | !=, =, <, > |
| Budget campagna | averageCpc | Double | !=, =, <, > |
| Budget campagna | impressions | BigInteger | BETWEEN, =, !=, <, >, <=, >= |
| Budget campagna | ctr | Double | !=, =, <, > |
| Budget campagna | updateMask | Stringa | |
| Budget campagna | creazione | Struct | |
| Budget campagna | aggiorna | Struct | |

Query di partizionamento

Se si desidera utilizzare la simultaneità in Spark, possono essere fornite opzioni Spark `PARTITION_FIELD`, `LOWER_BOUND`, `UPPER_BOUND`, `NUM_PARTITIONS` aggiuntive. Con questi parametri, la query originale viene suddivisa in `NUM_PARTITIONS` query secondarie che possono essere eseguite contemporaneamente dalle attività Spark.

- `PARTITION_FIELD`: il nome del campo da utilizzare per partizionare la query.
- `LOWER_BOUND`: un valore limite inferiore inclusivo del campo di partizione scelto.

  Per la data, accettiamo il formato di data di Spark utilizzato nelle query di Spark SQL. Esempio di valori validi: `"2024-02-06"`.
- `UPPER_BOUND`: un valore limite superiore esclusivo del campo di partizione scelto.

- NUM_PARTITIONS: numero di partizioni.

I dettagli di supporto del campo di partizionamento per entità sono riportati nella tabella seguente.

| Nome entità | Campo di partizionamento | Tipo di dati |
|---|---|---|
| Annuncio gruppo di annunci | data | Data |
| Gruppo di annunci | data | Data |
| Campagna | data | Data |
| Budget campagna | data | Data |

Esempio

```
googleads_read = glueContext.create_dynamic_frame.from_options(
    connection_type="googleads",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "campaign-3467***",
        "API_VERSION": "v16",
        "PARTITION_FIELD": "date"
        "LOWER_BOUND": "2024-01-01"
        "UPPER_BOUND": "2024-06-05"
        "NUM_PARTITIONS": "10"
    }
)
```

## Opzioni di connessione Google Ads

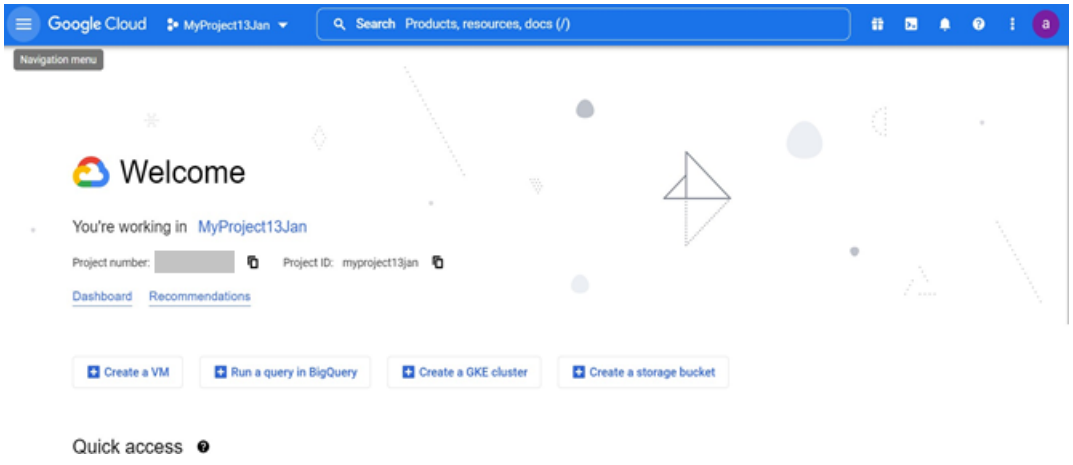
Le seguenti sono le opzioni di connessione per Google Ads:

- ENTITY_NAME(Stringa): (obbligatorio) utilizzato per la lettura/scrittura. Il nome del tuo oggetto in Google Ads.
- API_VERSION(Stringa): (obbligatorio) utilizzato per la lettura/scrittura. La versione dell'API Rest di Google Ads che desideri utilizzare. Esempio: v16.
- DEVELOPER_TOKEN(Stringa): (obbligatorio) utilizzato per la lettura/scrittura. Necessario per autenticare lo sviluppatore o l'applicazione che effettua richieste all'API.

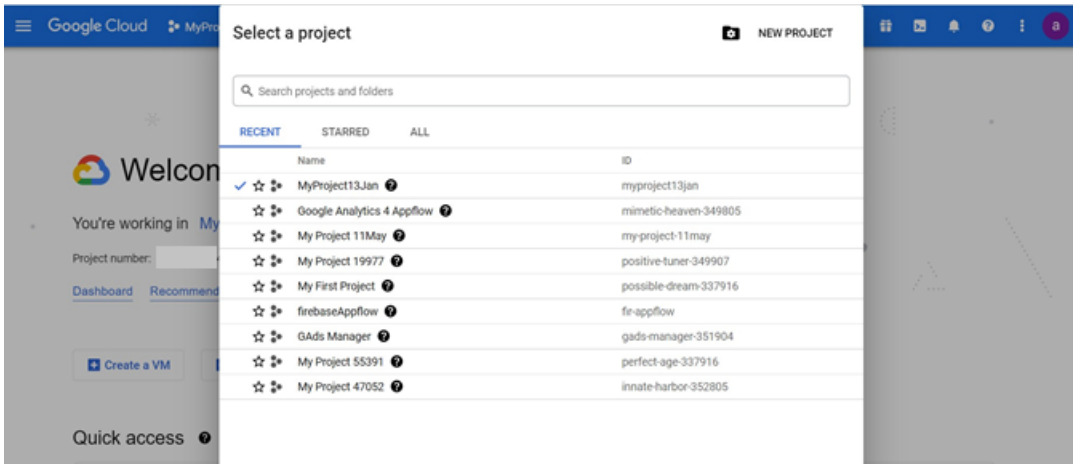
- `MANAGER_ID`(Stringa): utilizzato per la lettura/scrittura. Un identificatore univoco che ti consente di gestire più account Google Ads. Questo è l'ID cliente del gestore autorizzato. Se l'accesso all'account cliente avviene tramite un account amministratore, `MANAGER_ID` è necessario. Per ulteriori informazioni, consulta login-customer-id.
- `SELECTED_FIELDS`(List<String>): predefinito: vuoto (SELECT *). Utilizzato per la lettura. Colonne da selezionare per l'oggetto.
- `FILTER_PREDICATE`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Dev'essere in formato Spark SQL.
- `QUERY`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Query Spark SQL completa.
- `PARTITION_FIELD`(Stringa): utilizzato per la lettura. Campo da utilizzare per partizionare la query.
- `LOWER_BOUND`(Stringa): utilizzato per la lettura. Un valore limite inferiore inclusivo del campo di partizione scelto.
- `UPPER_BOUND`(Stringa): utilizzato per la lettura. Un valore limite superiore esclusivo del campo di partizione scelto.
- `NUM_PARTITIONS`(Numero intero): predefinito: 1. Utilizzato per la lettura. Numero di partizioni da leggere.

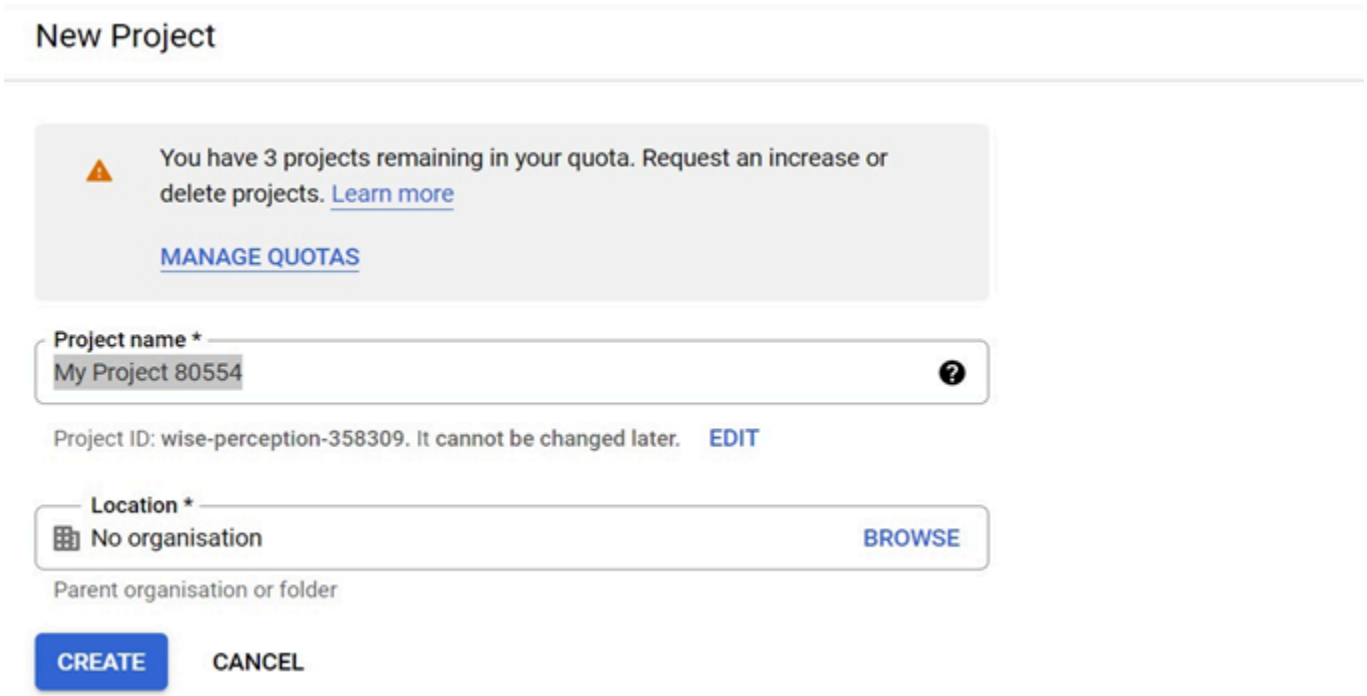
## Creazione di un account Google Ads

1. Accedi all'account sviluppatore Google Ads con le tue credenziali e vai a*MyProject.

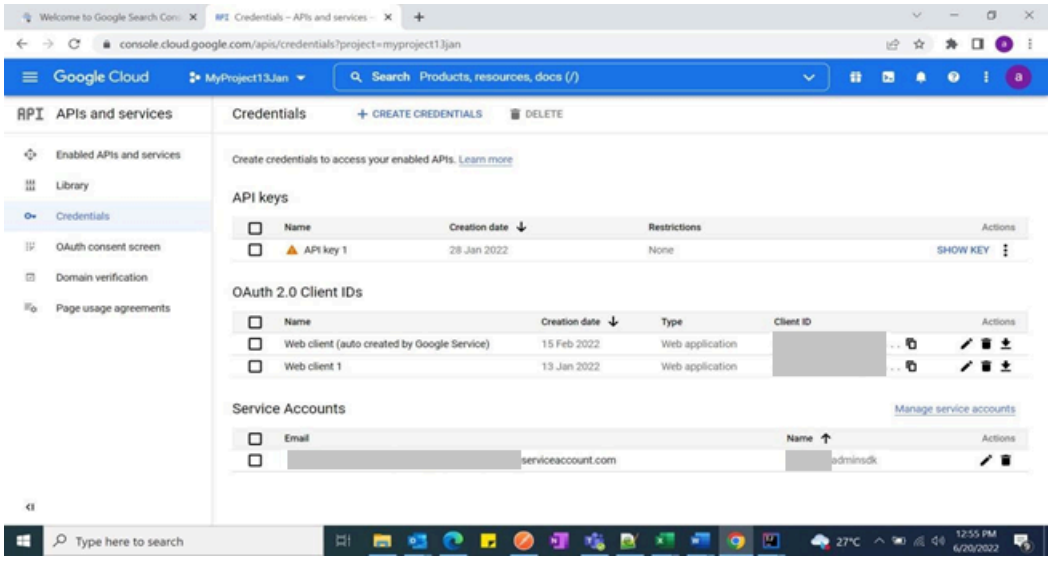



2. Scegli Nuovo progetto e fornisci le informazioni necessarie per creare il progetto Google se non hai alcuna applicazione registrata.

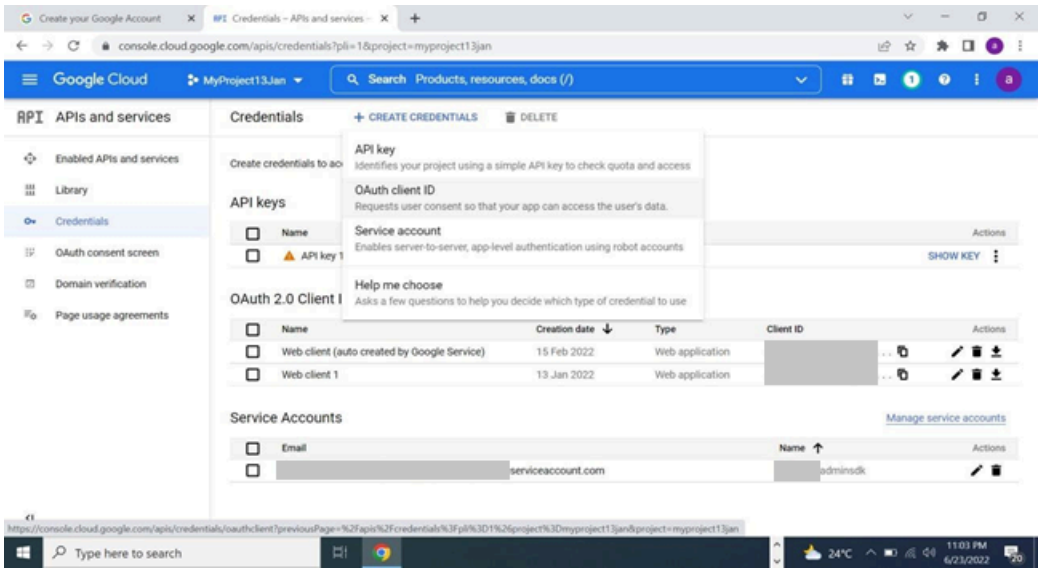


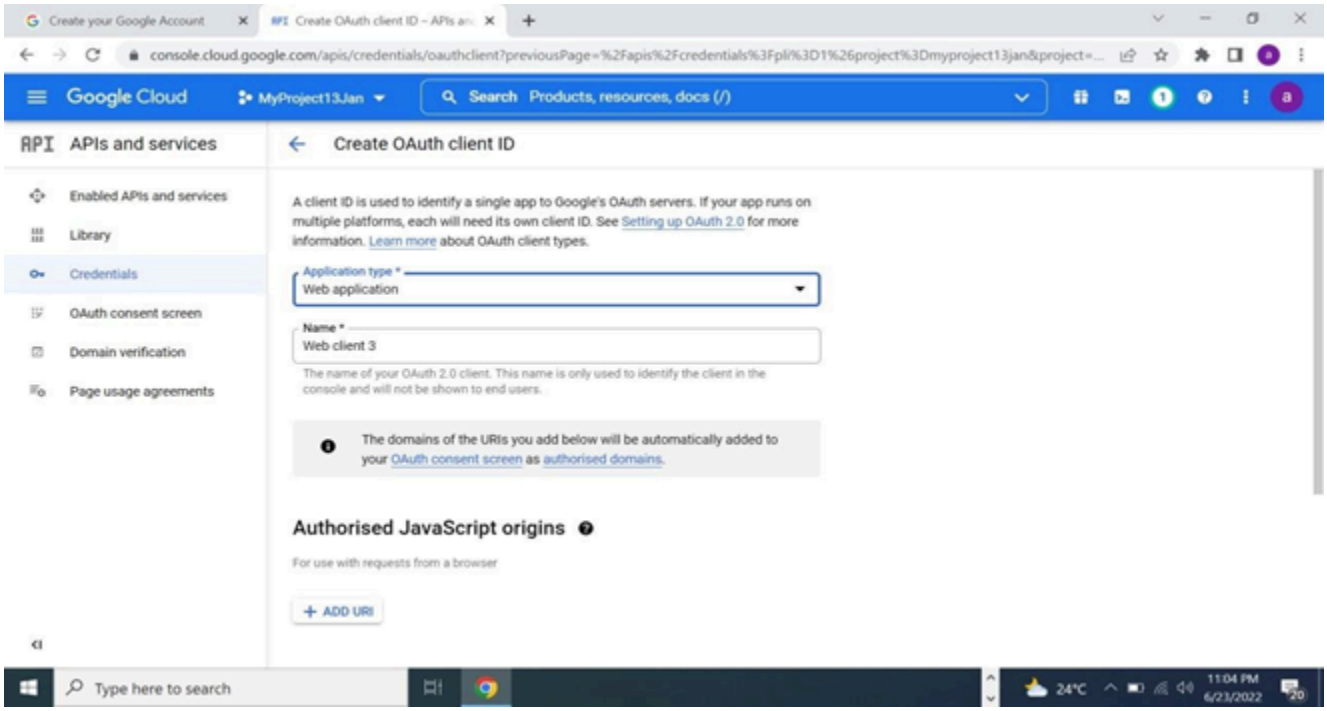

3. Scegli la scheda Navigazione, quindi API e impostazioni e Crea ID cliente. Questa operazione ClientSecretrichiederà un'ulteriore configurazione per la creazione di una connessione tra AWS Glue e GoogleAds. Per ulteriori informazioni, consulta Credenziali API.



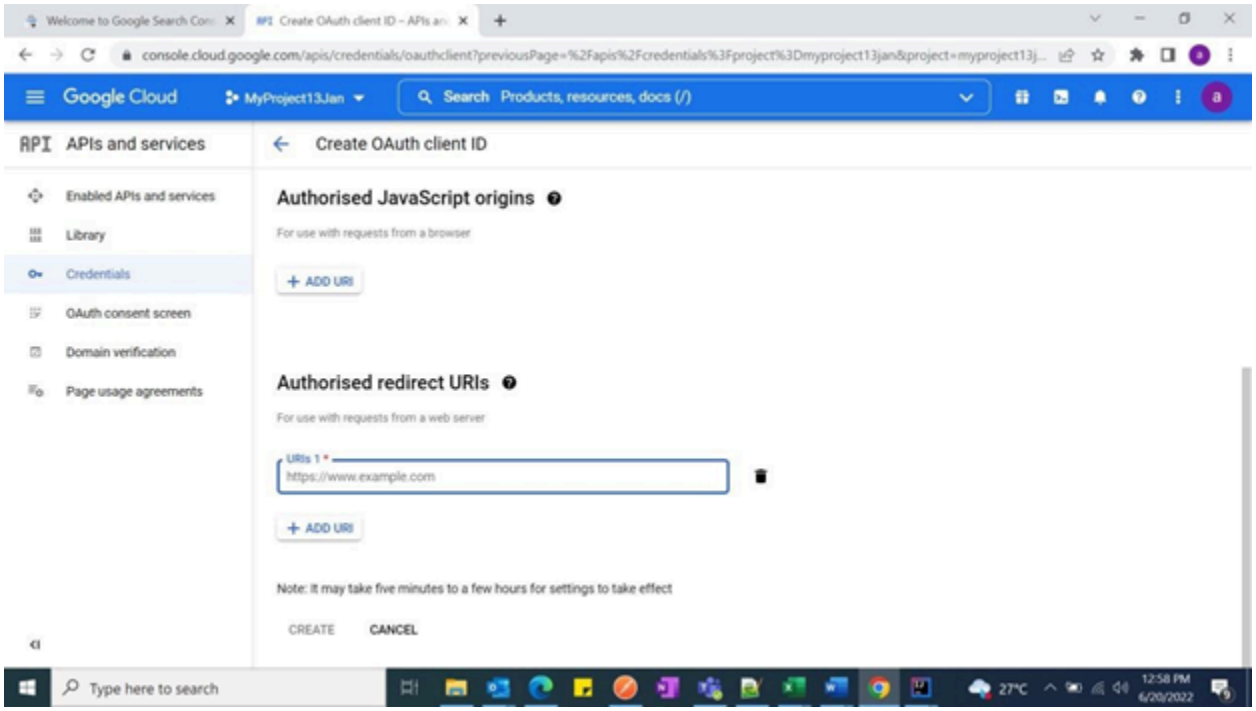

4. Scegli CREATE CREDENTIALS e scegli l'ID OAuth cliente.



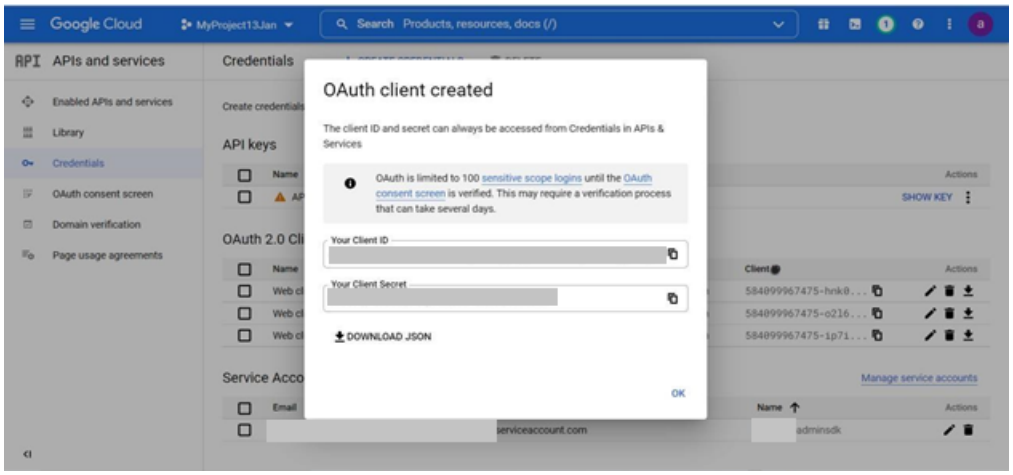

5. Seleziona il Tipo di applicazione come Applicazione web.



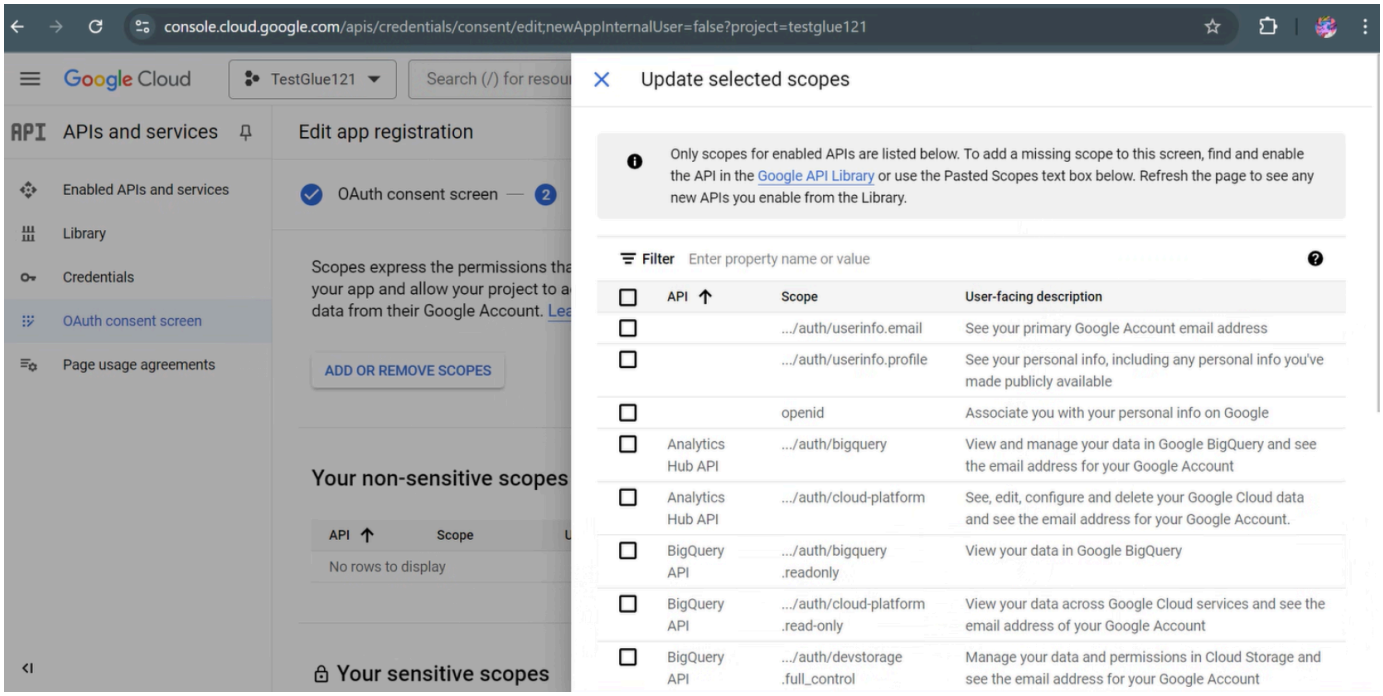

6. In Reindirizzamento autorizzato URIs, aggiungi il OAuth reindirizzamento URIs e scegli Crea. Se necessario, puoi aggiungere più reindirizzamenti URIs .




7. L'ID cliente e il segreto del cliente verranno generati durante la creazione di una connessione tra AWS Glue e Google Ads.




8. Aggiungi gli ambiti in base alle esigenze della tua applicazione, scegli la schermata di OAuth consenso e fornisci le informazioni richieste e aggiungi gli ambiti in base ai requisiti.




## Limitazioni

Di seguito sono riportate le limitazioni per il connettore di Google Ads.

- `MANAGER_ID` è un input opzionale per la creazione di una connessione. Ma quando si desidera accedere ai clienti sottostanti a un particolare gestore, allora `MANAGER_ID` è un input obbligatorio. La tabella seguente spiega le limitazioni di accesso in base al fatto che `MANAGER_ID` sia incluso o meno in una connessione.

| MANAGER_ID fornito durante la creazione della connessione? | Clienti accessibili |
| --- | --- |
| Sì | I clienti elencati sotto il MANAGER_ID fornito insieme al MANAGER_ID. |
| No | Verranno elencati tutti i clienti, ma i clienti sottostanti a qualsiasi gestore non saranno accessibili. |

- Quando si sceglie un account amministratore come oggetto, `Account` verrà visualizzato solo come oggetto secondario. Nel connettore Google Ads, entità come campagne, annunci e così via vengono recuperate in base ai singoli account cliente, non all'account amministratore.
- Non puoi recuperare le metriche per l'account amministratore. Puoi invece recuperare le metriche per i singoli account cliente.
- Ogni account può avere fino a 10.000 campagne, comprese quelle attive e quelle in pausa. Per ulteriori informazioni, consulta l'argomento relativo alle campagne per account.
- Quando crei un rapporto, se scegli determinate metriche da visualizzare, le righe le cui metriche selezionate sono tutte pari a zero non verranno restituite. Per ulteriori informazioni, consulta Zero Metrics .
- Con i seguenti campi, il flusso di mappatura completa non funzionerà per le entità Account, Ad Group e Ad Group Ad, in particolare per ConversionAction,. conversionActionCategory conversionActionName Per ulteriori informazioni, consulta Segmenti e metriche .
- Un filtro per intervallo di date è obbligatorio quando il campo `segments.date` è selezionato.

## Connessione a Google Analytics 4

Google Analytics 4 è un servizio di analisi che monitora e riporta le metriche relative alle interazioni dei visitatori con le tue app e i tuoi siti web. Queste metriche includono le visualizzazioni di pagina, gli utenti attivi e gli eventi. Se sei un utente di Google Analytics 4, puoi connetterti AWS Glue al tuo account Google Analytics 4. Puoi usare Google Analytics 4 come origine dati nei tuoi processi ETL. Esegui questi processi per trasferire dati da Google Analytics 4 ai AWS servizi o ad altre applicazioni supportate.

Argomenti

- AWS Glue supporto per Google Analytics 4
- Policy contenenti le operazioni API per creare e utilizzare le connessioni
- Configurazione di Google Analytics 4
- Configurazione delle connessioni di Google Analytics 4
- Lettura dalle entità di Google Analytics 4
- Opzioni di connessione di Google Analytics 4
- Creazione di un account Google Analytics 4
- Passaggi per creare un'app client e credenziali OAuth 2.0
- Considerazioni e limitazioni


## AWS Glue supporto per Google Analytics 4

AWS Glue supporta Google Analytics 4 come segue:

Supportato come origine?

Sì. Puoi utilizzare i job AWS Glue ETL per interrogare i dati di Google Analytics 4.

Supportato come destinazione?

No.

Versioni dell'API di Google Analytics 4 supportate

v1 Beta.

## Policy contenenti le operazioni API per creare e utilizzare le connessioni

La seguente politica di esempio descrive le AWS autorizzazioni richieste per la creazione e l'utilizzo delle connessioni. Se si sta creando un nuovo ruolo, creare una policy che contenga quanto segue:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
```

```
      "Action": [
        "glue:ListConnectionTypes",
        "glue:DescribeConnectionType",
        "glue:RefreshOAuth2Tokens",
        "glue:ListEntities",
        "glue:DescribeEntity"
      ],
      "Resource": "*"
    }
  ]
}
```

È anche possibile utilizzare le seguenti policy IAM per autorizzare l'accesso:

- AWSGlueServiceRole— Garantisce l'accesso alle risorse che diversi AWS Glue processi richiedono per l'esecuzione per conto dell'utente. Queste risorse includono Amazon S3 AWS Glue, IAM, CloudWatch Logs e Amazon EC2. Se segui la convenzione di denominazione delle risorse specificata in questa politica, AWS Glue i processi dispongono delle autorizzazioni necessarie. Questa policy è in genere collegata ai ruoli specificati quando si definiscono crawler, processi ed endpoint di sviluppo.
- AWSGlueConsoleFullAccess— Concede l'accesso completo alle AWS Glue risorse quando un'identità a cui è allegata la policy utilizza la AWS console di gestione. Se si segue la convenzione per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questa policy è in genere associata agli utenti della AWS Glue console.

## Configurazione di Google Analytics 4

Prima di poter AWS Glue utilizzare il trasferimento da Google Analytics 4, devi soddisfare questi requisiti:

Requisiti minimi

- Hai un account Google Analytics con uno o più flussi di dati che raccolgono i dati che desideri trasferire.
- Hai un account Google Cloud Platform e un progetto Google Cloud.
- Nel tuo progetto Google Cloud, hai abilitato quanto segue APIs:
  - API Google Analytics
  - API di amministrazione di Google Analytics

  - API di dati di Google Analytics
- Nel tuo progetto Google Cloud, hai configurato una schermata di OAuth consenso per gli utenti esterni. Per informazioni sulla schermata di OAuth consenso, consulta Configurazione della schermata di OAuth consenso nella guida della console di Google Cloud Platform.
- Nel tuo progetto Google Cloud, hai configurato un ID client OAuth 2.0. Per ulteriori informazioni, consulta Configurazione OAuth 2.0.

Se soddisfi questi requisiti, sei pronto per connetterti AWS Glue al tuo account Google Analytics 4.

## Configurazione delle connessioni di Google Analytics 4

Per configurare una connessione a Google Sheet:

1. In AWS Secrets Manager, crea un segreto con i seguenti dettagli. È necessario creare un segreto per ogni connessione in AWS Glue.
   a. Per il tipo di AuthorizationCode sovvenzione:
      - Per le app connesse gestite dal cliente: il segreto deve contenere l'app connessa Consumer Secret con `USER_MANAGED_CLIENT_APPLICATION_CLIENT_SECRET` come chiave.
2. In AWS Glue Glue Studio, crea una connessione in Connessioni dati seguendo i passaggi seguenti:
   a. Quando selezioni un tipo di connessione, seleziona Google Analytics 4.
   b. Fornisci `INSTANCE_URL` di Google Analytics 4 a cui desideri connetterti.
   c. Seleziona il ruolo IAM che AWS Glue può assumere e dispone delle autorizzazioni per le seguenti azioni:

      JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:DescribeSecret",
        "secretsmanager:GetSecretValue",
        "secretsmanager:PutSecretValue",
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces",
```

```
        "ec2:DeleteNetworkInterface"
      ],
      "Resource": "*"
    }
  ]
}
```

d. Seleziona quello `secretName` che desideri utilizzare per questa connessione per AWS Glue inserire i token.

e. Selezionare le opzioni di rete se si desidera utilizzare la propria rete.

3. Concedi il permesso di lettura `secretName` al ruolo IAM associato al tuo AWS Glue lavoro.

Tipo di concessione `AUTHORIZATION_CODE`.

Questo tipo di concessione è considerato «a tre gambe» in OAuth quanto si basa sul reindirizzamento degli utenti al server di autorizzazione di terze parti per autenticare l'utente. Viene utilizzato durante la creazione di connessioni tramite la console. AWS Glue La AWS Glue Console reindirizzerà l'utente a Google Analytics 4, dove l'utente deve effettuare il login e consentire AWS Glue le autorizzazioni richieste per accedere alla propria istanza di Google Analytics 4.

Gli utenti possono comunque scegliere di creare la propria app connessa in Google Analytics 4 e fornire il proprio ID client e il segreto del client quando creano connessioni tramite la AWS Glue Console. In questo scenario, verranno comunque reindirizzati a Google Analytics 4 per accedere e autorizzare l'accesso AWS Glue alle proprie risorse.

Questo tipo di concessione genera un token di aggiornamento e un token di accesso. Il token di accesso dura poco tempo e può essere aggiornato automaticamente senza l'interazione dell'utente utilizzando il token di aggiornamento.

Per ulteriori informazioni, consulta Utilizzo di Auth 2.0 per accedere a Google. APIs

## Lettura dalle entità di Google Analytics 4

Prerequisiti

- Un oggetto di Google Analytics 4 da cui si desidera leggere. Consulta la tabella delle entità supportate qui di seguito per verificare le entità disponibili.

Entità supportate

| Entità | Può essere Filtrato | Supporta Limite | Supporta Ordina per | Supporta Seleziona * | Supporta il partizionamento |
|---|---|---|---|---|---|
| Real-Time Report | Sì | Sì | Sì | Sì | No |
| Core Report | Sì | Sì | Sì | Sì | Sì |

Esempio

```
googleAnalytics4_read = glueContext.create_dynamic_frame.from_options(
    connection_type="GoogleAnalytics4",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "entityName",
        "API_VERSION": "v1beta"
    }
```

Dettagli sull'entità e sul campo di Google Analytics 4

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| Core Report | Campi dinamici | | |
| Core Report | Campi dimensionali | Stringa | LIKE, = |
| Core Report | Campi dimensionali | Data | LIKE, = |
| Core Report | Campi metrici | Stringa | >, <, >=, <=, = BETWEEN |
| Core Report | Dimensioni personalizzate e campi metrici personalizzati | Stringa | N/A |
| Real-Time Report | appVersion | Stringa | LIKE, = |
| Real-Time Report | audienceId | Stringa | LIKE, = |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| Real-Time Report | audienceName | Stringa | LIKE, = |
| Real-Time Report | città | Stringa | LIKE, = |
| Real-Time Report | cityId | Stringa | LIKE, = |
| Real-Time Report | paese | Stringa | LIKE, = |
| Real-Time Report | countryId | Stringa | LIKE, = |
| Real-Time Report | deviceCategory | Stringa | LIKE, = |
| Real-Time Report | eventName | Stringa | LIKE, = |
| Real-Time Report | minutesAgo | Stringa | LIKE, = |
| Real-Time Report | platform | Stringa | LIKE, = |
| Real-Time Report | streamId | Stringa | LIKE, = |
| Real-Time Report | streamName | Stringa | LIKE, = |
| Real-Time Report | unifiedScreenName | Stringa | LIKE, = |
| Real-Time Report | activeUsers | Stringa | >, <, >=, <=, = BETWEEN |
| Real-Time Report | conversions | Stringa | >, <, >=, <=, = BETWEEN |
| Real-Time Report | eventCount | Stringa | >, <, >=, <=, = BETWEEN |
| Real-Time Report | screenPageViews | Stringa | >, <, >=, <=, = BETWEEN |

Query di partizionamento

1. Partizione basata su filtri

Se si desidera utilizzare la simultaneità in Spark, possono essere fornite opzioni Spark PARTITION_FIELD, LOWER_BOUND, UPPER_BOUND, NUM_PARTITIONS aggiuntive. Con questi parametri, la query originale viene suddivisa in NUM_PARTITIONS query secondarie che possono essere eseguite contemporaneamente dalle attività Spark.

- PARTITION_FIELD: il nome del campo da utilizzare per partizionare la query.
- LOWER_BOUND: un valore limite inferiore inclusivo del campo di partizione scelto.

  Per la data, accettiamo il formato di data di Spark utilizzato nelle query di Spark SQL. Esempio di valori validi: "2024-02-06".
- UPPER_BOUND: un valore limite superiore esclusivo del campo di partizione scelto.
- NUM_PARTITIONS: numero di partizioni.

Esempio

```
googleAnalytics4_read = glueContext.create_dynamic_frame.from_options(
    connection_type="GoogleAnalytics4",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "entityName",
        "API_VERSION": "v1beta",
        "PARTITION_FIELD": "date"
        "LOWER_BOUND": "2022-01-01"
        "UPPER_BOUND": "2024-01-02"
        "NUM_PARTITIONS": "10"
    }
```

2. Partizione basata su record

Se desideri utilizzare la concorrenza in Spark, possono essere fornite le opzioni aggiuntive NUM_PARTITIONS. Con questi parametri, la query originale viene suddivisa in NUM_PARTITIONS query secondarie che possono essere eseguite contemporaneamente dalle attività Spark.

- NUM_PARTITIONS: numero di partizioni.

Esempio

```
googleAnalytics4_read = glueContext.create_dynamic_frame.from_options(
    connection_type="GoogleAnalytics4",
    connection_options={
        "connectionName": "connectionName",
```

```
        "ENTITY_NAME": "entityName",
        "API_VERSION": "v1beta",
        "NUM_PARTITIONS": "10"
    }
```

## Opzioni di connessione di Google Analytics 4

Le seguenti sono le opzioni di connessione per Google Analytics 4:

- `ENTITY_NAME`(String): (obbligatorio) utilizzato per la lettura. Il nome del tuo oggetto in Google Analytics 4.
- `API_VERSION`(String): (obbligatorio) utilizzato per la lettura. La versione dell'API Rest di Google Analytics 4 che desideri utilizzare.
- `SELECTED_FIELDS`(List<String>): predefinito: vuoto (SELECT *). Utilizzato per la lettura. Colonne da selezionare per l'oggetto.
- `FILTER_PREDICATE`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Dev'essere in formato Spark SQL.
- `QUERY`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Query Spark SQL completa.
- `PARTITION_FIELD`(Stringa): utilizzato per la lettura. Campo da utilizzare per partizionare la query.
- `LOWER_BOUND`(Stringa): utilizzato per la lettura. Un valore limite inferiore inclusivo del campo di partizione scelto.
- `UPPER_BOUND`(Stringa): utilizzato per la lettura. Un valore limite superiore esclusivo del campo di partizione scelto.
- `NUM_PARTITIONS`(Numero intero): predefinito: 1. Utilizzato per la lettura. Numero di partizioni da leggere.
- `INSTANCE_URL`(Numero intero): utilizzato per la lettura. (Facoltativo)

## Creazione di un account Google Analytics 4

Segui i passaggi per creare un account Google Analytics 4: https://support.google.com/analytics/answer/9304153? hl=it

## Passaggi per creare un'app client e credenziali OAuth 2.0

Per ulteriori informazioni, consulta la documentazione dell'API di Google Analytics4.

1. Crea e configura il tuo account accedendo all'account di Google Analytics con le tue credenziali. Quindi, vai su Amministratore > Crea account.
2. Crea una proprietà per l'account che hai creato scegliendo Crea proprietà. Imposta la proprietà con i dettagli richiesti. Una volta forniti tutti i dettagli, verrà generato l'ID della proprietà corrispondente.
3. Aggiungi il flusso di dati per la proprietà creata scegliendo Flussi di dati > Aggiungi flusso > Web dal menu a discesa. Fornisci i dettagli del sito Web come l'URL e altri campi obbligatori. Dopo aver fornito tutti i dettagli, verranno generati l'ID dello stream e l'ID di misurazione corrispondenti.
4. Configura Google Analytics sul tuo sito Web copiando l'ID di misurazione e aggiungendolo alla configurazione del tuo sito Web.
5. Crea un report da Google Analytics accedendo a Report e generando il report richiesto.
6. Autorizza la tua app accedendo a console.cloud.google.com e cerca l'API dei dati di Google Analytics, quindi abilitala.
    1. Vai alla pagina API e servizi e scegli Credenziali > setup OAuth 2.0 Client. IDs
    2. Fornisci l'URL di reindirizzamento aggiungendo l'URL di AWS Glue reindirizzamento.
7. Copia l'ID client e il segreto client, che richiederanno ulteriori informazioni per la creazione della connessione.

## Considerazioni e limitazioni

Di seguito sono riportate le limitazioni per il connettore di Google Analytics 4.

- Per l'entità Core Report, è consentito inviare solo 9 campi dimensionali e 10 campi metrici in una richiesta. Se viene superato il numero consentito di campi, la richiesta avrà esito negativo e il connettore genererà un messaggio di errore.
- Per l'entità Real-Time Report, è consentito inviare solo 4 campi dimensionali in una richiesta. Se viene superato il numero consentito di campi, la richiesta avrà esito negativo e il connettore genererà un messaggio di errore.
- Google Analytics 4 è uno strumento gratuito in versione beta, quindi ci saranno aggiornamenti regolari sulle nuove funzionalità, sul miglioramento delle entità, sull'aggiunta di nuovi campi e sulla deprecazione dei campi esistenti.
- I campi Core Report vengono compilati in maniera dinamica, quindi è possibile aggiungere, deprecare e rinominare i campi e imporre loro nuovi limiti in qualsiasi momento.

- La data di inizio predefinita è 30 giorni e la data di fine è ieri (un giorno prima della data corrente); queste date verranno sostituite nel codice di espressione del filtro se l'utente ha impostato il valore O se il flusso è incrementale.
- Secondo la documentazione, l'entità di report in tempo reale restituisce 10.000 record se il limite non viene superato nella richiesta, altrimenti l'API restituisce un massimo di 250.000 righe per richiesta, indipendentemente dal numero richiesto. Per ulteriori informazioni, vedere Metodo: proprietà. runRealtimeReportnella documentazione di Google Analytics.
- L'entità Real-Time Report non supporta la partizione basata su record in quanto non supporta l'impaginazione. Inoltre, non supporta la partizione basata sul campo poiché nessuno dei campi soddisfa i criteri definiti.
- A causa della limitazione del numero di campi che possono essere trasferiti in una richiesta. Stiamo impostando i campi dimensionali e metrici predefiniti entro i limiti designati. Se si sceglie "seleziona tutto", verranno recuperati solo i dati di quei campi predeterminati.
  - Core Report
    - In base alle limitazioni di SAAS, le richieste sono consentite solo fino a 9 dimensioni e fino a 10 metriche. In altre parole, una richiesta può contenere un massimo di 19 campi (metriche + dimensione).
    - Riguardo all'implementazione: se l'utente utilizza SELECT_ALL o se seleziona più di 25 campi, i campi predefiniti verranno passati nella richiesta.
    - I seguenti campi sono considerati campi predefiniti per Core Report: «country», «city», «eventName», «cityID», «browser», «date», «currencyCode», «deviceCategory», «transactionID», active1 «, «active28», «active7», «activeUsers», "«," User», "DayUsers«, «EngagedSessions», DayUsers «EventCount», DayUsers «EngagementRate». averagePurchaseRevenue averageRevenuePer averageSessionDuration
  - Real-Time Report
    - Come da limitazione, le richieste SAAS sono consentite fino a 4 dimensioni.
    - Se l'utente passa SELECT_ALL o se seleziona più di 15 campi, i campi predefiniti verranno passati nella richiesta.
    - I seguenti campi sono considerati campi predefiniti per RealTime Report: «country», «deviceCategory», «city», «cityID», «activeUsers», «conversions», «eventCount», "screenPageViews».
- Nell'entità Core Report, se la partizione sul campo data e il filtro su startDate sono presenti contemporaneamente. In tal caso il valore dateRange viene sovrascritto con il valore del filtro

startDate; tuttavia, poiché la partizione deve sempre essere la priorità, si elimina il filtro startDate se la partizione nel campo della data è già presente.

- Poiché ora anche cohortSpecs fa parte del corpo di richiesta di Core Report, abbiamo migliorato l'attuale entità Core Report per includere il supporto per l'attributo cohortSpec. Nel corpo della richiesta cohortSpecs, quasi tutti i campi richiedono l'input dell'utente. Per risolvere questo problema, abbiamo impostato valori predefiniti per questi valori attributes/fields e fornito all'utente la possibilità di sovrascrivere questi valori, se necessario.

| FieldName | Valori predefiniti | Query di esempio per passare alle opzioni filterPredicate per sovrascrivere i valori predefiniti |
|---|---|---|
| startDate | 30 giorni dalla data corrente | "startDate compresa tra "09/05/2023" e "10/05/2023" |
| endDate | 1 giorno fa dalla data corrente | "startDate compresa tra "09/05/2023" e "10/05/2023" |
| startOffset | 0 | startOffset=2 |
| endOffset | 1 | endOffset=10 |
| Granularità | GIORNALIERO | granularity= "SETTIMANALE" |

- Puoi anche passare tutti questi filtri insieme o con altri filtri.
  - Esempio 1: filterPredicate: startDate compresa tra "09/05/2023" e "10/05/2023" E startOffset=1 E endOffset=2 E granularity="SETTIMANALE"
  - Esempio 2: filterPredicate: city="xyz" E startOffset=1 E endOffset=2 E granularity="SETTIMANALE"
- Nella richiesta di coorte:
  - Se nella richiesta viene inserito cohortNthMonth '', internamente il valore di granularità verrà impostato come «MENSILE»
  - Allo stesso modo, se viene passato cohortNthWeek '', il valore di granularità verrà impostato come «WEEKLY»
  - Inoltre, per 'cohortNthDay' il valore di granularità sarà impostato come «GIORNALIERO». Per ulteriori informazioni, consulta:

- https://developers.google.com/analytics/devguides/reporting/data/v1/advanced
- https://developers.google.com/analytics/devguides/reporting/data/v1/rest/v1beta/CohortSpec

- È prevista la possibilità per l'utente di sovrascrivere il valore predefinito di dateRange e di granularità. Fare riferimento alla tabella precedente.

## Connessione a Google BigQuery in AWS Glue Studio

> **Note**
>
> Puoi utilizzare Spark AWS Glue per leggere e scrivere su tabelle in Google BigQuery nella AWS Glue versione 4.0 e successive. Per configurare Google BigQuery con i AWS Glue lavori a livello di codice, consulta BigQuery connessioni.

AWS Glue Studiofornisce un'interfaccia visiva a cui connettersi BigQuery, creare lavori di integrazione dei dati ed eseguirli nel runtime Spark AWS Glue Studio senza server.

Quando si crea una connessione a Google BigQuery in AWS Glue Studio, viene creata una connessione unificata. Per ulteriori informazioni, consulta Considerazioni.

Invece di creare un segreto con le credenziali in un formato specifico`{"credentials": "base64 encoded JSON"}`, ora con connessione unificata a Google BigQuery, puoi creare un segreto che include direttamente il JSON di Google:. BigQuery `{"type": "service-account", ...}`

Argomenti

- Creazione di una BigQuery connessione
- Creazione di un nodo BigQuery sorgente
- Creazione di un nodo BigQuery di destinazione
- Opzioni avanzate

### Creazione di una BigQuery connessione

Per connetterti a Google BigQuery da AWS Glue, dovrai creare e archiviare le tue credenziali di Google Cloud Platform in modo Gestione dei segreti AWS segreto, quindi associare tale segreto a una BigQuery AWS Glue connessione Google.

Per configurare una connessione a BigQuery:

1. In Google Cloud Platform, crea e identifica le risorse pertinenti:

   - Crea o identifica un progetto GCP contenente BigQuery le tabelle a cui desideri connetterti.
   - Abilita l' BigQuery API. Per ulteriori informazioni, consulta Utilizzare l'API BigQuery Storage Read per leggere i dati delle tabelle.

2. In Google Cloud Platform, crea ed esporta le credenziali dell'account del servizio:

   È possibile utilizzare la procedura guidata per le BigQuery credenziali per velocizzare questo passaggio: Creazione di credenziali.

   Per creare un account di servizio in GCP, segui il tutorial disponibile in Creazione di account di servizio.

   - Quando selezionate il progetto, selezionate il progetto contenente la tabella. BigQuery
   - Quando selezioni i ruoli GCP IAM per il tuo account di servizio, aggiungi o crea un ruolo che conceda le autorizzazioni appropriate per eseguire BigQuery lavori di lettura, scrittura o creazione BigQuery di tabelle.

   Per creare le credenziali per il tuo account di servizio, segui il tutorial disponibile in Creazione della chiave di un account di servizio.

   - Quando selezioni il tipo di chiave, seleziona JSON.

   Ora dovresti avere scaricato un file JSON con le credenziali per il tuo account di servizio. La schermata visualizzata dovrebbe risultare simile a quella nell'immagine seguente:

```
{
  "type": "service_account",
  "project_id": "*****",
  "private_key_id": "*****",
  "private_key": "*****",
  "client_email": "*****",
  "client_id": "*****",
  "auth_uri": "https://accounts.google.com/o/oauth2/auth",
  "token_uri": "https://oauth2.googleapis.com/token",
  "auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",
  "client_x509_cert_url": "*****",
```

```
  "universe_domain": "googleapis.com"
}
```

3. In Gestione dei segreti AWS, crea un segreto utilizzando il file di credenziali scaricato. Puoi scegliere la scheda Plaintext e incollare il contenuto del file in formato JSON. Per creare un segreto in Secrets Manager, segui il tutorial disponibile in Crea un Gestione dei segreti AWS segreto nella Gestione dei segreti AWS documentazione. Dopo aver creato il segreto, mantieni il nome del segreto *secretName* per il passaggio successivo.
4. Nel AWS Glue Data Catalog, crea una connessione seguendo la procedura riportata di seguitohttps://docs.aws.amazon.com/glue/latest/dg/console-connections.html. Dopo aver creato la connessione, mantieni il nome della connessione per il passaggio successivo. *connectionName*
   - Quando selezioni un tipo di connessione, seleziona Google BigQuery.
   - Quando selezioni un AWS segreto, fornisci*secretName*.
5. Concedi al ruolo IAM associato al tuo AWS Glue lavoro il permesso di lettura*secretName*.
6. Nella configurazione del tuo AWS Glue lavoro, fornisci *connectionName* una connessione di rete aggiuntiva.

## Creazione di un nodo BigQuery sorgente

### Prerequisiti necessari

- Una connessione BigQuery di tipo AWS Glue Data Catalog
- Un Gestione dei segreti AWS segreto per le tue BigQuery credenziali Google, utilizzate dalla connessione.
- Autorizzazioni appropriate sul processo per leggere il segreto utilizzato dalla connessione.
- Il nome e il set di dati della tabella e del progetto Google Cloud corrispondente che si desidera leggere.

### Aggiungere una fonte di BigQuery dati

Per aggiungere una fonte di dati — BigQuery nodo:

1. Scegli la connessione per la tua fonte di BigQuery dati. Dato che l'hai creato, dovrebbe essere disponibile nel menu a discesa. Se devi creare una connessione, scegli Crea BigQuery

connessione. Per ulteriori informazioni, consulta la pagina Overview of using connectors and connections.

Dopo aver scelto una connessione, puoi visualizzare le proprietà della connessione facendo clic su Visualizza proprietà.

2. Identifica BigQuery i dati che desideri leggere, quindi scegli un'opzione BigQuery Sorgente
   - Scegli una singola tabella: ti consente di estrarre tutti i dati da una tabella.
   - Inserisci una query personalizzata: ti consente di personalizzare i dati recuperati fornendo una query.
3. Descrivi i dati che desideri leggere

   (Obbligatorio) imposta il Progetto padre sul progetto contenente la tabella o su un progetto padre di fatturazione, se pertinente.

   Se hai scelto una tabella singola, imposta Table sul nome di una BigQuery tabella Google nel seguente formato: `[dataset].[table]`

   Se hai scelto una query, forniscila a Query. Nella tua query, fai riferimento alle tabelle con il loro nome di tabella completo, nel formato: `[project].[dataset].[tableName]`.
4. Fornisci BigQuery proprietà

   Se hai scelto una singola tabella, non è necessario fornire proprietà aggiuntive.

   Se hai scelto una query, devi fornire le seguenti  BigQuery proprietà Google personalizzate:
   - Imposta `viewsEnabled` su true.
   - Imposta `materializationDataset` su un set di dati. Il principale GCP autenticato dalle credenziali fornite tramite la AWS Glue connessione deve essere in grado di creare tabelle in questo set di dati.

## Creazione di un nodo BigQuery di destinazione

Prerequisiti necessari

- Una connessione BigQuery di tipo AWS Glue Data Catalog
- Un Gestione dei segreti AWS segreto per le tue BigQuery credenziali Google, utilizzate dalla connessione.

- Autorizzazioni appropriate sul processo per leggere il segreto utilizzato dalla connessione.
- Il nome e il set di dati della tabella e del progetto Google Cloud corrispondente su cui desideri scrivere.

Aggiungere un target di BigQuery dati

Per aggiungere un target di dati — BigQuery nodo:

1. Scegli la connessione per la tua destinazione BigQuery dati. Dato che l'hai creato, dovrebbe essere disponibile nel menu a discesa. Se devi creare una connessione, scegli Crea BigQuery connessione. Per ulteriori informazioni, consulta la pagina Overview of using connectors and connections.

   Dopo aver scelto una connessione, puoi visualizzare le proprietà della connessione facendo clic su Visualizza proprietà.
2. Identifica la BigQuery tabella su cui desideri scrivere, quindi scegli un metodo di scrittura.

   - Direct: scrive BigQuery direttamente utilizzando l'API BigQuery Storage Write.
   - Indiretto: scrive su Google Cloud Storage, quindi copia su BigQuery.

   Se desideri scrivere in modo indiretto, fornisci una posizione GCS di destinazione con un bucket GCS temporaneo. Dovrai fornire una configurazione aggiuntiva nella tua AWS Glue connessione. Per ulteriori informazioni, consulta Utilizzo della scrittura indiretta con Google BigQuery.
3. Descrivi i dati che desideri leggere

   (Obbligatorio) imposta il Progetto padre sul progetto contenente la tabella o su un progetto padre di fatturazione, se pertinente.

   Se hai scelto una tabella singola, imposta Table sul nome di una BigQuery tabella Google nel seguente formato: `[dataset].[table]`

## Opzioni avanzate

È possibile fornire opzioni avanzate durante la creazione di un BigQuery nodo. Queste opzioni sono le stesse disponibili durante la programmazione AWS Glue per gli script Spark.

Vedi il riferimento alle opzioni di BigQuery connessione nella guida per gli AWS Glue sviluppatori.

# Connessione a Google Search Console

Google Search Console è una piattaforma gratuita a disposizione dei proprietari di siti web per monitorare il modo in cui Google visualizza il sito e per ottimizzarne la presenza organica. Questo include la visualizzazione dei domini di riferimento, le prestazioni dei siti per dispositivi mobili, i risultati di ricerca completi e le query e le pagine con il traffico più elevato. Se sei un utente di Google Search Console, puoi connetterti AWS Glue al tuo account Google Search Console. È possibile utilizzare Google Search Console come origine dati nei processi ETL. Esegui questi processi per trasferire dati da Google Search Console ai AWS servizi o ad altre applicazioni supportate.

Argomenti


- AWS Glue supporto per Google Search Console
- Policy contenenti le operazioni API per creare e utilizzare le connessioni
- Configurazione di Google Search Console
- Configurazione delle connessioni a Google Search Console
- Lettura da entità di Google Search Console
- Opzioni di connessione di Google Search Console
- Limitazioni di Google Search Console


## AWS Glue supporto per Google Search Console

AWS Glue supporta Google Search Console come segue:

Supportato come origine?

Sì. Puoi utilizzare i lavori AWS Glue ETL per interrogare i dati da Google Search Console.

Supportato come destinazione?

No.

Versioni supportate dell'API Google Search Console

Le versioni dell'API Google Search Console supportate sono le seguenti:

- v3

## Policy contenenti le operazioni API per creare e utilizzare le connessioni

La seguente policy di esempio descrive le autorizzazioni AWS IAM richieste per la creazione e l'utilizzo delle connessioni. Se si sta creando un nuovo ruolo, creare una policy che contenga quanto segue:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:ListConnectionTypes",
        "glue:DescribeConnectionType",
        "glue:RefreshOAuth2Tokens",
        "glue:ListEntities",
        "glue:DescribeEntity"
      ],
      "Resource": "*"
    }
  ]
}
```

In alternativa, se non desideri utilizzare il metodo precedente, utilizza le seguenti policy IAM gestite:

- AWSGlueServiceRole— Garantisce l'accesso alle risorse che AWS Glue i vari processi richiedono per l'esecuzione per conto dell'utente. Queste risorse includono AWS Glue Amazon S3, IAM, CloudWatch Logs e Amazon. EC2 Se segui la convenzione di denominazione delle risorse specificata in questa politica, AWS Glue i processi dispongono delle autorizzazioni richieste. Questa policy è in genere collegata ai ruoli specificati quando si definiscono crawler, processi ed endpoint di sviluppo.
- AWSGlueConsoleFullAccess— Concede l'accesso completo alle AWS Glue risorse quando un'identità a cui è allegata la policy utilizza la AWS console di gestione. Se si segue la convenzione per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questa policy è in genere associata agli utenti della AWS Glue console.

## Configurazione di Google Search Console

Prima di poter AWS Glue utilizzare il trasferimento di dati da Google Search Console, devi soddisfare questi requisiti:

Requisiti minimi

I requisiti minimi sono i seguenti:

- Hai un account Google Search Console.
- Hai un account Google Cloud Platform e un progetto Google Cloud.
- Nel tuo progetto Google Cloud, hai abilitato l'API di Google Search Console.
- Nel tuo progetto Google Cloud, hai configurato una schermata di OAuth consenso per gli utenti esterni. Per ulteriori informazioni, consulta Configurazione della schermata di OAuth consenso nella guida della console di Google Cloud Platform.
- Nel tuo progetto Google Cloud, hai configurato un ID client OAuth 2.0. Vedi Configurazione OAuth 2.0 per le credenziali del client che AWS Glue utilizza per accedere ai tuoi dati in modo sicuro quando effettua chiamate autenticate al tuo account.

Se soddisfi questi requisiti, sei pronto per connetterti AWS Glue al tuo account Google Search Console. Per le connessioni tipiche, non è necessario fare nient'altro in Google Search Console.

## Configurazione delle connessioni a Google Search Console

Google Search Console supporta il tipo di concessione AUTHORIZATION_CODE per. OAuth2 Il tipo di concessione determina il modo in cui AWS Glue comunica con Google Search Console per richiedere l'accesso ai tuoi dati.

- Questo tipo di concessione è considerato «a tre gambe» in OAuth quanto si basa sul reindirizzamento degli utenti a un server di autorizzazione di terze parti per autenticare l'utente. Viene utilizzato durante la creazione di connessioni tramite la console. AWS Glue
- Gli utenti possono comunque scegliere di creare la propria app connessa in Google Search Console e fornire il proprio ID cliente e il segreto del client durante la creazione di connessioni tramite la AWS Glue console. In questo scenario, verranno comunque reindirizzati a Google Search Console per accedere e autorizzare l'accesso AWS Glue alle proprie risorse.
- Questo tipo di concessione genera un token di aggiornamento e un token di accesso. Il token di accesso dura poco tempo e può essere aggiornato automaticamente senza l'interazione dell'utente utilizzando il token di aggiornamento.

- Per la documentazione pubblica di Google Search Console sulla creazione di un'app connessa per il OAuth flusso del codice di autorizzazione, vedi Utilizzo della OAuth versione 2.0 per accedere a Google. APIs

Per configurare una connessione a Google Search Console:

1. In AWS Secrets Manager, crea un segreto con i seguenti dettagli:
   a. Per le app connesse gestite dal cliente: il segreto deve contenere l'app connessa Consumer Secret con `USER_MANAGED_CLIENT_APPLICATION_CLIENT_SECRET` come chiave.
   b. Nota: devi creare un segreto per le tue connessioni in AWS Glue.

1. In AWS Glue Glue Studio, crea una connessione in Connessioni dati seguendo i passaggi seguenti:
   a. Quando selezioni un tipo di connessione, seleziona Google Search Console.
   b. Seleziona il ruolo AWS IAM che AWS Glue può assumere e dispone delle autorizzazioni per le seguenti azioni:

      JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:DescribeSecret",
        "secretsmanager:GetSecretValue",
        "secretsmanager:PutSecretValue",
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DeleteNetworkInterface"
      ],
      "Resource": "*"
    }
  ]
}
```

   c. Seleziona quello `secretName` che desideri utilizzare per questa connessione per AWS Glue inserire i token.

d. Selezionare le opzioni di rete se si desidera utilizzare la propria rete.

2. Concedi il permesso di lettura `secretName` al ruolo IAM associato al tuo AWS Glue lavoro.

## Lettura da entità di Google Search Console

Prerequisito

Un oggetto di Google Search Console da cui si desidera leggere. È necessario il nome dell'oggetto.

Entità supportate come origine:

| Entità | Può essere filtrato | Supporta limite | Supporta Ordina per | Supporta Seleziona * | Supporta il partizionamento |
|---|---|---|---|---|---|
| Analisi della ricerca | Sì | Sì | No | Sì | No |
| Siti | No | No | No | Sì | No |
| Mappa del sito | No | No | No | Sì | No |

Esempio:

```
googleSearchConsole_read = glueContext.create_dynamic_frame.from_options(
    connection_type="googlesearchconsole",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "entityName",
        "API_VERSION": "v3"
    }
```

Dettagli dell'entità e dei campi di Google Search Console:

Google Search Console fornisce endpoint per recuperare i metadati in modo dinamico per le entità supportate. Di conseguenza, il supporto dell'operatore viene acquisito a livello di tipo di dati.

| Entità | Campo | Tipo di dati | Operatori supportati | Nota |
|---|---|---|---|---|
| Analisi della ricerca | keys | List | N/D | |
| | clic | Double | N/D | |
| | impression | Double | N/D | |
| | ctr | BigDecimal | N/D | Per il BigDecimal tipo di dati, il valore '0' è formattato come '0E-18' |
| | position | Double | N/D | |
| | start_end_date | Data | BETWEEN | Il valore predefinito per `start_end_date` è tra <30 days ago from the current date> AND <yesterday: that is, 1 day ago from the current date><br><br>Nota: si aspetta che tu passi un valore di data UTC.<br><br>Esempio: start_end_date between '2022-01-01' |

| Entità | Campo | Tipo di dati | Operatori supportati | Nota |
|---|---|---|---|---|
| | | | | AND '2024-09-09' |
| | paese | Stringa | EQUAL_TO, NOT_EQUAL_TO, CONTAINS | I valori validi sono “IND”, “CAN”, ecc. |
| | tipo | Stringa | EQUAL_TO, NOT_EQUAL_TO | I valori validi sono “discover”, “googleNews”, “news”, “image”, “video”, “web” |
| | searchAppearance | Stringa | EQUAL_TO, NOT_EQUAL_TO, CONTAINS | Vedi Search Appearance per un elenco di valori validi. |
| | dispositivo | Stringa | EQUAL_TO, NOT_EQUAL_TO, CONTAINS | I valori validi sono “DESKTOP”, “MOBILE”, “TABLET” |
| | dimensioni | Stringa | EQUAL_TO | I valori validi sono “country”, “device” |
| | page | Stringa | EQUAL_TO, NOT_EQUAL_TO, CONTAINS | |
| | query | Stringa | EQUAL_TO, NOT_EQUAL_TO, CONTAINS | |

| Entità | Campo | Tipo di dati | Operatori supportati | Nota |
|---|---|---|---|---|
| | dataState | Stringa | EQUAL_TO | I valori validi sono "all" e "final" |
| Siti | siteUrl | Stringa | N/A | |
| | permissionLevel | Stringa | N/A | |
| Mappa del sito | path | Stringa | N/A | |
| | tipo | Stringa | N/A | |
| | lastSubmitted | DateTime | N/D | |
| | isPending | Booleano | N/D | |
| | isSitemapsIndex | Booleano | N/D | |
| | lastDownloaded | DateTime | N/D | |
| | avvisi | Long | N/D | |
| | errori | Long | N/D | |
| | indice | List | N/D | |

> **Note**
>
> Per un elenco aggiornato dei valori validi per i filtri, consulta i documenti dell'API di Google Search Console.
> Il campo `start_end_date` è una combinazione di `start_date` e `end_date`.

Query di partizionamento

Il partizionamento basato su filtri e il partizionamento basato su record non sono supportati.

## Opzioni di connessione di Google Search Console

Le seguenti sono le opzioni di connessione per Google Search Console:

- `ENTITY_NAME`(String): (obbligatorio) utilizzato per la lettura. Il nome del tuo oggetto in Google Search Console.
- `API_VERSION`(String): (obbligatorio) utilizzato per la lettura. La versione dell'API Rest di Google Search Console che desideri utilizzare.
- `SELECTED_FIELDS`(List<String>): predefinito: vuoto (SELECT *). Utilizzato per la lettura. Colonne da selezionare per l'oggetto.
- `FILTER_PREDICATE`(Stringa): predefinito: "start_end_date between <30 days ago from current date> AND <yesterday: that is, 1 day ago from the current date>". Utilizzato per la lettura. Deve essere in formato Spark SQL.
- `QUERY`(Stringa): predefinito: "start_end_date between <30 days ago from current date> AND <yesterday: that is, 1 day ago from the current date>" utilizzato per la lettura. Query Spark SQL completa.
- `INSTANCE_URL`(Stringa): utilizzato per la lettura. Un URL di istanza di Google Search Console valido.

## Limitazioni di Google Search Console

Di seguito sono riportate le limitazioni o le note Google Search Console:

- Google Search Console impone limiti di utilizzo all'API. Per ulteriori informazioni, consulta Limiti d'uso.
- Quando non viene passato alcun filtro per l'entità `Search Analytics`, l'API riassume tutti i clic, le impressioni, il CTR e gli altri dati dell'intero sito entro l'intervallo di date predefinito specificato e li presenta come un singolo record.
- Per suddividere i dati in segmenti più piccoli, è necessario introdurre delle dimensioni nella query. Le dimensioni indicano all'API come vuoi segmentare i tuoi dati.
  - Ad esempio, se aggiungi `filterPredicate: dimensions="country"`, otterrai un record per ogni paese in cui il tuo sito ha ricevuto traffico durante il periodo specificato.
  - Esempio per passare a più dimensioni: `filterPredicate: dimensions="country" AND dimensions="device" AND dimensions="page"`. In questo caso otterrai una riga nella risposta per ogni combinazione unica di queste tre dimensioni.

- I valori predefiniti sono impostati per i campi `start_end_date` e `dataState`.

| Nome del campo | Espressione di filtro predefinita | Espressione per sovrascrivere i valori predefiniti |
|---|---|---|
| start_end_date | start_end_date between <30 days ago from current date> AND <yesterday: that is, 1 day ago from the current date> | start_end_date between "2024-01-01" AND "2024-05-05" |
| dateState | dataState="all | dataState="final |

## Connessione a Google Sheets

Google Sheets è un software per fogli di calcolo online che ti consente di organizzare grandi quantità di dati, creare report personalizzati, automatizzare i calcoli e collaborare con altri. Se sei un utente di Google Sheets, puoi AWS Glue connetterti al tuo account Google Sheets. Puoi usare Google Sheets come origine dati nei tuoi processi ETL. Esegui questi processi per trasferire dati tra Fogli Google e AWS servizi o altre applicazioni supportate.

Argomenti

- AWS Glue supporto per Google Sheets
- Policy contenenti le operazioni API per creare e utilizzare le connessioni
- Configurazione di Google Sheets
- Configurazione delle connessioni di Google Sheets
- Lettura da entità di Google Sheets
- Opzioni di connessione di Google Sheets
- Configura il OAuth flusso del codice di autorizzazione per Google Sheets
- Limitazioni per il connettore di Google Sheets

### AWS Glue supporto per Google Sheets

AWS Glue supporta Google Sheets come segue:

Supportato come origine?

Sì. Puoi utilizzare i lavori AWS Glue ETL per interrogare i dati da Google Sheets.

Supportato come destinazione?

No.

Versioni dell'API di Google Sheets supportate

API di Google Sheets v4 e API di Google Drive v3

## Policy contenenti le operazioni API per creare e utilizzare le connessioni

La seguente politica di esempio descrive le AWS autorizzazioni richieste per la creazione e l'utilizzo delle connessioni. Se si sta creando un nuovo ruolo, creare una policy che contenga quanto segue:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:ListConnectionTypes",
        "glue:DescribeConnectionType",
        "glue:RefreshOAuth2Tokens",
        "glue:ListEntities",
        "glue:DescribeEntity"
      ],
      "Resource": "*"
    }
  ]
}
```

È anche possibile utilizzare le seguenti policy IAM per autorizzare l'accesso:

- AWSGlueServiceRole— Garantisce l'accesso alle risorse che diversi AWS Glue processi richiedono per l'esecuzione per conto dell'utente. Queste risorse includono Amazon S3 AWS Glue, IAM, CloudWatch Logs e Amazon EC2. Se segui la convenzione di denominazione delle risorse specificata in questa politica, AWS Glue i processi dispongono delle autorizzazioni necessarie.

Questa policy è in genere collegata ai ruoli specificati quando si definiscono crawler, processi ed endpoint di sviluppo.

- AWSGlueConsoleFullAccess— Concede l'accesso completo alle AWS Glue risorse quando un'identità a cui è allegata la policy utilizza la AWS console di gestione. Se si segue la convenzione per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questa policy è in genere associata agli utenti della AWS Glue console.

## Configurazione di Google Sheets

Prima di poter AWS Glue utilizzare il trasferimento da Fogli Google, devi soddisfare questi requisiti:

Requisiti minimi

- Hai un account Google Sheets con e-mail e password.
- Il tuo account di Google Sheets è abilitato all'accesso tramite API. Tutti gli utilizzi dell'API di Google Sheets sono disponibili senza costi aggiuntivi.
- Il tuo account Google Sheets ti consente di installare app connesse. Se non hai accesso a questa funzionalità, contatta l'amministratore di Google Sheets.

Se soddisfi questi requisiti, sei pronto per connetterti AWS Glue al tuo account Google Sheets.

## Configurazione delle connessioni di Google Sheets

Per configurare una connessione a Google Sheet:

1. In AWS Secrets Manager, crea un segreto con i seguenti dettagli:
   a. Per il tipo di AuthorizationCode sovvenzione:
      - Per le app connesse gestite dal cliente: il segreto deve contenere l'app connessa Consumer Secret con `USER_MANAGED_CLIENT_APPLICATION_CLIENT_SECRET` come chiave.
2. In AWS Glue Glue Studio, crea una connessione in Connessioni dati seguendo i passaggi seguenti:
   a. Quando selezioni una origine dati, seleziona Fogli Google.
   b. Fornisci l'ambiente di Google Sheets.
      i. Seleziona quella `secretName` che desideri utilizzare per questa connessione AWS Glue per inserire i token.
      ii. Selezionare le opzioni di rete se si desidera utilizzare la propria rete.

3. Concedi il permesso di lettura `secretName` al ruolo IAM associato al tuo AWS Glue lavoro.

   JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:DescribeSecret",
        "secretsmanager:GetSecretValue",
        "secretsmanager:PutSecretValue",
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DeleteNetworkInterface"
      ],
      "Resource": "*"
    }
  ]
}
```

Tipo di concessione AUTHORIZATION_CODE

Questo tipo di concessione è considerato «a tre gambe» in OAuth quanto si basa sul reindirizzamento degli utenti al server di autorizzazione di terze parti per autenticare l'utente. Viene utilizzato durante la creazione di connessioni tramite la console. AWS Glue La AWS Glue Console reindirizzerà l'utente a Google Sheets, dove l'utente deve effettuare il login e consentire AWS Glue le autorizzazioni richieste per accedere alla propria istanza di Google Sheets.

Gli utenti possono scegliere di creare la propria app connessa in Google Sheets e fornire il proprio ID cliente e il segreto del client durante la creazione di connessioni tramite la AWS Glue Console. In questo scenario, verranno comunque reindirizzati a Google Sheets per effettuare il login e autorizzare l'accesso AWS Glue alle proprie risorse.

Questo tipo di concessione genera un token di aggiornamento e un token di accesso. Il token di accesso dura poco tempo e può essere aggiornato automaticamente senza l'interazione dell'utente utilizzando il token di aggiornamento.

Per ulteriori informazioni, consulta la documentazione pubblica di Google Sheets sulla creazione di un'app connessa per il flusso del codice di autorizzazione. OAuth

## Lettura da entità di Google Sheets

### Prerequisiti

- Un Google da SpreadSheet cui vorresti leggere. Avrai bisogno dell' SpreadSheet ID e del tabName del foglio di calcolo.

Dettagli sull'entità e sul campo di Google Sheets:

| Entità | Tipo di dati | Operatori supportati |
|---|---|---|
| Foglio di calcolo | Stringa | N/A (il filtro non è supportato) |

### Esempio

```
googleSheets_read = glueContext.create_dynamic_frame.from_options(
    connection_type="googlesheets",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "{SpreadSheetID}#{SheetTabName}",
        "API_VERSION": "v4"
    }
```

### Query di partizionamento

Solo per Record Base Partitioning, se desideri utilizzare la concorrenza in Spark, `NUM_PARTITIONS` può essere fornito come opzioni aggiuntive. Con questo parametro, la query originale viene suddivisa in `NUM_PARTITIONS` query secondarie che possono essere eseguite contemporaneamente dalle attività Spark.

### Esempio con **`NUM_PARTITIONS`**

```
googlesheets_read = glueContext.create_dynamic_frame.from_options(
    connection_type="googlesheets",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "{SpreadSheetID}#{SheetTabName}",
        "API_VERSION": "v4",
        "NUM_PARTITIONS": "10"
```

```
    }
```

## Opzioni di connessione di Google Sheets

Le seguenti sono le opzioni di connessione per Google Sheets:

- `ENTITY_NAME`(String): (obbligatorio) utilizzato per la lettura. `SpreadSheet ID` e `sheetTabName` in Google Sheets. Esempio: `{SpreadSheetID}#{SheetTabName}`.
- `API_VERSION`(String): (obbligatorio) utilizzato per la lettura. La versione dell'API Rest di Google Sheets che desideri utilizzare.
- `SELECTED_FIELDS`(List<String>): predefinito: vuoto (SELECT *). Utilizzato per la lettura. Colonne da selezionare per l'oggetto.
- `FILTER_PREDICATE`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Dev'essere in formato Spark SQL.
- `QUERY`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Query Spark SQL completa.
- `NUM_PARTITIONS`(Integer): predefinito: 1. Utilizzato per la lettura. Numero di partizioni da leggere.

## Configura il OAuth flusso del codice di autorizzazione per Google Sheets

Prerequisiti

- Un account di Google a cui puoi accedere per utilizzare l'app Google Sheets. Nel tuo account Google, Google Sheets contiene i dati che desideri trasferire.
- Un account Google Cloud Platform e un progetto Google Cloud. Per maggiori dettagli, vedi Crea un progetto di Google Cloud.

Per configurare il tuo account Google e ottenere le credenziali OAuth 2.0:

1. Una volta configurato il progetto Google Cloud, abilita l'API Google Sheets e Google Drive APIs nel progetto. Per i passaggi per abilitarli, consulta Abilita e disabilita APIs nella guida della console API per Google Cloud Platform.
2. Quindi, configura una schermata di OAuth consenso per gli utenti esterni. Per ulteriori informazioni sulla schermata di OAuth consenso, consulta Configurazione della schermata di OAuth consenso nella Guida della console di Google Cloud Platform.
3. Nella schermata di OAuth consenso, aggiungi i seguenti ambiti:

- L'ambito di sola lettura dell'API di Google Sheets
- L'ambito di sola lettura dell'API di Google Drive

Per ulteriori informazioni su questi ambiti, consulta OAuth 2.0 Scopes for Google APIs nella documentazione di Google Identity.

4. Genera ID client OAuth 2.0 e segreto. Per i passaggi per creare questo ID cliente, consulta Configurazione OAuth 2.0 nella guida della console di Google Cloud Platform.

   L'ID client OAuth 2.0 deve avere uno o più URLs reindirizzamenti autorizzati.

   I reindirizzamenti URLs hanno il seguente formato:

   - <aws-region>https://.console.aws.amazon. com/gluestudio/oauth
5. Annota l'ID client e il segreto del client nelle impostazioni per il tuo ID client OAuth 2.0.

## Limitazioni per il connettore di Google Sheets

Di seguito sono riportate le limitazioni per il connettore di Google Sheets.

- Il connettore Google Sheets non supporta Filtri. Pertanto, il partizionamento basato sui filtri non può essere supportato.
- In Record Base Partitioning, non è prevista la restituzione del conteggio esatto dei record da parte di SAAS. Di conseguenza, possono verificarsi scenari in cui vengono creati file con record vuoti.
- Poiché il connettore Google Sheets non supporta il partizionamento basato su filtri, `partitionField`, `lowerbound` e `upperbound` non sono opzioni di connessione valide. Se vengono fornite queste opzioni, il AWS Glue processo dovrebbe fallire.
- È essenziale designare la prima riga del foglio come riga di intestazione per evitare problemi di elaborazione dei dati.
  - Se non viene fornita, la riga di intestazione verrà sostituita con `Unnamed:1`, `Unnamed:2`, `Unnamed:3`... se il foglio contiene dati con la prima riga vuota.
  - Se viene fornita la riga di intestazione, i nomi delle colonne vuote verranno sostituiti con `Unnamed:<number of column>`. Ad esempio, se la riga di intestazione è `['ColumnName1', 'ColumnName2', '', '', 'ColumnName5', 'ColumnName6']`, allora diventerà `['ColumnName1', 'ColumnName2', 'Unnamed:3', 'Unnamed:4', 'ColumnName5', 'ColumnName6'].`

- Il connettore di Google Sheets non supporta il trasferimento incrementale.
- Il connettore di Google Sheets supporta solo il tipo di dati String.
- Le intestazioni duplicate in un foglio verranno rinominate in modo iterativo con un suffisso numerico. I nomi delle intestazioni forniti dall'utente avranno la precedenza durante la ridenominazione delle intestazioni duplicate. Ad esempio, se la riga di intestazione è ["Name", "Name", null, "Unnamed:6", ""], cambierà in: ["Name", "Unnamed:2", "Name1", "Unnamed:4", "Unnamed:6", "Unnamed:61"].
- Il connettore di Google Sheets non supporta gli spazi per un tabName.
- Il nome di una cartella non può contenere i seguenti caratteri speciali:
  - #
  - /

## Connessione a HubSpot

HubSpotla piattaforma CRM dispone di tutti gli strumenti e le integrazioni necessari per il marketing, le vendite, la gestione dei contenuti e il servizio clienti.

- Marketing Hub: software di marketing che aiuta a far crescere il traffico, convertire più visitatori ed eseguire campagne di inbound marketing complete su larga scala.
- Sales Hub: software CRM per le vendite che aiuta a ottenere informazioni più approfondite sui potenziali clienti, automatizzare le attività esistenti e concludere più trattative in meno tempo.
- Service Hub: software di assistenza clienti che aiuta a entrare in contatto con i clienti, superare le aspettative e trasformarli in promotori che fanno crescere l'attività.
- Operations Hub: software operativo che sincronizza le app, pulisce e crea i dati dei clienti e automatizza i processi, in modo che tutti i sistemi e i team lavorino meglio insieme.

Argomenti


- AWS Glue supporto per HubSpot
- Policy contenenti le operazioni API per creare e utilizzare le connessioni
- Configurazione HubSpot
- Configurazione delle connessioni HubSpot
- Lettura da HubSpot entità
- Scrivere alle HubSpot entità

- HubSpot opzioni di connessione
- Limitazioni e note per il HubSpot connettore

## AWS Glue supporto per HubSpot

AWS Glue supporta HubSpot quanto segue:

Supportato come origine?

Sì, sincrona e asincrona. È possibile utilizzare i job AWS Glue ETL da HubSpot cui interrogare i dati.

Supportato come destinazione?

Sì. È possibile utilizzare i job AWS Glue ETL su cui scrivere dati. HubSpot

Versioni HubSpot API supportate

Sono supportate le seguenti versioni HubSpot API:

- v1
- v2
- v3
- v4

Per il supporto delle entità per versione specifica, consulta Entità supportate per la fonte di sincronizzazione e Entità supportate come origine asincrona.

## Policy contenenti le operazioni API per creare e utilizzare le connessioni

La seguente policy di esempio descrive le autorizzazioni AWS IAM richieste per la creazione e l'utilizzo delle connessioni. Se si sta creando un nuovo ruolo, creare una policy che contenga quanto segue:

JSON

```
{
  "Version":"2012-10-17",
```

```
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:ListConnectionTypes",
        "glue:DescribeConnectionType",
        "glue:RefreshOAuth2Tokens",
        "glue:ListEntities",
        "glue:DescribeEntity"
      ],
      "Resource": "*"
    }
  ]
}
```

In alternativa, se non desideri utilizzare il metodo precedente, utilizza le seguenti policy IAM gestite:

- AWSGlueServiceRole— Garantisce l'accesso alle risorse che AWS Glue i vari processi richiedono per l'esecuzione per conto dell'utente. Queste risorse includono AWS Glue Amazon S3, IAM, CloudWatch Logs e Amazon. EC2 Se segui la convenzione di denominazione delle risorse specificata in questa politica, AWS Glue i processi dispongono delle autorizzazioni richieste. Questa policy è in genere collegata ai ruoli specificati quando si definiscono crawler, processi ed endpoint di sviluppo.
- AWSGlueConsoleFullAccess— Concede l'accesso completo alle AWS Glue risorse quando un'identità a cui è allegata la policy utilizza la AWS console di gestione. Se si segue la convenzione per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questa policy è in genere associata agli utenti della AWS Glue console.

## Configurazione HubSpot

Prima di poter AWS Glue utilizzare il trasferimento di dati da HubSpot, è necessario soddisfare i seguenti requisiti:

Requisiti minimi

I requisiti minimi sono i seguenti:

- Hai un HubSpot account. Per ulteriori informazioni, consulta Creazione di un HubSpot account.
- Il tuo HubSpot account è abilitato all'accesso all'API.

- Nel tuo account HubSpot sviluppatore dovresti avere un'app che fornisca le credenziali del client da AWS Glue utilizzare per accedere ai tuoi dati in modo sicuro quando effettua chiamate autenticate al tuo account. Per ulteriori informazioni, consulta Creazione di un'app per HubSpot sviluppatori.

Se soddisfi questi requisiti, sei pronto per AWS Glue connetterti al tuo account. HubSpot Per le connessioni tipiche, non devi fare nient'altro HubSpot.

Creazione di un HubSpot account

Per creare un HubSpot account:

1. Vai all' SignUp URL del HubSpot CRM.
2. Inserisci il tuo indirizzo e-mail e scegli Verifica e-mail (in alternativa, puoi scegliere di registrarti con un account Google, Microsoft o Apple).
3. Controlla nella tua casella di posta il codice di verifica di. HubSpot
4. Inserisci il codice di verifica a 6 cifre e fai clic su Avanti.
5. Inserisci una password e fai clic su Avanti.
6. Inserisci nome e cognome e fai clic su Avanti oppure registrati utilizzando il link Registrati con Google.
7. Inserisci il tuo settore e fai clic su Avanti.
8. Inserisci il tuo ruolo professionale e fai clic su Avanti.
9. Inserisci il nome della tua azienda e fai clic su Avanti.
10. Seleziona le dimensioni della tua azienda (numero di dipendenti che lavorano nella tua azienda) e fai clic su Avanti.
11. Inserisci il sito Web della tua azienda e fai clic su Avanti.
12. Seleziona dove devono essere ospitati i tuoi dati (Stati Uniti o Europa) e fai clic su Crea account.
13. Seleziona lo scopo della creazione del tuo account e fai clic su Avanti.
14. Scegli Connetti account Google o scegli di aggiungere contatti tu stesso per collegare i tuoi contatti al tuo HubSpot account.
15. Accedi al tuo account Google se hai scelto l'opzione Connect Google Account per collegare i tuoi contatti e iniziare a utilizzare il tuo HubSpot account.

### Creazione di un'app per HubSpot sviluppatori

Gli account per sviluppatori di app sono destinati alla creazione e alla gestione di app, integrazioni e account di test per sviluppatori. Le inserzioni di App Marketplace possono essere create e gestite da lì. Tuttavia, gli account sviluppatore di app e gli account di test associati non sono collegati a un HubSpot account standard. Non possono sincronizzare dati o risorse da o verso un altro HubSpot account. Per ottenere l'ID client e il segreto client devi creare un account sviluppatore.

1. Vai a https://developers.hubspot.com/
2. Scegli Crea un account sviluppatore e scorri verso il basso.
3. Ti verrà chiesto se desideri creare un account per sviluppatori di app, un account privato per app o un account CMS Developer Sandbox. Scegli Crea un account per sviluppatori di app.
4. Poiché hai già creato un account con HubSpot, puoi scegliere Continua con questo utente.
5. Fai clic su Inizia la registrazione.
6. Inserisci il tuo ruolo professionale e fai clic su Avanti.
7. Assegna un nome al tuo account sviluppatore e fai clic su Avanti, quindi su Ignora.
8. Scegli Create App (Crea app).
9. Una volta creata l'app, scegli Auth.
10. In Auth, prendi nota dell'ID client del segreto client.
11. Aggiungi l'URL di reindirizzamento specifico della tua regione come https: *//<aws-region>* .console.aws.amazon. com/gluestudio/oauth. For example, add https://us-east-1.console.aws.amazon.com/gluestudio/oauthper la regione us-east-1.
12. Scorri verso il basso e trova gli ambiti. È necessario selezionare due tipi di ambiti nelle intestazioni “CRM” e “Standard”.
13. Aggiungi i seguenti ambiti:

```
content
automation
oauth
crm.objects.owners.read
forms
tickets
crm.objects.contacts.write
e-commerce
crm.schemas.custom.read
crm.objects.custom.read
sales-email-read
```

```
crm.objects.custom.write
crm.objects.companies.write
crm.lists.write
crm.objects.companies.read
crm.lists.read
crm.objects.deals.read
crm.objects.deals.write
crm.objects.contacts.read
```

14. Fai clic su Salva: il tuo account di sviluppo è ora pronto per l'uso.
15. Scorri in alto per trovare l'ID client.
16. Nella stessa pagina, fai clic su Mostra per visualizzare il segreto client.

### Creazione di un account di test HubSpot per sviluppatori

All'interno degli account per sviluppatori di app, puoi creare account di test per sviluppatori per testare app e integrazioni senza influire sui HubSpot dati reali. Gli account di test per sviluppatori non rispecchiano gli account di produzione, ma hanno accesso a una versione di prova di 90 giorni delle versioni Enterprise di Marketing, Sales, Service, CMS e Operations Hub, che consente di testare la maggior parte degli strumenti e. HubSpot APIs

1. Fai clic su Home.
2. Fai clic su Crea account di prova.
3. Fai clic su Crea account di prova dell'app.
4. Viene visualizzata una nuova finestra. Inserisci il nome dell'account di prova dell'app e fai clic su Crea.

   Il account di prova dell'app è stato creato.

> **Note**
>
> L'account sviluppatore è correlato ad attività di sviluppo come l'integrazione delle API e l'account di prova dell'app viene utilizzato per visualizzare i dati creati o estratti dall'account sviluppatore.

## Configurazione delle connessioni HubSpot

HubSpot supporta il tipo di concessione AUTHORIZATION_CODE per. OAuth2

- Questo tipo di concessione è considerato «a tre gambe» in OAuth quanto si basa sul reindirizzamento degli utenti a un server di autorizzazione di terze parti per autenticare l'utente. Viene utilizzato durante la creazione di connessioni tramite la console. AWS Glue L'utente che crea una connessione deve fornire le informazioni OAuth correlate come Client ID e Client Secret per la propria applicazione HubSpot client. La AWS Glue console reindirizzerà l'utente al HubSpot punto in cui deve effettuare il login e consentirà AWS Glue alle autorizzazioni richieste di accedere alla propria HubSpot istanza.
- Gli utenti possono comunque scegliere di creare la propria app connessa HubSpot e fornire il proprio ID client e il segreto del client durante la creazione di connessioni tramite la AWS Glue console. In questo scenario, verranno comunque reindirizzati all'accesso e HubSpot all'autorizzazione ad accedere AWS Glue alle proprie risorse.
- Questo tipo di concessione genera un token di aggiornamento e un token di accesso. Il token di accesso dura poco tempo e può essere aggiornato automaticamente senza l'interazione dell'utente utilizzando il token di aggiornamento.
- Per la HubSpot documentazione pubblica sulla creazione di un'app connessa per il OAuth flusso del codice di autorizzazione, vedi App pubbliche.

Per configurare una HubSpot connessione:

1. In AWS Secrets Manager, crea un segreto con i seguenti dettagli:
   a. Per le app connesse gestite dal cliente: il segreto deve contenere l'app connessa Consumer Secret con `USER_MANAGED_CLIENT_APPLICATION_CLIENT_SECRET` come chiave.
   b. Nota: è necessario creare un segreto per la connessione in AWS Glue.
2. In AWS Glue Glue Studio, crea una connessione in Connessioni dati seguendo i passaggi seguenti:
   a. Quando si seleziona un tipo di connessione, selezionare HubSpot.
   b. Fornisci l' HubSpot ambiente.
   c. Seleziona il ruolo AWS IAM che AWS Glue può assumere e dispone delle autorizzazioni per le seguenti azioni:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:DescribeSecret",
        "secretsmanager:GetSecretValue",
        "secretsmanager:PutSecretValue",
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DeleteNetworkInterface"
      ],
      "Resource": "*"
    }
  ]
}
```

d. Seleziona quello `secretName` che desideri utilizzare per questa connessione per AWS Glue inserire i token.

e. Selezionare le opzioni di rete se si desidera utilizzare la propria rete.

3. Concedi il permesso di lettura `secretName` al ruolo IAM associato al tuo AWS Glue lavoro.

4. Nella configurazione del tuo AWS Glue lavoro, fornisci `connectionName` una connessione di rete aggiuntiva.

## Lettura da HubSpot entità

Prerequisito

Un HubSpot oggetto da cui vorresti leggere. È necessario il nome dell'oggetto, come contatto o attività. Nella tabella seguente sono illustrate le entità supportate per la fonte di sincronizzazione.

Entità supportate per la fonte di sincronizzazione

| Entità | Versione API | Può essere filtrato | Supporta limite | Supporta Ordina per | Supporta Seleziona * | Supporta il partizionamento |
|---|---|---|---|---|---|---|
| Campagne | v1 | No | Sì | No | Sì | No |
| Aziende | v3 | Sì | Sì | Sì | Sì | Sì |
| Contatti | v3 | Sì | Sì | Sì | Sì | Sì |
| Elenchi di contatti | v1 | No | Sì | No | Sì | No |
| Offerte | v3 | Sì | Sì | Sì | Sì | Sì |
| Pipeline CRM (Pipeline delle offerte) | v1 | No | No | No | Sì | No |
| Eventi e-mail | v1 | No | Sì | No | Sì | No |
| Chiamate | v3 | Sì | Sì | Sì | Sì | Sì |
| Note | v3 | Sì | Sì | Sì | Sì | Sì |
| E-mail | v3 | Sì | Sì | Sì | Sì | Sì |
| Riunioni | v3 | Sì | Sì | Sì | Sì | Sì |
| Processi | v3 | Sì | Sì | Sì | Sì | Sì |
| Posta elettronica | v3 | Sì | Sì | Sì | Sì | Sì |

| Entità | Versione API | Può essere filtrato | Supporta limite | Supporta Ordina per | Supporta Seleziona * | Supporta il partizionamento |
|---|---|---|---|---|---|---|
| Oggetti personalizzati | v3 | Sì | Sì | Sì | Sì | Sì |
| Moduli | v2 | No | No | No | Sì | No |
| Proprietari | v3 | No | Sì | No | Sì | No |
| Prodotti | v3 | Sì | Sì | Sì | Sì | Sì |
| Ticket | v3 | Sì | Sì | Sì | Sì | Sì |
| Flussi di lavoro | v3 | No | No | No | Sì | No |
| Associations | v4 | Sì | No | No | Sì | No |
| Etichette associazioni | v4 | No | No | No | Sì | No |

Esempio:

```
hubspot_read = glueContext.create_dynamic_frame.from_options(
    connection_type="hubspot",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "contact",
        "API_VERSION": "v3"
    }
```

Entità supportate come origine asincrona

| Entità | Versione API | Può essere filtrato | Supporta limite | Supporta Ordina per | Supporta Seleziona * | Supporta il partizionamento |
|---|---|---|---|---|---|---|
| Aziende | v3 | Sì | No | Sì | Sì | No |
| Contatti | v3 | Sì | No | Sì | Sì | No |
| Offerte | v3 | Sì | No | Sì | Sì | No |
| Chiamate | v3 | Sì | No | Sì | Sì | No |
| Note | v3 | Sì | No | Sì | Sì | No |
| E-mail | v3 | Sì | No | Sì | Sì | No |
| Riunioni | v3 | Sì | No | Sì | Sì | No |
| Processi | v3 | Sì | No | Sì | Sì | No |
| Posta elettronica | v3 | Sì | No | Sì | Sì | No |
| Oggetti personalizzati | v3 | Sì | No | Sì | Sì | No |
| Prodotti | v3 | Sì | No | Sì | Sì | No |
| Ticket | v3 | Sì | No | Sì | Sì | No |

Esempio:

```
hubspot_read = glueContext.create_dynamic_frame.from_options(
    connection_type="hubspot",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "contact",
```

```
        "API_VERSION": "v3",
        "TRANSFER_MODE": "ASYNC"
    }
```

HubSpot dettagli dell'entità e del campo:

HubSpot API v4:

| Entità | Versione API | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|---|
| Etichetta associazione | v4 | category | Stringa | N/A |
| | | typeId | Numero intero | N/D |
| | | etichetta | Stringa | N/A |
| Associations | | from | Struct | N/D |
| | | id | Stringa | "=" |
| | | in | List | N/D |

> **Note**
>
> Per l'`Associations`oggetto, per recuperare le associazioni tra due oggetti, è necessario fornire il 'from Id' (l'ID del primo oggetto) tramite un filtro obbligatorio durante la creazione di un lavoro. AWS Glue Se si desidera recuperare associazioni multiple da IDs in quel caso, è necessario fornire più IDs associazioni nella clausola. `where` Ad esempio: `Associations` per recuperare i contatti IDs «1» e «151», è necessario fornire un filtro come. `where id=1 AND id=151`

HubSpot API v3:

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| Owner | firstName | Stringa | N/A |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | lastName | Stringa | N/A |
| | createdAt | DateTime | N/D |
| | archived | Booleano | N/D |
| | team | List | N/D |
| | id | Stringa | N/A |
| | userId | Numero intero | N/D |
| | e-mail | Stringa | N/A |
| | updatedAt | DateTime | N/D |
| Flusso di lavoro | nome | Stringa | N/A |
| | id | Numero intero | N/D |
| | tipo | Stringa | N/A |
| | enabled | Booleano | N/D |
| | insertedAt | Long | N/D |
| | updatedAt | Long | N/D |
| | contactListIds | Struct | N/D |
| | personaTagIds | List | N/D |

Per le seguenti entità, HubSpot fornisce endpoint per recuperare i metadati in modo dinamico, in modo che il supporto dell'operatore venga acquisito a livello di tipo di dati per ciascuna entità.

> **Note**
>
> `DML_STATUS` è un campo virtuale aggiunto a ogni record al runtime per determinarne lo stato (CREATO/AGGIORNATO) in modalità sincrona. L'operatore `CONTAINS/LIKE` non è supportato nella modalità asincrona.

| Entità | Tipo di dati | Operatori supportati |
|---|---|---|
| Contatti | Numero intero | "=, !=, <, >, >=, <=" |
| | Long | "=, !=, <, >, >=, <=" |
| | Stringa | "=, !=, LIKE" |
| | Data | N/D |
| | DateTime | "between" |
| | Booleano | "=" |
| | List | N/D |
| | Struct | N/D |
| Azienda | Numero intero | "=, !=, <, >, >=, <=" |
| | Long | "=, !=, <, >, >=, <=" |
| | Stringa | "=, !=, LIKE" |
| | Data | N/D |
| | DateTime | "between" |
| | Booleano | "=" |
| | List | N/D |
| | Struct | N/D |

| Entità | Tipo di dati | Operatori supportati |
|---|---|---|
| Offerta | Numero intero | "=, !=, <, >, >=, <=" |
| | Long | "=, !=, <, >, >=, <=" |
| | Stringa | "=, !=, LIKE" |
| | Data | N/D |
| | DateTime | "between" |
| | Booleano | "=" |
| | List | N/D |
| | Struct | N/D |
| Ticket | Numero intero | "=, !=, <, >, >=, <=" |
| | Long | "=, !=, <, >, >=, <=" |
| | Stringa | "=, !=, LIKE" |
| | Data | N/D |
| | DateTime | "between" |
| | Booleano | "=" |
| | List | N/D |
| | Struct | N/D |
| Prodotto | Numero intero | "=, !=, <, >, >=, <=" |
| | Long | "=, !=, <, >, >=, <=" |
| | Stringa | "=, !=, LIKE" |
| | Data | N/D |

| Entità | Tipo di dati | Operatori supportati |
|---|---|---|
| | DateTime | "between" |
| | Booleano | "=" |
| | List | N/D |
| | Struct | N/D |
| Oggetto personalizzato | Numero intero | "=, !=, <, >, >=, <=" |
| | Long | "=, !=, <, >, >=, <=" |
| | Stringa | "=, !=, LIKE" |
| | Data | N/D |
| | DateTime | "between" |
| | Booleano | "=" |
| | List | N/D |
| | Struct | N/D |
| Esegui una chiamata a | Numero intero | "=, !=, <, >, >=, <=" |
| | Long | "=, !=, <, >, >=, <=" |
| | Stringa | "=, !=, LIKE" |
| | Data | N/D |
| | DateTime | "between" |
| | Booleano | "=" |
| | List | N/D |
| | Struct | N/D |

| Entità | Tipo di dati | Operatori supportati |
|---|---|---|
| Email | Numero intero | "=, !=, <, >, >=, <=" |
| | Long | "=, !=, <, >, >=, <=" |
| | Stringa | "=, !=, LIKE" |
| | Data | N/D |
| | DateTime | "between" |
| | Booleano | "=" |
| | List | N/D |
| | Struct | N/D |
| Riunione | Numero intero | "=, !=, <, >, >=, <=" |
| | Long | "=, !=, <, >, >=, <=" |
| | Stringa | "=, !=, LIKE" |
| | Data | N/D |
| | DateTime | "between" |
| | Booleano | "=" |
| | List | N/D |
| | Struct | N/D |
| Nota | Numero intero | "=, !=, <, >, >=, <=" |
| | Long | "=, !=, <, >, >=, <=" |
| | Stringa | "=, !=, LIKE" |
| | Data | N/D |

| Entità | Tipo di dati | Operatori supportati |
|---|---|---|
| | DateTime | "between" |
| | Booleano | "=" |
| | List | N/D |
| | Struct | N/D |
| Operazione | Numero intero | "=, !=, <, >, >=, <=" |
| | Long | "=, !=, <, >, >=, <=" |
| | Stringa | "=, !=, LIKE" |
| | Data | N/D |
| | DateTime | "between" |
| | Booleano | "=" |
| | List | N/D |
| | Struct | N/D |
| Posta elettronica | Numero intero | "=, !=, <, >, >=, <=" |
| | Long | "=, !=, <, >, >=, <=" |
| | Stringa | "=, !=, LIKE" |
| | Data | N/D |
| | DateTime | "between" |
| | Booleano | "=" |
| | List | N/D |
| | Struct | N/D |

HubSpot API v2:

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| Modulo | portalId | Numero intero | N/D |
| | guid | Stringa | N/A |
| | nome | Stringa | N/A |
| | metodo | Stringa | N/A |
| | cssClass | Stringa | N/A |
| | redirect | Stringa | N/A |
| | submitText | Stringa | N/A |
| | notifyRecipients | Stringa | N/A |
| | createdAt | Long | N/D |
| | updatedAt | Long | N/D |
| | ignoreCurrentValues | Booleano | N/D |
| | deletable | Booleano | N/D |
| | inlineMessage | Booleano | N/D |
| | captchaEnabled | Booleano | N/D |
| | cloneable | Booleano | N/D |
| | formFieldGroups | List | N/D |
| | editable | Booleano | N/D |
| | deletedAt | Numero intero | N/D |
| | themeName | Stringa | N/A |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | parentId | Numero intero | N/D |
| | stile | Stringa | N/A |
| | isPublished | Booleano | N/D |
| | publishAt | Numero intero | N/D |
| | unpublishAt | Numero intero | N/D |
| | publishedAt | Numero intero | N/D |
| | kickbackEmailWorkflowId | Stringa | N/A |
| | kickbackEmailsJson | Numero intero | N/D |
| | customUid | Stringa | N/A |
| | createMarketableContact | Booleano | N/D |
| | editVersion | Numero intero | N/D |
| | thankYouMessageJson | Stringa | N/A |
| | themeColor | Stringa | N/A |
| | alwaysCreateNewCompagnia | Booleano | N/D |
| | internalUpdatedAt | Long | N/D |
| | businessUnitId | Numero intero | N/D |
| | portableKey | Stringa | N/A |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | paymentSessionTemplateIds | List | N/D |
| | selectedExternalOptions | List | N/D |

HubSpot API v1:

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| Campagna | id | Numero intero | N/D |
| | appId | Numero intero | N/D |
| | appName | Stringa | N/A |
| | lastUpdatedTime | Long | N/D |
| Contact_List | dynamic | Booleano | N/D |
| | nome | Stringa | N/A |
| | portalId | Numero intero | N/D |
| | createdAt | Long | N/D |
| | listId | Numero intero | N/D |
| | updatedAt | Long | N/D |
| | ListType | Stringa | N/A |
| | filtri | List | N/D |
| | authorId | Numero intero | N/D |
| | metaData | Struct | N/D |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | archived | Booleano | N/D |
| | ilsFilterBranch | Stringa | N/A |
| | filterIds | List | N/D |
| | limitExempt | Booleano | N/D |
| | internal | Booleano | N/D |
| | readOnly | Booleano | N/D |
| | parentId | Numero intero | N/D |
| Email_Event | id | Stringa | N/A |
| | tipo | Stringa | N/A |
| | recipient | Stringa | N/A |
| | portalId | Numero intero | N/D |
| | appId | Numero intero | N/D |
| | appName | Stringa | N/A |
| | emailCampaignId | Long | N/D |
| | attempt | Numero intero | N/D |
| | created | Long | N/D |
| | sentBy | Struct | N/D |
| | smtpId | Stringa | N/A |
| | risposta | Stringa | N/A |
| | subject | Stringa | N/A |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | cc | List | N/D |
| | bcc | List | N/D |
| | replyTo | List | N/D |
| | from | Stringa | N/A |
| | dropReason | Stringa | N/A |
| | dropMessage | Stringa | N/A |
| | browser | Struct | N/D |
| | userAgent | Stringa | N/A |
| | durata | Long | N/D |
| | location | Struct | N/D |
| | filteredEvent | Booleano | N/D |
| | deviceType | Stringa | N/A |
| | suppressedReason | Stringa | N/A |
| | suppressedMessage | Stringa | N/A |
| CRM_Pipeline | pipelineId | Stringa | N/A |
| | createdAt | Long | N/D |
| | updatedAt | Long | N/D |
| | objectType | Stringa | N/A |
| | etichetta | Stringa | N/A |
| | displayOrder | Numero intero | N/D |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | attiva | Booleano | N/D |
| | fasi | List | N/D |
| | objectTypeId | Stringa | N/A |
| | default | Booleano | N/D |

Query di partizionamento

Puoi fornire le opzioni Spark aggiuntive `PARTITION_FIELD`, `LOWER_BOUND`, `UPPER_BOUND` e `NUM_PARTITIONS` se desideri utilizzare la concorrenza in Spark. Con questi parametri, la query originale viene suddivisa in `NUM_PARTITIONS` query secondarie che possono essere eseguite contemporaneamente dalle attività Spark.

- `PARTITION_FIELD`: il nome del campo da utilizzare per partizionare la query.
- `LOWER_BOUND`: un valore limite inferiore inclusivo del campo di partizione scelto.

  Per il DateTime campo, accettiamo il valore in formato ISO.

  Esempi di valori validi:

  ```
  “2024-01-01T10:00:00.115Z"
  ```
- `UPPER_BOUND`: un valore limite superiore esclusivo del campo di partizione scelto.
- `NUM_PARTITIONS`: il numero di partizioni.

Nella seguente tabella sono descritti i dettagli del campo di supporto del partizionamento delle entità:

| Nome dell'entità | Campi di partizionamento | Tipo di dati |
|---|---|---|
| contact | hs_object_id | Long |
| | createdate, lastmodifieddate | DateTime |
| company | hs_object_id | Long |

| Nome dell'entità | Campi di partizionamento | Tipo di dati |
|---|---|---|
| | createdate, hs_lastmodifieddate | DateTime |
| offerta | hs_object_id | Long |
| | createdate, hs_createdate, hs_lastmodifieddate | DateTime |
| ticket | hs_object_id | Long |
| | createdate, hs_lastmodifieddate | DateTime |
| product | hs_object_id | Long |
| | createdate, hs_lastmodifieddate | DateTime |
| custom_object | hs_object_id | Long |
| | createdate, hs_lastmodifieddate | DateTime |
| call | hs_object_id | Long |
| | createdate, hs_lastmodifieddate | DateTime |
| e-mail | hs_object_id | Long |
| | createdate, hs_lastmodifieddate | DateTime |
| meeting | hs_object_id | Long |
| | createdate, hs_lastmodifieddate | DateTime |
| note | hs_object_id | Long |

| Nome dell'entità | Campi di partizionamento | Tipo di dati |
|---|---|---|
| | createdate, hs_lastmodifieddate | DateTime |
| task | hs_object_id | Long |
| | createdate, hs_lastmodifieddate | DateTime |
| postal_mail | hs_object_id | Long |
| | createdate, hs_lastmodifieddate | DateTime |

Esempio:

```
hubspot_read = glueContext.create_dynamic_frame.from_options(
    connection_type="hubspot",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "company",
        "API_VERSION": "v3",
        "PARTITION_FIELD": "hs_object_id"
        "LOWER_BOUND": "50"
        "UPPER_BOUND": "16726619290"
        "NUM_PARTITIONS": "10"
    }
```

## Scrivere alle HubSpot entità

Prerequisiti

- Un HubSpot oggetto su cui desideri scrivere. È necessario il nome dell'oggetto, come contatto o ticket.
- Il HubSpot connettore supporta le seguenti operazioni di scrittura:
  - INSERT
  - UPDATE

- Quando si utilizza l'operazione di scrittura `UPDATE`, è necessario indicare l'opzione `ID_FIELD_NAMES` per specificare il campo ID per i record.

Entità supportate come destinazione di sincronizzazione:

| Entità | Versione API | Sarà supportato come connettore di destinazione | Può essere inserito | Può essere aggiornato |
|---|---|---|---|---|
| Aziende | v3 | Sì | Sì (singolo, in blocco) | Sì (singolo, in blocco) |
| Contatti | v3 | Sì | Sì (singolo, in blocco) | Sì (singolo, in blocco) |
| Offerte | v3 | Sì | Sì (singolo, in blocco) | Sì (singolo, in blocco) |
| Prodotti | v3 | Sì | Sì (singolo, in blocco) | Sì (singolo, in blocco) |
| Chiamate | v3 | Sì | Sì (singolo, in blocco) | Sì (singolo, in blocco) |
| Riunioni | v3 | Sì | Sì (singolo, in blocco) | Sì (singolo, in blocco) |
| Note | v3 | Sì | Sì (singolo, in blocco) | Sì (singolo, in blocco) |
| E-mail | v3 | Sì | Sì (singolo, in blocco) | Sì (singolo, in blocco) |
| Processi | v3 | Sì | Sì (singolo, in blocco) | Sì (singolo, in blocco) |
| Posta elettronica | v3 | Sì | Sì (singolo, in blocco) | Sì (singolo, in blocco) |

| Entità | Versione API | Sarà supportato come connettore di destinazione | Può essere inserito | Può essere aggiornato |
|---|---|---|---|---|
| Oggetti personalizzati | v3 | Sì | Sì (singolo, in blocco) | Sì (singolo, in blocco) |
| Ticket | v3 | Sì | Sì (singolo, in blocco) | Sì (singolo, in blocco) |
| Associations | v4 | Sì | Sì (singolo, in blocco) | No |
| Etichette associazioni | v4 | Sì | Sì (singolo, in blocco) | Sì (singolo, in blocco) |

Esempi:

Operazione INSERT

```
hubspot_write = glueContext.write_dynamic_frame.from_options(
    frame=frameToWrite,
    connection_type="hubspot",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "contact",
        "API_VERSION": "v3",
        "WRITE_OPERATION": "INSERT"
    }
)
```

Operazione UPDATE

```
hubspot_write = glueContext.write_dynamic_frame.from_options(
    frame=frameToWrite,
    connection_type="hubspot",
    connection_options={
```

```
        "connectionName": "connectionName",
        "ENTITY_NAME": "deal",
        "API_VERSION": "v3",
        "WRITE_OPERATION": "UPDATE",
        "ID_FIELD_NAMES": "hs_object_id"
    }
)
```

## HubSpot opzioni di connessione

Di seguito sono elencate le opzioni di connessione per HubSpot:

- `ENTITY_NAME`(String): (obbligatorio) utilizzato per la lettura. Il nome dell'oggetto in HubSpot.
- `API_VERSION`(String) - (Obbligatorio) Usato per la lettura. HubSpot Versione dell'API Rest che desideri utilizzare. Ad esempio: v1, v2, v3, v4.
- `SELECTED_FIELDS`(Elenco<Stringa>): predefinito: vuoto (SELECT *). Utilizzato per la lettura. Colonne da selezionare per l'oggetto.
- `FILTER_PREDICATE`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Dev'essere in formato Spark SQL.
- `QUERY`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Query Spark SQL completa.
- `PARTITION_FIELD`(Stringa): utilizzato per la lettura. Campo da utilizzare per partizionare la query.
- `LOWER_BOUND`(Stringa): utilizzato per la lettura. Un valore limite inferiore inclusivo del campo di partizione scelto.
- `UPPER_BOUND`(Stringa): utilizzato per la lettura. Un valore limite superiore esclusivo del campo di partizione scelto.
- `NUM_PARTITIONS`(Numero intero): predefinito: 1. Utilizzato per la lettura. Numero di partizioni da leggere.
- `TRANSFER_MODE`(Stringa): utilizzato per indicare se la query deve essere eseguita in modalità asincrona.
- `WRITE_OPERATION`(Stringa): predefinito: INSERT. Utilizzato per la scrittura. Il valore deve essere INSERT o UPDATE.
- `ID_FIELD_NAMES`(String): predefinito: nullo. Obbligatorio per UPDATE.

### Limitazioni e note per il HubSpot connettore

Di seguito sono riportate le limitazioni o le note relative al HubSpot connettore:

- Gli endpoint di ricerca sono limitati a 10.000 risultati totali per una determinata query. Qualsiasi partizione con più di 10.000 record genererà un errore 400.
- Altre importanti limitazioni per il connettore sono descritte in Limitazioni.
- Un massimo di tre istruzioni di filtraggio sono accettate da HubSpot.
- Attualmente, HubSpot supporta le associazioni tra HubSpot oggetti standard (ad esempio contact, company, deal o ticket) e oggetti personalizzati.
  - Per un account gratuito: puoi creare solo fino a 10 tipi di associazione tra ogni abbinamento di oggetti (ad esempio contatti e aziende).
  - Per un account Super Admin: puoi creare solo fino a 50 tipi di associazione tra ogni coppia di oggetti.
  - Per ulteriori informazioni, consulta Associazioni v4 e Crea e usa le etichette di associazione.
- Gli oggetti "Preventivi" e "Comunicazioni" non sono presenti per le associazioni in quanto attualmente non sono supportati nel connettore.
- Per l'asincrono, SaaS ordina i valori solo in ordine crescente.
- Per l'entità `Ticket`, SaaS non restituisce il campo `hs_object_id` in modalità asincrona.

## Connessione a Instagram Ads

Instagram è una popolare app per la condivisione di foto che consente di entrare in contatto con brand, celebrità, leader di pensiero, amici, familiari e altro ancora. Si tratta di un servizio di condivisione di foto e social network. Gli utenti possono scattare foto o brevi video e condividerli con i propri follower. Gli annunci Instagram sono post pagati dalle aziende per raggiungere gli utenti di Instagram.

Argomenti

- AWS Glue supporto per Instagram Ads
- Policy contenenti le operazioni API per creare e utilizzare le connessioni
- Configurazione di Instagram Ads
- Configurazione delle connessioni su Instagram Ads
- Lettura dalle entità di Instagram Ads

- Opzioni di connessione su Instagram Ads
- Limitazioni e note per il connettore di Instagram Ads

## AWS Glue supporto per Instagram Ads

AWS Glue supporta Instagram Ads come segue:

Supportato come origine?

Sì. Puoi utilizzare i lavori AWS Glue ETL per interrogare i dati da Instagram Ads.

Supportato come destinazione?

No.

Versioni dell'API di Instagram Ads supportate

Sono supportate le seguenti versioni dell'API Instagram Ads:

- v17.0
- v18.0
- v19.0
- v20.0

## Policy contenenti le operazioni API per creare e utilizzare le connessioni

La seguente policy di esempio descrive le autorizzazioni AWS IAM richieste per la creazione e l'utilizzo delle connessioni. Se si sta creando un nuovo ruolo, creare una policy che contenga quanto segue:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:ListConnectionTypes",
```

```
        "glue:DescribeConnectionType",
        "glue:RefreshOAuth2Tokens",
        "glue:ListEntities",
        "glue:DescribeEntity"
      ],
      "Resource": "*"
    }
  ]
}
```

In alternativa, se non desideri utilizzare il metodo precedente, utilizza le seguenti policy IAM gestite:

- AWSGlueServiceRole— Garantisce l'accesso alle risorse che AWS Glue i vari processi richiedono per l'esecuzione per conto dell'utente. Queste risorse includono AWS Glue Amazon S3, IAM, CloudWatch Logs e Amazon. EC2 Se segui la convenzione di denominazione delle risorse specificata in questa politica, AWS Glue i processi dispongono delle autorizzazioni richieste. Questa policy è in genere collegata ai ruoli specificati quando si definiscono crawler, processi ed endpoint di sviluppo.
- AWSGlueConsoleFullAccess— Concede l'accesso completo alle AWS Glue risorse quando un'identità a cui è allegata la policy utilizza la AWS console di gestione. Se si segue la convenzione per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questa policy è in genere associata agli utenti della AWS Glue console.

## Configurazione di Instagram Ads

Prima di poter AWS Glue utilizzare il trasferimento di dati da Instagram Ads, devi soddisfare questi requisiti:

Requisiti minimi

I requisiti minimi sono i seguenti:

- Gli account Instagram Standard sono accessibili indirettamente tramite Facebook.
- L'autenticazione dell'utente è necessaria per generare il token di accesso.
- Il connettore Instagram Ads SDK implementerà il OAuth flusso User Access Token.
- Stiamo utilizzando OAuth2 .0 per autenticare le nostre richieste API su Instagram Ads. Questa autenticazione basata sul Web rientra nell'architettura dell'autenticazione a più fattori (MFA), che è un superset di 2FA.

- L'utente deve concedere le autorizzazioni per accedere agli endpoint. Per accedere ai dati dell'utente, l'autorizzazione degli endpoint viene gestita tramite autorizzazioni e funzionalità.

Ottenere le credenziali 2.0 OAuth

Per ottenere le credenziali API così da effettuare chiamate autenticate alla tua istanza, consulta API Graph.

## Configurazione delle connessioni su Instagram Ads

Instagram Ads supporta il tipo di concessione AUTHORIZATION_CODE per. OAuth2

- Questo tipo di concessione è considerato a tre livelli in OAuth quanto si basa sul reindirizzamento degli utenti al server di autorizzazione di terze parti per autenticare l'utente. Viene utilizzato durante la creazione di connessioni tramite la console. AWS Glue
- Gli utenti possono scegliere di creare la propria app connessa in Instagram Ads e fornire il proprio ID cliente e il segreto del cliente quando creano connessioni tramite la AWS Glue console. In questo scenario, verranno comunque reindirizzati a Instagram Ads per accedere e autorizzare l'accesso AWS Glue alle proprie risorse.
- Questo tipo di concessione genera un token di accesso. Un token utente di sistema in scadenza è valido per 60 giorni dalla data di generazione o aggiornamento. Per creare continuità, lo sviluppatore deve aggiornare il token di accesso entro 60 giorni. In caso contrario, il token di accesso verrà perso e lo sviluppatore dovrà ottenerne un altro per poter accedere di nuovo all'API. Consulta Refresh Access Token.

Per configurare una connessione Instagram Ads:

1. In AWS Glue Glue Studio, crea una connessione in Connessioni dati seguendo i passaggi seguenti:
   a. In Tipo di connessione, seleziona Instagram Ads.
   b. Seleziona il ruolo AWS IAM che AWS Glue può assumere e dispone delle autorizzazioni per le seguenti azioni:

      JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
```

```
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:DescribeSecret",
        "secretsmanager:GetSecretValue",
        "secretsmanager:PutSecretValue",
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DeleteNetworkInterface"
      ],
      "Resource": "*"
    }
  ]
}
```

c. Fornisci l'ID client dell'applicazione client gestita dall'utente.

d. Seleziona quello `secretName` che desideri utilizzare per questa connessione per AWS Glue inserire i token. Il segreto selezionato deve avere una chiave `USER_MANAGED_CLIENT_APPLICATION_CLIENT_SECRET` il cui valore sia il segreto client dell'app connessa.

e. Seleziona le opzioni di rete se desideri utilizzare la tua rete.

2. Concedi il permesso di lettura `secretName` al ruolo IAM associato al tuo AWS Glue lavoro.

## Lettura dalle entità di Instagram Ads

Prerequisito

Un oggetto di Instagram Ads da cui si desidera leggere. È necessario il nome dell'oggetto. Le tabelle seguenti mostrano le entità supportate.

Entità supportate come origine:

| Entità | Può essere filtrato | Supporta limite | Supporta Ordina per | Supporta Seleziona * | Supporta il partizionamento |
|---|---|---|---|---|---|
| Campagna | Sì | Sì | No | Sì | Sì |
| Set di annunci | Sì | Sì | No | Sì | Sì |

| Entità | Può essere filtrato | Supporta limite | Supporta Ordina per | Supporta Seleziona * | Supporta il partizionamento |
|---|---|---|---|---|---|
| Inserzioni | Sì | Sì | No | Sì | Sì |
| Creative Ad | No | Sì | No | Sì | No |
| Informazioni approfondite: conto | No | Sì | No | Sì | No |
| Immagine inserzione | Sì | Sì | No | Sì | No |
| Informazioni approfondite: inserzione | Sì | Sì | No | Sì | Sì |
| Approfondimenti - AdSet | Sì | Sì | No | Sì | Sì |
| Informazioni approfondite: campagna | Sì | Sì | No | Sì | Sì |

Esempio:

```
instagramAds_read = glueContext.create_dynamic_frame.from_options(
    connection_type="instagramads",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "entityName",
        "API_VERSION": "v20.0"
    }
```

Dettagli dell'entità e del campo di Instagram Ads

Per ulteriori informazioni sui dettagli dell'entità e del campo, consultare:

- Campagna
- Set di annunci
- Inserzione
- Creative Ad
- Informazioni approfondite sull'account pubblicitario
- Immagine inserzione
- Informazioni approfondite inserzione
- AdSets Approfondimenti
- Informazioni approfondite campagna

Per ulteriori informazioni, consulta l'API Marketing.

> **Note**
>
> I tipi di dati Struct e List vengono convertiti in tipo di dati String nella risposta dei connettori.

Query di partizionamento

Puoi fornire le opzioni Spark aggiuntive `PARTITION_FIELD`, `LOWER_BOUND`, `UPPER_BOUND` e `NUM_PARTITIONS` se desideri utilizzare la concorrenza in Spark. Con questi parametri, la query originale viene suddivisa in `NUM_PARTITIONS` query secondarie che possono essere eseguite contemporaneamente dalle attività Spark.

- `PARTITION_FIELD`: il nome del campo da utilizzare per partizionare la query.
- `LOWER_BOUND`: un valore limite inferiore inclusivo del campo di partizione scelto.

  Per il DateTime campo, accettiamo il formato di timestamp Spark utilizzato nelle query SQL di Spark.

  Esempio di valore valido:

  ```
  "2022-01-01T00:00:00.000Z"
  ```

- UPPER_BOUND: un valore limite superiore esclusivo del campo di partizione scelto.

  Esempio di valore valido:

  ```
  "2024-01-02T00:00:00.000Z"
  ```

- NUM_PARTITIONS: il numero di partizioni.

Esempio:

```
instagramADs_read = glueContext.create_dynamic_frame.from_options(
    connection_type="instagramads",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "entityName",
        "API_VERSION": "v20.0",
        "PARTITION_FIELD": "created_time"
        "LOWER_BOUND": "2022-01-01T00:00:00.000Z"
        "UPPER_BOUND": "2024-01-02T00:00:00.000Z"
        "NUM_PARTITIONS": "10"
    }
```

## Opzioni di connessione su Instagram Ads

Di seguito sono riportate le opzioni di connessione per Instagram Ads:

- ENTITY_NAME(Stringa): (obbligatorio) utilizzato per la lettura. Il nome dell'oggetto in Instagram Ads.
- API_VERSION(Stringa): (obbligatorio) utilizzato per la lettura. Versione dell'API Graph che vuoi utilizzare su Instagram Ads. Ad esempio: v21.
- SELECTED_FIELDS(List<String>): predefinito: vuoto (SELECT *). Utilizzato per la lettura. Colonne da selezionare per l'oggetto.
- FILTER_PREDICATE(Stringa): predefinito: vuoto. Utilizzato per la lettura. Dev'essere in formato Spark SQL.
- QUERY(Stringa): predefinito: vuoto. Utilizzato per la lettura. Query Spark SQL completa.
- PARTITION_FIELD(Stringa): utilizzato per la lettura. Campo da utilizzare per partizionare la query.
- LOWER_BOUND(Stringa): utilizzato per la lettura. Un valore limite inferiore inclusivo del campo di partizione scelto.

- `UPPER_BOUND`(Stringa): utilizzato per la lettura. Un valore limite superiore esclusivo del campo di partizione scelto.
- `NUM_PARTITIONS`(Numero intero): predefinito: 1. Utilizzato per la lettura. Numero di partizioni da leggere.

### Limitazioni e note per il connettore di Instagram Ads

Di seguito sono riportate le limitazioni o le note per il connettore di Instagram Ads:

- Il conteggio delle chiamate di un'app è il numero di chiamate che un utente può effettuare durante una finestra continua di un'ora moltiplicando 200 per il numero di utenti. Per i dettagli sui limiti di velocità, consulta Limiti della frequenza e Limiti della frequenza dei casi d'uso business.

## Connessione a Intercom in AWS Glue Studio

Intercom è il sistema operativo Engagement, un canale aperto tra l'azienda e i clienti, sul prodotto, sul momento e alle condizioni stabilite, che crea un dialogo continuo che consente di sfruttare al meglio ogni coinvolgimento lungo tutto il percorso del cliente.

Argomenti

- AWS Glue supporto per Intercom
- Policy contenenti le operazioni API per creare e utilizzare le connessioni
- Configurazione di Intercom
- Configurazione delle connessioni Intercom
- Lettura da entità Intercom
- Opzioni di connessione Intercom
- Limitazioni
- Creazione di un nuovo account di Intercom e configurazione dell'app client

### AWS Glue supporto per Intercom

AWS Glue supporta Intercom come segue:

Supportato come origine?

Sì. È possibile utilizzare i job AWS Glue ETL per interrogare i dati da Intercom.

Supportato come destinazione?

No.

Versioni API di Intercom supportate

v2.5. Per il supporto delle entità per versione specifica, consulta Lettura da entità Intercom.

## Policy contenenti le operazioni API per creare e utilizzare le connessioni

Nella seguente policy di esempio sono descritte le autorizzazioni IAM per creare e utilizzare le connessioni. Se si sta creando un nuovo ruolo, creare una policy che contenga quanto segue:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:ListConnectionTypes",
        "glue:DescribeConnectionType",
        "glue:RefreshOAuth2Tokens",
        "glue:ListEntities",
        "glue:DescribeEntity"
      ],
      "Resource": "*"
    }
  ]
}
```

È anche possibile utilizzare le seguenti policy IAM per autorizzare l'accesso:

- AWSGlueServiceRole— Garantisce l'accesso alle risorse che AWS Glue i vari processi richiedono per l'esecuzione per conto dell'utente. Queste risorse includono AWS Glue Amazon S3 Amazon CloudWatch Logs, IAM e Amazon EC2. Se segui la convenzione di denominazione delle risorse specificata in questa politica, AWS Glue i processi dispongono delle autorizzazioni necessarie. Questa policy è in genere collegata ai ruoli specificati quando si definiscono crawler, processi ed endpoint di sviluppo.

- AWSGlueConsoleFullAccess— Concede l'accesso completo alle AWS Glue risorse quando un'identità a cui è allegata la policy utilizza la AWS console di gestione. Se si segue la convenzione per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questa policy è in genere associata agli utenti della AWS Glue console.

## Configurazione di Intercom

Prima di poter AWS Glue utilizzare il trasferimento da Intercom, è necessario soddisfare i seguenti requisiti:

Requisiti minimi

- Hai un account Intercom. Per ulteriori informazioni, consulta Creazione di un nuovo account di Intercom e configurazione dell'app client.
- L'account Intercom è abilitato per l'accesso alle API.
- Dovresti avere un'app nell'account sviluppatore Intercom che fornisca le credenziali del client da AWS Glue utilizzare per accedere ai tuoi dati in modo sicuro quando effettua chiamate autenticate verso il tuo account. Per ulteriori informazioni, consulta Intercom: passaggi per la creazione di nuovi account e app client.

Se soddisfi questi requisiti, sei pronto per AWS Glue connetterti al tuo account Intercom.

## Configurazione delle connessioni Intercom

Intercom supporta il tipo di `AUTHORIZATION_CODE` concessione per OAuth 2.

Questo tipo di concessione è considerato «a tre gambe» in OAuth quanto si basa sul reindirizzamento degli utenti al server di autorizzazione di terze parti per autenticare l'utente. Viene utilizzato durante la creazione di connessioni tramite la console. AWS Glue La AWS Glue Console reindirizzerà l'utente a Google Ads, dove l'utente deve effettuare il login e consentire AWS Glue le autorizzazioni richieste per accedere alla propria istanza Intercom.

Gli utenti devono fornire il proprio ID client e il proprio client secret quando creano connessioni tramite la Console. AWS Glue In questo scenario, verranno comunque reindirizzati a Intercom per effettuare il login e autorizzare l'accesso AWS Glue alle proprie risorse.

Questo tipo di concessione genera un token di aggiornamento e un token di accesso. Il token di accesso dura poco tempo e può essere aggiornato automaticamente senza l'interazione dell'utente utilizzando il token di aggiornamento.

Per ulteriori informazioni sulla creazione di un'app connessa per il OAuth flusso del codice di autorizzazione, consulta Ads API.

Per configurare una connessione Intercom:

1. In AWS Secrets Manager, crea un segreto con i seguenti dettagli. È necessario creare un segreto per ogni connessione in AWS Glue.
   a. Per l'app connessa gestita dal cliente, il segreto deve contenere il token di accesso all'app connessa, il token di aggiornamento, client_id e client_secret.
2. In AWS Glue Glue Studio, crea una connessione in Connessioni dati seguendo i passaggi seguenti:
   a. In Tipo di connessione, seleziona Intercom.
   b. Fornisci l'ambiente Intercom.
   c. Seleziona il ruolo IAM che AWS Glue può assumere e dispone delle autorizzazioni per le seguenti azioni:

      JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:DescribeSecret",
        "secretsmanager:GetSecretValue",
        "secretsmanager:PutSecretValue",
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DeleteNetworkInterface"
      ],
      "Resource": "*"
    }
  ]
}
```

   d. Seleziona quello `secretName` che desideri utilizzare per questa connessione per AWS Glue inserire i token.
   e. Selezionare le opzioni di rete se si desidera utilizzare la propria rete.
3. Concedi il permesso di lettura `secretName` al ruolo IAM associato al tuo AWS Glue lavoro.

## Lettura da entità Intercom

### Prerequisiti

- Un oggetto di Intercom da cui desideri leggere. Consultare la tabella delle entità supportate qui di seguito per verificare quelle disponibili.

### Entità supportate

| Entità | API_Vversione | Può essere Filtrato | Supporta Limite | Supporta Ordina per | Supporta Seleziona * | Supporta Partizionamento |
|---|---|---|---|---|---|---|
| Admins | v2.5 | No | No | No | Sì | No |
| Aziende | v2.5 | No | Sì | No | Sì | No |
| Conversazioni | v2.5 | Sì | Sì | Sì | Sì | Sì |
| Attributi dei dati | v2.5 | No | No | No | Sì | No |
| Contatti | v2.5 | Sì | Sì | Sì | Sì | Sì |
| Segmenti | v2.5 | No | No | No | Sì | No |
| Tag | v2.5 | No | No | No | Sì | No |
| Team | v2.5 | No | No | No | Sì | No |

### Esempio

```
Intercom_read = glueContext.create_dynamic_frame.from_options(
    connection_type="Intercom",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "company",
        "API_VERSION": "V2.5"
    }
```

```
)
```

Dettagli dell'entità e del campo di Intercom

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| Admins | tipo | Stringa | N/A |
| Admins | id | Stringa | N/A |
| Admins | avatar | Struct | N/A |
| Admins | nome | Stringa | N/A |
| Admins | e-mail | Stringa | N/A |
| Admins | away_mode_enabled | Booleano | N/A |
| Admins | away_mode_reassign | Booleano | N/A |
| Admins | has_inbox_seat | Booleano | N/A |
| Admins | teams_ids | List | N/A |
| Admins | job_title | Stringa | N/A |
| Aziende | tipo | Stringa | N/A |
| Aziende | id | Stringa | N/A |
| Aziende | app_id | Stringa | N/A |
| Aziende | created_at | DateTime | N/A |
| Aziende | remote_created_at | DateTime | N/A |
| Aziende | updated_at | DateTime | N/A |
| Aziende | last_request_at | DateTime | N/A |
| Aziende | pianificazione | Struct | N/A |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| Aziende | company_id | Stringa | N/A |
| Aziende | nome | Stringa | N/A |
| Aziende | custom_attributes | Struct | N/A |
| Aziende | session_count | Numero intero | N/A |
| Aziende | monthly_spend | Numero intero | N/A |
| Aziende | user_count | Numero intero | N/A |
| Aziende | Industria | Stringa | N/A |
| Aziende | formato | Numero intero | N/A |
| Aziende | website | Stringa | N/A |
| Aziende | tag | Struct | N/A |
| Aziende | segmenti | Struct | N/A |
| Contatti | id | Stringa | EQUAL_TO. NOT_EQUAL_TO |
| Contatti | tipo | Stringa | N/A |
| Contatti | workspace_id | Stringa | N/A |
| Contatti | external_id | Stringa | CONTAINS, EQUAL_TO, NOT_EQUAL_TO |
| Contatti | role | Stringa | EQUAL_TO. NOT_EQUAL_TO |
| Contatti | e-mail | Stringa | CONTAINS, EQUAL_TO, NOT_EQUAL_TO |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| Contatti | telefono | Stringa | CONTAINS, EQUAL_TO, NOT_EQUAL_TO |
| Contatti | nome | Stringa | CONTAINS, EQUAL_TO, NOT_EQUAL_TO |
| Contatti | avatar | Stringa | N/A |
| Contatti | owner_id | Numero intero | EQUAL_TO, NOT_EQUAL_TO, GREATER_THAN, LESS_THAN |
| Contatti | social_profiles | Struct | N/A |
| Contatti | has_hard_bounced | Booleano | EQUAL_TO |
| Contatti | marked_email_as_spam | Booleano | EQUAL_TO |
| Contatti | unsubscribed_from_emails | Booleano | EQUAL_TO |
| Contatti | created_at | DateTime | EQUAL_TO, GREATER_THAN, LESS_THAN |
| Contatti | updated_at | DateTime | EQUAL_TO, GREATER_THAN, LESS_THAN |
| Contatti | signed_up_at | DateTime | EQUAL_TO, GREATER_THAN, LESS_THAN |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| Contatti | last_seen_at | DateTime | EQUAL_TO, GREATER_THAN, LESS_THAN |
| Contatti | last_replied_at | DateTime | EQUAL_TO, GREATER_THAN, LESS_THAN |
| Contatti | last_contacted_at | DateTime | EQUAL_TO, GREATER_THAN, LESS_THAN |
| Contatti | last_email_opened_at | DateTime | EQUAL_TO, GREATER_THAN, LESS_THAN |
| Contatti | last_email_clicked_at | DateTime | EQUAL_TO, GREATER_THAN, LESS_THAN |
| Contatti | language_override | Stringa | CONTAINS, EQUAL_TO, NOT_EQUAL_TO |
| Contatti | browser | Stringa | CONTAINS, EQUAL_TO, NOT_EQUAL_TO |
| Contatti | browser_version | Stringa | CONTAINS, EQUAL_TO, NOT_EQUAL_TO |
| Contatti | browser_language | Stringa | CONTAINS, EQUAL_TO, NOT_EQUAL_TO |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| Contatti | so | Stringa | CONTAINS, EQUAL_TO, NOT_EQUAL_TO |
| Contatti | location | Struct | N/A |
| Contatti | location_country | Stringa | CONTAINS, EQUAL_TO, NOT_EQUAL_TO |
| Contatti | location_region | Stringa | CONTAINS, EQUAL_TO, NOT_EQUAL_TO |
| Contatti | location_city | Stringa | CONTAINS, EQUAL_TO, NOT_EQUAL_TO |
| Contatti | android_app_name | Stringa | CONTAINS, EQUAL_TO, NOT_EQUAL_TO |
| Contatti | android_app_version | Stringa | N/A |
| Contatti | android_device | Stringa | N/A |
| Contatti | android_os_version | Stringa | N/A |
| Contatti | android_sdk_version | Stringa | CONTAINS, EQUAL_TO, NOT_EQUAL_TO |
| Contatti | android_last_seen_at | Data | N/A |
| Contatti | ios_app_name | Stringa | CONTAINS, EQUAL_TO, NOT_EQUAL_TO |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| Contatti | ios_app_version | Stringa | N/A |
| Contatti | ios_device | Stringa | N/A |
| Contatti | ios_os_version | Stringa | CONTAINS, EQUAL_TO, NOT_EQUAL_TO |
| Contatti | ios_sdk_version | Stringa | CONTAINS, EQUAL_TO, NOT_EQUAL_TO |
| Contatti | ios_last_seen_at | DateTime | N/A |
| Contatti | custom_attributes | Struct | N/A |
| Contatti | tag | Struct | N/A |
| Contatti | notes | Struct | N/A |
| Contatti | aziende | Struct | N/A |
| Contatti | unsubscribed_from_sms | Booleano | N/A |
| Contatti | sms_consent | Booleano | N/A |
| Contatti | opted_out_subscription_types | Struct | N/A |
| Contatti | referrer | Stringa | N/A |
| Contatti | utm_campaign | Stringa | N/A |
| Contatti | utm_content | Stringa | N/A |
| Contatti | utm_medium | Stringa | N/A |
| Contatti | utm_source | Stringa | N/A |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| Contatti | utm_term | Stringa | N/A |
| Conversazioni | tipo | Stringa | N/A |
| Conversazioni | id | Numero intero | EQUAL_TO, NOT_EQUAL_TO, GREATER_THAN, LESS_THAN |
| Conversazioni | created_at | DateTime | EQUAL_TO, NOT_EQUAL_TO, GREATER_THAN, LESS_THAN |
| Conversazioni | updated_at | DateTime | EQUAL_TO, NOT_EQUAL_TO, GREATER_THAN, LESS_THAN |
| Conversazioni | origine | Struct | N/A |
| Conversazioni | source_id | Stringa | EQUAL_TO, NON_EQUAL_TO |
| Conversazioni | source_type | Stringa | EQUAL_TO, NOT_EQUAL_TO, |
| Conversazioni | source_delivered_as | Stringa | EQUAL_TO, NOT_EQUAL_TO, |
| Conversazioni | source_subject | Stringa | CONTAINS, EQUAL_TO, NOT_EQUAL_TO |
| Conversazioni | source_body | Stringa | CONTAINS, EQUAL_TO, NOT_EQUAL_TO |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| Conversazioni | source_author_id | Stringa | CONTAINS, EQUAL_TO, NOT_EQUAL_TO |
| Conversazioni | source_author_type | Stringa | CONTAINS, EQUAL_TO, NOT_EQUAL_TO |
| Conversazioni | source_author_name | Stringa | CONTAINS, EQUAL_TO, NOT_EQUAL_TO |
| Conversazioni | source_author_email | Stringa | CONTAINS, EQUAL_TO, NOT_EQUAL_TO |
| Conversazioni | source_url | Stringa | CONTAINS, EQUAL_TO, NOT_EQUAL_TO |
| Conversazioni | contatti | Struct | N/A |
| Conversazioni | compagni di squadra | Struct | N/A |
| Conversazioni | titolo | Stringa | N/A |
| Conversazioni | admin_assignee_id | Numero intero | EQUAL_TO, NOT_EQUAL_TO, GREATER_THAN, LESS_THAN |
| Conversazioni | team_assignee_id | Numero intero | CONTAINS, EQUAL_TO, NOT_EQUAL_TO |
| Conversazioni | custom_attributes | Struct | N/A |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| Conversazioni | aperto | Booleano | EQUAL_TO |
| Conversazioni | stato | Stringa | CONTAINS, EQUAL_TO, NOT_EQUAL_TO |
| Conversazioni | read | Booleano | EQUAL_TO |
| Conversazioni | waiting_since | DateTime | EQUAL_TO, NOT_EQUAL_TO, GREATER_THAN, LESS_THAN |
| Conversazioni | snoozed_until | DateTime | EQUAL_TO, NOT_EQUAL_TO, GREATER_THAN, LESS_THAN |
| Conversazioni | tag | Struct | N/A |
| Conversazioni | first_contact_reply | Struct | N/A |
| Conversazioni | priority | Stringa | EQUAL_TO, NON_EQUAL_TO |
| Conversazioni | topics | Struct | N/A |
| Conversazioni | sla_applied | Struct | N/A |
| Conversazioni | conversation_rating | Struct | N/A |
| Conversazioni | conversation_rating_requested_at | DateTime | EQUAL_TO, NOT_EQUAL_TO, GREATER_THAN, LESS_THAN |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| Conversazioni | conversation_rating_replied_at | DateTime | EQUAL_TO, NOT_EQUAL_TO, GREATER_THAN, LESS_THAN |
| Conversazioni | conversation_rating_score | Numero intero | EQUAL_TO, NOT_EQUAL_TO, GREATER_THAN, LESS_THAN |
| Conversazioni | conversation_rating_remark | Stringa | CONTAINS, EQUAL_TO, NOT_EQUAL_TO |
| Conversazioni | conversation_rating_contact_id | Stringa | CONTAINS, EQUAL_TO, NOT_EQUAL_TO |
| Conversazioni | conversation_rating_admin_id | Stringa | CONTAINS, EQUAL_TO, NOT_EQUAL_TO |
| Conversazioni | statistiche | Struct | N/A |
| Conversazioni | statistics_time_to_assignment | Numero intero | EQUAL_TO, NOT_EQUAL_TO, GREATER_THAN, LESS_THAN |
| Conversazioni | statistics_time_to_admin_reply | Numero intero | EQUAL_TO, NOT_EQUAL_TO, GREATER_THAN, LESS_THAN |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| Conversazioni | statistics_time_to_first_close | Numero intero | EQUAL_TO, NOT_EQUAL_TO, GREATER_THAN, LESS_THAN |
| Conversazioni | statistics_time_to_last_close | Numero intero | EQUAL_TO, NOT_EQUAL_TO, GREATER_THAN, LESS_THAN |
| Conversazioni | statistics_median_time_to_reply | Numero intero | EQUAL_TO, NOT_EQUAL_TO, GREATER_THAN, LESS_THAN |
| Conversazioni | statistics_first_contact_reply_at | DateTime | EQUAL_TO, NOT_EQUAL_TO, GREATER_THAN, LESS_THAN |
| Conversazioni | statistics_first_assignment_at | DateTime | EQUAL_TO, NOT_EQUAL_TO, GREATER_THAN, LESS_THAN |
| Conversazioni | statistics_first_admin_reply_at | DateTime | EQUAL_TO, NOT_EQUAL_TO, GREATER_THAN, LESS_THAN |
| Conversazioni | statistics_first_close_at | DateTime | EQUAL_TO, NOT_EQUAL_TO, GREATER_THAN, LESS_THAN |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| Conversazioni | statistics_last_assignment_at | DateTime | EQUAL_TO, NOT_EQUAL_TO, GREATER_THAN, LESS_THAN |
| Conversazioni | statistics_last_assignment_admin_reply_at | DateTime | EQUAL_TO, NOT_EQUAL_TO, GREATER_THAN, LESS_THAN |
| Conversazioni | statistics_last_contact_reply_at | DateTime | EQUAL_TO, NOT_EQUAL_TO, GREATER_THAN, LESS_THAN |
| Conversazioni | statistics_last_admin_reply_at | DateTime | EQUAL_TO, NOT_EQUAL_TO, GREATER_THAN, LESS_THAN |
| Conversazioni | statistics_last_close_at | DateTime | EQUAL_TO, NOT_EQUAL_TO, GREATER_THAN, LESS_THAN |
| Conversazioni | statistics_last_closed_by_id | Stringa | CONTAINS, EQUAL_TO, NOT_EQUAL_TO |
| Conversazioni | statistics_count_reopens | Numero intero | EQUAL_TO, NOT_EQUAL_TO, GREATER_THAN, LESS_THAN |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| Conversazioni | statistics_count_assignments | Numero intero | EQUAL_TO, NOT_EQUAL_TO, GREATER_THAN, LESS_THAN |
| Conversazioni | statistics_count_conversation_parts | Numero intero | EQUAL_TO, NOT_EQUAL_TO, GREATER_THAN, LESS_THAN |
| Conversazioni | conversation_parts | List | N/A |
| Attributi dei dati | id | Numero intero | N/A |
| Attributi dei dati | tipo | Stringa | N/A |
| Attributi dei dati | modello | Stringa | N/A |
| Attributi dei dati | nome | Stringa | N/A |
| Attributi dei dati | full_name | Stringa | N/A |
| Attributi dei dati | etichetta | Stringa | N/A |
| Attributi dei dati | description | Stringa | N/A |
| Attributi dei dati | data_type | Stringa | N/A |
| Attributi dei dati | options | List | N/A |
| Attributi dei dati | api_writable | Booleano | N/A |
| Attributi dei dati | ui_writable | Booleano | N/A |
| Attributi dei dati | personalizzato | Booleano | N/A |
| Attributi dei dati | archived | Booleano | N/A |
| Attributi dei dati | created_at | Booleano | N/A |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| Attributi dei dati | updated_at | DateTime | N/A |
| Attributi dei dati | admin_id | Stringa | N/A |
| Segmenti | tipo | Stringa | N/A |
| Segmenti | id | Stringa | N/A |
| Segmenti | nome | Stringa | N/A |
| Segmenti | created_at | DateTime | N/A |
| Segmenti | updated_at | DateTime | N/A |
| Segmenti | person_type | Stringa | N/A |
| Segmenti | count | Numero intero | N/A |
| Tag | tipo | Stringa | N/A |
| Tag | id | Stringa | N/A |
| Tag | nome | Stringa | N/A |
| Team | tipo | Stringa | N/A |
| Team | id | Stringa | N/A |
| Team | nome | Stringa | N/A |
| Team | admin_ids | List | N/A |

Query di partizionamento

Se si desidera utilizzare la simultaneità in Spark, possono essere fornite opzioni Spark `PARTITION_FIELD`, `LOWER_BOUND`, `UPPER_BOUND`, `NUM_PARTITIONS` aggiuntive. Con questi parametri, la query originale viene suddivisa in `NUM_PARTITIONS` query secondarie che possono essere eseguite contemporaneamente dalle attività Spark.

- PARTITION_FIELD: il nome del campo da utilizzare per partizionare la query.
- LOWER_BOUND: un valore limite inferiore inclusivo del campo di partizione scelto.

  Per la data, accettiamo il formato di data di Spark utilizzato nelle query di Spark SQL. Esempio di valori validi: "2024-02-06".
- UPPER_BOUND: un valore limite superiore esclusivo del campo di partizione scelto.
- NUM_PARTITIONS: numero di partizioni.

I dettagli di supporto del campo di partizionamento per entità sono riportati nella tabella seguente.

| Nome entità | Campo di partizionamento | Tipo di dati |
|---|---|---|
| Contatti | created_at, updated_at,last_seen_at | DateTime |
| Conversazioni | id | Numero intero |
| Conversazioni | created_at, updated_at | DateTime |

Esempio

```
Intercom_read = glueContext.create_dynamic_frame.from_options(
    connection_type="Intercom",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "conversation",
        "API_VERSION": "V2.5",
        "PARTITION_FIELD": "created_at"
        "LOWER_BOUND": "2022-07-13T07:55:27.065Z"
        "UPPER_BOUND": "2022-08-12T07:55:27.065Z"
        "NUM_PARTITIONS": "2"
    }
)
```

## Opzioni di connessione Intercom

Di seguito sono riportate le opzioni di connessione per Intercom:

- ENTITY_NAME(String): (obbligatorio) utilizzato per la lettura. Il nome dell'oggetto in Intercom.

- `API_VERSION`(String): (obbligatorio) utilizzato per la lettura. Versione di Rest API di Intercom da utilizzare. Esempio: v2.5.
- `SELECTED_FIELDS`(List<String>): predefinito: vuoto (SELECT *). Utilizzato per la lettura. Colonne da selezionare per l'oggetto.
- `FILTER_PREDICATE`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Dev'essere in formato Spark SQL.
- `QUERY`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Query Spark SQL completa.
- `PARTITION_FIELD`(Stringa): utilizzato per la lettura. Campo da utilizzare per partizionare la query.
- `LOWER_BOUND`(Stringa): utilizzato per la lettura. Un valore limite inferiore inclusivo del campo di partizione scelto.
- `UPPER_BOUND`(Stringa): utilizzato per la lettura. Un valore limite superiore esclusivo del campo di partizione scelto.
- `NUM_PARTITIONS`(Numero intero): predefinito: 1. Utilizzato per la lettura. Numero di partizioni da leggere.
- `INSTANCE_URL`(Stringa): URL dell'istanza in cui l'utente desidera eseguire le operazioni. Ad esempio: https://api.intercom.io.

## Limitazioni

Di seguito sono riportate le limitazioni per il connettore di Intercom:

- Quando si utilizza l'entità aziendale, è possibile restituire un limite di 10.000 aziende. Per ulteriori informazioni, consulta l'API Elenca tutte le aziende.
- Durante l'applicazione di “Order By”, il filtro è obbligatorio sia per le entità Contatto che per le entità Conversazione.
- MCA è supportato dal provider SaaS. Tuttavia, in base ai limiti di velocità delle API menzionati nella documentazione, non ospiteremo MCA in AWS Glue quanto potrebbe influire su altri carichi di lavoro e potenzialmente causare problemi di prestazioni a causa della contesa delle risorse.

## Creazione di un nuovo account di Intercom e configurazione dell'app client

Creazione di un account Intercom

1. Scegli l'URL di Intercom e scegli Inizia la prova gratuita nell'angolo in alto a destra della pagina.
2. Scegli il pulsante Prova gratis nell'angolo in alto a destra della pagina.

3. Scegli il tipo di attività di cui hai bisogno.
4. Immetti tutte le informazioni richieste nella pagina.
5. Dopo aver inserito tutte le informazioni, scegli Iscriviti.

Creazione di un'app per sviluppatori Intercom

Per ottenere l'ID client e il segreto client devi creare un account sviluppatore.

1. Vai su https://app.intercom.com/.
2. Inserisci l'ID e-mail e la password/Accedi utilizzando Google e accedi.
3. Scegli il profilo utente nell'angolo in basso a sinistra e scegli le impostazioni.
4. Scegli App e integrazione.
5. Scegli la scheda Developer Hub in App e integrazione.
6. Scegli Nuova app e crea l'app qui.
7. Fornisci il nome dell'app e scegli Crea app.
8. All'interno dell'app, vai alla sezione Autenticazione.
9. Scegli la modifica e aggiungi il reindirizzamento URIs. Aggiungi l'URL di reindirizzamento specifico della tua regione come `https://<aws-region>.console.aws.amazon.com/gluestudio/oauth`. Per un esempio, aggiungi `https://us-east-1.console.aws.amazon.com/gluestudio/oauth for the us-east-1 region`.
10. Ottieni l'ID client e il segreto client generati nella sezione Informazioni di base.

## Connessione a Jira Cloud

Jira Cloud è una piattaforma sviluppata da Atlassian. La piattaforma include prodotti per il monitoraggio dei problemi che aiutano i team a pianificare e monitorare i progetti Agile. Se sei un utente Jira Cloud, l'account contiene dati sui tuoi progetti, ad esempio problemi, flussi di lavoro ed eventi. Puoi utilizzarlo AWS Glue per trasferire i dati di Jira Cloud a determinati AWS servizi o altre applicazioni supportate.

Argomenti

- AWS Glue supporto per Jira Cloud

- Policy contenenti le operazioni API per creare e utilizzare le connessioni
- Configurazione Jira Cloud
- Configurazione delle connessioni Jira Cloud
- Lettura da entità Jira Cloud
- Opzioni di connessione Jira Cloud
- Limitazioni e note per il connettore Jira Cloud

## AWS Glue supporto per Jira Cloud

AWS Glue supporta Jira Cloud come segue:

Supportato come origine?

Sì. Puoi utilizzare i job AWS Glue ETL per interrogare i dati da Jira Cloud.

Supportato come destinazione?

No.

Versioni dell'API Jira Cloud supportate

Sono supportate le seguenti versioni API Jira Cloud:

- v3

Per il supporto delle entità per versione specifica, consulta Lettura da entità Jira Cloud.

## Policy contenenti le operazioni API per creare e utilizzare le connessioni

La seguente policy di esempio descrive le autorizzazioni AWS IAM richieste per la creazione e l'utilizzo delle connessioni. Se si sta creando un nuovo ruolo, creare una policy che contenga quanto segue:

JSON

```
{
  "Version":"2012-10-17",
```

```
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:ListConnectionTypes",
        "glue:DescribeConnectionType",
        "glue:RefreshOAuth2Tokens",
        "glue:ListEntities",
        "glue:DescribeEntity"
      ],
      "Resource": "*"
    }
  ]
}
```

In alternativa, se non desideri utilizzare il metodo precedente, utilizza le seguenti policy IAM gestite:

- AWSGlueServiceRole— Garantisce l'accesso alle risorse che AWS Glue i vari processi richiedono per l'esecuzione per conto dell'utente. Queste risorse includono AWS Glue Amazon S3, IAM, CloudWatch Logs e Amazon. EC2 Se segui la convenzione di denominazione delle risorse specificata in questa politica, AWS Glue i processi dispongono delle autorizzazioni richieste. Questa policy è in genere collegata ai ruoli specificati quando si definiscono crawler, processi ed endpoint di sviluppo.
- AWSGlueConsoleFullAccess— Concede l'accesso completo alle AWS Glue risorse quando un'identità a cui è allegata la policy utilizza la AWS console di gestione. Se si segue la convenzione per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questa policy è in genere associata agli utenti della AWS Glue console.

## Configurazione Jira Cloud

Prima di poterli utilizzare AWS Glue per trasferire dati da Jira Cloud alle destinazioni supportate, devi soddisfare questi requisiti:

Requisiti minimi

I requisiti minimi sono i seguenti:

- Hai un account Atlassian in cui utilizzi il prodotto software Jira in Jira Cloud. Per ulteriori informazioni, consulta Creazione di un account Jira Cloud.

- È necessario disporre di un AWS account creato con il servizio di accesso a AWS Glue.
- Questa app fornisce le credenziali del client che AWS Glue utilizza per accedere ai tuoi dati in modo sicuro quando effettua chiamate autenticate al tuo account. Per ulteriori informazioni, consulta Enabling OAuth 2.0 (3LO) nella documentazione di Atlassian Developer.

Se soddisfi questi requisiti, sei pronto per connetterti AWS Glue al tuo account Jira Cloud.

Creazione di un account Jira Cloud

Per creare un account Jira Cloud:

1. Vai all'URL di registrazione di Atlassian.
2. Inserisci l'e-mail di lavoro e il tuo nome, quindi scegli Accetto. Riceverai un'e-mail di verifica.
3. Dopo aver verificato la tua e-mail, puoi creare una password e scegliere Registrati.
4. Immetti il nome utente e la password, quindi scegli Registrati.
5. Verrai reindirizzato a una pagina in cui è necessario accedere al sito. Inserisci il nome del sito e scegli Accetto.

Una volta avviato il sito Atlassian Cloud, puoi configurare Jira rispondendo ad alcune domande in base alle preferenze del tipo di progetto.

Per accedere a un account esistente:

1. Vai all'URL di accesso di Atlassian e inserisci le credenziali.
2. Inserisci e-mail e password e fai clic su Accedi. Verrai reindirizzato alla dashboard Jira.

Creazione di un'app su Jira Cloud

Per creare un'app su Jira Cloud e ottenere l'ID client e il segreto client dall'app client gestita:

1. Vai all'URL di Jira Cloud e inserisci le credenziali.
2. Scegli Crea e seleziona l'opzione di integrazione OAuth 2.0.
3. Inserisci il nome dell'app, controlla Termini e condizioni e scegli Crea.
4. Vai alla sezione Distribuzione nel menu a sinistra e scegli Modifica.
5. Nella sezione Modifica i controlli di distribuzione:

a. Seleziona STATO DISTRIBUZIONE come Condivisione.
b. Inserisci il nome del fornitore.
c. Inserisci l'URL per la tua Informativa sulla privacy. Ad esempio, https://docs.aws.amazon.com/glue/ latest/dg/security -iam-awsmanpol.html
d. Inserisci l'URL per i tuoi Termini di servizio (facoltativo).
e. Inserisci l'URL del contatto per l'assistenza clienti (facoltativo).
f. Seleziona Yes/No dalla DICHIARAZIONE DEI DATI PERSONALI e scegli Salva modifiche.

6. Vai su Autorizzazioni nel menu a sinistra della rispettiva app.
7. Per l'API Jira, scegli Aggiungi. Una volta aggiunta, scegli l'opzione Configurazione.
8. In Ambiti classici > REST API della piattaforma Jira, scegli Modifica ambiti e controlla tutti gli ambiti. Fai clic su Salva.
9. In Ambiti granulari scegli Modifica ambiti e seleziona i seguenti:
10. Scorri verso il basso e trova gli ambiti. È necessario selezionare due tipi di ambiti nelle intestazioni "CRM" e "Standard".
11. Aggiungi i seguenti ambiti:

```
read:application-role:jira
read:audit-log:jira
read:avatar:jira
read:field:jira
read:group:jira
read:instance-configuration:jira
read:issue-details:jira
read:issue-event:jira
read:issue-link-type:jira
read:issue-meta:jira
read:issue-security-level:jira
read:issue-security-scheme:jira
read:issue-type-scheme:jira
read:issue-type-screen-scheme:jira
read:issue-type:jira
read:issue.time-tracking:jira
read:label:jira
read:notification-scheme:jira
read:permission:jira
read:priority:jira
read:project:jira
read:project-category:jira
```

```
read:project-role:jira
read:project-type:jira
read:project-version:jira
read:project.component:jir
read:project.property:jira
read:resolution:jira
read:screen:jira
read:status:jira
read:user:jira
read:workflow-scheme:jira
read:workflow:jira
read:field-configuration:jira
read:issue-type-hierarchy:jira
read:webhook:jira
```

12. Vai su Autenticazione nel menu a sinistra e scegli Aggiungi.
13. Inserisci un URL di callback come oauth https://us-east-1.console.aws.amazon.com/gluestudio/
14. Vai su Impostazioni nel menu a sinistra e scorri verso il basso per i dettagli di autenticazione. Prendi nota dell'ID client e del segreto.

## Configurazione delle connessioni Jira Cloud

Jira Cloud supporta il tipo di concessione AUTHORIZATION_CODE per. OAuth2

- Questo tipo di concessione è considerato «a tre gambe» in OAuth quanto si basa sul reindirizzamento degli utenti a un server di autorizzazione di terze parti per autenticare l'utente. Viene utilizzato durante la creazione di connessioni tramite la console. AWS Glue La AWS Glue console reindirizzerà l'utente a Jira Cloud, dove l'utente deve effettuare il login e consentire AWS Glue le autorizzazioni richieste per accedere alla propria istanza Jira Cloud.
- Gli utenti possono comunque scegliere di creare la propria app connessa in Jira Cloud e fornire il proprio ID client e il segreto del client durante la creazione di connessioni tramite la console. AWS Glue In questo scenario, verranno comunque reindirizzati a Jira Cloud per effettuare il login e autorizzare l'accesso AWS Glue alle proprie risorse.
- Questo tipo di concessione genera un token di aggiornamento e un token di accesso. Il token di accesso dura poco tempo e può essere aggiornato automaticamente senza l'interazione dell'utente utilizzando il token di aggiornamento.
- Per la documentazione pubblica di Jira Cloud sulla creazione di un'app connessa per il OAuth flusso del codice di autorizzazione, consulta Enabling OAuth 2.0 (3LO).

Per configurare una connessione Jira Cloud:

1. In AWS Secrets Manager, crea un segreto con i seguenti dettagli:
   a. Per le app connesse gestite dal cliente: il segreto deve contenere l'app connessa Consumer Secret con `USER_MANAGED_CLIENT_APPLICATION_CLIENT_SECRET` come chiave.
   b. Nota: è necessario creare un segreto per la connessione in AWS Glue.
2. In AWS Glue Glue Studio, crea una connessione in Connessioni dati seguendo i passaggi seguenti:
   a. In Tipo di connessione, seleziona Jira Cloud.
   b. Fornisci l'ambiente Jira Cloud.
   c. Seleziona il ruolo AWS IAM che AWS Glue può assumere e dispone delle autorizzazioni per le seguenti azioni:

      JSON

      ```
      {
        "Version":"2012-10-17",
        "Statement": [
          {
            "Effect": "Allow",
            "Action": [
              "secretsmanager:DescribeSecret",
              "secretsmanager:GetSecretValue",
              "secretsmanager:PutSecretValue",
              "ec2:CreateNetworkInterface",
              "ec2:DescribeNetworkInterfaces",
              "ec2:DeleteNetworkInterface"
            ],
            "Resource": "*"
          }
        ]
      }
      ```

   d. Seleziona quello `secretName` che desideri utilizzare per questa connessione per AWS Glue inserire i token.
   e. Selezionare le opzioni di rete se si desidera utilizzare la propria rete.
3. Concedi il permesso di lettura `secretName` al ruolo IAM associato al tuo AWS Glue lavoro.

## Lettura da entità Jira Cloud

Prerequisito

Un oggetto di Jira Cloud da cui desideri leggere. È necessario il nome dell'oggetto, come Record audit, o Problema. Nella tabella seguente sono illustrate le entità supportate.

Entità supportate come origine:

| Entità | Può essere filtrato | Supporta limite | Supporta Ordina per | Supporta Seleziona * | Supporta il partizionamento |
|---|---|---|---|---|---|
| Record audit | Sì | Sì | No | Sì | Sì |
| Problema | Sì | Sì | No | Sì | Sì |
| Campo Problema | No | No | No | Sì | No |
| Configurazione del campo Problema | Sì | Sì | No | Sì | Sì |
| Tipo di link del problema | No | No | No | Sì | No |
| Schema di notifica del problema | Sì | Sì | No | Sì | Sì |
| Schema di sicurezza del problema | No | No | No | Sì | No |
| Schema del tipo di problema | Sì | Sì | Sì | Sì | Sì |

| Entità | Può essere filtrato | Supporta limite | Supporta Ordina per | Supporta Seleziona * | Supporta il partizionamento |
|---|---|---|---|---|---|
| Schema del tipo di schermo del problema | Sì | Sì | Sì | Sì | Sì |
| Tipo di problema | No | No | No | Sì | No |
| Impostazioni Jira | Sì | No | No | Sì | No |
| Impostazioni avanzate di Jira | No | No | No | Sì | No |
| Impostazioni generali di Jira | No | No | No | Sì | No |
| Etichetta | No | No | No | Sì | Sì |
| Me stesso | Sì | No | No | Sì | No |
| Autorizzazione | No | No | No | Sì | No. |
| Progetto | Sì | Sì | Sì | Sì | Sì |
| Categoria di progetto | No | No | No | Sì | No |
| Tipo di progetto | No | No | No | Sì | No |

| Entità | Può essere filtrato | Supporta limite | Supporta Ordina per | Supporta Seleziona * | Supporta il partizionamento |
|---|---|---|---|---|---|
| Informazioni sul server | No | No | No | Sì | No |
| Utenti | No | No | No. | Sì | No |
| Flusso di lavoro | Sì | Sì | Sì | Sì | Sì |
| Schema del flusso di lavoro | No | Sì | No | Sì | Sì |
| Associazione di progetti dello schema del flusso di lavoro | Sì | No | No | Sì | No |
| Stato del flusso di lavoro | No | No | No | Sì | No |
| Categoria di stato del flusso di lavoro | No | No | No | Sì | No |

Esempio:

```
jiracloud_read = glueContext.create_dynamic_frame.from_options(
    connection_type="JiraCloud",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "audit-record",
        "API_VERSION": "v3"
```

```
}
```

Dettagli dell'entità e del campo di Jira Cloud:

| Oggetto | Campo | Tipo di dati | Operatori di filtro supportati |
|---|---|---|---|
| Record audit | filter | Stringa | "=" |
| | from | DateTime | "=" |
| | in | DateTime | "=" |
| | id | Numero intero | N/D |
| | riepilogo | Stringa | N/A |
| | remoteAddress | Stringa | N/A |
| | authorAccountId | Stringa | N/A |
| | created | Stringa | N/A |
| | category | Stringa | N/A |
| | eventSource | Stringa | N/A |
| | description | Stringa | N/A |
| | objectItem | Struct | N/D |
| | changedValues | List | N/D |
| | associatedItems | List | N/D |
| Gruppi | groupName | List | "=" |
| | nome | Stringa | N/A |
| | groupId | Stringa | "=" |
| Problema | affectedVersion | Stringa | "=, !=" |

| Oggetto | Campo | Tipo di dati | Operatori di filtro supportati |
|---|---|---|---|
| | assegnatario | Stringa | "=, !=" |
| | category | Stringa | "=, !=" |
| | componente | Stringa | "=, !=" |
| | creatore | Stringa | "=, !=" |
| | scadenza | DateTime | N/D |
| | epic_link | Stringa | "=, !=" |
| | filter | Stringa | "=, !=" |
| | fixVersion | Stringa | "=, !=" |
| | hierarchyLevel | Numero intero | "=, !=" |
| | issueKey | Stringa | "=, !=, >, <, >=, <=" |
| | issueLink | Stringa | "=, !=" |
| | issueLinkType | Stringa | "=, !=" |
| | labels | Stringa | "=, !=" |
| | lastViewed | DateTime | "=, >, <, >=, <=, between" |
| | level | Stringa | "=, !=" |
| | parent | Stringa | "=, !=" |
| | priority | Stringa | "=, !=" |
| | project | Stringa | "=, !=" |
| | projectType | Stringa | "=, !=" |

| Oggetto | Campo | Tipo di dati | Operatori di filtro supportati |
|---|---|---|---|
| | reporter | Stringa | "=, !=" |
| | risoluzione | Stringa | "=, !=" |
| | risolto | DateTime | "=, >, <, >=, <=, between" |
| | sprint | Stringa | "=, !=" |
| | status | Stringa | "=, !=" |
| | tipo | Stringa | "=, !=" |
| | updated | DateTime | "=, >, <, >=, <=, between" |
| | votante | Stringa | "=, !=" |
| | voti | Numero intero | "=, !=, <, >, <=, >=, between" |
| | osservatore | Stringa | "=, !=" |
| | watcher | Numero intero | "=, !=, <, >, <=, >=, between" |
| | workRatio | Numero intero | "=, !=, <, >, <=, >=, between" |
| | validateQuery | Stringa | "=" |
| | espansione | Stringa | "=" |
| | fieldByKeys | Booleano | "=" |
| | id | Stringa | N/A |
| | self | Stringa | N/A |

| Oggetto | Campo | Tipo di dati | Operatori di filtro supportati |
|---|---|---|---|
| | key | Stringa | N/A |
| | renderedFields | Struct | N/D |
| | properties | List | "=" |
| | nomi | Struct | N/D |
| | schema | Struct | N/D |
| | transizioni | List | N/D |
| | operazioni | Struct | N/D |
| | editmeta | Struct | N/D |
| | changelog | Struct | N/D |
| | versionedRepresentations | Struct | N/D |
| | campi | List | "=" |
| | fieldsToInclude | Struct | N/D |
| | warningMessages | List | N/D |
| | created | DateTime | N/D |
| | worklogDate | DateTime | N/D |
| IssueEvents | id | Numero intero | N/D |
| | nome | Stringa | N/A |
| Issue Fields | id | Stringa | N/A |
| | key | Stringa | N/A |

| Oggetto | Campo | Tipo di dati | Operatori di filtro supportati |
|---|---|---|---|
| | nome | Stringa | N/A |
| | personalizzato | Booleano | N/D |
| | ordinabile | Booleano | N/D |
| | navigabile | Booleano | N/D |
| | searchable | Booleano | N/D |
| | clauseNames | List | N/D |
| | scope | Struct | N/D |
| | schema | Struct | N/D |
| Configurazioni del campo Problema | IsDefault | Booleano | "=" |
| | query | Stringa | "=" |
| | id | Numero intero | "=" |
| | nome | Stringa | N/A |
| | description | Stringa | N/A |
| Tipo di link del problema | id | Stringa | N/A |
| | nome | Stringa | N/A |
| | verso l'interno | Stringa | N/A |
| | verso l'esterno | Stringa | N/A |
| | self | Stringa | N/A |
| Schemi di notifica del problema | espansione | Stringa | "=" |
| | self | Stringa | N/A |

| Oggetto | Campo | Tipo di dati | Operatori di filtro supportati |
|---|---|---|---|
| | id | Numero intero | N/D |
| | nome | Stringa | N/A |
| | description | Stringa | N/A |
| | notificationSchemeEvents | List | N/D |
| | scope | Struct | N/D |
| Priorità del problema | self | Stringa | N/A |
| | statusColor | Stringa | N/A |
| | description | Stringa | N/A |
| | iconUrl | Stringa | N/A |
| | nome | Stringa | N/A |
| | id | Stringa | N/A |
| | isDefault | Booleano | N/D |
| Risoluzione del problema | self | Stringa | N/A |
| | id | Stringa | N/A |
| | description | Stringa | N/A |
| | nome | Stringa | N/A |
| Schema di sicurezza del problema | self | Stringa | N/A |
| | id | Numero intero | N/D |
| | nome | Stringa | N/A |

| Oggetto | Campo | Tipo di dati | Operatori di filtro supportati |
|---|---|---|---|
| | description | Stringa | N/A |
| | defaultSecurityLevelId | Numero intero | N/D |
| | livelli | List | N/D |
| Tipo di problema | self | Stringa | N/A |
| | id | Stringa | N/A |
| | description | Stringa | N/A |
| | iconUrl | Stringa | N/A |
| | nome | Stringa | N/A |
| | attività secondaria | Booleano | N/D |
| | avatarId | Numero intero | N/D |
| | entityId | Stringa | N/A |
| | hierarchyLevel | Numero intero | N/D |
| | scope | Struct | N/D |
| Schema del tipo di problema | orderBy | Stringa | "=" |
| | espansione | Stringa | "=" |
| | queryString | Stringa | "=" |
| | id | Stringa | N/A |
| | nome | Stringa | N/A |
| | description | Stringa | N/A |
| | defaultIssueTypeId | Stringa | N/A |

| Oggetto | Campo | Tipo di dati | Operatori di filtro supportati |
|---|---|---|---|
| | isDefault | Booleano | N/D |
| Schema del tipo di schermo del problema | queryString | Stringa | "=" |
| | orderBy | Stringa | "=" |
| | espansione | Stringa | "=" |
| | id | Stringa | "=" |
| | nome | Stringa | N/A |
| | description | Stringa | N/A |
| Impostazioni Jira | key | Stringa | N/A |
| | keyFilter | Stringa | "=" |
| | id | Stringa | N/A |
| | value | Stringa | N/A |
| | nome | Stringa | N/A |
| | desc | Stringa | N/A |
| | tipo | Stringa | N/A |
| | defaultValue | Stringa | N/A |
| | example | Stringa | N/A |
| | allowedValues | List | N/D |
| Impostazioni avanzate di Jira | id | Stringa | N/A |
| | key | Stringa | N/A |
| | value | Stringa | N/A |

| Oggetto | Campo | Tipo di dati | Operatori di filtro supportati |
|---|---|---|---|
| | nome | Stringa | N/A |
| | desc | Stringa | N/A |
| | tipo | Stringa | N/A |
| | defaultValue | Stringa | N/A |
| | example | Stringa | N/A |
| | allowedValues | List | N/D |
| Impostazioni generali di Jira | votingEnabled | Booleano | N/D |
| | watchingEnabled | Booleano | N/D |
| | unassignedIssuesAllowed | Booleano | N/D |
| | subTasksEnabled | Booleano | N/D |
| | issueLinkingEnabled | Booleano | N/D |
| | timeTrackingEnabled | Booleano | N/D |
| | attachmentsEnabled | Booleano | N/D |
| | timeTrackingConfiguration | Struct | N/D |
| Etichetta | values | List | N/D |
| Me stesso | espansione | Stringa | "=" |
| | self | Stringa | N/A |
| | accountId | Stringa | N/A |
| | accountType | Stringa | N/A |

| Oggetto | Campo | Tipo di dati | Operatori di filtro supportati |
|---|---|---|---|
| | emailAddress | Stringa | N/A |
| | avatarUrls | Stringa | N/A |
| | displayName | Stringa | N/A |
| | attiva | Booleano | N/D |
| | timezone | Stringa | N/A |
| | locale | Stringa | N/A |
| | gruppi | Struct | N/D |
| | applicationRoles | Struct | N/D |
| Autorizzazione | id | Stringa | N/A |
| | key | Stringa | N/A |
| | nome | Stringa | N/A |
| | tipo | Stringa | N/A |
| | description | Stringa | N/A |
| | havePermission | Booleano | N/D |
| | deprecatedKey | Booleano | N/D |
| Progetto | orderBy | Stringa | "=" |
| | keys | List | "=" |
| | query | Stringa | "=" |
| | typeKey | Stringa | "=" |
| | categoryId | Numero intero | "=" |

| Oggetto | Campo | Tipo di dati | Operatori di filtro supportati |
|---|---|---|---|
| | operazione | Stringa | "=" |
| | espansione | Stringa | "=" |
| | status | List | "=" |
| | self | Stringa | N/A |
| | id | Numero intero | "=" |
| | key | Stringa | N/A |
| | description | Stringa | N/A |
| | lead | Struct | N/D |
| | componenti | List | N/D |
| | issueTypes | List | N/D |
| | url | Stringa | N/A |
| | e-mail | Stringa | N/A |
| | assigneeType | Stringa | N/A |
| | versioni | List | N/D |
| | nome | Stringa | N/A |
| | roles | Struct | N/D |
| | avatarUrls | Struct | N/D |
| | projectCategory | Struct | N/D |
| | projectTypeKey | Stringa | N/A |
| | simplified | Booleano | N/D |

| Oggetto | Campo | Tipo di dati | Operatori di filtro supportati |
|---|---|---|---|
| | stile | Stringa | N/A |
| | preferito | Booleano | N/D |
| | isPrivate | Booleano | N/D |
| | issueTypeHierarchy | Struct | N/D |
| | autorizzazioni | Struct | N/D |
| | properties | List | "=" |
| | uuid | Stringa | N/A |
| | informazione dettagliata | Struct | N/D |
| | deleted (eliminato) | Booleano | N/D |
| | retentionTillDate | Stringa | N/A |
| | deletedDate | Stringa | N/A |
| | deletedBy | Struct | N/D |
| | archived | Booleano | N/D |
| | archivedDate | Stringa | N/A |
| | archivedBy | Struct | N/D |
| | landedPageInfo | Struct | N/D |
| Categoria di progetto | self | Stringa | N/A |
| | id | Stringa | N/A |
| | nome | Stringa | N/A |

| Oggetto | Campo | Tipo di dati | Operatori di filtro supportati |
|---|---|---|---|
| | description | Stringa | N/A |
| Tipo di progetto | key | Stringa | N/A |
| | formattedKey | Stringa | N/A |
| | description | Stringa | N/A |
| | descriptionI18nKey | Stringa | N/A |
| | icon | Stringa | N/A |
| | color | Stringa | N/A |
| Informazioni sul server | baseUrl | Stringa | N/A |
| | version | Stringa | N/A |
| | versionNumbers | List | N/D |
| | deploymentType | Stringa | N/A |
| | buildNumber | Numero intero | N/D |
| | buildDate | DateTime | N/D |
| | serverTime | DateTime | N/D |
| | scmInfo | Stringa | N/A |
| | serverTitle | Stringa | N/A |
| | healthChecks | List | N/D |
| Utenti | self | Stringa | N/A |
| | accountId | Stringa | N/A |
| | accountType | Stringa | N/A |

| Oggetto | Campo | Tipo di dati | Operatori di filtro supportati |
|---|---|---|---|
| | emailAddress | Stringa | N/A |
| | avatarUrls | Struct | N/D |
| | displayName | Stringa | N/A |
| | attiva | Booleano | N/D |
| | timezone | Stringa | N/A |
| | locale | Stringa | N/A |
| | gruppi | Struct | N/D |
| | applicationRoles | Struct | N/D |
| | espansione | Stringa | N/A |
| Flusso di lavoro | workflowName | Stringa | "=" |
| | espansione | Stringa | "=" |
| | queryString | Stringa | "=" |
| | orderBy | Stringa | "=" |
| | isActive | Booleano | "=" |
| | id | Struct | N/D |
| | description | Stringa | N/A |
| | transizioni | List | N/D |
| | stati | List | N/D |
| | isDefault | Booleano | N/D |
| | schemi | List | N/D |

| Oggetto | Campo | Tipo di dati | Operatori di filtro supportati |
|---|---|---|---|
| | progetti | List | N/D |
| | hasDraftWorkflow | Booleano | N/D |
| | operazioni | Struct | N/D |
| | created | Stringa | N/A |
| | updated | Stringa | N/A |
| Schema del flusso di lavoro | self | Stringa | N/A |
| | id | Numero intero | N/D |
| | nome | Stringa | N/A |
| | description | Stringa | N/A |
| | defaultWorkflow | Stringa | N/A |
| | issueTypeMappings | Struct | N/D |
| | originalDefaultWorkflow | Stringa | N/A |
| | originalIssueTypeMappature | Struct | N/D |
| | bozza | Booleano | N/D |
| | lastModifiedUser | Struct | N/D |
| | lastModified | Stringa | N/A |
| | updateDraftIfServe | Booleano | N/D |
| | issueTypes | Struct | N/D |

| Oggetto | Campo | Tipo di dati | Operatori di filtro supportati |
|---|---|---|---|
| Associazione di progetti dello schema del flusso di lavoro | projectId | Numero intero | "=" |
| | projectIds | List | N/D |
| | workflowScheme | Struct | N/D |
| Stato del flusso di lavoro | self | Stringa | N/A |
| | description | Stringa | N/A |
| | iconUrl | Stringa | N/A |
| | nome | Stringa | N/A |
| | id | Stringa | N/A |
| | StatusCategory | Struct | N/D |
| Categoria di stato del flusso di lavoro | self | Stringa | N/A |
| | id | Stringa | N/A |
| | key | Stringa | N/A |
| | colorName | Stringa | N/A |
| | nome | Stringa | N/A |

Query di partizionamento

È possibile fornire l'opzione Spark aggiuntiva `NUM_PARTITIONS` se si desidera utilizzare la simultaneità in Spark. Con questo parametro, la query originale viene suddivisa in `NUM_PARTITIONS` query secondarie che possono essere eseguite contemporaneamente dalle attività Spark.

- `NUM_PARTITIONS`: il numero di partizioni.

Esempio:

```
jiraCloud_read = glueContext.create_dynamic_frame.from_options(
    connection_type="JiraCloud",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "issue",
        "API_VERSION": "v3",
        "NUM_PARTITIONS": "10"
    }
```

### Opzioni di connessione Jira Cloud

Di seguito sono riportate le opzioni di connessione per Jira Cloud:

- `ENTITY_NAME`(String): (obbligatorio) utilizzato per la lettura. Il nome del tuo oggetto in Jira Cloud.
- `API_VERSION`(String): (obbligatorio) utilizzato per la lettura. Versione dell'API Rest di Jira Cloud da utilizzare. Ad esempio: v3.
- `DOMAIN_URL`(Stringa): (obbligatorio) l'ID Jira Cloud che desideri utilizzare.
- `SELECTED_FIELDS`(List<String>): predefinito: vuoto (SELECT *). Utilizzato per la lettura. Colonne da selezionare per l'oggetto.
- `FILTER_PREDICATE`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Dev'essere in formato Spark SQL.
- `QUERY`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Query Spark SQL completa.
- `NUM_PARTITIONS`(Integer): predefinito: 1. Utilizzato per la lettura. Numero di partizioni da leggere.

### Limitazioni e note per il connettore Jira Cloud

Di seguito sono riportate le limitazioni o le note per il connettore Jira Cloud:

- L'`Contains`operatore non funziona con il `resourceName` campo, che è di tipo di `String` dati.
- Per impostazione predefinita, se non viene applicato alcun filtro esplicito, verranno sottoposti a scansione solo i problemi degli ultimi 30 giorni. Gli utenti hanno la possibilità di sostituire questo filtro predefinito specificando un filtro personalizzato.

## Connessione a Kustomer

Kustomer è una potente piattaforma per l'esperienza dei clienti che riunisce tutto ciò di cui hai bisogno per servire meglio i tuoi clienti in un unico strumento. easy-to-use

Argomenti


- AWS Glue supporto per Kustomer
- Policy contenenti le operazioni API per creare e utilizzare le connessioni
- Configurazione di Kustomer
- Configurazione delle connessioni Kustomer
- Lettura da entità Kustomer
- Opzioni di connessione Kustomer
- Limitazioni Kustomer


## AWS Glue supporto per Kustomer

AWS Glue supporta Kustomer come segue:

Supportato come origine?

Sì. È possibile utilizzare i job AWS Glue ETL per interrogare i dati di Kustomer.

Supportato come destinazione?

No.

Versioni API Kustomer supportate

Sono supportate le seguenti versioni API di Kustomer:

- v1

## Policy contenenti le operazioni API per creare e utilizzare le connessioni

La seguente policy di esempio descrive le autorizzazioni AWS IAM richieste per la creazione e l'utilizzo delle connessioni. Se si sta creando un nuovo ruolo, creare una policy che contenga quanto segue:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
```

```
      "Effect": "Allow",
      "Action": [
        "glue:ListConnectionTypes",
        "glue:DescribeConnectionType",
        "glue:RefreshOAuth2Tokens",
        "glue:ListEntities",
        "glue:DescribeEntity"
      ],
      "Resource": "*"
    }
  ]
}
```

In alternativa, se non desideri utilizzare il metodo precedente, utilizza le seguenti policy IAM gestite:

- AWSGlueServiceRole— Garantisce l'accesso alle risorse che AWS Glue i vari processi richiedono per l'esecuzione per conto dell'utente. Queste risorse includono AWS Glue Amazon S3, IAM, CloudWatch Logs e Amazon. EC2 Se segui la convenzione di denominazione delle risorse specificata in questa politica, AWS Glue i processi dispongono delle autorizzazioni richieste. Questa policy è in genere collegata ai ruoli specificati quando si definiscono crawler, processi ed endpoint di sviluppo.
- AWSGlueConsoleFullAccess— Concede l'accesso completo alle AWS Glue risorse quando un'identità a cui è allegata la policy utilizza la AWS console di gestione. Se si segue la convenzione per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questa policy è in genere associata agli utenti della AWS Glue console.

## Configurazione di Kustomer

Prima di poterli utilizzare AWS Glue per trasferire dati da Kustomer alle destinazioni supportate, devi soddisfare questi requisiti:

Requisiti minimi

I requisiti minimi sono i seguenti:

- Hai un account con Kustomer che contiene i dati che desideri trasferire.
- Nelle impostazioni del tuo account, hai creato una chiave API. Per ulteriori informazioni, consulta Creazione di una chiave API.
- Fornisci la chiave API a AWS Glue durante la creazione della connessione.

Se soddisfi questi requisiti, sei pronto per connetterti AWS Glue al tuo account Kustomer.

Creazione di una chiave API

Per creare una chiave API da utilizzare per creare una connessione per il connettore Kustomer in Studio: AWS Glue

1. Accedi alla dashboard di Kustomer utilizzando le tue credenziali.
2. Scegli l'icona Impostazioni dal menu a sinistra.
3. Espandi il menu a discesa Sicurezza e seleziona Chiavi API.
4. Nella pagina di creazione della chiave API, seleziona Aggiungi una chiave API nell'angolo in alto a destra.
5. Compila gli input obbligatori per la chiave API che stai creando.
   - Nome: qualsiasi nome per la chiave API.
   - Ruoli: 'org' deve essere selezionato per far funzionare APIs Kustomer.
   - Scadenza (in giorni): il numero di giorni in cui desideri che la chiave API sia valida. Puoi conservarlo come Non scade mai, se lo ritieni adatto al tuo caso d'uso.
6. Scegli Create (Crea).
7. Memorizza il valore della chiave API (token) per un ulteriore utilizzo per creare una connessione per il connettore Kustomer in Studio. AWS Glue

## Configurazione delle connessioni Kustomer

Per configurare una connessione Kustomer:

1. In AWS Secrets Manager, crea un segreto con i seguenti dettagli:
   a. Per le app connesse gestite dal cliente, il segreto deve contenere l'app connessa Consumer Secret con `apiKey` come chiave.
   b. Nota: devi creare un segreto per le tue connessioni in AWS Glue.

1. In AWS Glue Glue Studio, crea una connessione in Connessioni dati seguendo i passaggi seguenti:
   a. In Connessioni, scegli Crea connessione.
   b. Quando selezioni un'origine dati, seleziona Kustomer.

c. Seleziona il ruolo AWS IAM che AWS Glue può assumere e dispone delle autorizzazioni per le seguenti azioni:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:DescribeSecret",
        "secretsmanager:GetSecretValue",
        "secretsmanager:PutSecretValue",
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DeleteNetworkInterface"
      ],
      "Resource": "*"
    }
  ]
}
```

d. Seleziona quello `secretName` che desideri utilizzare per questa connessione per AWS Glue inserire i token.

e. Selezionare le opzioni di rete se si desidera utilizzare la propria rete.

2. Concedi il permesso di lettura `secretName` al ruolo IAM associato al tuo AWS Glue lavoro.

## Lettura da entità Kustomer

Prerequisito

Un oggetto Kustomer da cui si desidera leggere. È necessario il nome dell'oggetto, come Marchi o Schede. Nella tabella seguente sono illustrate le entità supportate.

Entità supportate come origine:

| Entità | Può essere filtrato | Supporta limite | Supporta Ordina per | Supporta Seleziona * | Supporta il partizionamento |
|---|---|---|---|---|---|
| Marchi | No | Sì | No | Sì | No |
| Schede | No | Sì | No | Sì | No |
| Impostazioni chat | No | No | No | Sì | No |
| Aziende | Sì | Sì | Sì | Sì | Sì |
| Conversazioni | Sì | Sì | Sì | Sì | Sì |
| Clienti | Sì | Sì | Sì | Sì | Sì |
| Ricerche ancorate dei clienti | No | Sì | No | Sì | No |
| Posizione di ricerca del cliente | No | No | No | Sì | No |
| Hook e-mail | No | Sì | No | Sì | No |
| Hook Web | No | Sì | No | Sì | No |
| Articoli KB | No | Sì | No | Sì | No |
| Categorie KB | No | Sì | No | Sì | No |
| Moduli KB | No | Sì | No | Sì | No |
| Percorsi KB | No | Sì | No | Sì | No |
| Tag KB | No | Sì | No | Sì | No |

| Entità | Può essere filtrato | Supporta limite | Supporta Ordina per | Supporta Seleziona * | Supporta il partizionamento |
|---|---|---|---|---|---|
| Modelli KB | No | Sì | No | Sì | No |
| Temi KB | No | Sì | No | Sì | No |
| Klasses | No | Sì | No | Sì | No |
| KViews | No | Sì | No | Sì | No |
| Messaggi | Sì | Sì | Sì | Sì | Sì |
| Note | Sì | Sì | Sì | Sì | Sì |
| Notifications | No | Sì | No | Sì | No |

Esempio:

```
Kustomer_read = glueContext.create_dynamic_frame.from_options(
    connection_type="kustomer",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "brands",
        "API_VERSION": "v1"
    }
```

Dettagli dell'entità e del campo di Kustomer

Per ulteriori informazioni sui dettagli dell'entità e del campo, consultare:

- Marchi
- Carte
- Impostazioni chat
- Aziende
- Conversazioni
- Clienti

- Ricerche ancorate dei clienti
- Posizione di ricerca del cliente
- Hook e-mail
- Hook Web
- Articoli KB
- Categorie KB
- Moduli KB
- Percorsi KB
- Tag KB
- Modelli KB
- Temi KB
- Klasses
- KViews
- Messaggi
- Note
- Notifiche

API Kustomer v1

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| Marchi | id | Stringa | N/A |
| | nome | Stringa | N/A |
| | iconUrl | Stringa | N/A |
| | createdAt | DateTime | N/D |
| | updatedAt | DateTime | N/D |
| | modifiedAt | DateTime | N/D |
| | default | Booleano | N/D |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| Schede | id | Stringa | N/A |
| | nome | Stringa | N/A |
| | createdAt | DateTime | N/D |
| | updatedAt | DateTime | N/D |
| | modifiedAt | DateTime | N/D |
| | description | Stringa | N/A |
| | url | Stringa | N/A |
| | contesti | List | N/D |
| Impostazioni chat | id | Stringa | N/A |
| | settingsVersion | Numero intero | N/D |
| | widgetType | Stringa | N/A |
| | version | Numero intero | N/D |
| | teamName | Stringa | N/A |
| | saluto | Stringa | N/A |
| | autoreply | Stringa | N/A |
| | embedIconUrl | Stringa | N/A |
| | embedIconColor | Stringa | N/A |
| | fallbackEmailSubject | Stringa | N/A |
| | fallbackEmailIntroduction | Stringa | N/A |
| | enabled | Booleano | N/D |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | outboundChatEnabled | Booleano | N/D |
| | updatedAt | DateTime | N/D |
| | modifiedAt | DateTime | N/D |
| | offhoursMessage | Stringa | N/A |
| | offhoursImageUrl | Stringa | N/A |
| | closableChat | Booleano | N/D |
| | noHistory | Booleano | N/D |
| | disableAttachments | Booleano | N/D |
| | volumeControl | Struct | N/D |
| | singleSessonChat | Booleano | N/D |
| | showTypingIndicatorWeb | Booleano | N/D |
| Aziende | id | Stringa | N/A |
| | nome | Stringa | =, !=, CONTAINS |
| | createdAt | DateTime | N/D |
| | updatedAt | DateTime | N/D |
| | modifiedAt | DateTime | =, !=, <, <=, >, >=, BETWEEN |
| | tag | List | N/D |
| | domains | List | N/D |
| | e-mail | List | N/D |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | telefoni | List | N/D |
| | whatsapp | List | N/D |
| | social | List | N/D |
| | url | List | N/D |
| | posizioni | List | N/D |
| | roleGroupVersions | List | N/D |
| | rev | Numero intero | N/D |
| Conversazioni | id | Stringa | N/A |
| | nome | Stringa | =, !=, CONTAINS |
| | anteprima | Stringa | N/A |
| | canali | List | N/D |
| | status | Stringa | =, !=, CONTAINS |
| | messageCount | Numero intero | =, !=, >, >=, <, <= |
| | noteCount | Numero intero | =, !=, >, >=, <, <= |
| | soddisfazione | Numero intero | =, !=, >, >=, <, <= |
| | satisfactionLevel | Struct | N/D |
| | createdAt | DateTime | =, !=, <, <=, >, >=, BETWEEN |
| | updatedAt | DateTime | =, !=, <, <=, >, >=, BETWEEN |
| | modifiedAt | DateTime | =, !=, <, <=, >, >=, BETWEEN |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | lastActivityAt | DateTime | N/D |
| | spam | Booleano | N/D |
| | terminato | Booleano | =, != |
| | endedAt | DateTime | =, !=, <, <=, >, >=, BETWEEN |
| | endedReason | Stringa | CONTAINS |
| | endedByType | Stringa | N/A |
| | importedAt | Stringa | N/A |
| | tag | List | N/D |
| | suggestedTags | List | N/D |
| | sentiment | Stringa | N/A |
| | previsioni | List | N/D |
| | suggestedShortcuts | List | N/D |
| | firstMessageIn | Struct | N/D |
| | firstMessageOut | Struct | N/D |
| | lastMessageIn | Struct | N/D |
| | lastMessageOut | Struct | N/D |
| | lastMessageAt | DateTime | =, !=, <, <=, >, >=, BETWEEN |
| | lastMessageUnrespondedPer | Struct | N/D |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | lastMessageUnrespondedToSinceLastDone | Struct | N/D |
| | assignedUsers | List | N/D |
| | assignedTeams | List | N/D |
| | firstResponse | Struct | N/D |
| | firstResponseSinceLastDone | Struct | N/D |
| | lastResponse | Struct | N/D |
| | firstDone | Struct | N/D |
| | lastDone | Struct | N/D |
| | direzione | Stringa | =, !=, CONTAINS |
| | lastMessageDirection | Stringa | N/A |
| | outboundMessageCount | Numero intero | N/D |
| | inboundMessageCount | Numero intero | N/D |
| | rev | Numero intero | N/D |
| | priority | Numero intero | =, !=, >, >=, <, <= |
| | roleGroupVersions | List | N/D |
| | accessOverride | List | N/D |
| | assistente | Struct | N/D |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | fase | Stringa | N/A |
| | Competenze | List | N/D |
| | matchedTi meBasedRegole | List | N/D |
| Clienti | id | Stringa | N/A |
| | nome | Stringa | =, !=, CONTAINS |
| | displayName | Stringa | N/A |
| | displayColor | Stringa | N/A |
| | displayIcon | Stringa | N/A |
| | externalId | Stringa | =, !=, CONTAINS |
| | externalIds | List | N/D |
| | sharedExternalIds | List | N/D |
| | e-mail | List | N/D |
| | sharedEmails | List | N/D |
| | telefoni | List | N/D |
| | sharedPhones | List | N/D |
| | whatsapp | List | N/D |
| | facebookIds | List | N/D |
| | instagramIds | List | N/D |
| | social | List | N/D |
| | sharedSocials | List | N/D |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | url | List | N/D |
| | posizioni | List | N/D |
| | activeUsers | List | N/D |
| | watcher | List | N/D |
| | recentLocation | Struct | N/D |
| | locale | Stringa | =, !=, CONTAINS |
| | timezone | Stringa | N/A |
| | gender | Stringa | =, !=, CONTAINS |
| | createdAt | DateTime | =, !=, <, <=, >, >=, BETWEEN |
| | updatedAt | DateTime | =, !=, <, <=, >, >=, BETWEEN |
| | modifiedAt | DateTime | =, !=, <, <=, >, >=, BETWEEN |
| | lastActivityAt | DateTime | N/D |
| | deleted (eliminato) | Booleano | N/D |
| | lastConversation | Struct | N/D |
| | conversationCounts | Struct | N/D |
| | anteprima | Struct | N/D |
| | tag | List | N/D |
| | progressiveStatus | Stringa | =, !=, CONTAINS |
| | verified | Booleano | N/D |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | rev | Numero intero | N/D |
| | recentItems | List | N/D |
| | defaultLang | Stringa | =, !=, CONTAINS |
| | satisfactionLevel | Struct | N/D |
| | roleGroupVersions | List | N/D |
| | accessOverride | List | N/D |
| | companyName | Stringa | N/A |
| | firstName | Stringa | N/A |
| | lastName | Stringa | N/A |
| Ricerche ancorate dei clienti | id | Stringa | N/A |
| | cerca | Stringa | N/A |
| | createdAt | DateTime | N/D |
| Posizione di ricerca del cliente | id | Stringa | N/A |
| | posizioni | List | N/D |
| | figli | List | N/D |
| | createdAt | DateTime | N/D |
| | updatedAt | DateTime | N/D |
| | modifiedAt | DateTime | N/D |
| | rev | Numero intero | N/D |
| Hook e-mail | id | Stringa | N/A |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | description | Stringa | N/A |
| | debug | Booleano | N/D |
| | e-mail | Stringa | N/A |
| | eventName | Stringa | N/A |
| | titolo | Stringa | N/A |
| | hash | Stringa | N/A |
| | key | Stringa | N/A |
| | createdAt | DateTime | N/D |
| | modifiedAt | DateTime | N/D |
| | updatedAt | DateTime | N/D |
| Hook Web | id | Stringa | N/A |
| | description | Stringa | N/A |
| | eventName | Stringa | N/A |
| | hash | Stringa | N/A |
| | url | Stringa | N/A |
| | createdAt | DateTime | N/D |
| | modifiedAt | DateTime | N/D |
| | updatedAt | DateTime | N/D |
| | titolo | Stringa | N/A |
| | version | Numero intero | N/D |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | debug | Booleano | N/D |
| Articoli KB | id | Stringa | N/A |
| | hash | Stringa | N/A |
| | titolo | Stringa | N/A |
| | origine | Stringa | N/A |
| | status | Stringa | N/A |
| | scope | Stringa | N/A |
| | createdAt | DateTime | N/D |
| | updatedAt | DateTime | N/D |
| | deleted (eliminato) | Booleano | N/D |
| | deletedAt | DateTime | N/D |
| | modifiedAt | DateTime | N/D |
| | publishedAt | DateTime | N/D |
| | tag | List | N/D |
| | categorie | List | N/D |
| | knowledgeBases | List | N/D |
| | metaTitle | Stringa | N/A |
| | metaDescription | Stringa | N/A |
| | metaKeywords | List | N/D |
| | langVersions | Struct | N/D |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | latestLangs | Struct | N/D |
| Categorie KB | id | Stringa | N/A |
| | hash | Stringa | N/A |
| | createdAt | DateTime | N/D |
| | modifiedAt | DateTime | N/D |
| | updatedAt | DateTime | N/D |
| | pubblicato | Booleano | N/D |
| | posizioni | List | N/D |
| | categoryPositions | List | N/D |
| | root | Booleano | N/D |
| | langs | Struct | N/D |
| Moduli KB | id | Stringa | N/A |
| | nome | Stringa | N/A |
| | slug | Stringa | N/A |
| | hash | Stringa | N/A |
| | body | Stringa | N/A |
| | layout | List | N/D |
| | layoutV2 | List | N/D |
| | componentsV2 | Struct | N/D |
| | condizioni | Struct | N/D |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | avanzato | Booleano | N/D |
| | createdAt | DateTime | N/D |
| | updatedAt | DateTime | N/D |
| | publishedAt | DateTime | N/D |
| | modifiedAt | Stringa | N/A |
| | pubblicato | Booleano | N/D |
| | frammenti | List | N/D |
| | recaptcha | Booleano | N/D |
| | klass | Stringa | N/A |
| | channel | Stringa | N/A |
| | deflection | Booleano | N/D |
| | formHookEnabled | Booleano | N/D |
| | replyFrom | Stringa | N/A |
| | wcag | Booleano | N/D |
| Percorsi KB | id | Stringa | N/A |
| | url | Stringa | N/A |
| | routableType | Stringa | N/A |
| | routableId | Stringa | N/A |
| | createdAt | DateTime | N/D |
| | updatedAt | DateTime | N/D |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | modifiedAt | DateTime | N/D |
| Tag KB | id | Stringa | N/A |
| | nome | Stringa | N/A |
| | createdAt | DateTime | N/D |
| | updatedAt | DateTime | N/D |
| | modifiedAt | DateTime | N/D |
| Modelli KB | id | Stringa | N/A |
| | titolo | Stringa | N/A |
| | description | Stringa | N/A |
| | beta | Booleano | N/D |
| | manifest | Struct | N/D |
| | jsxSnippets | List | N/D |
| | images | List | N/D |
| | version | Stringa | N/A |
| | createdAt | DateTime | N/D |
| | updatedAt | DateTime | N/D |
| Temi KB | id | Stringa | N/A |
| | nome | Stringa | N/A |
| | attiva | Booleano | N/D |
| | default | Booleano | N/D |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | lastfileUpdatedAt | DateTime | N/D |
| | personalizzato | Booleano | N/D |
| | status | Stringa | N/A |
| | templateVersionId | Stringa | N/A |
| | templateTitle | Stringa | N/A |
| | templateVersion | Stringa | N/A |
| | manifest | Struct | N/D |
| | configSnippets | List | N/D |
| | jsxSnippets | List | N/D |
| | createdAt | DateTime | N/D |
| | updatedAt | DateTime | N/D |
| | modifiedAt | DateTime | N/D |
| | rev | Numero intero | N/D |
| Klasses | id | Stringa | N/A |
| | nome | Stringa | N/A |
| | icon | Stringa | N/A |
| | color | Stringa | N/A |
| | appDisabled | Booleano | N/D |
| | status | Stringa | N/A |
| | updatedAt | DateTime | N/D |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | createdAt | DateTime | N/D |
| | s3 DataUrl | Stringa | N/A |
| KViews | id | Stringa | N/A |
| | risorsa | Stringa | N/A |
| | modello | Stringa | N/A |
| | context | Stringa | N/A |
| | meta | Struct | N/D |
| | appDisabled | Booleano | N/D |
| | enabled | Booleano | N/D |
| | avanzato | Booleano | N/D |
| | layout | List | N/D |
| | componenti | Struct | N/D |
| | condizioni | Struct | N/D |
| | rev | Numero intero | N/D |
| | createdAt | DateTime | N/D |
| | modifiedAt | DateTime | N/D |
| | updatedAt | DateTime | N/D |
| Notifications | id | Stringa | N/A |
| | nome | Stringa | N/A |
| | status | Stringa | N/A |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | evento | Struct | N/D |
| | createdAt | DateTime | N/D |
| | updatedAt | DateTime | N/D |
| Messaggi | id | Stringa | N/A |
| | externalId | Stringa | N/A |
| | channel | Stringa | =, !=, CONTAINS |
| | app | Stringa | N/A |
| | formato | Numero intero | =, !=, >, >=, <, <= |
| | direzione | Stringa | =, !=, CONTAINS |
| | anteprima | Stringa | N/A |
| | subject | Stringa | N/A |
| | meta | Struct | N/D |
| | status | Stringa | =, !=, CONTAINS |
| | directionType | Stringa | =, !=, CONTAINS |
| | assignedTeams | List | N/D |
| | assignedUsers | List | N/D |
| | errorAt | DateTime | =, !=, <, <=, >, >=, BETWEEN |
| | auto | Booleano | =, != |
| | sentAt | DateTime | =, !=, <, <=, >, >=, BETWEEN |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | createdAt | DateTime | =, !=, <, <=, >, >=, BETWEEN |
| | updatedAt | DateTime | N/D |
| | modifiedAt | DateTime | N/D |
| | redatto | Booleano | N/D |
| | createdByTeams | List | N/D |
| | rev | Numero intero | N/D |
| | reazioni | List | N/D |
| | intentDetections | List | N/D |
| Note | id | Stringa | N/A |
| | body | Stringa | CONTAINS |
| | createdAt | DateTime | =, !=, <, <=, >, >=, BETWEEN |
| | updatedAt | DateTime | =, !=, <, <=, >, >=, BETWEEN |
| | modifiedAt | DateTime | =, !=, <, <=, >, >=, BETWEEN |
| | createdByTeams | List | N/D |

Query di partizionamento

Partizionamento basato sul campo

Puoi fornire le opzioni Spark aggiuntive `PARTITION_FIELD`, `LOWER_BOUND`, `UPPER_BOUND` e `NUM_PARTITIONS` se desideri utilizzare la concorrenza in Spark. Con questi parametri, la query

originale viene suddivisa in NUM_PARTITIONS query secondarie che possono essere eseguite contemporaneamente dalle attività Spark.

- PARTITION_FIELD: il nome del campo da utilizzare per partizionare la query.
- LOWER_BOUND: un valore limite inferiore inclusivo del campo di partizione scelto.

  Per il DateTime campo, accettiamo il valore in formato ISO.

  Esempio di valore valido:

  ```
  "2023-01-15T11:18:39.205Z"
  ```

- UPPER_BOUND: un valore limite superiore esclusivo del campo di partizione scelto.
- NUM_PARTITIONS: il numero di partizioni.

I dettagli di supporto del campo di partizionamento per entità sono riportati nella tabella seguente:

| Nome dell'entità | Campi di partizionamento | Tipo di dati |
|---|---|---|
| Aziende | modifiedAt | DateTime |
| Conversazioni | CreatedAt, updatedAt , ModifiedAt, EndedAt, lastMessageAt | DateTime |
| | messageCount, noteCount | BigInteger |
| | priority | Numero intero |
| Clienti | createdAt, updatedAt, modifiedAt | DateTime |
| Messaggi | errorAt, sentAt, createdAt | DateTime |
| | formato | BigInteger |
| Note | createdAt, updatedAt, modifiedAt | DateTime |

Esempio:

```
Kustomer_read = glueContext.create_dynamic_frame.from_options(
    connection_type="kustomer",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "conversation",
        "API_VERSION": "v1",
        "PARTITION_FIELD": "createdAt"
        "LOWER_BOUND": "2023-01-15T11:18:39.205Z"
        "UPPER_BOUND": "2023-02-15T11:18:39.205Z"
        "NUM_PARTITIONS": "2"
    }
```

## Opzioni di connessione Kustomer

Di seguito sono riportate le opzioni di connessione per Kustomer:

- `ENTITY_NAME`(String): (obbligatorio) utilizzato per la lettura. Il nome dell'oggetto in Kustomer.
- `API_VERSION`(String): (obbligatorio) utilizzato per la lettura. Versione di Rest API di Kustomer da utilizzare.
- `SELECTED_FIELDS`(List<String>): predefinito: vuoto (SELECT *). Utilizzato per la lettura. Colonne da selezionare per l'oggetto.
- `FILTER_PREDICATE`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Dev'essere in formato Spark SQL.
- `QUERY`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Query Spark SQL completa.
- `PARTITION_FIELD`(Stringa): utilizzato per la lettura. Campo da utilizzare per partizionare la query.
- `LOWER_BOUND`(Stringa): utilizzato per la lettura. Un valore limite inferiore inclusivo del campo di partizione scelto.
- `UPPER_BOUND`(Stringa): utilizzato per la lettura. Un valore limite superiore esclusivo del campo di partizione scelto.
- `NUM_PARTITIONS`(Numero intero): predefinito: 1. Utilizzato per la lettura. Numero di partizioni da leggere.
- `INSTANCE_URL`(Stringa): (obbligatorio) utilizzato per la lettura. URL dell'istanza del cliente.

## Limitazioni Kustomer

Di seguito sono riportate le limitazioni o le note di Kustomer:

- L'entità `Customer Searches` non è supportata poiché la documentazione dell'API Kustomer non ne ha dichiarato alcun endpoint.
- Il supporto della filtrazione e del trasferimento incrementale sull'entità `Klasses` non è supportato.
- Order By può essere supportato su più campi applicabili in un'unica richiesta.

  Tuttavia, è stato osservato che l'ordine per funzionalità su più campi si comporta in modo incoerente dal lato SaaS per alcune combinazioni. È imprevedibile in quanto potrebbero esserci "n" combinazioni che possono mostrare risultati di ordinamento errati. Esempio:

  Per l'entità `Customers`, Order By `progressiveStatus desc, name asc` non produce il risultato ordinato corretto. Viene ordinato solo in base all'ordine `progressiveStatus`. Se si osserva tale comportamento, è possibile utilizzare un singolo campo in base al quale ordinare.
- Order By sul campo "ID" è supportato solo dalle entità `Conversations` e `Messages` come parametro di query. Ad esempio: https://api.kustomerapp.com/v1/ conversazioni? sort=desc (Ordina i risultati per 'id' in ordine decrescente).

  Inoltre, qualsiasi altro filtro o ordinamento su qualsiasi altro campo viene tradotto in un corpo di richiesta POST con l'endpoint API POST https://api.kustomerapp.com/v1/ customers/search. Per consentire il supporto dell'ordinamento per «id» in `Conversations` e`Messages`, dovrebbe essere presente solo l'ordine per id o qualsiasi altro filtro ordinato per ogni altro campo applicabile. and/or
- Kustomer consente di recuperare un massimo di 10.000 record indipendentemente da una richiesta filtrata o non filtrata. A causa di questa limitazione, si verificherà una perdita di dati per qualsiasi entità che detenga più di 10.000 record. Esistono due possibili soluzioni alternative che è possibile eseguire per mitigare parzialmente questo problema:
  - Applica filtri per recuperare un set specifico di record.
  - Se ci sono più di 10.000 record con un filtro applicato, applica un valore di filtro successivo in una nuova richiesta successiva o applica intervalli nei filtri. Esempio:

    filterExpression della prima richiesta: `modifiedAt >= 2022-03-15T05:26:23.000Z and modifiedAt < 2023-03-15T05:26:23.000Z`

    Si supponga che questo esaurisca il limite di record di 10.000.

Un'altra richiesta può essere attivata con filterExpression: `modifiedAt >= 2023-03-15T05:26:23.000Z`

- Come comportamento SaaS, l'operatore `CONTAINS` di Kustomer supporta la corrispondenza solo su parole complete e non le corrispondenze parziali all'interno di una parola. Ad esempio: “body CONTAINS 'test record'” corrisponderà a un record con 'test' nel campo 'body'. Tuttavia, «body CONTAINS 'test'» non corrisponderà a un record con '' nel campo testAnotherRecord 'body'.

## Connessione a LinkedIn

LinkedIn è uno strumento di marketing a pagamento che offre l'accesso ai LinkedIn social network attraverso vari post sponsorizzati e altri metodi. LinkedIn è un potente strumento di marketing per le aziende B2B per creare lead, riconoscere online, condividere contenuti e altro ancora.

Argomenti


- AWS Glue supporto per LinkedIn
- Policy contenenti le operazioni API per creare e utilizzare le connessioni
- Configurazione LinkedIn
- Configurazione delle connessioni LinkedIn
- Lettura da LinkedIn entità
- LinkedIn opzioni di connessione
- Creare un LinkedIn account
- Limitazioni


### AWS Glue supporto per LinkedIn

AWS Glue supporta LinkedIn quanto segue:

Supportato come origine?

Sì. È possibile utilizzare i job AWS Glue ETL da LinkedIn cui interrogare i dati.

Supportato come destinazione?

No.

Versioni LinkedIn API supportate

202406 (giugno 2024)

## Policy contenenti le operazioni API per creare e utilizzare le connessioni

La seguente politica di esempio descrive le AWS autorizzazioni richieste per la creazione e l'utilizzo delle connessioni. Se si sta creando un nuovo ruolo, creare una policy che contenga quanto segue:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:ListConnectionTypes",
        "glue:DescribeConnectionType",
        "glue:RefreshOAuth2Tokens",
        "glue:ListEntities",
        "glue:DescribeEntity"
      ],
      "Resource": "*"
    }
  ]
}
```

In alternativa, se non desideri utilizzare il metodo precedente, utilizza le seguenti policy IAM:

- AWSGlueServiceRole — Garantisce l'accesso alle risorse che diversi AWS Glue processi richiedono per l'esecuzione per conto dell'utente. Queste risorse includono Amazon S3 AWS Glue, IAM, CloudWatch Logs e Amazon EC2. Se segui la convenzione di denominazione delle risorse specificata in questa politica, AWS Glue i processi dispongono delle autorizzazioni necessarie. Questa policy è in genere collegata ai ruoli specificati quando si definiscono crawler, processi ed endpoint di sviluppo.
- AWSGlueConsoleFullAccess — Concede l'accesso completo alle AWS Glue risorse quando un'identità a cui è allegata la policy utilizza la AWS console di gestione. Se si segue la convenzione per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questa policy è in genere associata agli utenti della AWS Glue console.

## Configurazione LinkedIn

Prima di poter utilizzare AWS Glue per il trasferimento da LinkedIn, è necessario soddisfare i seguenti requisiti:

Requisiti minimi

- Hai un LinkedIn account. Per ulteriori informazioni sulla creazione di un account, consulta Creazione di un LinkedIn account.
- Il tuo LinkedIn account è abilitato all'accesso tramite API.
- Hai creato un'`OAuth2 API`integrazione nel tuo LinkedIn account. Questa integrazione fornisce le credenziali del client che AWS Glue utilizza per accedere ai dati in modo sicuro quando effettua chiamate autenticate al vostro account. Per ulteriori informazioni, consulta the section called "Creare un LinkedIn account".

Se soddisfi questi requisiti, sei pronto per AWS Glue connetterti al tuo account. LinkedIn Per le connessioni tipiche, non è necessario fare nient'altro LinkedIn.

## Configurazione delle connessioni LinkedIn

LinkedIn supporta il tipo di `AUTHORIZATION_CODE` concessione per OAuth2.

Questo tipo di concessione è considerato `OAuth` "a tre gambe" in quanto si basa sul reindirizzamento degli utenti al server di autorizzazione di terzi per l'autenticazione. Gli utenti possono scegliere di creare la propria app connessa LinkedIn e fornire il proprio ID client e il segreto del client durante la creazione di connessioni tramite la AWS Glue console. In questo scenario, verranno comunque reindirizzati al login e LinkedIn all'autorizzazione ad accedere AWS Glue alle proprie risorse.

Questo tipo di concessione genera sia un token di aggiornamento che un token di accesso. Il token di accesso scade 60 giorni dopo la creazione. È possibile ottenere un nuovo token di accesso utilizzando il token di aggiornamento.

Per la LinkedIn documentazione pubblica sulla creazione di un'app connessa per `Authorization Code OAuth` flow, consulta Authorization Code Flow ( OAuth3-legged).

Configurazione di una connessione LinkedIn

1. In AWS Secrets Manager, crea un segreto con i seguenti dettagli:

- Per le app connesse gestite dal cliente: il segreto deve contenere l'app connessa Consumer Secret con `USER_MANAGED_CLIENT_APPLICATION_CLIENT_SECRET` come chiave.
- Per l'app AWS connessa gestita: segreto o segreto vuoto con un valore temporaneo.

> **Note**
>
> È necessario creare un segreto per ogni connessione in AWS Glue.

2. Nel AWS Glue Studio, crea una connessione in Connessioni dati seguendo i passaggi seguenti:

   1. Quando si seleziona un tipo di connessione, selezionare LinkedIn.
   2. Fornisci l' LinkedIn ambiente.
   3. Seleziona il ruolo IAM per il quale AWS Glue può assumere e dispone delle autorizzazioni per le seguenti azioni:

      JSON

      ```
      {
        "Version":"2012-10-17",
        "Statement": [
          {
            "Effect": "Allow",
            "Action": [
              "secretsmanager:DescribeSecret",
              "secretsmanager:GetSecretValue",
              "secretsmanager:PutSecretValue",
              "ec2:CreateNetworkInterface",
              "ec2:DescribeNetworkInterfaces",
              "ec2:DeleteNetworkInterface"
            ],
            "Resource": "*"
          }
        ]
      }
      ```

   4. Seleziona quello `secretName` che desideri utilizzare per questa connessione per AWS Glue inserire i token.
   5. Seleziona le opzioni di rete se desideri utilizzare la tua rete.

3. Concedi il permesso di lettura `secretName` al ruolo IAM associato al tuo AWS Glue lavoro.

## Lettura da LinkedIn entità

### Prerequisiti

Un LinkedIn oggetto da cui vorresti leggere. Consultare la tabella delle entità supportate qui di seguito per verificare quelle disponibili.

### Entità supportate

| Entità | Può essere Filtrato | Supporta Limite | Supporta Ordina per | Supporta Seleziona * | Supporta Partizionamento |
|---|---|---|---|---|---|
| Account pubblicitari | Sì | Sì | Sì | Sì | No |
| Campagne | Sì | Sì | Sì | Sì | No |
| Gruppi di campagne | Sì | Sì | Sì | Sì | No |
| Creativi | Sì | Sì | Sì | Sì | No |
| Analisi degli annunci | Sì | No | No | Sì | No |
| Ad Analytics All AdAcocunts | Sì | No | No | Sì | No |
| Analisi degli annunci (tutte le campagne) | Sì | No | No | Sì | No |
| Ad Analytics Tutti CampaignGroups | Sì | No | No | Sì | No |

| Entità | Può essere Filtrato | Supporta Limite | Supporta Ordina per | Supporta Seleziona * | Supporta Partizionamento |
|---|---|---|---|---|---|
| Ad Analytics Tutti AdCreatives | Sì | No | No | Sì | No |
| Condividi statistiche | Sì | No | No | Sì | No |
| Pagina delle statistiche | Sì | No | No | Sì | No |
| Statistiche dei follower | Sì | No | No | Sì | No |

Esempio

```
netsuiteerp_read = glueContext.create_dynamic_frame.from_options(
    connection_type="linkedin",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "adaccounts",
        "API_VERSION": "202406"
    }
)
```

LinkedIn dettagli sull'entità e sul campo

| Tipo di dati dei campi | Operatori di filtro supportati |
|---|---|
| Stringa | = |
| DateTime | BETWEEN, = |
| Numerico | = |
| Booleano | = |

## LinkedIn opzioni di connessione

Di seguito sono elencate le opzioni di connessione per LinkedIn:

- `ENTITY_NAME` (stringa): (obbligatorio) utilizzato per lettura/scrittura. Il nome del tuo oggetto in LinkedIn. Ad esempio, adAccounts.
- `API_VERSION`(String) — (Obbligatorio) Utilizzato per la lettura/scrittura. LinkedIn Versione dell'API Rest che desideri utilizzare. Il valore sarà 202406, poiché LinkedIn attualmente supporta solo la versione 202406.
- `SELECTED_FIELDS` (List<String>): predefinito: vuoto (SELECT *). Utilizzato per la lettura. Colonne che si desidera selezionare per l'entità selezionata.
- `FILTER_PREDICATE`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Dev'essere in formato Spark SQL.
- `QUERY`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Query Spark SQL completa.

## Creare un LinkedIn account

Creazione di un' LinkedIn app e OAuth credenziali

1. Vai alla pagina LinkedIn Developer Network e accedi con le credenziali del tuo LinkedIn account.
2. Vai alla pagina Le mie app e scegli Crea applicazione per creare una nuova LinkedIn app.
3. Inserisci i seguenti dettagli nel modulo di registrazione dell'app:
   - Nome dell'azienda: seleziona un'azienda esistente o creane una nuova.
   - Nome: inserisci il nome dell'applicazione.
   - Descrizione: inserisci la descrizione dell'applicazione.
   - Logo dell'applicazione: seleziona un file immagine come logo dell'applicazione.
   - Uso dell'applicazione: seleziona l'uso dell'applicazione.
   - URL del sito Web: inserisci l'URL del sito Web che contiene informazioni dettagliate sull'applicazione.
   - E-mail aziendale: inserisci il tuo indirizzo e-mail aziendale.
   - Telefono aziendale: inserisci il numero di telefono aziendale.
   - LinkedIn Termini di utilizzo dell'API: leggi e accetta.
4. Dopo aver completato il modulo di registrazione dell'app, scegli Invia.

Verrai reindirizzato alla pagina di autenticazione, dove verranno visualizzate le chiavi di autenticazione (ID client e segreto client) e altri dettagli pertinenti.

5. Se la tua applicazione web richiede l'accesso all'indirizzo e-mail dell'utente dal suo LinkedIn account, seleziona l'`r_emailaddress`autorizzazione. Inoltre, puoi specificare il reindirizzamento autorizzato URLs per la tua LinkedIn applicazione.

Creazione di una pagina nell'account LinkedIn

1. Vai a Prodotti per LinkedIn sviluppatori.
2. Nell'angolo in alto a destra della pagina Prodotti per LinkedIn sviluppatori, seleziona Le mie app.
3. Nell'angolo in alto a destra della pagina Le mie app, seleziona Crea app.
4. Nella pagina Crea un'app immetti il nome dell'app nel campo Nome.
5. Nel campo LinkedIn Pagina, inserisci il nome o l'URL della pagina aziendale.

> **Note**
>
> Se non hai una LinkedIn pagina, puoi crearne una selezionando Crea una nuova LinkedIn.

6. Nel campo URL dell'informativa sulla privacy, inserisci l'URL dell'Informativa sulla privacy.
7. Scegli Carica un logo per caricare un'immagine da mostrare agli utenti quando effettuano l'autorizzazione con la tua app.
8. Nella sezione Accordo legale, seleziona Ho letto e accetto questi termini.
9. Scegli Crea app.

   La tua nuova app verrà creata e sarà disponibile nella scheda Le mie app.

Pubblicazione di annunci relativi alla campagna in LinkedIn

1. Accedi a Gestione della campagna.
2. Seleziona un Gruppo campagna esistente oppure scegli Crea per crearne uno nuovo.
3. Seleziona il tuo obiettivo.
4. Seleziona il gruppo, il budget e il programma.
5. Costruisci il tuo pubblico di riferimento.

6. Seleziona il formato dell'annuncio.
7. Seleziona il budget e il programma.
8. Configura i tuoi annunci.
9. Rivedi e avvia.

### Limitazioni

Per i campi di analisi `ad_analytics_all_adAccounts`, `ad_analytics_all_campaigns`, `ad_analytics_all_campaign_groups` e `ad_analytics_all_adCreatives` è obbligatorio un filtro per recuperare i record.

## Connessione a Mailchimp

Mailchimp è una piattaforma all-in-one di marketing che ti aiuta a gestire e parlare con i tuoi clienti, clienti e altre parti interessate. Il suo approccio al marketing si concentra su pratiche di gestione dei contatti sane, e-mail dal design accattivante, eccezionali flussi di lavoro automatizzati e potenti analisi dei dati. Se sei un utente Mailchimp, puoi connetterti AWS Glue al tuo account Mailchimp. Quindi, puoi usare Mailchimp come origine dati nei tuoi processi ETL. Esegui questi processi per trasferire dati tra Mailchimp e AWS servizi o altre applicazioni supportate.

Argomenti


- AWS Glue supporto per Mailchimp
- Policy contenenti le operazioni API per creare e utilizzare le connessioni
- Configurazione di Mailchimp
- Configurazione delle connessioni di Mailchimp
- Lettura da entità di Mailchimp
- Opzioni di connessione di Mailchimp
- Creazione di un account Mailchimp
- Limitazioni


### AWS Glue supporto per Mailchimp

AWS Glue supporta Mailchimp come segue:

Supportato come origine?

Sì. È possibile utilizzare i job AWS Glue ETL per interrogare i dati da Mailchimp.

Supportato come destinazione?

No.

Versioni API di Mailchimp supportate

3.0

## Policy contenenti le operazioni API per creare e utilizzare le connessioni

La seguente politica di esempio descrive le AWS autorizzazioni richieste per la creazione e l'utilizzo delle connessioni. Se si sta creando un nuovo ruolo, creare una policy che contenga quanto segue:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:ListConnectionTypes",
        "glue:DescribeConnectionType",
        "glue:RefreshOAuth2Tokens",
        "glue:ListEntities",
        "glue:DescribeEntity"
      ],
      "Resource": "*"
    }
  ]
}
```

In alternativa, se non desideri utilizzare il metodo precedente, utilizza le seguenti policy IAM:

- AWSGlueServiceRole — Garantisce l'accesso alle risorse che diversi AWS Glue processi richiedono per l'esecuzione per conto dell'utente. Queste risorse includono Amazon S3 AWS Glue, IAM, CloudWatch Logs e Amazon EC2. Se segui la convenzione di denominazione delle risorse specificata in questa politica, AWS Glue i processi dispongono delle autorizzazioni necessarie.

Questa policy è in genere collegata ai ruoli specificati quando si definiscono crawler, processi ed endpoint di sviluppo.

- AWSGlueConsoleFullAccess — Concede l'accesso completo alle AWS Glue risorse quando un'identità a cui è allegata la politica utilizza la AWS console di gestione. Se si segue la convenzione per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questa policy è in genere associata agli utenti della AWS Glue console.

## Configurazione di Mailchimp

Prima di poter AWS Glue utilizzare il trasferimento da Mailchimp, devi soddisfare i seguenti requisiti:

Requisiti minimi

- Hai un account Mailchimp con e-mail e password. Per ulteriori informazioni sulla creazione di account, consulta Creazione di un account Mailchimp.
- È necessario avere un AWS account creato con il servizio di accesso a. AWS Glue
- Assicurati di aver creato una delle seguenti risorse. Queste risorse forniscono le credenziali da AWS Glue utilizzare per accedere in modo sicuro ai dati quando si effettuano chiamate autenticate al proprio account:
  - Un'app per sviluppatori che supporta OAuth l'autenticazione 2.0. Per ulteriori informazioni sulla creazione di un'app per gli sviluppatori, consulta Creazione di un account Mailchimp.

Se soddisfi questi requisiti, sei pronto per connetterti AWS Glue al tuo account Mailchimp. Per le connessioni tipiche, non è necessario fare nient'altro in Mailchimp.

## Configurazione delle connessioni di Mailchimp

Mailchimp supporta i due tipi di meccanismo di autenticazione seguenti:

- Mailchimp supporta il tipo di concessione `AUTHORIZATION_CODE`.
  - Questo tipo di concessione è considerato `OAuth` "a tre gambe" in quanto si basa sul reindirizzamento degli utenti al server di autorizzazione di terzi per l'autenticazione. Viene utilizzato durante la creazione di connessioni tramite la console AWS Glue . Per impostazione predefinita, l'utente che crea una connessione può fare affidamento su un'app connessa di proprietà di AWS Glue in cui non è necessario fornire alcuna informazione relativa a `OAuth` a eccezione dell'ID client di Mailchimp e del segreto client. La AWS Glue Console reindirizzerà

l'utente a Mailchimp dove l'utente deve effettuare il login e consentire le autorizzazioni richieste per accedere AWS Glue alla propria istanza Mailchimp.

- Gli utenti possono comunque scegliere di creare la propria app connessa in Mailchimp e fornire il proprio Client ID e Client Secret durante la creazione di connessioni tramite la Console. AWS Glue In questo scenario, verranno comunque reindirizzati a Mailchimp per accedere e AWS Glue autorizzare l'accesso alle proprie risorse.
- Per la documentazione pubblica di Mailchimp sulla creazione di un'app connessa per `AUTHORIZATION_CODE` `OAuth` flow, consulta Accedere ai dati per conto di altri utenti con 2. OAuth

- Autenticazione personalizzata: per la documentazione pubblica di Mailchimp sulla generazione delle chiavi API richieste per l'autorizzazione personalizzata, consulta Informazioni sulle chiavi API.

Per configurare una connessione di Mailchimp:

1. In AWS Secrets Manager, crea un segreto con i seguenti dettagli:
   - Autenticazione `OAuth`: per le app connesse gestite dal cliente: il segreto deve contenere l'app connessa Consumer Secret con `USER_MANAGED_CLIENT_APPLICATION_CLIENT_SECRET` come chiave.
   - Autenticazione personalizzata: per le app connesse gestite dal cliente: il segreto deve contenere l'app connessa Consumer Secret con “api_key” come chiave.

   > **Note**
   >
   > È necessario creare un segreto per ogni connessione in AWS Glue.

2. In AWS Glue Studio, crea una connessione in Connessioni dati seguendo i passaggi seguenti:
   a. In Connessioni, seleziona Crea connessione.
   b. Quando selezioni un'origine dati, seleziona Mailchimp.
   c. Fornisci `instanceUrl` di Mailchimp.
   d. Seleziona il ruolo IAM per il quale AWS Glue può assumere e dispone delle autorizzazioni per le seguenti azioni:

      JSON

```
{
```

```
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:DescribeSecret",
        "secretsmanager:GetSecretValue",
        "secretsmanager:PutSecretValue",
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DeleteNetworkInterface"
      ],
      "Resource": "*"
    }
  ]
}
```

e. Seleziona il tipo di autenticazione per connetterti a Mailchimp:

- Per l'0`Auth`autenticazione: fornisci l'URL del token, l'applicazione client gestita dagli utenti ClientId di Mailchimp a cui desideri connetterti.
- Per l'autenticazione personalizzata: seleziona il tipo di autenticazione CUSTOM per connetterti a Mailchimp.

f. Seleziona quello `secretName` che desideri utilizzare per questa connessione per AWS Glue inserire i token.

g. Selezionare le opzioni di rete se si desidera utilizzare la propria rete.

3. Concedi il permesso di lettura `secretName` al ruolo IAM associato al tuo AWS Glue lavoro.
4. Nella configurazione del AWS Glue lavoro, fornisci `connectionName` una connessione di rete aggiuntiva.

## Lettura da entità di Mailchimp

Prerequisiti

Un oggetto di Mailchimp da cui si desidera leggere. Consulta la tabella delle entità supportate qui di seguito per verificare le entità disponibili.

Entità supportate

- Report di abuso

- Automazione
- Campagne
- Dettagli clic
- Elenchi
- Membri
- Dettagli aperti
- Segmenti
- Archiviazioni
- Iscrizione annullata

| Entità | Può essere Filtrato | Supporta Limite | Supporta Ordina per | Supporta Seleziona * | Supporta Partizionamento |
|---|---|---|---|---|---|
| Automazione | Sì | Sì | Sì | Sì | Sì |
| Campagne | No | No | No | No | No |
| Elenchi | Sì | Sì | No | Sì | Sì |
| Segnala un abuso | No | Sì | No | Sì | Sì |
| Report aperti | No | Sì | No | Sì | Sì |
| Report clic | Sì | Sì | No | Sì | Sì |
| Report annullamento iscrizione | No | Sì | No | Sì | Sì |
| Segment | No | Sì | No | Sì | Sì |
| Membri del segmento | Sì | Sì | No | Sì | No |
| Archiviazioni | Sì | Sì | Sì | Sì | No |

Esempio

```
mailchimp_read = glueContext.create_dynamic_frame.from_options(
            connection_type="mailchimp",
            connection_options={

                  "connectionName": "connectionName",

                  "ENTITY_NAME": "stores",
"INSTANCE_URL": "https://us14.api.mailchimp.com",
                  "API_VERSION": "3.0"
               })
```

Dettagli dell'entità e del campo di Mailchimp

- Report di utilizzo
- Automazione
- Campagne
- Dettagli clic
- Elenchi
- Membri
- Dettagli aperti
- Segmenti
- Archiviazioni
- Iscrizione annullata

Query di partizionamento

Puoi fornire le opzioni Spark aggiuntive `PARTITION_FIELD`, `LOWER_BOUND`, `UPPER_BOUND` e `NUM_PARTITIONS` se desideri utilizzare la concorrenza in Spark. Con questi parametri, la query originale viene suddivisa in `NUM_PARTITIONS` query secondarie che possono essere eseguite contemporaneamente dalle attività Spark.

- `PARTITION_FIELD`: il nome del campo da utilizzare per partizionare la query.
- `LOWER_BOUND`: un valore limite inferiore inclusivo del campo di partizione scelto.

  Per il DateTime campo, accettiamo il valore in formato ISO.

Esempio di valore valido:

```
"2024-07-01T00:00:00.000Z"
```

- UPPER_BOUND: un valore limite superiore esclusivo del campo di partizione scelto.
- NUM_PARTITIONS: il numero di partizioni.

Nella seguente tabella sono descritti i dettagli del campo di supporto del partizionamento delle entità:

| Nome dell'entità | Campi di partizionamento | Tipo di dati |
|---|---|---|
| | | |
| | | |
| | | |

Esempio:

```
read_read = glueContext.create_dynamic_frame.from_options(
    connection_type="mailchimp",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "automations",
        "API_VERSION": "3.0",
        "INSTANCE_URL": "https://us14.api.mailchimp.com",
        "PARTITION_FIELD": "create_time",
        "LOWER_BOUND": "2024-02-05T14:09:30.115Z",
        "UPPER_BOUND": "2024-06-07T13:30:00.134Z",
        "NUM_PARTITIONS": "3"
    }
```

## Opzioni di connessione di Mailchimp

Le seguenti sono le opzioni di connessione per Mailchimp:

- ENTITY_NAME (Stringa): (obbligatorio) utilizzato per la lettura/scrittura. Il nome del tuo oggetto in Mailchimp.

- `INSTANCE_URL`(Stringa): (obbligatorio) un URL di istanza di Mailchimp valido.
- `API_VERSION`(String): (obbligatorio) utilizzato per la lettura. Versione della versione dell'API Engage Rest di Mailchimp che desideri utilizzare. Ad esempio: 3.0.
- `SELECTED_FIELDS`(List<String>): predefinito: vuoto (SELECT *). Utilizzato per la lettura. Colonne da selezionare per l'oggetto.
- `FILTER_PREDICATE`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Dev'essere in formato Spark SQL.
- `QUERY`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Query Spark SQL completa.
- `PARTITION_FIELD`(Stringa): utilizzato per la lettura. Campo da utilizzare per partizionare la query.
- `LOWER_BOUND`(Stringa): utilizzato per la lettura. Un valore limite inferiore inclusivo del campo di partizione scelto.
- `UPPER_BOUND`(Stringa): utilizzato per la lettura. Un valore limite superiore esclusivo del campo di partizione scelto.
- `NUM_PARTITIONS`(Numero intero): predefinito: 1. Utilizzato per la lettura. Numero di partizioni da leggere.

## Creazione di un account Mailchimp

1. Vai alla pagina di accesso di Mailchimp, inserisci il tuo ID e-mail e la password, quindi scegli Registrati.
2. Apri l'e-mail di conferma di Mailchimp e scegli il link di conferma per verificare il tuo account.

> **Note**
>
> Il tempo necessario per ricevere l'e-mail di attivazione può variare. Se non hai ricevuto l'e-mail di attivazione, controlla la cartella spam e leggi i nostri suggerimenti per la risoluzione dei problemi relativi all'e-mail di attivazione. Mailchimp blocca le registrazioni da indirizzi e-mail basati su ruoli, come admin@pottedplanter.com o security@example.com.

La prima volta che accedi al tuo account, Mailchimp chiede le informazioni necessarie. Mailchimp utilizza queste informazioni per garantire che il tuo account sia conforme ai loro Termini di utilizzo e per fornire indicazioni rilevanti per le tue esigenze e quelle della tua azienda.

3. Inserisci le tue informazioni, segui o prompt per completare il processo di attivazione e inizia a utilizzare il tuo nuovo account di Mailchimp.

### Registrazione di un'applicazione **`OAuth2.0`**

1. Vai alla pagina di accesso di Mailchimp, inserisci il tuo ID e-mail e la password, quindi scegli Accedi.
2. Seleziona l'icona Utente nell'angolo in alto a destra, quindi scegli Account e fatturazione dal menu a discesa.
3. Seleziona Extra e scegli App registrate dal menu a discesa.
4. Individua e scegli Registra un'app.
5. Inserisci i seguenti dettagli:
   - Nome dell'app: nome dell'app.
   - Azienda/Organizzazione: nome dell'azienda o dell'organizzazione.
   - Sito Web dell'app: sito Web dell'app.
   - URI di reindirizzamento: un pattern URI di reindirizzamento è un percorso URI (o un elenco di percorsi separati da virgole) a cui Mailchimp può reindirizzare (se richiesto) quando il flusso di accesso è completo. Ad esempio, `https://ap-southeast-2\\.console\\.aws\\.amazon\\.com`
6. Scegli Create (Crea).
7. L'ID client e il segreto client saranno ora visibili. Copia i dati e salvali in un posto sicuro. Quindi, scegli Fine.

> **Note**
>
> Le stringhe Client ID e Client Secret sono credenziali utilizzate per stabilire una connessione con questo connettore quando si utilizza AppFlow o. AWS Glue

### Generazione di una chiave API

1. Vai alla pagina di accesso di Mailchimp, inserisci il tuo ID e-mail e la password, quindi scegli Accedi.

2. Seleziona l'icona Utente nell'angolo in alto a destra, quindi scegli Account e fatturazione dal menu a discesa.
3. Seleziona Extra e scegli Chiavi API dal menu a discesa.
4. Scegli Crea una chiave.
5. Inserisci un nome per la chiave e seleziona Genera chiave.

   La pagina successiva mostra la chiave API generata.
6. Copia la chiave, archiviala in modo sicuro e scegli Fine.

### Limitazioni

Di seguito sono riportate le limitazioni per il connettore di Mailchimp:

- La filtrazione è supportata solo dalle entità `Campaigns`, `Automations`, `Lists`, `Open Details`, `Members` e `Segments`.
- Durante l'utilizzo di un filtro sul campo del tipo di dati `DateTime`, è necessario passare valori in questo formato: `yyyy-mm-ddThh:MM:ssZ`

## Connessione a Microsoft Dynamics 365 CRM

Microsoft Dynamics 365 è una linea di prodotti composta da applicazioni aziendali intelligenti per la pianificazione delle risorse aziendali e la gestione delle relazioni con i clienti.

Argomenti


- AWS Glue supporto per Microsoft Dynamics 365
- Policy contenenti le operazioni API per creare e utilizzare le connessioni
- Configurazione di Microsoft Dynamics 365 CRM
- Configurazione delle connessioni Microsoft Dynamics 365 CRM
- Lettura da entità Microsoft Dynamics 365 CRM
- Riferimento all'opzione di connessione Microsoft Dynamics 365 CRM
- Limitazioni


### AWS Glue supporto per Microsoft Dynamics 365

AWS Glue supporta Microsoft Dynamics 365 come segue:

Supportato come origine?

Sì. Puoi utilizzare i processi AWS Glue ETL per interrogare i dati da Microsoft Dynamics 365.

Supportato come destinazione?

No.

Versioni dell'API Microsoft Dynamics 365 CRM supportate

v9.2.

## Policy contenenti le operazioni API per creare e utilizzare le connessioni

La seguente politica di esempio descrive le AWS autorizzazioni richieste per la creazione e l'utilizzo delle connessioni. Se si sta creando un nuovo ruolo, creare una policy che contenga quanto segue:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:ListConnectionTypes",
        "glue:DescribeConnectionType",
        "glue:RefreshOAuth2Tokens",
        "glue:ListEntities",
        "glue:DescribeEntity"
      ],
      "Resource": "*"
    }
  ]
}
```

È anche possibile utilizzare le seguenti policy IAM per autorizzare l'accesso:

- AWSGlueServiceRole— Garantisce l'accesso alle risorse che diversi AWS Glue processi richiedono per l'esecuzione per conto dell'utente. Queste risorse includono Amazon S3 AWS Glue,

IAM, CloudWatch Logs e Amazon EC2. Se segui la convenzione di denominazione delle risorse specificata in questa politica, AWS Glue i processi dispongono delle autorizzazioni necessarie. Questa policy è in genere collegata ai ruoli specificati quando si definiscono crawler, processi ed endpoint di sviluppo.

- AWSGlueConsoleFullAccess— Concede l'accesso completo alle AWS Glue risorse quando un'identità a cui è allegata la policy utilizza la AWS console di gestione. Se si segue la convenzione per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questa policy è in genere associata agli utenti della AWS Glue console.

## Configurazione di Microsoft Dynamics 365 CRM

Prima di poter utilizzare AWS Glue per trasferire dati da Microsoft Dynamics 365 CRM, devi soddisfare questi requisiti:

Requisiti minimi

- Hai un account sviluppatore Microsoft Dynamics 365 CRM con ClientId and Secret.
- Il tuo account Microsoft Dynamics 365 CRM dispone dell'accesso all'API con una licenza valida.

Se soddisfi questi requisiti, sei pronto per connetterti AWS Glue al tuo account Microsoft Dynamics 365 CRM. Per le connessioni tipiche, non è necessario fare nient'altro in Microsoft Dynamics 365 CRM.

## Configurazione delle connessioni Microsoft Dynamics 365 CRM

Tipo di concessione AUTHORIZATION_CODE

- Questo tipo di concessione è considerato «a tre gambe» in OAuth quanto si basa sul reindirizzamento degli utenti al server di autorizzazione di terze parti per autenticare l'utente. Viene utilizzato durante la creazione di connessioni tramite la console. AWS Glue La AWS Glue Console reindirizzerà l'utente a Microsoft Dynamics 365 CRM dove l'utente deve accedere e consentire AWS Glue le autorizzazioni richieste per accedere alla propria istanza di Microsoft Dynamics 365 CRM.
- Gli utenti possono scegliere di creare la propria app connessa in Microsoft Dynamics 365 CRM e fornire il proprio ID client e il segreto del client durante la creazione di connessioni tramite la AWS Glue Console. In questo scenario, verranno comunque reindirizzati a Microsoft Dynamics 365 CRM per accedere e autorizzare l'accesso AWS Glue alle proprie risorse.

- Questo tipo di concessione genera un token di aggiornamento e un token di accesso. Il token di accesso dura poco tempo e può essere aggiornato automaticamente senza l'interazione dell'utente utilizzando il token di aggiornamento.
- Per la documentazione pubblica di Microsoft Dynamics 365 CRM sulla creazione di un'app connessa per il OAuth flusso del codice di autorizzazione, vedi | Microsoft Learn. Registrazione dell'app Microsoft.

Microsoft Dynamics 365 CRM supporta l' OAuth2autenticazione 2.0.

Per configurare una connessione Microsoft Dynamics 365 CRM:

1. In Gestione dei segreti AWS, crea un segreto con i seguenti dettagli. È necessario creare un segreto per ogni connessione in AWS Glue.
   - Per il tipo di AuthorizationCode sovvenzione:

     Per le app connesse gestite dal cliente: il segreto deve contenere l'app connessa al segreto client con `USER_MANAGED_CLIENT_APPLICATION_CLIENT_SECRET` come chiave.
2. In AWS Glue Studio, crea una connessione in Connessioni dati seguendo i passaggi seguenti:
   a. Quando selezioni un'origine dati, seleziona Microsoft Dynamics 365 CRM.
   b. Seleziona l'INSTANCE_URL dell'istanza di Microsoft Dynamics 365 CRM a cui desideri connetterti.
   c. Seleziona il ruolo IAM che AWS Glue può assumere e dispone delle autorizzazioni per le seguenti azioni:

      JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:DescribeSecret",
        "secretsmanager:GetSecretValue",
        "secretsmanager:PutSecretValue",
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DeleteNetworkInterface"
      ],
```

```
      "Resource": "*"
    }
  ]
}
```

d. Seleziona URL del token e URL del codice di autorizzazione per accedere all'area di lavoro di Microsoft Dynamics 365 CRM.

e. Fornisci l'applicazione client gestita dagli utenti ClientId della tua app Microsoft Dynamics 365 CRM.

f. Seleziona quello `secretName` che desideri utilizzare per questa connessione per AWS Glue inserire i token.

g. Selezionare le opzioni di rete se si desidera utilizzare la propria rete.

3. Concedi il permesso di lettura `secretName` al ruolo IAM associato al tuo AWS Glue lavoro. Scegli Next (Successivo).
4. Nella configurazione del AWS Glue lavoro, fornisci `connectionName` una connessione di rete aggiuntiva.

## Lettura da entità Microsoft Dynamics 365 CRM

Prerequisiti

- Un oggetto Microsoft Dynamics 365 CRM da cui si desidera leggere. È necessario il nome dell'oggetto, come contatti o account. Nella tabella seguente sono illustrate le entità supportate.

Entità supportate

| Entità | Può essere Filtrato | Supporta Limite | Supporta Ordina per | Supporta Seleziona * | Supporta Partizionamento |
|---|---|---|---|---|---|
| Entità dinamica | Sì | Sì | Sì | Sì | Sì |

Esempio

```
dynamics365_read = glueContext.create_dynamic_frame.from_options(
```

```
    connection_type="microsoftdynamics365crm",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "dynamic_entity",
        "API_VERSION": "v9.2",
        "INSTANCE_URL": "https://{tenantID}.api.crm.dynamics.com"
    }
```

Dettagli dell'entità e del campo di Microsoft Dynamics 365 CRM

Entità con metadati dinamici:

Microsoft Dynamics 365 CRM fornisce endpoint per recuperare i metadati in modo dinamico. Pertanto, per le entità dinamiche, il supporto dell'operatore viene acquisito a livello di tipo di dati.

| Entità | Tipo di dati | Operatori supportati |
|---|---|---|
| Entità dinamica | DateTime | =, <, <=, >, >=, BETWEEN |
| | Data | =, <, <=, >, >= |
| | Stringa | =, != |
| | Double | =, <, <=, >, >= |
| | Numero intero | =, <, <=, >, >= |
| | Decimale | =, <, <=, >, >= |
| | Long | =, <, <=, >, >= |
| | BigInteger | =, <, <=, >, >= |
| | List | N/A |
| | Struct | N/A |
| | Eseguire la mappatura | N/A |

Query di partizionamento

Microsoft Dynamics 365 CRM supporta solo il partizionamento basato sul campo.

Se desideri utilizzare la concorrenza in Spark, possono essere fornite le opzioni aggiuntive `PARTITION_FIELD`, `LOWER_BOUND`, `UPPER_BOUND` e `NUM_PARTITIONS`. Con questi parametri, la query originale viene suddivisa in `NUM_PARTITIONS` query secondarie che possono essere eseguite contemporaneamente dalle attività Spark.

- `PARTITION_FIELD`: il nome del campo da utilizzare per partizionare la query.
- `LOWER_BOUND`: un valore limite inferiore inclusivo del campo di partizione scelto.

  Per Datetime, accettiamo il formato di timestamp Spark utilizzato nelle query SQL di Spark. Esempio di valori validi: `"2024-01-30T06:47:51.000Z"`.
- `UPPER_BOUND`: un valore limite superiore esclusivo del campo di partizione scelto.
- `NUM_PARTITIONS`: numero di partizioni.

I dettagli di supporto del campo di partizionamento per entità sono riportati nella tabella di seguito:

| Nome entità | Campi di partizionamento | DataType |
|---|---|---|
| Entità dinamica (entità standard) | DateTime Campi dinamici interrogabili | createdon, modifiedon |
| Entità dinamica (entità personalizzata) | createdon, modifiedon | createdon, modifiedon |

Esempio

```
dynamics365_read = glueContext.create_dynamic_frame.from_options(
    connection_type="microsoftdynamics365crm",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "dynamic_entity",
        "API_VERSION": "v9.2",
        "instanceUrl": "https://{tenantID}.api.crm.dynamics.com"
        "PARTITION_FIELD": "createdon"
        "LOWER_BOUND": "2024-01-30T06:47:51.000Z"
        "UPPER_BOUND": "2024-06-30T06:47:51.000Z"
        "NUM_PARTITIONS": "10"
    }
```

## Riferimento all'opzione di connessione Microsoft Dynamics 365 CRM

Di seguito sono riportate le opzioni di connessione per Microsoft Dynamics 365 CRM:

- `ENTITY_NAME`(String): (obbligatorio) utilizzato per la lettura. Il nome dell'oggetto in Microsoft Dynamics 365 CRM.
- `API_VERSION`(String): (obbligatorio) utilizzato per la lettura. La versione dell'API Rest di Microsoft Dynamics 365 CRM che desideri utilizzare.
- `SELECTED_FIELDS`(List<String>): predefinito: vuoto (SELECT *). Utilizzato per la lettura. Colonne da selezionare per l'oggetto.
- `FILTER_PREDICATE`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Dev'essere in formato Spark SQL.
- `QUERY`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Query Spark SQL completa.
- `INSTANCE_URL`(Stringa): (obbligatorio) Un URL di istanza di Microsoft Dynamics 365 CRM valido con il formato: `https://{tenantID}.api.crm.dynamics.com`
- `NUM_PARTITIONS`(Numero intero): predefinito: 1. Utilizzato per la lettura. Numero di partizioni da leggere.
- `PARTITION_FIELD`(Stringa): utilizzato per la lettura. Campo da utilizzare per partizionare la query.
- `LOWER_BOUND`(Stringa): utilizzato per la lettura. Un valore limite inferiore inclusivo del campo di partizione scelto. Esempio: `2024-01-30T06:47:51.000Z`.
- `UPPER_BOUND`(Stringa): utilizzato per la lettura. Un valore limite superiore esclusivo del campo di partizione scelto. Esempio: `2024-06-30T06:47:51.000Z`.

## Limitazioni

Di seguito sono riportate le limitazioni per il connettore Microsoft Dynamics 365 CRM:

- In Microsoft Dynamics 365 CRM, il partizionamento basato su record non è supportato in quanto non supporta un parametro offset e pertanto il partizionamento basato su record non può essere supportato.
- L'impaginazione è impostata su un massimo di 500 record per pagina per evitare eccezioni del server interno da SaaS dovute a una combinazione di limiti di dimensione e velocità dei dati.
  - Documentazione SaaS sull'impaginazione
  - Documentazione SaaS sui limiti di velocità

- Microsoft Dynamics 365 CRM supporta `order by` solo sui campi padre per tutte le entità. `order by` non è supportato nei campi secondari.
  - Sono supportate entrambe le direzioni ASC e DESC.
  - `order by` è supportato su più campi.
- Il filtraggio sul campo "createddatetime" dell'entità `aadusers` standard genera un errore di richiesta errata da SaaS anche se supporta la filtrazione. Non esiste alcuna identificazione specifica di nessun'altra entità con un problema simile a causa della natura dinamica dei metadati e non è nota la causa principale. Pertanto, non può essere gestito.
- I tipi di oggetti complessi come Struct, List e Map, non supportano la filtrazione.
- Molti campi che possono essere recuperati da una risposta hanno `isRetrievable` contrassegnato come `false` nella risposta dinamica ai metadati. Per evitare la perdita di dati, `isRetrievable` è impostato su `true` per tutti i campi.
- Il partizionamento basato sui campi sarà supportato su tutte le entità quando rispetta i seguenti criteri:
  - DateTime i campi interrogabili devono essere presenti nelle entità standard o `createdon` `modifiedon` i campi (generati dal sistema) nelle entità personalizzate.
  - Non esiste un'identificazione esclusiva dei campi generati dal sistema o della proprietà nullable da qualsiasi metadato SaaS APIs, tuttavia è prassi generale che solo i campi disponibili per impostazione predefinita siano filtrabili e non annullabili. Pertanto, il suddetto criterio di selezione dei campi è considerato nullo e sicuro e, se filtrabile, sarà idoneo per il partizionamento.

## Connessione a Microsoft Teams

Microsoft Teams è uno spazio di lavoro collaborativo all'interno di Microsoft 365 che funge da hub centrale per le conversazioni sul posto di lavoro, il lavoro di squadra collaborativo, le chat video e la condivisione di documenti, il tutto progettato per sostenere la produttività dei lavoratori in una suite di strumenti unificata.

Argomenti

- AWS Glue supporto per Microsoft Teams
- Policy contenenti le operazioni API per creare e utilizzare le connessioni
- Configurazione di Microsoft Teams
- Configurazione delle connessioni Microsoft Teams
- Lettura da entità Microsoft Teams

- Indicazioni di riferimento alle opzioni di connessione di Microsoft Teams
- Limitazioni
- Creare un nuovo account Microsoft Teams:

## AWS Glue supporto per Microsoft Teams

AWS Glue supporta Microsoft Teams come segue:

Supportato come origine?

Sì. È possibile utilizzare i processi AWS Glue ETL per interrogare i dati di Microsoft Teams.

Supportato come destinazione?

No.

Versioni dell'API di Microsoft Teams supportate

v1. Per il supporto delle entità per le versioni specifiche, consulta Entità supportate per l'origine.

## Policy contenenti le operazioni API per creare e utilizzare le connessioni

La seguente politica di esempio descrive le AWS autorizzazioni richieste per la creazione e l'utilizzo delle connessioni. Se si sta creando un nuovo ruolo, creare una policy che contenga quanto segue:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:ListConnectionTypes",
        "glue:DescribeConnectionType",
        "glue:RefreshOAuth2Tokens",
        "glue:ListEntities",
        "glue:DescribeEntity"
      ],
      "Resource": "*"
```

```
    }
  ]
}
```

È anche possibile utilizzare le seguenti policy IAM per autorizzare l'accesso:

- AWSGlueServiceRole— Garantisce l'accesso alle risorse che diversi AWS Glue processi richiedono per l'esecuzione per conto dell'utente. Queste risorse includono Amazon S3 AWS Glue, IAM, CloudWatch Logs e Amazon EC2. Se segui la convenzione di denominazione delle risorse specificata in questa politica, AWS Glue i processi dispongono delle autorizzazioni necessarie. Questa policy è in genere collegata ai ruoli specificati quando si definiscono crawler, processi ed endpoint di sviluppo.
- AWSGlueConsoleFullAccess— Concede l'accesso completo alle AWS Glue risorse quando un'identità a cui è allegata la policy utilizza la AWS console di gestione. Se si segue la convenzione per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questa policy è in genere associata agli utenti della AWS Glue console.

## Configurazione di Microsoft Teams

Prima di poter AWS Glue utilizzare il trasferimento di dati da Microsoft Teams, devi soddisfare questi requisiti:

Requisiti minimi

- Hai un account sviluppatore Microsoft Teams con e-mail e password. Per ulteriori informazioni, consulta Creare un nuovo account Microsoft Teams:.
- Dovresti aver configurato un' OAuth2 app nel tuo account Microsoft che fornisca l'ID client e le credenziali segrete da AWS Glue utilizzare per accedere ai tuoi dati in modo sicuro quando effettua chiamate autenticate al tuo account. Per ulteriori informazioni, consulta Creare un nuovo account Microsoft Teams:.

Se soddisfi questi requisiti, sei pronto per connetterti AWS Glue al tuo account Microsoft Teams. Per le connessioni tipiche, non è necessario fare nient'altro in Microsoft Teams.

## Configurazione delle connessioni Microsoft Teams

Microsoft Teams supporta i due tipi di meccanismo di autenticazione seguenti:

1. OAuth Autenticazione: Microsoft Teams supporta il tipo di concessione AUTHORIZATION_CODE per. OAuth2
   - Questo tipo di concessione è considerato «a tre gambe» in OAuth quanto si basa sul reindirizzamento degli utenti al server di autorizzazione di terze parti per autenticare l'utente. Viene utilizzato durante la creazione di connessioni tramite la console. AWS Glue Per impostazione predefinita, l'utente che crea una connessione può fare affidamento su un'app connessa di AWS Glue proprietà in cui non è necessario fornire alcuna informazione OAuth correlata ad eccezione dell'url dell'istanza di Microsoft Teams. La AWS Glue Console reindirizzerà l'utente a Microsoft Teams dove l'utente deve accedere e consentire AWS Glue le autorizzazioni richieste per accedere alla propria istanza di Microsoft Teams.
   - Gli utenti possono scegliere di creare la propria app connessa in Microsoft Teams e fornire il proprio ID client e il segreto del client durante la creazione di connessioni tramite la AWS Glue Console. In questo scenario, verranno comunque reindirizzati a Microsoft Teams per accedere e autorizzare l'accesso AWS Glue alle proprie risorse.
   - Questo tipo di concessione genera un token di aggiornamento e uno di accesso. Il token di accesso è attivo per un'ora e può essere aggiornato automaticamente senza l'interazione dell'utente utilizzando il token di aggiornamento.
   - Per la documentazione pubblica di Microsoft Teams sulla creazione di un'app connessa per il OAuth flusso del codice di autorizzazione, vedi | Microsoft Learn. Registrare un'applicazione con la piattaforma di identità Microsoft: Microsoft Graph.

Per configurare una connessione Microsoft Teams:

1. Nel Gestione dei segreti AWS, crea un segreto con i seguenti dettagli. È necessario creare un segreto per ogni connessione in AWS Glue.
   a. Per l' OAuth autenticazione:
      - Per l'app connessa gestita dal cliente, il segreto deve contenere l'app connessa Consumer Secret con USER_MANAGED_CLIENT_APPLICATION_CLIENT_SECRET come chiave.
2. In AWS Glue Studio, crea una connessione in Connessioni dati seguendo i passaggi seguenti:
   a. In Connessioni dati, scegli Crea connessione.
   b. Quando selezioni un'origine dati, seleziona Microsoft Teams.
   c. Fornisci il tuo ID tenant di Microsoft Teams.
   d. Seleziona il ruolo IAM che AWS Glue può assumere e dispone delle autorizzazioni per le seguenti azioni:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:DescribeSecret",
        "secretsmanager:GetSecretValue",
        "secretsmanager:PutSecretValue",
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DeleteNetworkInterface"
      ],
      "Resource": "*"
    }
  ]
}
```

e. Fornisci l'applicazione client gestita dagli utenti ClientId dell'app Microsoft Teams.

f. Seleziona quello `secretName` che desideri utilizzare per questa connessione AWS Glue per inserire i token.

g. Selezionare le opzioni di rete se si desidera utilizzare la propria rete.

3. Concedi il permesso di lettura `secretName` al ruolo IAM associato al tuo AWS Glue lavoro. Scegli Next (Successivo).
4. Nella configurazione del AWS Glue lavoro, fornisci `connectionName` una connessione di rete aggiuntiva.

## Lettura da entità Microsoft Teams

Prerequisiti

- Un oggetto Microsoft Teams da cui si desidera leggere. Sarà necessario il nome dell'oggetto, ad esempio team o channel-message. Nella tabella seguente sono illustrate le entità supportate.

Entità supportate come origine

Tutte le entità sono supportate con l'API versione 1.0.

| Entità | Può essere Filtrato | Supporta Limite | Supporta Ordina per | Supporta Seleziona * | Supporta Partizionamento |
|---|---|---|---|---|---|
| Team | No | No | No | Sì | No |
| Membri del team | Sì | Sì | No | Sì | Sì |
| Gruppi | Sì | Sì | Sì | Sì | Sì |
| Membri del gruppo | Sì | Sì | No | Sì | No |
| Canali | Sì | No | No | Sì | Sì |
| Messaggi del canale | No | Sì | No | Sì | No |
| Risposte ai messaggi del canale | No | Sì | No | Sì | No |
| Tabulazioni del canale | Sì | No | No | Sì | No |
| Chat | Sì | Sì | Sì | Sì | Sì |
| Eventi del calendario | Sì | Sì | Sì | Sì | Sì |

## Esempio

```
MicrosoftTeams_read = glueContext.create_dynamic_frame.from_options(
    connection_type="MicrosoftTeams",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "company",
        "API_VERSION": "v1.0"
```

```
    }
```

Dettagli dell'entità e dei campi di Microsoft Teams

Elenco delle entità:

- Squadra: https://docs.microsoft.com/en-us/graph/api/user-list-joinedteams? view=graph-rest-1.0
- Membro del team: -list-members? https://docs.microsoft.com/en-us/ graph/api/team view=graph-rest-1.0
- Gruppo: -list? https://docs.microsoft.com/en-us/ graph/api/group view=graph-rest-1.0
- Membro del gruppo: -list-members? https://docs.microsoft.com/en-us/ graph/api/group view=graph-rest-1.0
- Canale: -list? https://docs.microsoft.com/en-us/ graph/api/channel view=graph-rest-1.0
- Messaggio del canale: -list-messaggi? https://docs.microsoft.com/en-us/ graph/api/channel view=graph-rest-1.0
- Channel-Message-Reply: -list-replies? https://docs.microsoft.com/en-us/ graph/api/chatmessage view=graph-rest-1.0
- Channel-Tab: -list-tabs? https://docs.microsoft.com/en-us/ graph/api/channel view=graph-rest-1.0
- Chat: -list? https://docs.microsoft.com/en-us/ graph/api/chat view=graph-rest-1.0
- Evento del calendario: -list-eventi? https://docs.microsoft.com/en-us/ graph/api/group view=graph-rest-1.0

Query di partizionamento

Se si desidera utilizzare la simultaneità in Spark, possono essere fornite opzioni Spark `PARTITION_FIELD`, `LOWER_BOUND`, `UPPER_BOUND`, `NUM_PARTITIONS` aggiuntive. Con questi parametri, la query originale viene suddivisa in `NUM_PARTITIONS` query secondarie che possono essere eseguite contemporaneamente dalle attività Spark.

- `PARTITION_FIELD`: il nome del campo da utilizzare per partizionare la query.
- `LOWER_BOUND`: un valore limite inferiore inclusivo del campo di partizione scelto.

  Per la data, accettiamo il formato di data di Spark utilizzato nelle query di Spark SQL. Esempio di valori validi: `"2024-02-06"`.
- `UPPER_BOUND`: un valore limite superiore esclusivo del campo di partizione scelto.
- `NUM_PARTITIONS`: numero di partizioni.

I dettagli di supporto del campo di partizionamento per entità sono riportati nella tabella di seguito:

| Nome entità | Campi di partizionamento | Tipo di dati |
|---|---|---|
| Membri del team | visibleHistoryStartDateTime | DateTime |
| Gruppi | createdDateTime | DateTime |
| Canali | createdDateTime | DateTime |
| Chat | createdDateTime, Ora lastModifiedDate | DateTime |
| Eventi del calendario | createdDateTime, lastModif iedDate Ora, inizio originale | DateTime |

Esempio

```
microsoftteams_read = glueContext.create_dynamic_frame.from_options(
    connection_type="MicrosoftTeams",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "group",
        "API_VERSION": "v1.0",
        "PARTITION_FIELD": "createdDateTime"
        "LOWER_BOUND": "2022-07-13T07:55:27.065Z"
        "UPPER_BOUND": "2022-08-12T07:55:27.065Z"
        "NUM_PARTITIONS": "2"
    }
```

## Indicazioni di riferimento alle opzioni di connessione di Microsoft Teams

Di seguito sono riportate le opzioni di connessione per Microsoft Teams:

- `ENTITY_NAME`(String): (obbligatorio) utilizzato per la lettura. Il nome del tuo oggetto in Microsoft Teams.
- `API_VERSION`(String): (obbligatorio) utilizzato per la lettura. La versione dell'API Rest di Microsoft Teams che desideri utilizzare. Esempio: v1.0.

- `SELECTED_FIELDS`(List<String>): predefinito: vuoto (SELECT *). Utilizzato per la lettura. Colonne da selezionare per l'oggetto.
- `FILTER_PREDICATE`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Dev'essere in formato Spark SQL.
- `QUERY`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Query Spark SQL completa.
- `PARTITION_FIELD`(Stringa): utilizzato per la lettura. Campo da utilizzare per partizionare la query.
- `LOWER_BOUND`(Stringa): utilizzato per la lettura. Un valore limite inferiore inclusivo del campo di partizione scelto.
- `UPPER_BOUND`(Stringa): utilizzato per la lettura. Un valore limite superiore esclusivo del campo di partizione scelto.
- `NUM_PARTITIONS`(Numero intero): predefinito: 1. Utilizzato per la lettura. Numero di partizioni da leggere.

## Limitazioni

Di seguito sono riportate le limitazioni per il connettore Microsoft Teams:

- L'API di Microsoft Teams restituisce un numero di record inferiore a quello specificato per le entità Chat e Membro del team. Il problema è stato segnalato al supporto di Microsoft Teams ed è oggetto di indagine.

## Creare un nuovo account Microsoft Teams:

1. Vai alla home page di Microsoft Teams https://account.microsoft.com/account/e scegli Accedi.
2. Scegli Fare clic qui per crearne uno.
3. Inserisci le informazioni richieste per creare l'account e crea un nuovo account.
4. Vai al sito Web di Microsoft Teams all'indirizzo https://www.microsoft.com/en-in/microsoft-teams/log-in.
5. Accedi utilizzando l'account Microsoft appena creato.
6. Dopo aver effettuato con successo la registrazione su Teams, vai su /services. https://account.microsft.com
7. Scegli Prova Microsoft 365.
8. Attiva uno degli abbonamenti Microsoft 365 o Microsoft Teams riportati di seguito per accedere a tutte le funzionalità richieste dal connettore Microsoft Teams:

- Microsoft Teams Essentials
- Microsoft 365 Business
- Microsoft 365 Business Basic
- Microsoft 365 Business Standard
- Microsoft 365 Business Premium

Creare un'applicazione client gestita:

1. Per creare un'applicazione gestita, devi registrare una nuova OAuth app su Microsoft Entra (precedentemente Azure Active Directory):
2. Accedi al centro di amministrazione di Microsoft Entra.
3. Se hai accesso a più tenant, usa l'icona Impostazioni nel menu in alto per passare al tenant in cui desideri registrare l'applicazione dal menu delle sottoscrizioni Directory +.
4. Vai su Identità > Applicazioni > Registrazioni app e seleziona Nuova registrazione.
5. Inserisci un nome da visualizzare per la tua applicazione.
6. Specifica chi può utilizzare l'applicazione nella sezione Tipi di account supportati. Per rendere l'app globale, seleziona "Account in qualsiasi directory organizzativa" o "Account in qualsiasi directory organizzativa e account Microsoft personali".
7. Inserisci l'URI di reindirizzamento `https://{region}.console.aws.amazon.com/appflow/oauth`. Ad esempio, per `us-west-2 region`, aggiungi `https://us-west-2.console.aws.amazon.com/appflow/oauth`. Puoi aggiungerne più di una URLs per le diverse aree che desideri utilizzare.
8. Registra l'app.
9. Prendi nota dell'ID client per usarlo in futuro.
10. Scegli Aggiungi certificato o segreto nella sezione Essentials.
11. Scegli Nuovo segreto client.
12. Inserisci la descrizione e la durata della scadenza.
13. Copia e salva il segreto client per usarlo in futuro.
14. Nel menu a sinistra, seleziona Autorizzazioni API.
15. Scegli Aggiungi un'autorizzazione.
16. Seleziona "Microsoft Graph".
17. Seleziona "Autorizzazioni delegate".

18. Controlla tutte le seguenti autorizzazioni:

    - User.Read
    - Offline_access
    - User.Read.All
    - Utente. ReadWrite.Tutti
    - TeamsTab.ReadWriteForTeam
    - TeamsTab.ReadWriteForChat
    - TeamsTab. ReadWrite.Tutto
    - TeamsTab.Leggi tutto
    - TeamSettings. ReadWrite.Tutto
    - TeamSettings.Leggi tutto
    - TeamMember. ReadWrite.Tutto
    - TeamMember.Leggi tutto
    - Squadra. ReadBasic.Tutti
    - GroupMember. ReadWrite.Tutto
    - GroupMember.Leggi tutto
    - Gruppo. ReadWrite.Tutti
    - Group.Read.All
    - Elenco. ReadWrite.Tutti
    - Directory.Read.All
    - Elenco. AccessAsUser.Tutti
    - Chiacchierare ReadWrite
    - Chiacchierare ReadBasic
    - Chat.Read
    - ChannelSettings. ReadWrite.Tutto
    - ChannelSettings.Leggi tutto
    - ChannelMessage.Leggete tutto
    - Canale. ReadBasic.Tutti

19. Scegli Add Permissions (Aggiungi autorizzazioni). Adesso la tua app è configurata correttamente. Puoi usare l'ID client e il segreto client per creare una nuova connessione. Per ulteriori informazioni, vedere https://learn.microsoft.com/en-us/graph/ 2 auth-register-app-v.

# Connessione a Mixpanel

Mixpanel è una potente piattaforma di analisi in tempo reale che aiuta le aziende a misurare e a ottimizzare il coinvolgimento degli utenti. Mixpanel è un'applicazione usata per tracciare il comportamento dei clienti. Consente di monitorare il modo in cui gli utenti interagiscono con il prodotto e di analizzare i dati ottenuti con report interattivi che permettono di interrogare e visualizzare i risultati in pochi clic. Come utente Mixpanel, puoi connetterti AWS Glue al tuo account Mixpanel. Quindi, puoi usare Mixpanel come origine dati nei tuoi processi ETL. Esegui questi lavori per trasferire dati tra Mixpanel e AWS servizi o altre applicazioni supportate.

Argomenti


- AWS Glue supporto per Mixpanel
- Policy contenenti le operazioni API per creare e utilizzare le connessioni
- Configurazione di Mixpanel
- Configurazione delle connessioni Mixpanel
- Lettura da entità di Mixpanel
- Opzioni di connessione di Mixpanel
- Creazione di un account di Mixpanel e configurazione dell'app client
- Limitazioni


## AWS Glue supporto per Mixpanel

AWS Glue supporta Mixpanel come segue:

Supportato come origine?

Sì. È possibile utilizzare i job AWS Glue ETL per interrogare i dati da Mixpanel.

Supportato come destinazione?

No.

Versioni API di Mixpanel supportate

2.0

## Policy contenenti le operazioni API per creare e utilizzare le connessioni

La seguente politica di esempio descrive le AWS autorizzazioni richieste per la creazione e l'utilizzo delle connessioni. Se si sta creando un nuovo ruolo, creare una policy che contenga quanto segue:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:ListConnectionTypes",
        "glue:DescribeConnectionType",
        "glue:RefreshOAuth2Tokens",
        "glue:ListEntities",
        "glue:DescribeEntity"
      ],
      "Resource": "*"
    }
  ]
}
```

In alternativa, se non desideri utilizzare il metodo precedente, utilizza le seguenti policy IAM:

- AWSGlueServiceRole— Garantisce l'accesso alle risorse che diversi AWS Glue processi richiedono per l'esecuzione per conto dell'utente. Queste risorse includono Amazon S3 AWS Glue, IAM, CloudWatch Logs e Amazon EC2. Se segui la convenzione di denominazione delle risorse specificata in questa politica, AWS Glue i processi dispongono delle autorizzazioni necessarie. Questa policy è in genere collegata ai ruoli specificati quando si definiscono crawler, processi ed endpoint di sviluppo.
- AWSGlueConsoleFullAccess— Concede l'accesso completo alle AWS Glue risorse quando un'identità a cui è allegata la policy utilizza la AWS console di gestione. Se si segue la convenzione per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questa policy è in genere associata agli utenti della AWS Glue console.

## Configurazione di Mixpanel

Prima di poter utilizzare AWS Glue il trasferimento da Mixpanel, è necessario soddisfare i seguenti requisiti:

Requisiti minimi

- Avere un account Mixpanel. Per ulteriori informazioni sulla creazione di account, consultare Creazione di un account Mixpanel.
- L'account di Mixpanel è abilitato all'accesso tramite API. L'accesso alle API è abilitato per impostazione predefinita per le edizioni Enterprise, Unlimited, Developer e Performance.

Se soddisfi questi requisiti, sei pronto per connetterti AWS Glue al tuo account Mixpanel. Per le connessioni tipiche, non è necessario fare nient'altro in Mixpanel.

## Configurazione delle connessioni Mixpanel

Mixpanel supporta nome utente e password per `BasicAuth`. L'autenticazione di base è un metodo di autenticazione semplice in cui i client forniscono direttamente le credenziali per accedere alle risorse protette. AWS Glue è in grado di utilizzare il nome utente e la password per autenticare Mixpanel. APIs

Per la documentazione pubblica di Mixpanel sul flusso `BasicAuth`, consultare gli account del servizio Mixpanel.

Per configurare una connessione Mixpanel:

1. In AWS Secrets Manager, crea un segreto con i seguenti dettagli:
   - Per l'autenticazione di base, il segreto deve contenere l'app connessa Consumer Secret con `USERNAME` e `PASSWORD` come chiave.

   > **Note**
   >
   > È necessario creare un segreto per ogni connessione in AWS Glue.

2. In AWS Glue Studio, crea una connessione in Connessioni dati seguendo i passaggi seguenti:
   a. Quando si seleziona un Tipo di connessione, selezionare Mixpanel.
   b. Fornire il `INSTANCE_URL` di Mixpanel a cui si desidera connettersi.

c. Seleziona il ruolo IAM per il quale AWS Glue può assumere e dispone delle autorizzazioni per le seguenti azioni:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:DescribeSecret",
        "secretsmanager:GetSecretValue",
        "secretsmanager:PutSecretValue",
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DeleteNetworkInterface"
      ],
      "Resource": "*"
    }
  ]
}
```

d. Seleziona quello `secretName` che desideri utilizzare per questa connessione per AWS Glue inserire i token.

e. Selezionare Opzioni di rete se si desidera utilizzare la propria rete.

3. Concedi il permesso di lettura `secretName` al ruolo IAM associato al tuo AWS Glue lavoro.

## Lettura da entità di Mixpanel

Prerequisiti

È necessario disporre di un oggetto di Mixpanel, ad esempio `Funnels`, `Retention`, o`Retention Funnels`, da cui si desidera leggere i dati. Inoltre, è necessario conoscere il nome dell'oggetto.

Entità supportate

| Entità | Può essere Filtrato | Supporta Limite | Supporta Ordina per | Supporta Seleziona * | Supporta Partizionamento |
|---|---|---|---|---|---|
| Imbuti | Sì | No | No | Sì | No |
| Retention | Sì | No | No | Sì | No |
| Segmentazione | Sì | No | No | Sì | No |
| Somma di segmentazione | Sì | No | No | Sì | No |
| Media di segmentazione | Sì | No | No | Sì | No |
| Coorti | Sì | No | No | Sì | No |
| Integra | No | Sì | No | Sì | No |
| Eventi | Sì | No | No | Sì | No |
| Eventi principali | Sì | No | No | Sì | No |
| Nomi eventi | Sì | No | No | Sì | No |
| Proprietà degli eventi | Sì | No | No | Sì | No |
| Proprietà degli eventi principali | Sì | No | No | Sì | No |

| Entità | Può essere Filtrato | Supporta Limite | Supporta Ordina per | Supporta Seleziona * | Supporta Partizionamento |
|---|---|---|---|---|---|
| Valori delle proprietà degli eventi | Sì | No | No | Sì | No |
| Annotazioni | Sì | No | No | Sì | No |
| Attività dell'evento del profilo | Sì | No | No | Sì | No |

Esempio

```
mixpanel_read = glueContext.create_dynamic_frame.from_options(
    connection_type="mixpanel",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "/cohorts/list?project_id=2603353",
        "API_VERSION": "2.0",
        "INSTANCE_URL": "https://www.mixpanel.com/api/app/me"
    }
```

Dettagli dell'entità e del campo di Mixpanel

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| Imbuto | funnel_id | Numero intero | '=' |
| | workspace_id | Numero intero | '=' |
| | from_date | Data | '=' |
| | to_date | Data | '=’ |
| | length | Numero intero | '=' |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | length_unit | Stringa | '=' |
| | intervallo | Numero intero | '=' |
| | unità | Stringa | '=' |
| | limit | Numero intero | '=' |
| | data | Struct | |
| | meta | Struct | |
| Retention | workspace_id | Numero intero | '=' |
| | unità | Stringa | '=' |
| | addiction_unit | Stringa | '=' |
| | from_date | Data | '=' |
| | to_date | Data | '=' |
| | evento | Stringa | '=' |
| | limit | Numero intero | '=' |
| | data | Struct | |
| Segmentazione | workspace_id | Numero intero | '=' |
| | evento | Stringa | '=' |
| | from_date | Data | '=' |
| | to_date | Data | '=' |
| | unità | Stringa | '=' |
| | intervallo | Numero intero | '=' |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | limit | Numero intero | '=' |
| | tipo | Stringa | '=' |
| | serie | List | |
| | values | Struct | |
| | data | Struct | |
| Segmentazione numerica | workspace_id | Numero intero | '=' |
| | evento | Stringa | '=' |
| | on | Stringa | '=' |
| | from_date | Data | '=' |
| | to_date | Data | '=' |
| | unità | Stringa | '=' |
| | tipo | Stringa | '=' |
| | serie | List | |
| | values | Struct | |
| Somma di segmentazione | workspace_id | Numero intero | '=' |
| | evento | Stringa | '=' |
| | on | Stringa | '=' |
| | from_date | Data | '=' |
| | to_date | Data | '=' |
| | unità | Stringa | '=' |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | metadati | Struct | |
| | results | Struct | |
| Media di segmentazione | workspace_id | Numero intero | '=' |
| | evento | Stringa | '=' |
| | on | Stringa | '=' |
| | from_date | Data | '=' |
| | to_date | Data | '=' |
| | unità | Stringa | '=' |
| | metadati | Struct | |
| | results | Struct | |
| Coorti | count | Numero intero | |
| | is_visible | Numero intero | |
| | description | Stringa | |
| | created | DateTime | |
| | project_id | Numero intero | |
| | id | BigInteger | |
| | nome | Stringa | |
| | data_group_id | Stringa | |
| Integra | distinct_id | Stringa | |
| | | properties | Struct |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| Event | workspace | Numero intero | '=' |
| | evento | Stringa | '=' |
| | tipo | Stringa | '=' |
| | unità | Stringa | '=' |
| | intervallo | Numero intero | '=' |
| | from_date | Data | '=' |
| | to_date | Data | '=' |
| | serie | List | |
| | values | Struct | |
| Eventi principali | tipo | Stringa | '=' |
| | workspace_id | Numero intero | '=' |
| | limit | Numero intero | '=' |
| | quantità | Numero intero | |
| | evento | Stringa | |
| | percent_change | Float | |
| Nome evento | data | List | |
| | workspace_id | Numero intero | '=' |
| | tipo | Stringa | '=' |
| | limit | Numero intero | '=' |
| Proprietà dell'evento | workspace_id | Numero intero | '=' |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | evento | Stringa | '=' |
| | nome | Stringa | '=' |
| | tipo | Stringa | '=' |
| | unità | Stringa | '=' |
| | intervallo | Numero intero | '=' |
| | from_date | Data | '=' |
| | to_date | Data | '=' |
| | limit | Numero intero | '=' |
| | data | Struct | |
| | serie | List | |
| | values | Struct | |
| Proprietà dell'evento principale | workspace_id | Numero intero | '=' |
| | evento | Stringa | '=' |
| | limit | Numero intero | '=' |
| | data | Struct | |
| Valori delle proprietà dell'evento | workspace_id | Numero intero | '=' |
| | evento | Stringa | '=' |
| | limit | Numero intero | '=' |
| | nome | Stringa | '=' |
| | data | List | |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| Annotazione | workspace_id | Numero intero | |
| | data | DateTime | |
| | project_id | Numero intero | |
| | id | BigInteger | |
| | description | Stringa | |
| | from_date | Data | BETWEEN |
| Attività dell'evento del profilo | workspace_id | Numero intero | '=' |
| | distinct_ids | Stringa | '=' |
| | from_date | Data | '=' |
| | to_date | Data | '=' |
| | evento | Stringa | |
| | properties | Struct | |

## Opzioni di connessione di Mixpanel

Le seguenti sono le opzioni di connessione per Mixpanel:

- `ENTITY_NAME` (Stringa): (obbligatorio) utilizzato per la lettura/scrittura. Il nome del tuo oggetto in Mixpanel.
- `API_VERSION` (Stringa): (obbligatorio) utilizzato per la lettura/scrittura. API Rest di Mixpanel da utilizzare. Ad esempio, v2.0.
- `SELECTED_FIELDS`(Elenco<Stringa>): predefinito: vuoto (SELECT *). Utilizzato per la lettura. Colonne da selezionare per l'oggetto.
- `FILTER_PREDICATE`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Dev'essere in formato Spark SQL.
- `QUERY`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Query Spark SQL completa.

## Creazione di un account di Mixpanel e configurazione dell'app client

### Creazione di un account di Mixpanel

1. Vai alla home page di Mixpanel.
2. Nella home page di Mixpanel, scegli Iscriviti nell'angolo in alto a destra della pagina.
3. Nella pagina Inizia, completa le seguenti azioni:
   - Immetti l'indirizzo e-mail nell'apposito campo.
   - Seleziona la casella di controllo richiesta per accettare i termini.
   - Scegli Inizia per procedere.

   Una volta completato con successo, riceverai un'e-mail di verifica.
4. Controlla la tua casella di posta elettronica per un messaggio di verifica, apri l'e-mail e segui le istruzioni per verificare il tuo indirizzo e-mail.
5. Nella pagina di verifica, scegli Verifica e-mail per completare la verifica.
6. Nella pagina Dai un nome alla tua organizzazione, inserisci il nome della tua organizzazione e scegli Avanti.
7. Nella pagina Il tuo primo progetto, inserisci i dettagli del progetto e scegli Crea.
8. Nella pagina successiva, scegli Inizia per completare la creazione del tuo account.

### Accesso a un account Mixpanel

1. Vai alla pagina di accesso di Mixpanel.
2. Immetti l'indirizzo e-mail e scegli Continua.
3. Controlla la tua casella di posta elettronica per un messaggio di verifica, apri l'e-mail e segui le istruzioni per verificare il tuo indirizzo e-mail.
4. Nella pagina successiva, scegli il pulsante Accedi per accedere al tuo account.

### Acquisto di un piano Mixpanel

1. Nella pagina di Mixpanel, seleziona l'icona Impostazioni nell'angolo in alto a destra della pagina.
2. Dall'elenco di opzioni, seleziona Dettagli del piano e fatturazione.
3. Nella pagina Dettagli del piano e fatturazione, seleziona Aggiorna o Modifica.

4. Nella pagina successiva, seleziona il piano da acquistare.

   Questo completa la creazione dell'account e il processo di acquisto del piano.

Creazione di un nome utente e di un segreto client (per registrare l'app)

1. Nella pagina di Mixpanel, seleziona l'icona Impostazioni nell'angolo in alto a destra della pagina.
2. Dall'elenco di opzioni, seleziona Impostazioni del progetto.
3. Nella pagina Impostazioni del progetto, seleziona Account di servizio, quindi seleziona Aggiungi account di servizio.
4. Dall'elenco a discesa Account di servizio, seleziona l'account di servizio o inserisci il nome per creare, aggiungi il ruolo del progetto, specifica le scadenze e seleziona Aggiungi.

> **Important**
>
> Dopo aver completato il passaggio precedente, la pagina successiva mostra la chiave segreta dell'account di servizio. Assicurati di salvare la chiave segreta dell'account di servizio. Non potrai accedervi di nuovo dopo questo punto.

## Limitazioni

Di seguito sono riportate le limitazioni per il connettore di Mixpanel:

- Per l'entità `Segmentation Numeric`, l'API di Mixpanel genera un errore `400 (Bad Request)` se non vengono trovati dati numerici per i filtri obbligatori. Stiamo trattando questo come una risposta 0K per prevenire l'interruzione del flusso.
- Il campo `limit`, su cui è possibile eseguire la query, è stato rimosso dalle entità supportate perché:
  - Causava errori perché era interpretato come funzionalità limite dell'SDK
  - Il filtro non aveva alcuno scopo pratico
  - Le funzionalità equivalenti sono ora coperte dall'implementazione della funzionalità limite
- Il partizionamento basato sul campo non può essere supportato a causa dell'assenza degli operatori richiesti (>=, <=, <, >, `between`) per il partizionamento dalla piattaforma SaaS. Sebbene supporti l'operatore `between`, i campi per i quali lo supporta non sono recuperabili. Pertanto, i criteri per il partizionamento basato sul campo non sono soddisfatti.

- Poiché non è previsto l'utilizzo di un valore di "offset" per le entità che supportano la paginazione, non è possibile supportare il partizionamento basato su record per Mixpanel.
- L'entità `Cohorts` supporta solo il campo `CreatedDate/Time` e non vi è alcun campo per identificare `UpdatedDate/Time`; di conseguenza, `DML_Status` non può essere identificato. Inoltre, non esiste un endpoint per identificare i record eliminati. Pertanto, CDC non può essere supportato.
- Per eseguire un AWS Glue lavoro per le entità menzionate di seguito, sono necessari filtri obbligatori. Consulta la tabella seguente per i nomi delle entità e i relativi filtri richiesti.

  Nome dell'entità e filtri richiesti

| Nome dell'entità | Filtri obbligatori |
|---|---|
| Annotazioni | from_date, to_date |
| Coorti | Nessuno |
| Integra | Nessuno |
| Event | evento, tipo, unità, from_date, to_date, il |
| Nome eventi | tipo |
| Proprietà degli eventi | evento, tipo, unità, from_date, to_date |
| Proprietà degli eventi principali | evento |
| Valori delle proprietà degli eventi | evento, nome |
| Eventi principali | tipo |
| Imbuti | funnel_id, from_date, to_date |
| Attività dell'evento del profilo | distinct_ids, from_date, to_date |
| Retention | from_date, to_date, unità, addiction_unit |
| Segmentazione | evento, from_date, to_date |
| Media di segmentazione | event, from_date, to_date, on |

| Nome dell'entità | Filtri obbligatori |
| --- | --- |
| Segmentazione numerica | event, from_date, to_date, on |
| Somma di segmentazione | event, from_date, to_date, on |

# Connessione al Monday

Monday.com è un sistema operativo versatile che semplifica la gestione dei progetti e la collaborazione tra i team. Offre flussi di lavoro personalizzabili, pannelli di controllo visivi e strumenti di automazione per migliorare la produttività. Gli utenti possono monitorare le attività, gestire le risorse e comunicare in modo efficace in un'unica piattaforma integrata.

Argomenti


- AWS Glue supporto per lunedì
- Policy contenenti le operazioni API per creare e utilizzare le connessioni
- Configurazione Monday
- Configurazione delle connessioni di Monday
- Lettura dalle entità di Monday
- Indicazioni di riferimento alle opzioni di connessione a Monday
- Limitazioni
- Per creare un nuovo account Monday:


## AWS Glue supporto per lunedì

AWS Glue supporta il lunedì come segue:

Supportato come origine?

Sì. È possibile utilizzare i job AWS Glue ETL per interrogare i dati di lunedì.

Supportato come destinazione?

No.

Versioni API di Monday supportate

v2.

## Policy contenenti le operazioni API per creare e utilizzare le connessioni

La seguente politica di esempio descrive le AWS autorizzazioni richieste per la creazione e l'utilizzo delle connessioni. Se si sta creando un nuovo ruolo, creare una policy che contenga quanto segue:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:ListConnectionTypes",
        "glue:DescribeConnectionType",
        "glue:RefreshOAuth2Tokens",
        "glue:ListEntities",
        "glue:DescribeEntity"
      ],
      "Resource": "*"
    }
  ]
}
```

È anche possibile utilizzare le seguenti policy IAM per autorizzare l'accesso:

- AWSGlueServiceRole— Garantisce l'accesso alle risorse che diversi AWS Glue processi richiedono per l'esecuzione per conto dell'utente. Queste risorse includono Amazon S3 AWS Glue, IAM, CloudWatch Logs e Amazon EC2. Se segui la convenzione di denominazione delle risorse specificata in questa politica, AWS Glue i processi dispongono delle autorizzazioni necessarie. Questa policy è in genere collegata ai ruoli specificati quando si definiscono crawler, processi ed endpoint di sviluppo.
- AWSGlueConsoleFullAccess— Concede l'accesso completo alle AWS Glue risorse quando un'identità a cui è allegata la policy utilizza la AWS console di gestione. Se si segue la convenzione per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questa policy è in genere associata agli utenti della AWS Glue console.

## Configurazione Monday

Prima di poter utilizzare AWS Glue il trasferimento di dati a partire da lunedì, è necessario soddisfare i seguenti requisiti:

Requisiti minimi

- Hai un account sviluppatore Monday con e-mail e password. Per ulteriori informazioni, consulta Per creare un nuovo account Monday:.
- Il tuo account sviluppatore Monday è abilitato all'accesso alle API. Tutti gli utilizzi del lunedì APIs sono disponibili senza costi aggiuntivi durante il periodo di prova. Una volta terminato il periodo di prova, è necessario acquistare un abbonamento per creare e accedere ai dati. Per ulteriori informazioni, consulta la pagina delle licenze di Monday.

Se soddisfi questi requisiti, sei pronto per connetterti AWS Glue al tuo account Monday. Per le connessioni tipiche, non è necessario fare nient'altro in Monday.

## Configurazione delle connessioni di Monday

Monday supporta i due tipi di meccanismo di autenticazione seguenti:

1. OAuth Autenticazione: Monday supporta il tipo di concessione AUTHORIZATION_CODE per. OAuth2
   - Questo tipo di concessione è considerato «a tre gambe» in OAuth quanto si basa sul reindirizzamento degli utenti al server di autorizzazione di terze parti per autenticare l'utente. Viene utilizzato durante la creazione di connessioni tramite la console. AWS Glue Per impostazione predefinita, l'utente che crea una connessione può fare affidamento su un'app connessa di AWS Glue proprietà in cui non è necessario fornire alcuna informazione OAuth correlata ad eccezione dell'url dell'istanza di lunedì. La AWS Glue console reindirizzerà l'utente a Monday, dove l'utente deve effettuare il login e concedere le autorizzazioni richieste per accedere AWS Glue all'istanza di Monday.
   - Gli utenti devono scegliere di creare la propria app connessa lunedì e fornire il proprio ID client e il segreto del client quando creano connessioni tramite la AWS Glue Console. In questo scenario, verranno comunque reindirizzati a lunedì per effettuare il login e autorizzare l'accesso AWS Glue alle proprie risorse.
   - Questo tipo di concessione genera un token di aggiornamento e uno di accesso. Il token di accesso è attivo per un'ora e può essere aggiornato automaticamente senza l'interazione dell'utente utilizzando il token di aggiornamento.

- Per ulteriori informazioni, consulta la documentazione sulla creazione di un'app connessa per il flusso OAuth AUTHORIZATION_CODE.

2. Autenticazione personalizzata:

- Per la documentazione pubblica del lunedì sulla generazione delle chiavi API richieste per l'autorizzazione personalizzata, consulta docs/authentication#. https://developer.monday.com/api-reference/ api-token-permissions

Per configurare una connessione a Monday:

1. Nel Gestione dei segreti AWS, crea un segreto con i seguenti dettagli. È necessario creare un segreto per ogni connessione in AWS Glue.
   a. Per l' OAuth autenticazione:
      - Per l'app connessa gestita dal cliente, il segreto deve contenere l'app connessa Consumer Secret con USER_MANAGED_CLIENT_APPLICATION_CLIENT_SECRET come chiave.
   b. Per l'autenticazione personalizzata:
      - Per le app connesse gestite dal cliente: il segreto deve contenere l'app connessa Consumer Secret con `personalAccessToken` come chiave.
2. In AWS Glue Studio, crea una connessione in Connessioni dati seguendo i passaggi seguenti:
   a. In Connessioni dati, scegliere Crea connessione.
   b. Quando si seleziona una Origine dati, selezionare Monday.
   c. Fornire la InstanceURL di Monday.
   d. Seleziona il ruolo IAM che AWS Glue può assumere e dispone delle autorizzazioni per le seguenti azioni:

      JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:DescribeSecret",
        "secretsmanager:GetSecretValue",
        "secretsmanager:PutSecretValue",
        "ec2:CreateNetworkInterface",
```

```
            "ec2:DescribeNetworkInterfaces",
            "ec2:DeleteNetworkInterface"
        ],
        "Resource": "*"
      }
    ]
}
```

e. Selezionare il tipo di autenticazione per connetterti a Monday
   - Per l' OAuth autenticazione: fornisci l'URL del token e l'applicazione client gestita dagli utenti ClientId del lunedì a cui desideri connetterti.
   - Per l'autenticazione personalizzata: selezionare il tipo di autenticazione CUSTOM per connettersi a Monday.

f. Seleziona quello `secretName` che desideri utilizzare per questa connessione per AWS Glue inserire i token.

g. Selezionare le opzioni di rete se si desidera utilizzare la propria rete.

3. Concedi il permesso di lettura `secretName` al ruolo IAM associato al tuo AWS Glue lavoro. Scegli Next (Successivo).
4. Nella configurazione del tuo AWS Glue lavoro, fornisci `connectionName` una connessione di rete aggiuntiva.

## Lettura dalle entità di Monday

Prerequisiti

- Un oggetto di Monday da cui desideri leggere. Consulta la tabella delle entità supportate qui di seguito per verificare le entità disponibili.

Entità supportate come origine

Elenco delle entità:

- Conto: https://developer.monday.com/api-reference/docs/account #queries
- Bacheca: docs/boards #queries https://developer.monday.com/api-reference/
- Colonna: docs/columns #queries https://developer.monday.com/api-reference/
- Documenti: docs/docs #queries https://developer.monday.com/api-reference/
- Blocco di documento: docs/blocks #queries https://developer.monday.com/api-reference/

- File: docs/files #queries https://developer.monday.com/api-reference/
- Cartelle: docs/folders #queries https://developer.monday.com/api-reference/
- Gruppi: docs/groups #queries https://developer.monday.com/api-reference/
- Elemento: docs/items #queries https://developer.monday.com/api-reference/
- Sottoelementi: docs/subitems #queries https://developer.monday.com/api-reference/
- Tag: docs/tags-queries #queries https://developer.monday.com/api-reference/
- https://developer.monday.com/api-reference/Squadre: docs/teams #queries
- Aggiornamenti: docs/updates #queries https://developer.monday.com/api-reference/
- Utenti: docs/users #queries https://developer.monday.com/api-reference/
- Spazi di lavoro: docs/workspaces #queries https://developer.monday.com/api-reference/

| Entità | Può essere Filtrato | Supporta Limite | Supporta Ordina per | Supporta Seleziona * | Supporta Partizionamento |
|---|---|---|---|---|---|
| Account | No | No | No | Sì | No |
| Bacheche | Sì | Sì | No | Sì | No |
| Colonne | No | No | No | Sì | No |
| Documenti | Sì | Sì | No | Sì | No |
| Blocchi documenti | No | Sì | No | Sì | No |
| File | Sì | No | No | Sì | No |
| Gruppi | No | No | No | Sì | No |
| Elemento | Sì | Sì | No | Sì | No |
| Elementi secondari | No | No | No | Sì | No |
| Tag | Sì | No | No | Sì | Sì |

| Entità | Può essere Filtrato | Supporta Limite | Supporta Ordina per | Supporta Seleziona * | Supporta Partizionamento |
|---|---|---|---|---|---|
| Team | Sì | No | No | Sì | No |
| Aggiornamenti | No | Sì | No | Sì | No |
| Utenti | Sì | Sì | No | Sì | No |
| WorkSpace | Sì | Sì | No | Sì | No |
| Cartelle | Sì | Sì | No | Sì | No |

Esempio

```
monday_read = glueContext.create_dynamic_frame.from_options(
     connection_type="monday",
     connection_options={
         "connectionName": "connectionName",
         "ENTITY_NAME": "account",
         "API_VERSION": "v2"
     }
```

## Indicazioni di riferimento alle opzioni di connessione a Monday

Di seguito sono riportate le opzioni di connessione per Monday:

- `ENTITY_NAME`(Stringa): (obbligatorio) utilizzato per lettura/scrittura. Il nome del tuo oggetto in Monday.
- `API_VERSION`(Stringa): (obbligatorio) utilizzato per lettura/scrittura. Versione dell'API Rest di Monday da utilizzare. Esempio: v2.
- `SELECTED_FIELDS`(List<String>): predefinito: vuoto (SELECT *). Utilizzato per la lettura. Colonne da selezionare per l'oggetto.
- `FILTER_PREDICATE`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Dev'essere in formato Spark SQL.
- `QUERY`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Query Spark SQL completa.

## Limitazioni

Di seguito sono riportate le limitazioni per il connettore di Monday:

- La risposta dinamica ai metadati presenta alcuni conflitti con la documentazione, così come indicato qui di seguito:
  - L'entità Group, Column supporta le operazioni di filtro, ma non è presente nell'endpoint dinamico dei metadati, quindi viene mantenuta come entità non filtrabile.
  - L'endpoint dinamico è composto da più di 15.000 righe e restituisce i metadati di tutte le entità in un'unica risposta. Di conseguenza, il caricamento dei campi impiega in media 10 secondi, quindi ciò richiede del tempo aggiuntivo durante l'esecuzione di un processo.
  - Consulta la tabella seguente per il limite tariffario di Monday. La dimensione significativa dei dati di risposta dell'entità dinamica causa un notevole ritardo, con i campi che richiedono in media 10 secondi per il caricamento.

| | |
|---|---|
| Limite di complessità | 5.000.000 (5 milioni) di punti di complessità |
| Limite chiamate giornaliere | 10.000 per il piano Pro |
| Limite minuti | 500 query al minuto |
| Limite di concorrenza | 100 richieste simultanee massime per il piano Pro |

## Per creare un nuovo account Monday:

1. Vai alla home page di lunedì, https://monday.com/e scegli Accedi.
2. Verrai reindirizzato alla pagina di accesso. Nella parte inferiore della pagina, seleziona Registrati.
3. Inserisci il tuo indirizzo e-mail e seleziona Continua. In alternativa, puoi accedere con Google.
4. Inserisci i dati richiesti e seleziona Continua.
5. Rispondi alle domande del sondaggio e segui i passaggi indicati per completare la procedura di creazione dell'account.

Registra un' OAuth applicazione:

1. Accedi al tuo account monday.com. Fai clic sul tuo avatar (l'icona con l'immagine) nell'angolo in basso a sinistra dello schermo.
2. Seleziona Sviluppatore.
3. Seleziona Crea app.
4. Compila i campi obbligatori relativi al nome e alla descrizione.
5. Vai alla sezione «OAuth» presente sul lato destro, aggiungi gli ambiti e scegli «Salva funzione».
6. Passa alla scheda "Reindirizza URL" accanto all'ambito, quindi aggiungi l'URL di reindirizzamento e seleziona "Salva funzionalità".
7. URLsNella scheda Redirect, fornisci l'URL della tua app. Dovrebbe essere https://{region-code}.console.aws.amazon.com/appflow/ oauth. Ad esempio, se stai usando `us-east-1` , puoi aggiungere `https://us-east-1.console.aws.amazon.com/appflow/oauth`.
8. A questo punto, l'applicazione è pronta per l'uso. Puoi trovare le tue credenziali nella sezione "Informazioni di base". Prendi nota delle stringhe ID client e segreto client. Queste stringhe vengono utilizzate per stabilire una connessione con questa app utilizzando un connettore. AppFlow

Per generare un token di accesso personale:

Al momento, monday.com offre solo i nostri token API V2, che sono tutti token personali. Per accedere ai tuoi token API, puoi usare uno dei due metodi disponibili a seconda del tuo livello utente. Gli utenti admin possono usare entrambi i metodi per acquisire i propri token API. Gli utenti membri possono accedere ai propri token API dalle schede Sviluppatore.

Admin: se sei un utente admin del tuo account monday.com, puoi accedere ai token API dalla scheda "Admin" seguendo questi passaggi:

1. Accedi al tuo account monday.com. Fai clic sul tuo avatar (l'icona con l'immagine) nell'angolo in basso a sinistra dello schermo.
2. Seleziona "Admin" dal menu che appare (per farlo devi avere i permessi di admin).
3. Vai alla sezione "API" e genera un "token API V2". Puoi copiare il tuo token e usarlo.

Sviluppatore: se sei un utente membro del tuo account monday.com, puoi accedere ai tuoi token API dalla scheda Sviluppatore seguendo questi passaggi:

1. Accedi al tuo account monday.com. Fai clic sul tuo avatar (l'icona con l'immagine) nell'angolo in basso a sinistra dello schermo.
2. Seleziona "Sviluppatori" dal menu che appare.
3. Nel menu in alto, seleziona il menu a discesa "Sviluppatore". Seleziona la prima opzione nel menu a discesa intitolato "I miei token di accesso".

## Connessione a MongoDB in AWS Glue Studio

AWS Glue fornisce il supporto integrato per MongoDB. AWS Glue Studio fornisce un'interfaccia visiva per connettersi a MongoDB, creare processi di integrazione dei dati ed eseguirli sul runtime Spark serverless AWS Glue Studio.

Argomenti

- Creazione di una connessione MongoDB
- Creazione di un nodo di origine MongoDB
- Creazione di un nodo di destinazione MongoDB
- Opzioni avanzate

### Creazione di una connessione MongoDB

Prerequisiti:

- Se la tua istanza MongoDB si trova in un Amazon VPC, configura Amazon VPC per consentire al AWS Glue job di comunicare con l'istanza MongoDB senza che il traffico attraversi la rete Internet pubblica.

  In Amazon VPC, identifica o crea un VPC, una sottorete e un gruppo di sicurezza da utilizzare durante l'esecuzione del AWS Glue lavoro. Inoltre, assicurati che Amazon VPC sia configurato per consentire il traffico di rete tra l'istanza MongoDB e questa posizione. In base al layout della rete, ciò potrebbe richiedere modifiche alle regole del gruppo di sicurezza, alla rete ACLs, ai gateway NAT e alle connessioni peering.

Per configurare una connessione a MongoDB:

1. Facoltativamente Gestione dei segreti AWS, crea un segreto usando le tue credenziali MongoDB. Per creare un segreto in Secrets Manager, segui il tutorial disponibile in Crea un

Gestione dei segreti AWS segreto nella Gestione dei segreti AWS documentazione. Dopo aver creato il segreto, mantieni il nome del segreto *secretName* per il passaggio successivo.

- Quando selezionate le coppie chiave/valore, create una coppia per la chiave `username` con il valore. *mongodbUser*

  Quando selezionate le coppie chiave/valore, create una coppia per la chiave `password` con il valore. *mongodbPass*

2. Nella AWS Glue console, crea una connessione seguendo la procedura riportata di seguito. the section called "Aggiungere una AWS Glue connessione" Dopo aver creato la connessione, conserva il nome della connessione*connectionName*, per utilizzi futuri in AWS Glue.

   - Quando selezioni un tipo di connessione, seleziona MongoDB o MongoDB Atlas.
   - Quando selezioni l'URL MongoDB o URL MongoDB Atlas, fornisci il nome host dell'istanza MongoDB.

     Un URL MongoDB viene fornito nel formato `mongodb://`*mongoHost*`:`*mongoPort*`/`*mongoDBname*.

     Un URL MongoDB Atlas viene fornito nel formato `mongodb +srv://`*mongoHost*`/`*mongoDBname*.
   - Se hai scelto di creare un segreto di Secrets Manager, scegli il tipo di Gestione dei segreti AWS credenziale.

     Quindi, in AWS Secret fornisci*secretName*.
   - Se scegli di fornire nome utente e password, fornisci *mongodbUser* e*mongodbPass*.
3. Nelle seguenti situazioni, potresti aver bisogno di una configurazione aggiuntiva:

   - Per le istanze MongoDB ospitate su AWS un Amazon VPC
     - Dovrai fornire le informazioni di connessione Amazon VPC alla AWS Glue connessione che definisce le tue credenziali di sicurezza MongoDB. Durante la creazione o l'aggiornamento della connessione, imposta VPC, sottorete e Gruppi di sicurezza nelle opzioni di rete.

Dopo aver creato una AWS Glue connessione MongoDB, dovrai eseguire i seguenti passaggi prima di eseguire il job: AWS Glue

- Quando lavori con AWS Glue lavori nell'editor visivo, devi fornire le informazioni sulla connessione Amazon VPC affinché il lavoro possa connettersi a MongoDB. Identifica una posizione adatta in Amazon VPC e forniscila alla tua connessione MongoDB AWS Glue .
- Se hai scelto di creare un segreto di Secrets Manager, concedi al ruolo IAM associato al tuo AWS Glue lavoro il permesso di lettura*secretName*.

## Creazione di un nodo di origine MongoDB

Prerequisiti necessari

- Una connessione AWS Glue MongoDB, come descritto nella sezione precedente,. the section called "Creazione di una connessione MongoDB"
- Se hai scelto di creare un segreto di Secrets Manager, autorizzazioni appropriate sul tuo processo per leggere il segreto usato dalla connessione.
- Una raccolta MongoDB da cui desideri leggere. Avrai bisogno delle informazioni di identificazione per la raccolta.

  Una raccolta MongoDB è identificata da un nome di database e da un nome di raccolta,,. *mongodbName* *mongodbCollection*

Aggiunta di un'origine dati MongoDB

Per aggiungere un nodo origine dati: MongoDB:

1. Scegli la connessione per la tua origine dati MongoDB. Dato che l'hai creato, dovrebbe essere disponibile nel menu a discesa. Se devi creare una connessione, scegli Crea connessione MongoDB. Per ulteriori informazioni, consulta la sezione the section called "Creazione di una connessione MongoDB" precedente.

   Dopo aver scelto una connessione, puoi visualizzare le proprietà della connessione facendo clic su Visualizza proprietà.
2. Scegli un Database. Specificare *mongodbName*.
3. Scegli una Raccolta. Specificare *mongodbCollection*.
4. Scegli il tuo partizionatore, la dimensione della partizione (MB) e la chiave di partizione. Per ulteriori informazioni sui parametri di partizione, consulta the section called ""connectionType": "mongodb" come sorgente".

5. In Proprietà personalizzate di MongoDB, inserisci i parametri e i valori necessari.

## Creazione di un nodo di destinazione MongoDB

Prerequisiti necessari

- Una connessione AWS Glue MongoDB, configurata con Gestione dei segreti AWS un segreto, come descritto nella sezione precedente,. the section called "Creazione di una connessione MongoDB"
- Autorizzazioni appropriate sul processo per leggere il segreto utilizzato dalla connessione.
- Una tabella MongoDB su cui scrivere,. *`tableName`*

Aggiunta di una destinazione dati MongoDB

Per aggiungere un nodo di destinazione dati: MongoDB:

1. Scegli la connessione per la tua origine dati MongoDB. Dato che l'hai creato, dovrebbe essere disponibile nel menu a discesa. Se devi creare una connessione, scegli Crea connessione MongoDB. Per ulteriori informazioni, consulta la sezione the section called "Creazione di una connessione MongoDB" precedente.

   Dopo aver scelto una connessione, puoi visualizzare le proprietà della connessione facendo clic su Visualizza proprietà.
2. Scegli un Database. Specificare *`mongodbName`*.
3. Scegli una Raccolta. Specificare *`mongodbCollection`*.
4. Scegli il tuo partizionatore, la dimensione della partizione (MB) e la chiave di partizione. Per ulteriori informazioni sui parametri di partizione, consulta the section called ""connectionType": "mongodb" come sorgente".
5. Se lo desideri, scegli Riprova a scrivere.
6. In Proprietà personalizzate di MongoDB, inserisci i parametri e i valori necessari.

## Opzioni avanzate

È possibile fornire opzioni avanzate durante la creazione di un nodo MongoDB. Queste opzioni sono le stesse disponibili durante la programmazione AWS Glue per gli script Spark.

Per informazioni, consulta the section called "Connessione MongoDB".

# Connessione a Oracle NetSuite

Oracle NetSuite è una soluzione di gestione aziendale all-in-one cloud che aiuta le organizzazioni a operare in modo più efficace automatizzando i processi principali e fornendo visibilità in tempo reale sulle prestazioni operative e finanziarie. Con un'unica suite integrata di applicazioni per la gestione della contabilità, l'elaborazione degli ordini, la gestione dell'inventario, la produzione, la catena di approvvigionamento e le operazioni di magazzino, Oracle NetSuite offre alle aziende una chiara visibilità dei propri dati e un controllo più stretto sulle proprie attività.

Argomenti


- AWS Glue supporto per Oracle NetSuite
- Policy contenenti le operazioni API per creare e utilizzare le connessioni
- Configurazione di Oracle NetSuite
- Configurazione delle connessioni Oracle NetSuite
- Lettura da NetSuite entità Oracle
- Opzioni di NetSuite connessione Oracle
- Limitazioni e note per il NetSuite connettore Oracle


## AWS Glue supporto per Oracle NetSuite

AWS Glue supporta Oracle NetSuite come segue:

Supportato come origine?

Sì. È possibile utilizzare i job AWS Glue ETL per interrogare i dati da Oracle NetSuite.

Supportato come destinazione?

No.

Versioni Oracle NetSuite API supportate

Sono supportate le seguenti versioni NetSuite dell'API Oracle:

- v1

Per il supporto delle entità per le versioni specifiche, consulta Supported entities for Source.

## Policy contenenti le operazioni API per creare e utilizzare le connessioni

La seguente policy di esempio descrive le autorizzazioni AWS IAM richieste per la creazione e l'utilizzo delle connessioni. Se si sta creando un nuovo ruolo, creare una policy che contenga quanto segue:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:ListConnectionTypes",
        "glue:DescribeConnectionType",
        "glue:RefreshOAuth2Tokens",
        "glue:ListEntities",
        "glue:DescribeEntity"
      ],
      "Resource": "*"
    }
  ]
}
```

In alternativa, se non desideri utilizzare il metodo precedente, utilizza le seguenti policy IAM gestite:

- AWSGlueServiceRole— Garantisce l'accesso alle risorse che AWS Glue i vari processi richiedono per l'esecuzione per conto dell'utente. Queste risorse includono AWS Glue Amazon S3, IAM, CloudWatch Logs e Amazon. EC2 Se segui la convenzione di denominazione delle risorse specificata in questa politica, AWS Glue i processi dispongono delle autorizzazioni richieste. Questa policy è in genere collegata ai ruoli specificati quando si definiscono crawler, processi ed endpoint di sviluppo.
- AWSGlueConsoleFullAccess— Concede l'accesso completo alle AWS Glue risorse quando un'identità a cui è allegata la policy utilizza la AWS console di gestione. Se si segue la convenzione per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questa policy è in genere associata agli utenti della AWS Glue console.

## Configurazione di Oracle NetSuite

Prima di poter utilizzare il trasferimento AWS Glue di dati da Oracle NetSuite, è necessario soddisfare i seguenti requisiti:

Requisiti minimi

I requisiti minimi sono i seguenti:

- Hai un NetSuite account Oracle. Per ulteriori informazioni, consulta Creazione di un NetSuite account Oracle.
- Il tuo NetSuite account Oracle è abilitato per l'accesso all'API.
- Hai creato un'integrazione API OAuth 2.0 nel tuo account NetSuite sviluppatore Oracle. Questa integrazione fornisce le credenziali del client che AWS Glue utilizza per accedere ai dati in modo sicuro quando effettua chiamate autenticate al vostro account. Per ulteriori informazioni, consulta Creazione di un'app NetSuite client Oracle e credenziali OAuth 2.0.

Se soddisfi questi requisiti, sei pronto per connetterti al tuo account AWS Glue Oracle. NetSuite

Creazione di un NetSuite account Oracle

Vai su NetSuiteOracle e scegli Free Product Tour. Inserire i dettagli richiesti per ottenere una dimostrazione gratuita del prodotto, tramite la quale è possibile contattare un fornitore. Di seguito è riportato il procedimento per l'acquisizione di un account:

- L'acquisizione di un NetSuite account viene effettuata tramite un fornitore, che fornisce un account form/quote che deve essere esaminato legalmente.
- L'account da acquistare per Oracle NetSuite Connector è di Standard Cloud Service.
- Questo account viene creato dal fornitore e le credenziali temporanee vengono condivise da quest'ultimo. Riceverai un'e-mail di benvenuto NetSuite < billing@notification.netsuite.com > < system@sent-via.netsuite.com > con dettagli come il tuo nome utente e un link per impostare la password.
- Utilizzare il link Imposta password per impostare la password per il nome utente fornito dal fornitore.

Creazione di un'app NetSuite client Oracle e credenziali OAuth 2.0

Per ottenere l'ID client e il segreto del cliente, crei un'app NetSuite client Oracle:

1. Accedi al tuo NetSuite account tramite il login NetSuite del cliente.
2. Scegliere Configurazione > Azienda > Abilita funzionalità.
3. Vai alla SuiteCloudsezione e seleziona la casella di controllo REST WEB SERVICES sotto SuiteTalk (Servizi Web).
4. Selezionare la casella di spunta OAUTH 2.0 in Gestisci autenticazione. Fai clic su Salva.
5. Vai a Configurazione > Integrazione > Gestione delle integrazioni e scegli Nuovo per creare un'applicazione OAuth2 2.0.
6. Inserire un nome a scelta e mantenere lo STATO come Abilitato.
7. Se selezionate, deselezionare le caselle di spunta TBA: FLUSSO DI AUTORIZZAZIONE e AUTENTICAZIONE BASATA SU TOKEN visualizzate in Autenticazione basata su token.
8. Seleziona le caselle di controllo AUTHORIZATION CODE GRANT e PUBLIC CLIENT nella sezione 2.0. OAuth
9. In Auth, prendere nota dell'ID client del segreto client.
10. Inserire un URI DI REINDIRIZZAMENTO. Ad esempio, oauth https://us-east-1.console.aws.amazon.com/gluestudio/
11. Selezionare la casella di spunta REST WEB SERVICES in AMBITO.
12. Selezionare la casella di spunta CREDENZIALI UTENTE in Credenziali utente. Scegli Save (Salva).
13. Annota l' KEY/CLIENT ID CONSUMATORE e IL SECRET/CLIENT SEGRETO DEL CONSUMATORE nelle credenziali del cliente. Questi valori vengono visualizzati una sola volta.
14. Se necessario, creare un ruolo AMMINISTRATORE andando su Utente/Ruoli > Gestisci ruoli > Nuovo.
15. Quando si crea un ruolo personalizzato, aggiungere l'accesso completo nella scheda Autorizzazioni per le seguenti entità/funzionalità:
    - «Deposito», «Articoli», «Gestione articoli», «Registra registrazione», «Ordine di acquisto», «Filiali», «Fornitori», «Fatture», «Autorizzazione alla restituzione del fornitore», «Track Time», «Pagamento cliente», «Inserimenti di record personalizzati», «Tipi di record personalizzati», «Servizi Web REST», «Gestione applicazioni autorizzate OAuth 2.0», «Campi di accesso personalizzati per entità», «Accesso tramite OAuth 2.0 Gettoni».

Per ulteriori informazioni, vedere OAuth 2.0 nella documentazione di NetSuite Applications Suite.

## Configurazione delle connessioni Oracle NetSuite

Oracle NetSuite supporta il tipo di concessione AUTHORIZATION_CODE per. OAuth2 Il tipo di concessione determina il modo in cui AWS Glue comunica con Oracle NetSuite per richiedere l'accesso ai dati.

- Questo tipo di concessione è considerato «a tre gambe» in OAuth quanto si basa sul reindirizzamento degli utenti a un server di autorizzazione di terze parti per autenticare l'utente. Viene utilizzato durante la creazione di connessioni tramite la console. AWS Glue Per impostazione predefinita, l'utente che crea una connessione può fare affidamento su un'app connessa di AWS Glue proprietà (applicazione client AWS Glue gestita) in cui non è necessario fornire alcuna informazione OAuth correlata ad eccezione dell'URL dell' NetSuite istanza Oracle. La AWS Glue console reindirizzerà l'utente a Oracle, NetSuite dove l'utente deve effettuare il login e AWS Glue concedere le autorizzazioni richieste per accedere alla propria istanza Oracle. NetSuite
- Gli utenti possono comunque scegliere di creare la propria app connessa in Oracle NetSuite e fornire il proprio ID client e il segreto del client durante la creazione di connessioni tramite la AWS Glue console. In questo scenario, verranno comunque reindirizzati a Oracle per effettuare NetSuite il login e autorizzare l'accesso AWS Glue alle proprie risorse.
- Questo tipo di concessione genera un token di aggiornamento e un token di accesso. Il token di accesso dura poco tempo e può essere aggiornato automaticamente senza l'interazione dell'utente utilizzando il token di aggiornamento.
- Per la NetSuite documentazione pubblica di Oracle sulla creazione di un'app connessa per il OAuth flusso del codice di autorizzazione, vedi App pubbliche.

Per configurare una NetSuite connessione Oracle:

1. In AWS Secrets Manager, crea un segreto con i seguenti dettagli:
   a. Per le app connesse gestite dal cliente: il segreto deve contenere l'app connessa Consumer Secret con `USER_MANAGED_CLIENT_APPLICATION_CLIENT_SECRET` come chiave.
   b. Nota: è necessario creare un segreto per la connessione in AWS Glue.
2. In AWS Glue Glue Studio, crea una connessione in Connessioni dati seguendo i passaggi seguenti:
   a. Quando selezioni un tipo di connessione, seleziona Oracle NetSuite.
   b. Fornisci l' NetSuite ambiente Oracle.

c. Seleziona il ruolo AWS IAM che AWS Glue può assumere e dispone delle autorizzazioni per le seguenti azioni:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:DescribeSecret",
        "secretsmanager:GetSecretValue",
        "secretsmanager:PutSecretValue",
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DeleteNetworkInterface"
      ],
      "Resource": "*"
    }
  ]
}
```

d. Seleziona quello `secretName` che desideri utilizzare per questa connessione per AWS Glue inserire i token.

e. Selezionare le opzioni di rete se si desidera utilizzare la propria rete.

3. Concedi il permesso di lettura `secretName` al ruolo IAM associato al tuo AWS Glue lavoro.

## Lettura da NetSuite entità Oracle

Prerequisito

Un NetSuite oggetto Oracle da cui desideri leggere. È necessario il nome dell'oggetto, come `deposit` o `timebill`. Nella tabella seguente sono illustrate le entità supportate.

Entità supportate come origine:

| Entità | Può essere filtrato | Supporta Ordina per | Supporta Limite | Supporta SELEZIONA * | Supporta il partizionamento |
|---|---|---|---|---|---|
| Deposito | Sì | No | Sì | Sì | Sì |
| Descrizione articolo | Sì | No | Sì | Sì | Sì |
| Articolo di inventario | Sì | No | Sì | Sì | Sì |
| Evasione dell'articolo | Sì | No | Sì | Sì | Sì |
| Gruppo di articoli | Sì | No | Sì | Sì | Sì |
| Registrazione nel diario | Sì | No | Sì | Sì | Sì |
| Articolo di acquisto non in inventario | Sì | No | Sì | Sì | Sì |
| Articolo di rivendita non in inventario | Sì | No | Sì | Sì | Sì |
| Articolo di vendita non in inventario | Sì | No | Sì | Sì | Sì |
| Ordine di acquisto | Sì | No | Sì | Sì | Sì |
| Affiliata | Sì | No | Sì | Sì | Sì |
| Fornitori | Sì | No | Sì | Sì | Sì |

| Entità | Può essere filtrato | Supporta Ordina per | Supporta Limite | Supporta SELEZIONA * | Supporta il partizionamento |
|---|---|---|---|---|---|
| Fattura fornitore | Sì | No | Sì | Sì | Sì |
| Autorizzazione per il reso al fornitore | Sì | No | Sì | Sì | Sì |
| Ora di fatturazione | Sì | No | Sì | Sì | Sì |
| Pagamento cliente | Sì | No | Sì | Sì | Sì |
| Richiesta di evasione | Sì | No | Sì | Sì | Sì |
| Elemento | Sì | Sì | Sì | Sì | Sì |
| Linea di transazione | Sì | Sì | Sì | Sì | Sì |
| Linea di contabilità di transazione | Sì | Sì | Sì | Sì | Sì |
| Tipi di record personalizzati (dinamici) | Sì | Sì | Sì | Sì | Sì |

Esempio:

```
netsuiteerp_read = glueContext.create_dynamic_frame.from_options(
    connection_type="netsuiteerp",
    connection_options={
```

```
        "connectionName": "connectionName",
        "ENTITY_NAME": "deposit",
        "API_VERSION": "v1"
    }
)
```

Dettagli NetSuite dell'entità e dei campi Oracle:

Oracle carica NetSuite dinamicamente i campi disponibili nell'entità selezionata. A seconda del tipo di dati di campo, supporta i seguenti operatori di filtro.

| Tipo di dati dei campi | Operatori di filtro supportati |
|---|---|
| Stringa | LIKE, =, != |
| Data | BETWEEN, =, <, <=, >, >= |
| DateTime | BETWEEN, <, <=, >, >= |
| Numerico | =, !=, <, <=, >, >= |
| Booleano | =, != |

Formato di input previsto per i valori booleani in Espressione filtro:

| Entità | Formato di valore booleano “true” | Formato di valore booleano “false” | Esempio |
|---|---|---|---|
| Entità Articolo, Linea di transazione, Linea di contabilità di transazione e Tipo di record personalizzato | T o t | F o f | isinactive = “T” oppure isinactive = “t” |
| Tutte le altre entità | true | false | isinactive = true |

Query di partizionamento

Partizionamento basato sul campo

Il NetSuite connettore Oracle dispone di metadati dinamici in modo che i campi supportati per il partizionamento basato sui campi vengano scelti dinamicamente. Il partizionamento basato sui campi è supportato nei campi con il tipo di dati Integer, Date o. BigInteger DateTime

Puoi fornire le opzioni Spark aggiuntive `PARTITION_FIELD`, `LOWER_BOUND`, `UPPER_BOUND` e `NUM_PARTITIONS` se desideri utilizzare la concorrenza in Spark. Con questi parametri, la query originale viene suddivisa in `NUM_PARTITIONS` query secondarie che possono essere eseguite contemporaneamente dalle attività Spark.

- `PARTITION_FIELD`: il nome del campo da utilizzare per partizionare la query.
- `LOWER_BOUND`: un valore limite inferiore inclusivo del campo di partizione scelto.

  Per il campo timestamp, accettiamo il formato di timestamp Spark utilizzato nelle query SQL di Spark.

  Esempi di valori validi:

  ```
  "TIMESTAMP \"1707256978123\""
  "TIMESTAMP \"1702600882\""
  "TIMESTAMP '2024-02-06T22:00:00.000Z'"
  "TIMESTAMP '2024-02-06T22:00:00Z'"
  "TIMESTAMP '2024-02-06'"
  ```

- `UPPER_BOUND`: un valore limite superiore esclusivo del campo di partizione scelto.
- `NUM_PARTITIONS`: il numero di partizioni.

Esempio:

```
netsuiteerp_read = glueContext.create_dynamic_frame.from_options(
    connection_type="netsuiteerp",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "deposit",
        "API_VERSION": "v1",
        "PARTITION_FIELD": "id",
        "LOWER_BOUND": "1",
        "UPPER_BOUND": "10000",
        "NUM_PARTITIONS": "10"
    }
```

Partizionamento basato su record

È possibile fornire l'opzione Spark aggiuntiva NUM_PARTITIONS se si desidera utilizzare la simultaneità in Spark. Con questo parametro, la query originale viene suddivisa in NUM_PARTITIONS query secondarie che possono essere eseguite contemporaneamente dalle attività Spark.

Nel partizionamento basato sui record, il numero totale di record presenti viene interrogato dall' NetSuite API Oracle e diviso per il NUM_PARTITIONS numero fornito, il numero di record risultante viene quindi recuperato contemporaneamente da ciascuna sottoquery.

- NUM_PARTITIONS: il numero di partizioni.

Esempio:

```
netsuiteerp_read = glueContext.create_dynamic_frame.from_options(
    connection_type="netsuiteerp",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "deposit",
        "API_VERSION": "v1",
        "NUM_PARTITIONS": "3"
    }
```

## Opzioni di NetSuite connessione Oracle

Di seguito sono riportate le opzioni di connessione per Oracle NetSuite:

- ENTITY_NAME(String): (obbligatorio) utilizzato per la lettura. Il nome dell' NetSuite entità Oracle. Esempio: deposito.
- API_VERSION(String): (obbligatorio) utilizzato per la lettura. Versione dell' NetSuite API Oracle Rest che desideri utilizzare. Il valore sarà v1, poiché Oracle NetSuite attualmente supporta solo la versione v1.
- SELECTED_FIELDS(List<String>): predefinito: vuoto (SELECT *). Utilizzato per la lettura. Elenco di colonne separate da virgole che si desidera selezionare per l'entità selezionata.
- FILTER_PREDICATE(String): predefinito: vuoto. Utilizzato per la lettura. Dev'essere in formato Spark SQL.
- QUERY(Stringa): predefinito: vuoto. Utilizzato per la lettura. Query Spark SQL completa.

- PARTITION_FIELD(Stringa): utilizzato per la lettura. Campo da utilizzare per partizionare la query (partizionamento basato sul campo).
- LOWER_BOUND(Stringa): utilizzato per la lettura. Un valore limite inferiore inclusivo del campo di partizione scelto, usato nel partizionamento basato sul campo.
- UPPER_BOUND(Stringa): utilizzato per la lettura. Un valore limite superiore esclusivo del campo di partizione scelto, usato nel partizionamento basato sul campo.
- NUM_PARTITIONS(Numero intero): predefinito: 1. Utilizzato per la lettura. Numero di partizioni da leggere. Utilizzato sia nel partizionamento basato sul campo che su quello basato sui record.
- INSTANCEE_URL(String): un URL di NetSuite istanza valido con formato. https://{account-id}.suitetalk.api.netsuite.com

## Limitazioni e note per il NetSuite connettore Oracle

Di seguito sono riportate le limitazioni o le note per il NetSuite connettore Oracle:

- I valori dei parametri access_token e refresh_token sono in formato JSON Web Token (JWT). Il token di accesso è valido per 60 minuti mentre il refresh_token è valido per sette giorni.
- Durante la generazione dell'ID client e del segreto client, se si seleziona “CLIENT PUBBLICO” insieme a “CONCESSIONE DI CODICE DI AUTORIZZAZIONE”, il token di aggiornamento è valido solo per tre ore e può essere utilizzato una sola volta.
- È possibile recuperare al massimo 1.00.000 record utilizzando il connettore. Per ulteriori informazioni, fare riferimento a Esecuzione di query SuiteQL tramite Web REST Services.
- Le partizioni vengono create in modo tale che ogni partizione recuperi i record in multipli di 1000, tranne possibilmente l'ultima che recupererà i record rimanenti.
- Per gli oggetti Articolo, Linea di transizione e Linea di contabilità di transazione, il connettore non supporterà alcuni operatori per i seguenti motivi:
  - L'applicazione degli operatori di filtro EQUAL_TO, NOT_EQUAL_TO ai campi di tipo Data fornisce risultati inaffidabili.
  - L'applicazione dell'operatore di filtro LESS_THAN_OR_EQUAL_TO ai campi di tipo Data fornisce risultati inaffidabili e ha un comportamento simile all'operatore LESS_THAN.
  - L'applicazione dell'operatore di filtro GREATER_THAN ai campi di tipo Data= fornisce risultati inaffidabili e ha un comportamento simile all'operatore GREATER_THAN_OR_EQUAL_TO.

- Per gli oggetti Item, Transaction Line, Transaction Accounting Line e Custom Record Type, i valori booleani sono disponibili nel formato T/F anziché in true/false. The connector maps the t/f values to true/false quello standard per garantire la coerenza dei dati.

## Connessione al OpenSearch servizio in AWS Glue Studio

AWS Gluefornisce supporto integrato per Amazon OpenSearch Service. AWS Glue Studiofornisce un'interfaccia visiva per connettersi ad Amazon OpenSearch Service, creare lavori di integrazione dei dati ed eseguirli sul runtime Spark AWS Glue Studio senza server. Questa funzionalità non è compatibile con OpenSearch Service serverless.

AWS Glue Studio crea una connessione unificata per Amazon OpenSearch Service. Per ulteriori informazioni, consulta Considerazioni.

Argomenti

- Creazione di una connessione OpenSearch al servizio
- Creazione di un nodo OpenSearch di origine del servizio
- Creazione di un nodo OpenSearch di destinazione del servizio
- Opzioni avanzate

### Creazione di una connessione OpenSearch al servizio

Prerequisiti:

- Identifica l'endpoint *aosEndpoint* e la porta del dominio da *aosPort* cui desideri leggere o crea la risorsa seguendo le istruzioni nella documentazione di Amazon OpenSearch Service. Per ulteriori informazioni sulla creazione di un dominio, consulta Creazione e gestione di domini Amazon OpenSearch Service nella documentazione di Amazon OpenSearch Service.

  Un endpoint OpenSearch di dominio Amazon Service avrà il seguente modulo predefinito, https://search - *domainName* -*unstructuredIdContent*. *region*.es.amazonaws.com. Per ulteriori informazioni sull'identificazione dell'endpoint del tuo dominio, consulta Creazione e gestione dei domini Amazon OpenSearch Service nella documentazione di Amazon OpenSearch Service.

  Identifica o genera credenziali di autenticazione HTTP di base *aosUser* e *aosPassword* per il tuo dominio.

Per configurare una connessione al OpenSearch servizio:

1. In Gestione dei segreti AWS, crea un segreto utilizzando le tue credenziali OpenSearch di servizio. Per creare un segreto in Secrets Manager, segui il tutorial disponibile in Crea un Gestione dei segreti AWS segreto nella Gestione dei segreti AWS documentazione. Dopo aver creato il segreto, mantieni il nome del segreto *secretName* per il passaggio successivo.

   - Quando selezionate le coppie chiave/valore, create una coppia per la chiave USERNAME con il valore. *aosUser*
   - Quando selezionate le coppie chiave/valore, create una coppia per la chiave PASSWORD con il valore. *aosPassword*

2. Nella AWS Glue console, crea una connessione seguendo la procedura riportata di seguito. the section called "Aggiungere una AWS Glue connessione" Dopo aver creato la connessione, conserva il nome della connessione*connectionName*, per utilizzi futuri in AWS Glue.

   - Quando selezioni un tipo di connessione, seleziona OpenSearch Servizio.
   - Quando selezioni un endpoint di dominio, fornisci*aosEndpoint*.
   - Quando selezioni una porta, fornisci*aosPort*.
   - Quando selezioni un AWS segreto, fornisci*secretName*.

## Creazione di un nodo OpenSearch di origine del servizio

Prerequisiti necessari

- Una connessione AWS Glue OpenSearch al servizio, configurata con un Gestione dei segreti AWS segreto, come descritto nella sezione precedente,the section called "Creazione di una connessione OpenSearch al servizio".
- Autorizzazioni appropriate sul processo per leggere il segreto utilizzato dalla connessione.
- Un indice dei OpenSearch servizi da cui desideri leggere,*aosIndex*.

Aggiungere una fonte OpenSearch di dati di servizio

Per aggiungere un'origine dati — nodo OpenSearch di servizio:

1. Scegli la connessione per la fonte OpenSearch di dati del servizio. Dato che l'hai creato, dovrebbe essere disponibile nel menu a discesa. Se devi creare una connessione, scegli Crea

connessione OpenSearch al servizio. Per ulteriori informazioni, consulta la sezione the section called "Creazione di una connessione OpenSearch al servizio" precedente.

Dopo aver scelto una connessione, puoi visualizzare le proprietà della connessione facendo clic su Visualizza proprietà.

2. Nel campo Indice, fornisci l'indice che desideri leggere.
3. Facoltativamente, fornisci Query, una OpenSearch query per fornire risultati più specifici. Per ulteriori informazioni sulla scrittura di OpenSearch interrogazioni, consulta. the section called "Leggi dal servizio OpenSearch "
4. Nelle proprietà OpenSearch del servizio personalizzato, immettete i parametri e i valori necessari.

## Creazione di un nodo OpenSearch di destinazione del servizio

Prerequisiti necessari

- Una connessione AWS Glue OpenSearch al servizio, configurata con un Gestione dei segreti AWS segreto, come descritto nella sezione precedente,the section called "Creazione di una connessione OpenSearch al servizio".
- Autorizzazioni appropriate sul processo per leggere il segreto utilizzato dalla connessione.
- Un indice di OpenSearch servizio su cui desideri scrivere,`aosIndex`.

Aggiungere un target OpenSearch di dati del servizio

Per aggiungere un target di dati — nodo OpenSearch di servizio:

1. Scegli la connessione per la fonte OpenSearch di dati del servizio. Dato che l'hai creato, dovrebbe essere disponibile nel menu a discesa. Se devi creare una connessione, scegli Crea connessione OpenSearch al servizio. Per ulteriori informazioni, consulta la sezione the section called "Creazione di una connessione OpenSearch al servizio" precedente.

   Dopo aver scelto una connessione, puoi visualizzare le proprietà della connessione facendo clic su Visualizza proprietà.
2. Nel campo Indice, fornisci l'indice che desideri leggere.
3. Nelle proprietà OpenSearch del servizio personalizzato, inserisci i parametri e i valori necessari.

### Opzioni avanzate

È possibile fornire opzioni avanzate durante la creazione di un nodo OpenSearch di servizio. Queste opzioni sono le stesse disponibili durante la programmazione AWS Glue per gli script Spark.

Per informazioni, consulta the section called "OpenSearch Connessioni di servizio".

## Connessione a Okta

L'API Okta è l'interfaccia programmatica di Okta, usata per gestire campagne e account Okta complessi o di grandi dimensioni. Se sei un utente Okta, puoi connetterti AWS Glue al tuo account Okta. Quindi, puoi usare Okta come origine dati nei tuoi processi ETL. Esegui questi lavori per trasferire dati tra Okta e i AWS servizi o altre applicazioni supportate.

Argomenti


- AWS Glue supporto per Okta
- Policy contenenti le operazioni API per creare e utilizzare le connessioni
- Configurare Okta
- Configurazione delle connessioni Okta
- Lettura dalle entità di Okta
- Indicazioni di riferimento alle opzioni di connessione a Okta
- Procedura di creazione di un nuovo account e di un'app per sviluppatori di Okta
- Limitazioni


### AWS Glue supporto per Okta

AWS Glue supporta Okta come segue:

Supportato come origine?

Sì. È possibile utilizzare i job AWS Glue ETL per interrogare i dati di Okta.

Supportato come destinazione?

No.

Versioni delle API Okta supportate

v1.

## Policy contenenti le operazioni API per creare e utilizzare le connessioni

La seguente politica di esempio descrive le AWS autorizzazioni richieste per la creazione e l'utilizzo delle connessioni. Se si sta creando un nuovo ruolo, creare una policy che contenga quanto segue:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:ListConnectionTypes",
        "glue:DescribeConnectionType",
        "glue:RefreshOAuth2Tokens",
        "glue:ListEntities",
        "glue:DescribeEntity"
      ],
      "Resource": "*"
    }
  ]
}
```

È anche possibile utilizzare le seguenti policy IAM per autorizzare l'accesso:

- AWSGlueServiceRole— Garantisce l'accesso alle risorse che diversi AWS Glue processi richiedono per l'esecuzione per conto dell'utente. Queste risorse includono Amazon S3 AWS Glue, IAM, CloudWatch Logs e Amazon EC2. Se segui la convenzione di denominazione delle risorse specificata in questa politica, AWS Glue i processi dispongono delle autorizzazioni necessarie. Questa policy è in genere collegata ai ruoli specificati quando si definiscono crawler, processi ed endpoint di sviluppo.
- AWSGlueConsoleFullAccess— Concede l'accesso completo alle AWS Glue risorse quando un'identità a cui è allegata la policy utilizza la AWS console di gestione. Se si segue la convenzione per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questa policy è in genere associata agli utenti della AWS Glue console.

## Configurare Okta

Prima di poter AWS Glue utilizzare il trasferimento di dati da o verso Okta, devi soddisfare questi requisiti:

Requisiti minimi

- Avere un account Okta. Per ulteriori informazioni sulla creazione di un account, consultare Procedura di creazione di un nuovo account e di un'app per sviluppatori di Okta.
- L'account Okta è abilitato per l'accesso alle API.
- Hai creato un'integrazione OAuth2 API nel tuo account Okta. Questa integrazione fornisce le credenziali del client che AWS Glue utilizza per accedere ai dati in modo sicuro quando effettua chiamate autenticate al vostro account. Per ulteriori informazioni, consulta Passaggi per creare un'app client e credenziali OAuth2 2.0: Okta New Account e Passaggi per la creazione di un'app per sviluppatori
- Hai un account Okta con un. OktaApiToken Consultare la documentazione di Okta.

Se soddisfi questi requisiti, sei pronto per connetterti AWS Glue al tuo account Okta. Per le connessioni tipiche, non è necessario fare nient'altro in Okta.

## Configurazione delle connessioni Okta

Okta supporta due tipi di meccanismi di autenticazione:

- OAuth auth: Okta supporta il tipo di `AUTHORIZATION_CODE` concessione.
  - Questo tipo di concessione è considerato «a tre gambe» in OAuth quanto si basa sul reindirizzamento degli utenti al server di autorizzazione di terze parti per autenticare l'utente. Viene utilizzato durante la creazione di connessioni tramite la console. AWS Glue La AWS Glue Console reindirizzerà l'utente a Okta, dove l'utente deve effettuare il login e consentire AWS Glue le autorizzazioni richieste per accedere alla propria istanza Okta.
  - Gli utenti possono scegliere di creare la propria app connessa in Okta e fornire il proprio ID client e il segreto del client durante la creazione di connessioni tramite la Console. AWS Glue In questo scenario, verranno comunque reindirizzati a Okta per accedere e autorizzare l'accesso AWS Glue alle proprie risorse.
  - Questo tipo di concessione genera un token di aggiornamento e un token di accesso. Il token di accesso dura poco tempo e può essere aggiornato automaticamente senza l'interazione dell'utente utilizzando il token di aggiornamento.

  - Per ulteriori informazioni, consulta la documentazione pubblica di Okta sulla creazione di un'app connessa per il flusso del codice di autorizzazione. OAuth
- Autenticazione personalizzata:
  - Per la documentazione pubblica di Okta sulla generazione delle chiavi API richieste per l'autorizzazione personalizzata, consultare la documentazione di Okta.

Per configurare una connessione Okta:

1. In AWS Secrets Manager, crea un segreto con i seguenti dettagli. È necessario creare un segreto per ogni connessione in AWS Glue.
   a. Per l' OAuth autenticazione:
      - Per le app connesse gestite dal cliente: il segreto deve contenere l'app connessa Consumer Secret con `USER_MANAGED_CLIENT_APPLICATION_CLIENT_SECRET` come chiave.
   b. Per l'autenticazione personalizzata:
      - Per le app connesse gestite dal cliente: il segreto deve contenere l'app connessa Consumer Secret con `OktaApiToken` come chiave.
2. In AWS Glue Studio, crea una connessione in Connessioni dati seguendo i passaggi seguenti:
   a. In Connessioni, scegliere Crea connessione.
   b. Quando si seleziona una Origine dati, selezionare Okta.
   c. Fornire il proprio sottodominio Okta.
   d. Selezionare l'URL del dominio Okta dell'account Okta.
   e. Seleziona il ruolo IAM che AWS Glue può assumere e dispone delle autorizzazioni per le seguenti azioni:

      JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:DescribeSecret",
        "secretsmanager:GetSecretValue",
        "secretsmanager:PutSecretValue",
        "ec2:CreateNetworkInterface",
```

```
        "ec2:DescribeNetworkInterfaces",
        "ec2:DeleteNetworkInterface"
      ],
      "Resource": "*"
    }
  ]
}
```

f. Selezionare il tipo di autenticazione per connetterti a un'origine dati.

g. Per il tipo di OAuth2 autenticazione, fornisci l'applicazione User Managed Client ClientId dell'app Okta.

h. Seleziona quello `secretName` che desideri utilizzare per questa connessione per AWS Glue inserire i token.

i. Selezionare le opzioni di rete se si desidera utilizzare la propria rete.

3. Concedi il permesso di lettura `secretName` al ruolo IAM associato al tuo AWS Glue lavoro.
4. Nella configurazione del tuo AWS Glue lavoro, fornisci `connectionName` una connessione di rete aggiuntiva.

## Lettura dalle entità di Okta

Prerequisiti

- Un oggetto di Okta da cui si desidera leggere. Consultare la tabella delle entità supportate qui di seguito per verificare quelle disponibili.

Entità supportate

| Entità | Può essere Filtrato | Supporta Limite | Supporta Ordina per | Supporta Seleziona * | Supporta Partizionamento |
|---|---|---|---|---|---|
| Applicazioni | Sì | Sì | No | Sì | No |
| dispositivi | Sì | Sì | No | Sì | Sì |
| Gruppi | Sì | Sì | Sì | Sì | Sì |
| Utenti | Sì | Sì | Sì | Sì | Sì |

| Entità | Può essere Filtrato | Supporta Limite | Supporta Ordina per | Supporta Seleziona * | Supporta Partizionamento |
|---|---|---|---|---|---|
| Tipi di utente | No | No | No | Sì | No |

Esempio

```
okta_read = glueContext.create_dynamic_frame.from_options(
    connection_type="Okta",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "applications",
        "API_VERSION": "v1"
    }
```

Dettagli dell'entità e del campo di Okta

Elenco delle entità:

- Applicazione: https://developer.okta.com/docs/api/openapi/okta-management/management/tag/Application/
- Dispositivo: https://developer.okta.com/docs/api/openapi/okta-management/management/tag/Device/
- Gruppo: https://developer.okta.com/docs/api/openapi/okta-management/management/tag/Group/
- Utente: https://developer.okta.com/docs/api/openapi/okta-management/management/tag/User/
- Tipo di utente: https://developer.okta.com/docs/api/openapi/okta-management/management/tag/UserType/

Query di partizionamento

Se si desidera utilizzare la simultaneità in Spark, possono essere fornite opzioni Spark `PARTITION_FIELD`, `LOWER_BOUND`, `UPPER_BOUND`, `NUM_PARTITIONS` aggiuntive. Con questi parametri, la query originale viene suddivisa in `NUM_PARTITIONS` query secondarie che possono essere eseguite contemporaneamente dalle attività Spark.

- `PARTITION_FIELD`: il nome del campo da utilizzare per partizionare la query.

- LOWER_BOUND: un valore limite inferiore inclusivo del campo di partizione scelto.

  Per la data, accettiamo il formato di data di Spark utilizzato nelle query di Spark SQL. Esempio di valori validi: "2024-02-06".
- UPPER_BOUND: un valore limite superiore esclusivo del campo di partizione scelto.
- NUM_PARTITIONS: numero di partizioni.

Esempio

```
okta_read = glueContext.create_dynamic_frame.from_options(
    connection_type="okta",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "lastUpdated",
        "API_VERSION": "v1",
        "PARTITION_FIELD": "lastMembershipUpdated"
        "LOWER_BOUND": "2022-08-10T10:28:46.000Z"
        "UPPER_BOUND": "2024-08-10T10:28:46.000Z"
        "NUM_PARTITIONS": "10"
    }
```

## Indicazioni di riferimento alle opzioni di connessione a Okta

Di seguito sono riportate le opzioni di connessione per Okta:

- ENTITY_NAME(Stringa): (obbligatorio) utilizzato per lettura/scrittura. Il nome dell'oggetto in Okta.
- API_VERSION(Stringa): (obbligatorio) utilizzato per lettura/scrittura. Versione dell'API Rest di Okta da utilizzare. Ad esempio: v1.
- SELECTED_FIELDS(List<String>): predefinito, vuoto (SELECT *). Utilizzato per la lettura. Colonne da selezionare per l'oggetto.
- FILTER_PREDICATE(Stringa): predefinito: vuoto. Utilizzato per la lettura. Dev'essere in formato Spark SQL.
- QUERY(Stringa): predefinito: vuoto. Utilizzato per la lettura. Query Spark SQL completa.
- PARTITION_FIELD(Stringa): utilizzato per la lettura. Campo da utilizzare per partizionare la query.
- LOWER_BOUND(Stringa): utilizzato per la lettura. Un valore limite inferiore inclusivo del campo di partizione scelto.

- `UPPER_BOUND`(Stringa): utilizzato per la lettura. Un valore limite superiore esclusivo del campo di partizione scelto.
- `NUM_PARTITIONS`(Numero intero): predefinito: 1. Utilizzato per la lettura. Numero di partizioni da leggere.

## Procedura di creazione di un nuovo account e di un'app per sviluppatori di Okta

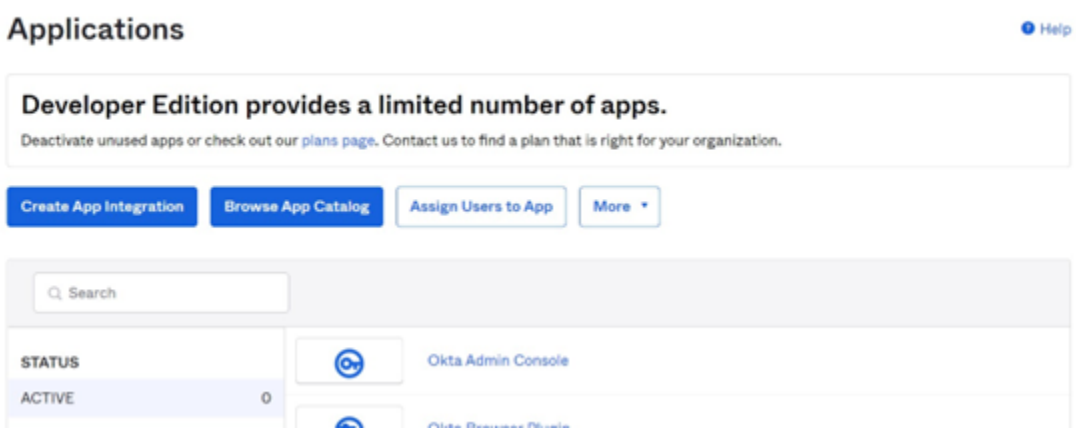
Creare un account sviluppatore su Okta per accedere all'API Okta. Un account sviluppatore Okta gratuito consente di accedere alla maggior parte delle principali funzionalità di sviluppo necessarie per accedere all'API Okta.

Per creare un account sviluppatore su Okta

1. Accedi a https://developer.okta.com/signup/.
2. Inserire le informazioni sull'account, l'e-mail, il nome, il cognome e il Paese/la regione). Scegliere Non sono un robot e poi Registrati.
3. Un'e-mail di verifica viene inviata all'indirizzo di posta registrato. Si riceverà un link tramite e-mail per attivare l'account sviluppatore Okta. Selezionare Attiva.
4. Si verrà reindirizzati alla pagina di reimpostazione della password. Inserire la nuova password due volte e scegliere Reimposta password.
5. Si verrà reindirizzati alla dashboard dell'account sviluppatore Okta.

Per creare un'app client e credenziali OAuth 2.0

1. Nella dashboard per sviluppatori, scegliere Crea integrazione app.



2. Apparirà la finestra Crea una nuova integrazione app che presenta vari metodi di accesso. Selezionare OIDC –OpenID Connect.
3. Scorrere in basso fino alla sezione Tipo di applicazione. Selezionare come Applicazione Web e scegliere Successivo.
4. Nella schermata “Nuova integrazione app web”, inserire le seguenti informazioni:
   - Il nome dell'integrazione app: inserire il nome dell'app.
   - Tipo di concessione: scegliere Codice di autorizzazione e Aggiorna token dall'elenco.
   - Reindirizzamento dell'accesso URIs : scegli Aggiungi URI e aggiungi. `https://{regioncode}.console.aws.amazon.com/appflow/oauth` Ad esempio, se si sta utilizzando `us-west-2 (Oregon)`, è possibile aggiungere `https://us-east-1.console.aws.amazon.com/appflow/oauth`.
   - Accesso controllato: assegnare l'app ai gruppi di utenti in base alle proprie esigenze e scegliere Salva.
5. Vengono generati l'ID cliente e il segreto client.

### Limitazioni

Di seguito sono riportate le limitazioni per il connettore di Okta:

- Per l'entità “Applicazioni” può essere applicato un solo filtro. Se viene applicato più di un filtro, viene restituito 400 Bad Request con un riepilogo dell'errore: “Criteri di ricerca non validi”.
- “Ordina per” può essere supportato solo con le query di ricerca. Ad esempio, `http://dev-15940405.okta.com/api/v1/groups?search=type e.q. "OKTA_GROUP"&sortBy=lastUpdated&sortOrder=asc`

## Connessione a PayPal

PayPal è un sistema di pagamento che facilita i trasferimenti di denaro online tra le parti, come i trasferimenti tra clienti e fornitori online. Se sei un PayPal utente, il tuo account contiene dati sulle tue transazioni, come i pagatori, le date e lo stato. Puoi utilizzarli AWS Glue per trasferire dati PayPal da determinati AWS servizi o altre applicazioni supportate.

Argomenti

- AWS Glue supporto per PayPal

- Policy contenenti le operazioni API per creare e utilizzare le connessioni
- Configurazione PayPal
- Configurazione delle connessioni PayPal
- Lettura da PayPal entità
- PayPal opzioni di connessione
- Limitazioni e note per il PayPal connettore


## AWS Glue supporto per PayPal

AWS Glue supporta PayPal quanto segue:

Supportato come origine?

Sì. È possibile utilizzare i job AWS Glue ETL da PayPal cui interrogare i dati.

Supportato come destinazione?

No.

Versioni PayPal API supportate

Sono supportate le seguenti versioni PayPal API:

- v1

## Policy contenenti le operazioni API per creare e utilizzare le connessioni

La seguente policy di esempio descrive le autorizzazioni AWS IAM richieste per la creazione e l'utilizzo delle connessioni. Se si sta creando un nuovo ruolo, creare una policy che contenga quanto segue:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
```

```
        "glue:ListConnectionTypes",
        "glue:DescribeConnectionType",
        "glue:RefreshOAuth2Tokens",
        "glue:ListEntities",
        "glue:DescribeEntity"
      ],
      "Resource": "*"
    }
  ]
}
```

In alternativa, se non desideri utilizzare il metodo precedente, utilizza le seguenti policy IAM gestite:

- AWSGlueServiceRole— Garantisce l'accesso alle risorse che AWS Glue i vari processi richiedono per l'esecuzione per conto dell'utente. Queste risorse includono AWS Glue Amazon S3, IAM, CloudWatch Logs e Amazon. EC2 Se segui la convenzione di denominazione delle risorse specificata in questa politica, AWS Glue i processi dispongono delle autorizzazioni richieste. Questa policy è in genere collegata ai ruoli specificati quando si definiscono crawler, processi ed endpoint di sviluppo.
- AWSGlueConsoleFullAccess— Concede l'accesso completo alle AWS Glue risorse quando un'identità a cui è allegata la policy utilizza la AWS console di gestione. Se si segue la convenzione per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questa policy è in genere associata agli utenti della AWS Glue console.

## Configurazione PayPal

Prima di poter utilizzare AWS Glue per trasferire dati da PayPal, è necessario soddisfare i seguenti requisiti:

Requisiti minimi

I requisiti minimi sono i seguenti:

- Hai un PayPal account con le credenziali del cliente.
- Il tuo PayPal account ha accesso all'API con una licenza valida.

Se soddisfi questi requisiti, sei pronto per connetterti AWS Glue al tuo PayPal account. Per le connessioni tipiche, non devi fare nient'altro PayPal.

## Configurazione delle connessioni PayPal

PayPal supporta il tipo di concessione CLIENT CREDENTIALS per. OAuth2

- Questo tipo di concessione è considerato OAuth 2.0 a 2 gambe in quanto viene utilizzato dai client per ottenere un token di accesso al di fuori del contesto di un utente. AWS Glue è in grado di utilizzare l'ID client e il client secret per autenticare i PayPal APIs servizi forniti dai servizi personalizzati definiti dall'utente.
- Ogni servizio personalizzato è di proprietà di un utente che utilizza solo API e dispone di una serie di ruoli e autorizzazioni che autorizzano il servizio a eseguire azioni specifiche. Un token di accesso è associato a un singolo servizio personalizzato.
- Questo tipo di concessione determina un token di accesso di breve durata e che può essere rinnovato chiamando nuovamente l'endpoint `/v2/oauth2/token`.
- Per la PayPal documentazione pubblica per la OAuth versione 2.0 con le credenziali del client, vedere Autenticazione.

Per configurare una PayPal connessione:

1. In AWS Secrets Manager, crea un segreto con i seguenti dettagli:
   a. Per le app connesse gestite dal cliente: il segreto deve contenere l'app connessa Consumer Secret con `USER_MANAGED_CLIENT_APPLICATION_CLIENT_SECRET` come chiave.
   b. Nota: devi creare un segreto per le tue connessioni in AWS Glue.

1. In AWS Glue Glue Studio, crea una connessione in Connessioni dati seguendo i passaggi seguenti:
   a. Quando si seleziona un tipo di connessione, selezionare PayPal.
   b. Fornisci `INSTANCE_URL` l' PayPal istanza a cui desideri connetterti.
   c. Seleziona il ruolo AWS IAM che AWS Glue può assumere e dispone delle autorizzazioni per le seguenti azioni:

      JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
```

```
      "Action": [
        "secretsmanager:DescribeSecret",
        "secretsmanager:GetSecretValue",
        "secretsmanager:PutSecretValue",
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DeleteNetworkInterface"
      ],
      "Resource": "*"
    }
  ]
}
```

d. Seleziona quello `secretName` che desideri utilizzare per questa connessione per AWS Glue inserire i token.

e. Selezionare le opzioni di rete se si desidera utilizzare la propria rete.

2. Concedi il permesso di lettura `secretName` al ruolo IAM associato al tuo AWS Glue lavoro.

Ottenere le credenziali OAuth 2.0

Per chiamare l'API Rest, occorre scambiare l'ID client e il segreto del client con un token di accesso. Per ulteriori informazioni, consulta Guida introduttiva a PayPal REST APIs.

## Lettura da PayPal entità

Prerequisito

Un PayPal oggetto da cui desideri leggere. È necessario il nome dell'oggetto, `transaction`.

Entità supportate come origine:

| Entità | Può essere filtrato | Supporta limite | Supporta Ordina per | Supporta Seleziona * | Supporta il partizionamento |
|---|---|---|---|---|---|
| transaction | Sì | Sì | No | Sì | Sì |

Esempio:

```
paypal_read = glueContext.create_dynamic_frame.from_options(
```

```
    connection_type="paypal",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "transaction",
        "API_VERSION": "v1",
        "INSTANCE_URL": "https://api-m.paypal.com"
    }
```

PayPal dettagli dell'entità e del campo:

Entità con metadati statici:

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| transaction | transaction_initiation_date | DateTime | Tra |
| | last_refreshed_datetime | Stringa | N/A |
| | payment_instrument_type | Stringa | = |
| | balance_affecting_records_only | Stringa | = |
| | store_id | Stringa | = |
| | terminal_id | Stringa | = |
| | transaction_currency | Stringa | = |
| | transaction_id | Stringa | N/A |
| | transaction_status | Stringa | N/A |
| | transaction_type | Stringa | N/A |
| | transaction_info | Struct | N/D |
| | payer_info | Struct | N/D |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | shipping_info | Struct | N/D |
| | cart_info | Struct | N/D |
| | store_info | Struct | N/D |
| | auction_info | Struct | N/D |
| | incentive_info | Struct | N/D |

Query di partizionamento

Puoi fornire le opzioni Spark aggiuntive `PARTITION_FIELD`, `LOWER_BOUND`, `UPPER_BOUND` e `NUM_PARTITIONS` se desideri utilizzare la concorrenza in Spark. Con questi parametri, la query originale viene suddivisa in `NUM_PARTITIONS` query secondarie che possono essere eseguite contemporaneamente dalle attività Spark.

- `PARTITION_FIELD`: il nome del campo da utilizzare per partizionare la query.
- `LOWER_BOUND`: un valore limite inferiore inclusivo del campo di partizione scelto.

  Per il campo Datetime, accettiamo il valore in formato ISO.

  Esempi di valori validi:

  ```
  "2024-07-01T00:00:00.000Z"
  ```
- `UPPER_BOUND`: un valore limite superiore esclusivo del campo di partizione scelto.
- `NUM_PARTITIONS`: il numero di partizioni.

Il seguente campo è supportato per il partizionamento per entità:

| Nome dell'entità | Campi di partizionamento | Tipo di dati |
|---|---|---|
| transaction | transaction_initiation_date | DateTime |

Esempio:

```
paypal_read = glueContext.create_dynamic_frame.from_options(
    connection_type="paypal",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "transaction",
        "API_VERSION": "v1",
        "PARTITION_FIELD": "transaction_initiation_date"
        "LOWER_BOUND": "2024-07-01T00:00:00.000Z"
        "UPPER_BOUND": "2024-07-02T00:00:00.000Z"
        "NUM_PARTITIONS": "10"
    }
```

## PayPal opzioni di connessione

Di seguito sono elencate le opzioni di connessione per PayPal:

- `ENTITY_NAME`(String): (obbligatorio) utilizzato per la lettura. Il nome dell'oggetto in PayPal.
- `API_VERSION`(String) - (Obbligatorio) Usato per la lettura. PayPal Versione dell'API Rest che desideri utilizzare.
- `SELECTED_FIELDS`(List<String>): predefinito: vuoto (SELECT *). Utilizzato per la lettura. Colonne da selezionare per l'oggetto.
- `FILTER_PREDICATE`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Dev'essere in formato Spark SQL.
- `QUERY`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Query Spark SQL completa.
- `PARTITION_FIELD`(Stringa): utilizzato per la lettura. Campo da utilizzare per partizionare la query.
- `LOWER_BOUND`(Stringa): utilizzato per la lettura. Un valore limite inferiore inclusivo del campo di partizione scelto.
- `UPPER_BOUND`(Stringa): utilizzato per la lettura. Un valore limite superiore esclusivo del campo di partizione scelto.
- `NUM_PARTITIONS`(Numero intero): predefinito: 1. Utilizzato per la lettura. Numero di partizioni da leggere.

## Limitazioni e note per il PayPal connettore

Di seguito sono riportate le limitazioni o le note relative al PayPal connettore:

- La documentazione PayPal sulle transazioni indica che sono necessarie un massimo di tre ore prima che le transazioni eseguite vengano visualizzate nella chiamata delle transazioni dell'elenco. Tuttavia, è stato osservato che richiede più tempo a seconda del `last_refreshed_datetime`. `last_refreshed_datetime`Ecco il periodo di tempo fino al quale i dati saranno disponibili da. APIs
- Se `last_refreshed_datetime` è inferiore al `end_date` richiesto, `end_date` diventa uguale a `last_refreshed_datetime` in quanto abbiamo solo dati fino a quel momento.
- Il campo `transaction_initiation_date` è un filtro obbligatorio da fornire per l'entità `transaction` e l'intervallo di date massimo supportato per questo campo è 31 giorni.
- Quando si chiama la richiesta API dell'entità `transaction` con filtri (parametri di query) diversi dal campo `transaction_initiation_date`, si prevede che il valore del campo `ending_balance` non verrà recuperato nella risposta.

## Connessione a Pendo

Pendo offre un ricco archivio di dati relativi alle interazioni degli utenti. I clienti trasferiranno questi dati AWS in modo da poterli unire ad altri dati di prodotto, eseguire analisi e dashboard aggiuntive e impostare avvisi, se lo desiderano.

Argomenti

- AWS Glue supporto per Pendo
- Policy contenenti le operazioni API per creare e utilizzare le connessioni
- Configurazione di Pendo
- Configurazione delle connessioni Pendo
- Lettura dalle entità di Pendo
- Opzioni di connessione di Pendo
- Limitazioni

### AWS Glue supporto per Pendo

AWS Glue supporta Pendo come segue:

Supportato come origine?

Sì. È possibile utilizzare i job AWS Glue ETL per interrogare i dati da Pendo.

Supportato come destinazione?

No.

Versioni API Pendo supportate

v1

## Policy contenenti le operazioni API per creare e utilizzare le connessioni

La seguente politica di esempio descrive le AWS autorizzazioni richieste per la creazione e l'utilizzo delle connessioni. Se si sta creando un nuovo ruolo, creare una policy che contenga quanto segue:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:ListConnectionTypes",
        "glue:DescribeConnectionType",
        "glue:RefreshOAuth2Tokens",
        "glue:ListEntities",
        "glue:DescribeEntity"
      ],
      "Resource": "*"
    }
  ]
}
```

In alternativa, se non desideri utilizzare il metodo precedente, utilizza le seguenti policy IAM:

- AWSGlueServiceRole — Garantisce l'accesso alle risorse che diversi AWS Glue processi richiedono per l'esecuzione per conto dell'utente. Queste risorse includono Amazon S3 AWS Glue, IAM, CloudWatch Logs e Amazon EC2. Se segui la convenzione di denominazione delle risorse specificata in questa politica, AWS Glue i processi dispongono delle autorizzazioni necessarie. Questa policy è in genere collegata ai ruoli specificati quando si definiscono crawler, processi ed endpoint di sviluppo.

- AWSGlueConsoleFullAccess — Concede l'accesso completo alle AWS Glue risorse quando un'identità a cui è allegata la politica utilizza la AWS console di gestione. Se si segue la convenzione per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questa policy è in genere associata agli utenti della AWS Glue console.

## Configurazione di Pendo

Prima di poter AWS Glue utilizzare il trasferimento da Pendo, devi soddisfare i seguenti requisiti:

Requisiti minimi

- Avere un account Pendo con un `apiKey` con `write access` abilitato.
- L'account Pendo dispone dell'accesso API con una licenza valida.

Se soddisfi questi requisiti, sei pronto per connetterti AWS Glue al tuo account Pendo. Per le connessioni tipiche, non è necessario fare nient'altro in Pendo.

## Configurazione delle connessioni Pendo

Pendo supporta l'autenticazione personalizzata.

Per la documentazione pubblica di Pendo sulla generazione delle chiavi API richieste per l'autorizzazione personalizzata, consultare Autenticazione: documentazione di Pendo REST API

Per configurare una connessione Pendo:

1. In AWS Secrets Manager, crea un segreto con i seguenti dettagli:
   - Per le app connesse gestite dal cliente: il segreto deve contenere l'app connessa Consumer Secret con `apiKey` come chiave.

   > **Note**
   >
   > È necessario creare un segreto per ogni connessione in AWS Glue.

2. In AWS Glue Studio, crea una connessione in Connessioni dati seguendo i passaggi seguenti:
   a. Quando si seleziona una Origine dati, selezionare Pendo.
   b. Fornire `instanceUrl` dell'istanza Pendo a cui connettersi.

c. Seleziona il ruolo IAM per il quale AWS Glue può assumere e dispone delle autorizzazioni per le seguenti azioni:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:DescribeSecret",
        "secretsmanager:GetSecretValue",
        "secretsmanager:PutSecretValue",
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DeleteNetworkInterface"
      ],
      "Resource": "*"
    }
  ]
}
```

d. Seleziona quello `secretName` che desideri utilizzare per questa connessione per AWS Glue inserire i token.

e. Selezionare le opzioni di rete se si desidera utilizzare la propria rete.

3. Concedi il permesso di lettura `secretName` al ruolo IAM associato al tuo AWS Glue lavoro.
4. Nella configurazione del tuo AWS Glue lavoro, fornisci `connectionName` una connessione di rete aggiuntiva.

## Lettura dalle entità di Pendo

Prerequisiti

Un oggetto di Pendo da cui desideri leggere. Consultare la tabella delle entità supportate qui di seguito per verificare le entità disponibili.

Entità supportate

- Caratteristica

- Guida
- Pagina
- Report
- Report di dati
- Visitatore
- Account
- Evento
- Evento principale
- Evento guida
- Evento della pagina
- Evento di sondaggio
- Tracciare l'evento

| Entità | Può essere Filtrato | Supporta Limite | Supporta Ordina per | Supporta Seleziona * | Supporta il partizionamento |
|---|---|---|---|---|---|
| Funzionalità | No | No | No | Sì | No |
| Guida | No | No | No | Sì | No |
| Pagina | No | No | No | Sì | No |
| Report | No | No | No | Sì | No |
| Report di dati | No | No | No | Sì | No |
| Visitatore (API di aggregazione) | Sì | No | Sì | Sì | No |
| Account (API di aggregazione) | Sì | No | Sì | Sì | No |

| Entità | Può essere Filtrato | Supporta Limite | Supporta Ordina per | Supporta Seleziona * | Supporta il partizionamento |
|---|---|---|---|---|---|
| Evento (API di aggregazione) | Sì | No | Sì | Sì | No |
| Evento principale (API di aggregazione) | Sì | No | Sì | Sì | Sì |
| Evento guida (API di aggregazione) | Sì | No | Sì | Sì | Sì |
| Account (API di aggregazione) | Sì | No | Sì | Sì | Sì |
| Evento della pagina (API di aggregazione) | Sì | No | Sì | Sì | Sì |
| Evento di sondaggio (API di aggregazione) | Sì | No | Sì | Sì | Sì |
| Tracciare l'evento (API di aggregazione) | Sì | No | Sì | Sì | Sì |

Esempio

```
Pendo_read = glueContext.create_dynamic_frame.from_options(
    connection_type="glue.spark.Pendo",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "feature",
        "API_VERSION": "v1",
        "INSTANCE_URL": "instanceUrl"
    }
```

Query di partizionamento

Puoi fornire le opzioni Spark aggiuntive PARTITION_FIELD, LOWER_BOUND, UPPER_BOUND e NUM_PARTITIONS se desideri utilizzare la concorrenza in Spark. Con questi parametri, la query originale viene suddivisa in NUM_PARTITIONS query secondarie che possono essere eseguite contemporaneamente dalle attività Spark.

- PARTITION_FIELD: il nome del campo da utilizzare per partizionare la query.
- LOWER_BOUND: un valore limite inferiore inclusivo del campo di partizione scelto.

  Per il DateTime campo, accettiamo il valore in formato ISO.

  Esempio di valore valido:

  ```
  "2024-07-01T00:00:00.000Z"
  ```
- UPPER_BOUND: un valore limite superiore esclusivo del campo di partizione scelto.
- NUM_PARTITIONS: il numero di partizioni.

Nella seguente tabella sono descritti i dettagli del campo di supporto del partizionamento delle entità:

| Nome dell'entità |
| --- |
| Event |
| Evento principale |
| Evento guida |

| Nome dell'entità |
| --- |
| Evento della pagina |
| Evento di sondaggio |
| Tracciare l'evento |

Esempio:

```
pendo_read = glueContext.create_dynamic_frame.from_options(
    connection_type="glue.spark.pendo",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "event",
        "API_VERSION": "v1",
        "INSTANCE_URL": "instanceUrl"
        "NUM_PARTITIONS": "10",
        "PARTITION_FIELD": "appId"
        "LOWER_BOUND": "4656"
        "UPPER_BOUND": "7788"
    }
```

## Opzioni di connessione di Pendo

Di seguito sono riportate le opzioni di connessione per Pendo:

- `ENTITY_NAME` (stringa): (obbligatorio) utilizzato per lettura/scrittura. Il nome dell'oggetto in Pendo.
- `INSTANCE_URL`(String): (obbligatorio) Un URL di istanza Pendo valido con i seguenti valori consentiti:
  - Impostazione predefinita
  - Europa
  - US1
- `API_VERSION`(String): (obbligatorio) utilizzato per la lettura. Versione dell'API Engage Rest di Pendo che si desidera utilizzare. Ad esempio: 3.0.
- `SELECTED_FIELDS`(List<String>): predefinito: vuoto (SELECT *). Utilizzato per la lettura. Colonne da selezionare per l'oggetto.

- `FILTER_PREDICATE`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Dev'essere in formato Spark SQL.
- `QUERY`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Query Spark SQL completa.
- `PARTITION_FIELD`(Stringa): utilizzato per la lettura. Campo da utilizzare per partizionare la query.
- `LOWER_BOUND`(Stringa): utilizzato per la lettura. Un valore limite inferiore inclusivo del campo di partizione scelto.
- `UPPER_BOUND`(Stringa): utilizzato per la lettura. Un valore limite superiore esclusivo del campo di partizione scelto.
- `NUM_PARTITIONS`(Numero intero): predefinito: 1. Utilizzato per la lettura. Numero di partizioni da leggere.

### Limitazioni

Di seguito sono riportate le limitazioni per il connettore di Pendo:

- La paginazione non è supportata in Pendo.
- La filtrazione è supportata solo dagli oggetti dell'API Aggregate (`Account`, `Event`, `Feature Event`, `Guide Events`, `Page Event`, `Poll Event`, `Track Event` e `Visitor`)
- DateTimeRange è un parametro di filtro obbligatorio per gli oggetti API aggregati (`Event`,`Feature Event`,, `Guide EventsPage Event`, `Poll Event`,`Track Event`)
- Il periodo dayRange verrà arrotondato per difetto all'inizio del periodo nel fuso orario. Ad esempio, se il filtro fornito è `2023-01-12T07:55:27.065Z`, questo periodo di tempo verrà arrotondato all'inizio del periodo, ovvero `2023-01-12T00:00:00Z`.

## Connessione a Pipedrive

Pipedrive è un CRM per la gestione della pipeline di vendita progettato per aiutare le piccole imprese a gestire i lead, monitorare le attività di vendita e concludere più affari. Pipedrive permette ai team di vendita delle piccole imprese di semplificare i processi e consolidare i dati di vendita in un unico strumento CRM unificato. Se sei un utente Pipedrive, puoi connetterti AWS Glue al tuo account Pipedrive. Quindi, puoi usare Pipedrive come origine dati nei tuoi processi ETL. Esegui questi lavori per trasferire dati tra Pipedrive e AWS servizi o altre applicazioni supportate.

Argomenti

- AWS Glue supporto per Pipedrive

- Policy contenenti le operazioni API per creare e utilizzare le connessioni
- Configurazione di Pipedrive
- Configurazione delle connessioni Pipedrive
- Lettura da entità Pipedrive
- Indicazioni di riferimento alle opzioni di connessione a Pipedrive
- Limitazioni


## AWS Glue supporto per Pipedrive

AWS Glue supporta Pipedrive come segue:

Supportato come origine?

Sì. Puoi utilizzare i job AWS Glue ETL per interrogare i dati da Pipedrive.

Supportato come destinazione?

No.

Versioni API Pipedrive supportate

v1.

## Policy contenenti le operazioni API per creare e utilizzare le connessioni

La seguente politica di esempio descrive le AWS autorizzazioni richieste per la creazione e l'utilizzo delle connessioni. Se si sta creando un nuovo ruolo, creare una policy che contenga quanto segue:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:ListConnectionTypes",
        "glue:DescribeConnectionType",
        "glue:RefreshOAuth2Tokens",
        "glue:ListEntities",
```

```
        "glue:DescribeEntity"
      ],
      "Resource": "*"
    }
  ]
}
```

È anche possibile utilizzare le seguenti policy IAM per autorizzare l'accesso:

- AWSGlueServiceRole— Garantisce l'accesso alle risorse che diversi AWS Glue processi richiedono per l'esecuzione per conto dell'utente. Queste risorse includono Amazon S3 AWS Glue, IAM, CloudWatch Logs e Amazon EC2. Se segui la convenzione di denominazione delle risorse specificata in questa politica, AWS Glue i processi dispongono delle autorizzazioni necessarie. Questa policy è in genere collegata ai ruoli specificati quando si definiscono crawler, processi ed endpoint di sviluppo.
- AWSGlueConsoleFullAccess— Concede l'accesso completo alle AWS Glue risorse quando un'identità a cui è allegata la policy utilizza la AWS console di gestione. Se si segue la convenzione per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questa policy è in genere associata agli utenti della AWS Glue console.

## Configurazione di Pipedrive

Prima di poter utilizzare AWS Glue per trasferire dati da Pipedrive, devi soddisfare questi requisiti:

Requisiti minimi

- Avere un account Pipedrive.
- L'account Pipedrive è abilitato per l'accesso alle API.

Se soddisfi questi requisiti, sei pronto per connetterti AWS Glue al tuo account Pipedrive. Per le connessioni tipiche, non è necessario fare nient'altro in Pipedrive.

## Configurazione delle connessioni Pipedrive

Pipedrive supporta il tipo di concessione AUTHORIZATION_CODE per. OAuth2

- Questo tipo di concessione è considerato «a tre gambe» in OAuth quanto si basa sul reindirizzamento degli utenti al server di autorizzazione di terze parti per autenticare l'utente. Viene

utilizzato durante la creazione di connessioni tramite la console. AWS Glue Per impostazione predefinita, l'utente che crea una connessione può fare affidamento su un'app connessa AWS Glue di proprietà in cui non è necessario fornire alcuna informazione OAuth correlata ad eccezione dell'url dell'istanza di Pipedrive. La AWS Glue Console reindirizzerà l'utente a Pipedrive dove l'utente deve effettuare il login e consentire AWS Glue le autorizzazioni richieste per accedere alla propria istanza Pipedrive.

- Gli utenti devono scegliere di creare la propria app connessa in Pipedrive e fornire il proprio ID client e il segreto del client quando creano connessioni tramite la Console. AWS Glue In questo scenario, verranno comunque reindirizzati a Pipedrive per accedere e AWS Glue autorizzare l'accesso alle proprie risorse.
- Questo tipo di concessione genera un token di aggiornamento e uno di accesso. Il token di accesso è attivo per un'ora e può essere aggiornato automaticamente senza l'interazione dell'utente utilizzando il token di aggiornamento.
- Per ulteriori informazioni, consulta la documentazione sulla creazione di un'app connessa per il flusso AUTHORIZATION_CODE. OAuth

Per configurare una connessione a Pipedrive:

1. In AWS Secrets Manager, crea un segreto con i seguenti dettagli. È necessario creare un segreto per ogni connessione in AWS Glue.
   a. Per l'app connessa gestita dal cliente, il segreto deve contenere l'app connessa Consumer Secret con USER_MANAGED_CLIENT_APPLICATION_CLIENT_SECRET come chiave.
2. In AWS Glue Studio, crea una connessione in Connessioni dati seguendo i passaggi seguenti:
   a. In Connessioni dati, scegliere Crea connessione.
   b. Quando si seleziona una Origine dati, selezionare Pipedrive.
   c. Fornire instanceURL di Pipedrive.
   d. Seleziona il ruolo IAM che AWS Glue può assumere e dispone delle autorizzazioni per le seguenti azioni:

      JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
```

```
      "Action": [
        "secretsmanager:DescribeSecret",
        "secretsmanager:GetSecretValue",
        "secretsmanager:PutSecretValue",
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DeleteNetworkInterface"
      ],
      "Resource": "*"
    }
  ]
}
```

e. Fornisci l'applicazione client gestita dagli utenti ClientId del Pipedrive a cui desideri connetterti.

f. Seleziona quello `secretName` che desideri utilizzare per questa connessione per AWS Glue inserire i token.

g. Selezionare le opzioni di rete se si desidera utilizzare la propria rete.

3. Concedi il permesso di lettura `secretName` al ruolo IAM associato al tuo AWS Glue lavoro. Scegli Next (Successivo).
4. Fornire connectionName e scegliere Successivo.
5. Nella pagina successiva, scegliere Crea connessione. Si dovrà accedere a Pipedrive. Fornire nome utente e password e scegliere Accedi.
6. Una volta effettuato l'accesso, scegliere Continua sull'app. Ora la connessione è pronta per essere utilizzata.
7. Nella configurazione del AWS Glue lavoro, fornisci `connectionName` una connessione di rete aggiuntiva.

## Lettura da entità Pipedrive

### Prerequisiti

- Un oggetto di Pipedrive da cui si desidera leggere. Consultare la tabella delle entità supportate qui di seguito per verificare quelle disponibili.

### Entità supportate

| Entità | Può essere Filtrato | Supporta Limite | Supporta Ordina per | Supporta Seleziona * | Supporta Partizionamento |
|---|---|---|---|---|---|
| Attività | Sì | Sì | No | Sì | Sì |
| Tipo di attività | No | No | No | Sì | No |
| Log delle chiamate | No | No | No | Sì | No |
| Valute | Sì | Sì | No | Sì | No |
| Offerte | Sì | Sì | Sì | Sì | Sì |
| Lead | Sì | Sì | Sì | Sì | No |
| Fonti di lead | No | Sì | No | Sì | No |
| Etichette di lead | No | No | No | No | No |
| Note | Sì | Sì | Sì | Sì | Sì |
| Organizzazione | Sì | Sì | No | Sì | Sì |
| Set di autorizzazioni | Sì | No | No | Sì | No |
| Persone | Sì | Sì | Sì | Sì | Sì |
| Pipelines | No | Sì | No | Sì | No |
| Prodotti | Sì | Sì | No | Sì | Sì |
| Roles | No | Sì | No | Sì | No |
| Stage | Sì | Sì | No | Sì | No |
| Utenti | No | No | No | Sì | No |

Esempio

```
pipedrive_read= glueContext.create_dynamic_frame.from_options(
    connection_type="PIPEDRIVE",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "activites",
        "API_VERSION": "v1"
    }
```

Dettagli dell'entità e del campo di Pipedrive

Elenco delle entità:

- Attività: https://developers.pipedrive.com/docs/api/v1/Activities
- Tipo di attività: https://developers.pipedrive.com/docs/api/v1/ActivityTypes
- Registri delle chiamate: https://developers.pipedrive.com/docs/api/v1/CallLogs
- Valute: https://developers.pipedrive.com/docs/api/v1/Currencies
- Offerte: https://developers.pipedrive.com/docs/api/v1/Deals
- Conduce: https://developers.pipedrive.com/docs/api/v1/Leads
- Fonti di piombo: https://developers.pipedrive.com/docs/api/v1/LeadSources
- Etichette di piombo: https://developers.pipedrive.com/docs/api/v1/LeadLabels
- Note: https://developers.pipedrive.com/docs/api/v1/Notes
- Organizzazioni: https://developers.pipedrive.com/docs/api/v1/Organizations
- Set di autorizzazioni: https://developers.pipedrive.com/docs/api/v1/PermissionSets
- Persone: https://developers.pipedrive.com/docs/api/v1/Persons
- Condutture: https://developers.pipedrive.com/docs/api/v1/Pipelines
- Prodotti: https://developers.pipedrive.com/docs/api/v1/Products
- Ruoli: https://developers.pipedrive.com/docs/api/v1/Roles
- Fasi: https://developers.pipedrive.com/docs/api/v1/Stages
- Utenti: https://developers.pipedrive.com/docs/api/v1/Users

| Entità | Tipo di dati | Operatori supportati |
|---|---|---|
| Attività, Offerte, Note, Organizzazione, Persone e Prodotti. | Data | '=' |
| | Numero intero | '=' |
| | Stringa | '=' |
| | Booleano | '=' |

Query di partizionamento

In Pipedrive, solo un campo (due_date) dell'entità Attività supporta il partizionamento basato sul campo. È un campo Data.

Se si desidera utilizzare la simultaneità in Spark, possono essere fornite opzioni Spark `PARTITION_FIELD`, `LOWER_BOUND`, `UPPER_BOUND`, `NUM_PARTITIONS` aggiuntive. Con questi parametri, la query originale viene suddivisa in `NUM_PARTITIONS` query secondarie che possono essere eseguite contemporaneamente dalle attività Spark.

- `PARTITION_FIELD`: il nome del campo da utilizzare per partizionare la query.
- `LOWER_BOUND`: un valore limite inferiore inclusivo del campo di partizione scelto.

  Per la data, accettiamo il formato di data di Spark utilizzato nelle query di Spark SQL. Esempio di valori validi: `"2024-02-06"`.
- `UPPER_BOUND`: un valore limite superiore esclusivo del campo di partizione scelto.
- `NUM_PARTITIONS`: numero di partizioni.

Esempio

```
pipedrive_read = glueContext.create_dynamic_frame.from_options(
    connection_type="PIPEDRIVE",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "activites",
        "API_VERSION": "v1",
```

```
        "PARTITION_FIELD": "due_date"
        "LOWER_BOUND": "2023-09-07T02:03:00.000Z"
        "UPPER_BOUND": "2024-05-07T02:03:00.000Z"
        "NUM_PARTITIONS": "10"
    }
```

## Indicazioni di riferimento alle opzioni di connessione a Pipedrive

Di seguito sono riportate le opzioni di connessione per Pipedrive:

- `ENTITY_NAME`(String): (obbligatorio) utilizzato per la lettura/scrittura. Il nome dell'oggetto in Pipedrive.
- `API_VERSION`(String): (obbligatorio) utilizzato per la lettura/scrittura. Versione dell'API Rest di Pipedrive da utilizzare. Esempio: v1.
- `INSTANCE_URL`(String): (obbligatorio) URL dell'istanza in cui l'utente desidera eseguire le operazioni. Ad esempio: v1.
- `SELECTED_FIELDS`(List<String>): predefinito, vuoto (SELECT *). Utilizzato per la lettura. Colonne da selezionare per l'oggetto.
- `FILTER_PREDICATE`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Dev'essere in formato Spark SQL.
- `QUERY`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Query Spark SQL completa.
- `PARTITION_FIELD`(Stringa): utilizzato per la lettura. Campo da utilizzare per partizionare la query.
- `LOWER_BOUND`(Stringa): utilizzato per la lettura. Un valore limite inferiore inclusivo del campo di partizione scelto.
- `UPPER_BOUND`(Stringa): utilizzato per la lettura. Un valore limite superiore esclusivo del campo di partizione scelto.
- `NUM_PARTITIONS`(Numero intero): predefinito: 1. Utilizzato per la lettura. Numero di partizioni da leggere.

## Limitazioni

Di seguito sono riportate le limitazioni per il connettore di Pipedrive:

- Pipedrive supporta il partizionamento basato sul campo solo per un'entità (Attività).
- Pipedrive supporta il partizionamento basato su record per le entità Attività, Offerte, Note, Persone, Organizzazioni e Prodotti.

- Nell'entità Offerte, il campo dello stato come filtro restituirà tutti i record se viene utilizzato un valore di filtro con valore non valido.
- Nell'entità Offerte, gli ordini con più campi non sono supportati.
- Per ottenere dati sulle prestazioni, utilizziamo un AWS account locale. Tuttavia, a causa della limitazione dell'aggiornamento locale del token di accesso, il AWS Glue processo di elaborazione di 1 GB di dati non riesce. Di conseguenza, abbiamo ottimizzato il test delle prestazioni con 179 MB di dati e i risultati sopra riportati si basano su questa ottimizzazione. Tuttavia, abbiamo osservato che con un numero crescente di partizioni, l'endpoint SaaS impiega più tempo rispetto a una singola partizione. Abbiamo consultato il team di supporto di Pipedrive in merito a questo comportamento e ci hanno informato che Pipedrive sta limitando silenziosamente le richieste e ritardando la risposta. Pertanto, quando si esegue il AWS Glue processo con set di dati di grandi dimensioni o si chiama più volte lo stesso endpoint API, potrebbe verificarsi un problema di timeout dovuto all'implementazione dell'API Pipedrive. Tuttavia, i tempi di risposta del connettore e dello shim stanno diminuendo come previsto con l'aumento del numero di partizioni.

## Connessione a Productboard

Productboard è il sistema di gestione dei prodotti che aiuta i team di prodotto a lanciare sul mercato i prodotti giusti in modo più rapido. Oltre 3.000 aziende moderne e orientate ai prodotti, come Zendesk UiPath e Microsoft, utilizzano Productboard per capire di cosa hanno realmente bisogno gli utenti, dare priorità a cosa creare in futuro e coinvolgere tutti attorno alla propria roadmap.

Argomenti


- AWS Glue supporto per Productboard
- Policy contenenti le operazioni API per creare e utilizzare le connessioni
- Configurazione di Productboard
- Configurazione delle connessioni di Productboard
- Lettura da entità Productboard
- Opzioni di connessione a Productboard
- Creazione di un account Productboard
- Limitazioni


### AWS Glue supporto per Productboard

AWS Glue supporta Productboard come segue:

Supportato come origine?

Sì. È possibile utilizzare i lavori AWS Glue ETL per interrogare i dati da Productboard.

Supportato come destinazione?

No.

Versioni API Productboard supportate

v1

## Policy contenenti le operazioni API per creare e utilizzare le connessioni

La seguente politica di esempio descrive le AWS autorizzazioni richieste per la creazione e l'utilizzo delle connessioni. Se si sta creando un nuovo ruolo, creare una policy che contenga quanto segue:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:ListConnectionTypes",
        "glue:DescribeConnectionType",
        "glue:RefreshOAuth2Tokens",
        "glue:ListEntities",
        "glue:DescribeEntity"
      ],
      "Resource": "*"
    }
  ]
}
```

In alternativa, se non desideri utilizzare il metodo precedente, utilizza le seguenti policy IAM:

- AWSGlueServiceRole — Garantisce l'accesso alle risorse che diversi AWS Glue processi richiedono per l'esecuzione per conto dell'utente. Queste risorse includono Amazon S3 AWS Glue, IAM, CloudWatch Logs e Amazon EC2. Se segui la convenzione di denominazione delle risorse

specificata in questa politica, AWS Glue i processi dispongono delle autorizzazioni necessarie. Questa policy è in genere collegata ai ruoli specificati quando si definiscono crawler, processi ed endpoint di sviluppo.

- AWSGlueConsoleFullAccess — Concede l'accesso completo alle AWS Glue risorse quando un'identità a cui è allegata la politica utilizza la AWS console di gestione. Se si segue la convenzione per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questa policy è in genere associata agli utenti della AWS Glue console.

## Configurazione di Productboard

Prima di poter AWS Glue utilizzare il trasferimento da Productboard, devi soddisfare i seguenti requisiti:

Requisiti minimi

- Avere un account Productboard con e-mail e password. Per ulteriori informazioni sulla creazione di account, consultare Creazione di un account Productboard.
- È necessario aver creato un AWS account con accesso al servizio a AWS Glue.
- Disponi dei dettagli di autenticazione di un account Productboard: token JWT se si desidera utilizzare l'autenticazione personalizzata o l'ID client e segreto se si desidera utilizzare .0. OAuth2
- Se l'utente desidera utilizzarla`OAuth2.0`, registra l'applicazione con Productboard e configura l'applicazione seguendo le istruzioni in, Come integrare con Productboard tramite - documentazione per sviluppatori. OAuth2

Se soddisfi questi requisiti, sei pronto per connetterti AWS Glue al tuo account Productboard. Per le connessioni tipiche, non è necessario fare nient'altro in Productboard.

## Configurazione delle connessioni di Productboard

Productboard supporta l'autenticazione personalizzata e `OAuth2.0`. Per Productboard `OAuth2.0` supporta il tipo di concessione `AUTHORIZATION_CODE`.

- Questo tipo di concessione è considerato `OAuth` “a tre gambe” in quanto si basa sul reindirizzamento degli utenti al server di autorizzazione di terzi per l'autenticazione. Viene utilizzato durante la creazione di connessioni tramite la AWS Glue console. Per impostazione predefinita, l'utente che crea una connessione può fare affidamento su un'app connessa di AWS Glue proprietà in cui non è necessario fornire alcuna informazione `OAuth` correlata ad eccezione

del Productboard Client ID e del Client Secret. La AWS Glue Console reindirizzerà l'utente a Productboard, dove l'utente deve effettuare il login e consentire le autorizzazioni richieste per accedere AWS Glue alla propria istanza di Productboard.

- Gli utenti possono comunque scegliere di creare la propria app connessa in Productboard e fornire il proprio Client ID e Client Secret durante la creazione di connessioni tramite la Console. AWS Glue In questo scenario, verranno comunque reindirizzati a Productboard per accedere e autorizzare l'accesso AWS Glue alle proprie risorse.
- Questo tipo di concessione genera un token di aggiornamento e un token di accesso. Il token di accesso dura poco tempo e può essere aggiornato automaticamente senza l'interazione dell'utente utilizzando il token di aggiornamento.
- Per la documentazione pubblica di Productboard sulla creazione di un'app connessa per `AUTHORIZATION_CODE OAuth` flow, vedi Come integrarsi con Productboard tramite - documentazione per sviluppatori. OAuth2

Per configurare una connessione Productboard:

1. In AWS Secrets Manager, crea un segreto con i seguenti dettagli:
   - Per l'autenticazione `OAuth`: per le app connesse gestite dal cliente: il segreto deve contenere l'app connessa Consumer Secret con `USER_MANAGED_CLIENT_APPLICATION_CLIENT_SECRET` come chiave.
   - Per `Custom auth` - per le app connesse gestite dal cliente: il segreto deve contenere l'app connessa `JWT token` con `access_token` come chiave.

   > **Note**
   >
   > È necessario creare un segreto per ogni connessione in AWS Glue.

2. In AWS Glue Studio, crea una connessione in Connessioni dati seguendo i passaggi seguenti:
   a. Quando si seleziona una Origine dati, selezionare Productboard.
   b. Seleziona il ruolo IAM per il quale AWS Glue può assumere e dispone delle autorizzazioni per le seguenti azioni:

      JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
```

```
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:DescribeSecret",
        "secretsmanager:GetSecretValue",
        "secretsmanager:PutSecretValue",
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DeleteNetworkInterface"
      ],
      "Resource": "*"
    }
  ]
}
```

c. Selezionare il tipo di autenticazione per connetterti a un'origine dati:

   - Per l'autenticazione `OAuth`: fornire `Token URL` e `User Managed Client Application ClientId` dell'app Productboard.

d. Seleziona quello `secretName` che desideri utilizzare per questa connessione per AWS Glue inserire i token.

e. Selezionare le opzioni di rete se si desidera utilizzare la propria rete.

3. Concedi il permesso di lettura `secretName` al ruolo IAM associato al tuo AWS Glue lavoro.
4. Nella configurazione del AWS Glue lavoro, fornisci `connectionName` una connessione di rete aggiuntiva.

## Lettura da entità Productboard

Prerequisiti

Un oggetto di Productboard da cui si desidera leggere. Consultare la tabella delle entità supportate qui di seguito per verificare quelle disponibili.

Entità supportate

- Report di abuso
- Automazione
- Campagne
- Dettagli clic

- Elenchi
- Membri
- Dettagli aperti
- Segmenti
- Archiviazioni
- Iscrizione annullata

| Entità | Può essere Filtrato | Supporta Limite | Supporta Ordina per | Supporta Seleziona * | Supporta il partizionamento |
|---|---|---|---|---|---|
| Funzionalità | Sì | Sì | No | Sì | Sì |
| Componenti | No | Sì | No | Sì | No |
| Prodotti | No | Sì | No | Sì | No |
| Stati delle funzionalità | No | Sì | No | Sì | Sì |
| Definizioni di campi personalizzati | No | Sì | No | Sì | No |
| Valori di campi personalizzati | Sì | Sì | No | Sì | No |

Esempio

```
Productboard_read = glueContext.create_dynamic_frame.from_options(
    connection_type="Productboard",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "feature",
        "API_VERSION": "1"
```

```
    }
```

Dettagli di entità e campo di Productboard

- Funzionalità
- Componenti
- Stati delle funzionalità
- Prodotti
- Definizioni di campi personalizzati
- Valori di campi personalizzati

## Opzioni di connessione a Productboard

Di seguito sono riportate le opzioni di connessione per Productboard:

- `ENTITY_NAME`(String): (obbligatorio) utilizzato per la lettura/scrittura. Il nome dell'oggetto in Productboard.
- `API_VERSION`(String): (obbligatorio) utilizzato per la lettura. Versione dell'API Engage Rest di Productboard che si desidera utilizzare. Ad esempio: 3.0.
- `SELECTED_FIELDS`(List<String>): predefinito: vuoto (SELECT *). Utilizzato per la lettura. Colonne da selezionare per l'oggetto.
- `FILTER_PREDICATE`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Dev'essere in formato Spark SQL.
- `QUERY`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Query Spark SQL completa.

## Creazione di un account Productboard

1. Andare alla pagina di registrazione di Productboard, inserire il proprio ID e-mail e la password, quindi scegliere Accedi.
2. Nel campo Nome account, inserire il nome dell'account Productboard, poi selezionare la casella di spunta di Accetto l'Informativa sulla privacy.
3. Nella pagina Ora crea spazio di lavoro, nel campo URL dello spazio di lavoro, inserire l'URL del nuovo spazio di lavoro. Quindi scegliere Continua per passare alla pagina successiva e fornire i dettagli rimanenti.

Così viene creato l'account di prova. L'account di prova è gratuito per 15 giorni. Una volta scaduto il periodo di prova, è possibile acquistare un piano a pagamento. Prendere nota dell'indirizzo e-mail, della password e dell'URL dello spazio di lavoro. Queste informazioni saranno necessarie per accedere all'account in futuro."

Registrazione di un'applicazione **`OAuth2.0`**

1. Andare alla pagina di registrazione di Productboard, inserire il proprio ID e-mail e la password, quindi scegliere Accedi.
2. Selezionare l'icona Utente nell'angolo in alto a destra, quindi scegliere Account e fatturazione dal menu a discesa.
3. Seleziona Extra e scegli App registrate dal menu a discesa.
4. Individua e scegli Registra un'app.
5. Inserisci i seguenti dettagli:
   - Nome dell'app: nome dell'app.
   - Azienda/Organizzazione: nome dell'azienda o dell'organizzazione.
   - Sito web dell'app: sito web dell'app.
   - URI di reindirizzamento: un pattern URI di reindirizzamento è un percorso URI (o un elenco di percorsi separati da virgole) a cui Productboard può reindirizzare (se richiesto) quando il flusso di accesso è completo. Ad esempio, `https://ap-southeast-2\\.console\\.aws\\.amazon\\.com`
6. Scegli Create (Crea).
7. L'ID client e il segreto client saranno ora visibili. Copia i dati e salvali in un posto sicuro. Quindi, scegli Fine.

> **Note**
>
> Le stringhe Client ID e Client Secret sono credenziali utilizzate per stabilire una connessione con questo connettore quando si utilizza AppFlow o. AWS Glue

Recupero delle credenziali CustomAuth

1. Andare alla pagina di registrazione di Productboard, inserire il proprio ID e-mail e la password, quindi scegliere Accedi.

   Si ritornerà alla home page.

2. Nella home page, vai a Impostazioni dell'area di lavoro > Integrazioni > Pubblico > Token di accesso. APIs

   > **Note**
   >
   > Se la APIs sezione Pubblico non è visibile, è possibile che il tuo account utilizzi il piano Essentials. L'accesso ai token API richiede almeno un piano Pro. Le caratteristiche e i nomi dei piani sono soggetti a modifiche. Per ulteriori informazioni sui pacchetti, consultare i prezzi di Productboard.

3. Scegliere + per generare un nuovo token e assicurarsi di archiviarlo in modo sicuro per riferimenti futuri.

Creazione delle credenziali **`OAuth2.0`**

Per utilizzare l'autenticazione `OAuth2.0` con il connettore Productboard, è necessario registrare l'applicazione sulla piattaforma Productboard e generare un `Client ID` e un `Client Secret`.

1. Andare alla pagina di registrazione di Productboard, inserire il proprio ID e-mail e la password, quindi scegliere Accedi.
2. Per registrare una nuova OAuth2 applicazione con il tuo account Productboard, vai alla pagina Producboard.
3. Completare i campi obbligatori e selezionare gli ambiti necessari per ogni entità a cui si desidera accedere.

   > **Note**
   >
   > Sono stati scelti i seguenti quattro ambiti, necessari per le sei entità supportate.

4. L'URL di reindirizzamento deve avere il formato seguente: `https://ap-southeast-2\\.console\\.aws\\.amazon\\.com`

> **Note**
>
> I reindirizzamenti di Appflow URLs sono soggetti a modifiche. Una volta disponibile, aggiorna il reindirizzamento URLs per la piattaforma. AWS Glue

5. L'ID client e il segreto client saranno ora visibili. Copia i dati e salvali in un posto sicuro.
6. Puoi configurare e verificare `OAuth2` seguendo i passaggi indicati nella sezione Come integrare con Productboard tramite la documentazione per OAuth2 sviluppatori.

### Limitazioni

Di seguito sono riportate le limitazioni per il connettore di Productboard:

- Productboard non supporta il partizionamento basato su campi o record.

## Connessione a QuickBooks

QuickBooks è un'applicazione di contabilità leader per le piccole e medie imprese. QuickBooks le applicazioni di contabilità risalgono agli anni '80 come uno dei primi prodotti di Intuit e, di conseguenza, erano originariamente software desktop. Oggi QuickBooks offre diverse applicazioni contabili e finanziarie aziendali sia come software installabile che come software SaaS basato su cloud. Come QuickBooks utente, puoi AWS Glue connetterti al tuo account. QuickBooks Quindi, puoi utilizzarlo QuickBooks come fonte di dati nei tuoi lavori ETL. Esegui questi processi per trasferire dati tra QuickBooks AWS servizi o altre applicazioni supportate.

Argomenti

- AWS Glue supporto per QuickBooks
- Policy contenenti le operazioni API per creare e utilizzare le connessioni
- Configurazione QuickBooks
- Configurazione delle connessioni QuickBooks
- Lettura da QuickBooks entità
- QuickBooks opzioni di connessione
- Limitazioni e note per il QuickBooks connettore

## AWS Glue supporto per QuickBooks

AWS Glue supporta QuickBooks quanto segue:

Supportato come origine?

Sì. È possibile utilizzare i job AWS Glue ETL da QuickBooks cui interrogare i dati.

Supportato come destinazione?

No.

Versioni QuickBooks API supportate

Sono supportate le seguenti versioni QuickBooks API:

- v3

## Policy contenenti le operazioni API per creare e utilizzare le connessioni

La seguente policy di esempio descrive le autorizzazioni AWS IAM richieste per la creazione e l'utilizzo delle connessioni. Se si sta creando un nuovo ruolo, creare una policy che contenga quanto segue:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:ListConnectionTypes",
        "glue:DescribeConnectionType",
        "glue:RefreshOAuth2Tokens",
        "glue:ListEntities",
        "glue:DescribeEntity"
      ],
      "Resource": "*"
    }
  ]
}
```

In alternativa, se non desideri utilizzare il metodo precedente, utilizza le seguenti policy IAM gestite:

- AWSGlueServiceRole— Garantisce l'accesso alle risorse che AWS Glue i vari processi richiedono per l'esecuzione per conto dell'utente. Queste risorse includono AWS Glue Amazon S3, IAM, CloudWatch Logs e Amazon. EC2 Se segui la convenzione di denominazione delle risorse specificata in questa politica, AWS Glue i processi dispongono delle autorizzazioni richieste. Questa policy è in genere collegata ai ruoli specificati quando si definiscono crawler, processi ed endpoint di sviluppo.
- AWSGlueConsoleFullAccess— Concede l'accesso completo alle AWS Glue risorse quando un'identità a cui è allegata la policy utilizza la AWS console di gestione. Se si segue la convenzione per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questa policy è in genere associata agli utenti della AWS Glue console.

## Configurazione QuickBooks

Prima di poter AWS Glue utilizzare il trasferimento di dati da QuickBooks, è necessario soddisfare i seguenti requisiti:

Requisiti minimi

I requisiti minimi sono i seguenti:

- Hai un QuickBooks account.
- Il tuo QuickBooks account è abilitato all'accesso all'API.

Per ulteriori informazioni, consulta i seguenti argomenti nella QuickBooks documentazione:

- Creare un account Intuit
- Creare e iniziare a sviluppare la propria app

Se soddisfi questi requisiti, sei pronto per connetterti AWS Glue al tuo QuickBooks account. Per le connessioni tipiche, non è necessario fare nient'altro QuickBooks.

## Configurazione delle connessioni QuickBooks

QuickBooks supporta il tipo di concessione AUTHORIZATION_CODE per. OAuth2 Il tipo di concessione determina la modalità di AWS Glue comunicazione con cui richiedere l'accesso QuickBooks ai dati.

- Questo tipo di concessione è considerato «a tre gambe» in OAuth quanto si basa sul reindirizzamento degli utenti a un server di autorizzazione di terze parti per autenticare l'utente. Viene utilizzato durante la creazione di connessioni tramite la console. AWS Glue
- Gli utenti possono comunque scegliere di creare la propria app connessa QuickBooks e fornire il proprio ID client e il segreto del client durante la creazione di connessioni tramite la AWS Glue console. In questo scenario, verranno comunque reindirizzati all'accesso e QuickBooks all'autorizzazione ad accedere AWS Glue alle proprie risorse.
- Questo tipo di concessione genera un token di aggiornamento e un token di accesso. Il token di accesso dura poco tempo e può essere aggiornato automaticamente senza l'interazione dell'utente utilizzando il token di aggiornamento.
- Per la QuickBooks documentazione pubblica sulla creazione di un'app connessa per il OAuth flusso del codice di autorizzazione, vedi OAuth Configurazione 2.0.

Per configurare una QuickBooks connessione:

1. In AWS Glue Glue Studio, crea una connessione in Connessioni dati seguendo i passaggi seguenti:
   a. Quando si seleziona un tipo di connessione, selezionare QuickBooks.
   b. Fornisci l'URL dell'istanza e l'ID aziendale dell' QuickBooks istanza a cui desideri connetterti.
   c. Seleziona il ruolo AWS IAM che AWS Glue può assumere e dispone delle autorizzazioni per le seguenti azioni:

      JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:DescribeSecret",
        "secretsmanager:GetSecretValue",
        "secretsmanager:PutSecretValue",
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DeleteNetworkInterface"
      ],
      "Resource": "*"
```

```
    }
  ]
}
```

d. Seleziona quello `secretName` che desideri utilizzare per questa connessione per AWS Glue inserire i token.

e. Selezionare le opzioni di rete se si desidera utilizzare la propria rete.

2. Concedi il permesso di lettura `secretName` al ruolo IAM associato al tuo AWS Glue lavoro.

## Lettura da QuickBooks entità

Prerequisito

Un QuickBooks oggetto da cui vorresti leggere.

Entità supportate come origine:

| Entità | Può essere filtrato | Supporta limite | Supporta Ordina per | Supporta Seleziona * | Supporta Partizionamento |
|---|---|---|---|---|---|
| Account | Sì | Sì | Sì | Sì | Sì |
| Conto | Sì | Sì | Sì | Sì | Sì |
| Informazioni aziendali | No | No | No | Sì | No |
| Cliente | Sì | Sì | Sì | Sì | Sì |
| Dipendente | Sì | Sì | Sì | Sì | Sì |
| Stima | Sì | Sì | Sì | Sì | Sì |
| Fattura | Sì | Sì | Sì | Sì | Sì |
| Elemento | Sì | Sì | Sì | Sì | Sì |
| Pagamento | Sì | Sì | Sì | Sì | Sì |

| Entità | Può essere filtrato | Supporta limite | Supporta Ordina per | Supporta Seleziona * | Supporta Partizionamento |
|---|---|---|---|---|---|
| Preferenze | No | No | No | Sì | No |
| Profitti e perdite | Sì | No | No | Sì | No |
| Agenzia fiscale | Sì | Sì | Sì | Sì | Sì |
| Fornitori | Sì | Sì | Sì | Sì | Sì |

Esempio:

```
QuickBooks_read = glueContext.create_dynamic_frame.from_options(
    connection_type="quickbooks",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "Account",
        "API_VERSION": "v3"
    }
```

QuickBooks dettagli dell'entità e del campo:

Per ulteriori informazioni sui dettagli dell'entità e del campo, consultare:

- Account
- Conto
- CompanyInfo
- Cliente
- Dipendente
- Stima
- Fattura
- Elemento
- Pagamento

- Preferenze
- ProfitAndLoss
- TaxAgency
- Fornitori

Query di partizionamento

Partizionamento basato sul campo:

In QuickBooks, i campi Integer e DateTime datatype supportano il partizionamento basato sul campo.

Puoi fornire le opzioni Spark aggiuntive `PARTITION_FIELD`, `LOWER_BOUND`, `UPPER_BOUND` e `NUM_PARTITIONS` se desideri utilizzare la concorrenza in Spark. Con questi parametri, la query originale viene suddivisa in `NUM_PARTITIONS` query secondarie che possono essere eseguite contemporaneamente dalle attività Spark.

- `PARTITION_FIELD`: il nome del campo da utilizzare per partizionare la query.
- `LOWER_BOUND`: un valore limite inferiore inclusivo del campo di partizione scelto.

  Per il campo Datetime accettiamo il formato di timestamp Spark utilizzato nelle query SQL di Spark.

  Esempi di valori validi:

  ```
  "2024-05-07T02:03:00.00Z"
  ```
- `UPPER_BOUND`: un valore limite superiore esclusivo del campo di partizione scelto.
- `NUM_PARTITIONS`: il numero di partizioni.

Esempio:

```
QuickBooks_read = glueContext.create_dynamic_frame.from_options(
    connection_type="quickbooks",
    connection_options={
        "connectionName": "connectionName",
        "REALMID": "12345678690123456789",
        "ENTITY_NAME": "Account",
        "API_VERSION": "v3",
        "PARTITION_FIELD": "MetaData_CreateTime"
        "LOWER_BOUND": "2023-09-07T02:03:00.000Z"
```

```
        "UPPER_BOUND": "2024-05-07T02:03:00.000Z"
        "NUM_PARTITIONS": "10"
    }
```

Partizionamento basato su record:

La query originale viene suddivisa in NUM_PARTITIONS query secondarie che possono essere eseguite contemporaneamente dalle attività Spark.

- NUM_PARTITIONS: il numero di partizioni.

Esempio:

```
QuickBooks_read = glueContext.create_dynamic_frame.from_options(
    connection_type="quickbooks",
    connection_options={
        "connectionName": "connectionName",
        "REALMID": "1234567890123456789",
        "ENTITY_NAME": "Bill",
        "API_VERSION": "v3",
        "NUM_PARTITIONS": "10"
    }
```

## QuickBooks opzioni di connessione

Di seguito sono elencate le opzioni di connessione per QuickBooks:

- ENTITY_NAME(String): (obbligatorio) utilizzato per la lettura. Il nome dell'oggetto in QuickBooks.
- INSTANCE_URL(String) - (Obbligatorio) Un URL di QuickBooks istanza valido.
- API_VERSION(String) - (Obbligatorio) Usato per la lettura. QuickBooks Versione dell'API Rest che desideri utilizzare.
- REALM_ID(Stringa): un ID che identifica una singola società QuickBooks online a cui si inviano richieste.
- SELECTED_FIELDS(List<String>): predefinito: vuoto (SELECT *). Utilizzato per la lettura. Colonne da selezionare per l'oggetto.
- FILTER_PREDICATE(Stringa): predefinito: vuoto. Utilizzato per la lettura. Dev'essere in formato Spark SQL.
- QUERY(Stringa): predefinito: vuoto. Utilizzato per la lettura. Query Spark SQL completa.

- `PARTITION_FIELD`(Stringa): utilizzato per la lettura. Campo da utilizzare per partizionare la query.
- `LOWER_BOUND`(Stringa): utilizzato per la lettura. Un valore limite inferiore inclusivo del campo di partizione scelto.
- `UPPER_BOUND`(Stringa): utilizzato per la lettura. Un valore limite superiore esclusivo del campo di partizione scelto.
- `NUM_PARTITIONS`(Numero intero): predefinito: 1. Utilizzato per la lettura. Numero di partizioni da leggere.

### Limitazioni e note per il QuickBooks connettore

Di seguito sono riportate le limitazioni o le note relative al QuickBooks connettore:

- Nell'API `taxAgency`, il filtraggio "Ordina per" non funziona come previsto.

## Connessione a un'API REST

AWS Glue consente di configurare un file AWS Glue ConnectionType che può essere utilizzato per connettersi AWS Glue a qualsiasi fonte di dati basata su API REST. Questo può essere usato come fonte di dati nei tuoi lavori ETL. È possibile eseguire questi processi per trasferire dati tra l'origine dati basata sull'API REST e i AWS servizi o altre applicazioni supportate.

Argomenti

- AWS Glue supporto per l'API REST
- Politiche contenenti le operazioni API per la registrazione dei tipi di connessione e creating/using delle connessioni
- Configurazione di un'API REST ConnectionType
- Configurazione di una connessione API REST
- Tutorial: creazione di un'API REST ConnectionType e di una connessione
- Limitazioni

### AWS Glue supporto per l'API REST

AWS Glue supporta l'API REST come segue:

Supportato come origine?

Sì. È possibile utilizzare i job AWS Glue ETL per interrogare i dati da un'origine dati basata su API REST.

Supportato come destinazione?

No.

## Politiche contenenti le operazioni API per la registrazione dei tipi di connessione e creating/using delle connessioni

La seguente policy IAM di esempio descrive le autorizzazioni richieste per la registrazione, la creazione, la gestione e l'utilizzo delle connessioni API REST all'interno AWS Glue dei job ETL. Se si sta creando un nuovo ruolo, creare una policy che contenga quanto segue:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "glue:RegisterConnectionType",
                "glue:ListConnectionTypes",
                "glue:DescribeConnectionType",
                "glue:CreateConnection",
                "glue:RefreshOAuth2Tokens",
                "glue:ListEntities",
                "glue:DescribeEntity"
            ],
            "Resource": "*"
        }
    ]
}
```

È anche possibile utilizzare le seguenti policy IAM per autorizzare l'accesso:

- AWSGlueServiceRole— Garantisce l'accesso alle risorse che AWS Glue i vari processi richiedono per l'esecuzione per tuo conto. Queste risorse includono Amazon S3 AWS Glue, IAM, CloudWatch Logs e Amazon EC2. Se segui la convenzione di denominazione delle risorse specificata in questa politica, AWS Glue i processi dispongono delle autorizzazioni necessarie. Questa policy è in genere collegata ai ruoli specificati quando si definiscono crawler, processi ed endpoint di sviluppo.

- AWSGlueConsoleFullAccess— Concede l'accesso completo alle AWS Glue risorse quando un'identità a cui è allegata la policy utilizza la AWS console di gestione. Se si segue la convenzione per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questa policy è in genere associata agli utenti della AWS Glue console.

Se si forniscono opzioni di rete durante la creazione di una connessione API REST, nel ruolo IAM devono essere incluse anche le seguenti azioni:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:DescribeSecret",
                "secretsmanager:GetSecretValue",
                "secretsmanager:PutSecretValue",
                "ec2:CreateNetworkInterface",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DeleteNetworkInterface"
            ],
            "Resource": "*"
        }
    ]
}
```

## Configurazione di un'API REST ConnectionType

Prima di poterli utilizzare AWS Glue per trasferire dati dall'origine dati basata sull'API REST, devi soddisfare questi requisiti:

Requisiti minimi

I requisiti minimi sono i seguenti:

- Hai configurato e registrato un tipo di connessione API AWS Glue REST. Vedi Connessione a REST APIs.
- Se utilizzi le credenziali OAuth2 del cliente, il codice di autorizzazione o JWT, configura l'app client di conseguenza.

Se soddisfi questi requisiti, sei pronto per connetterti AWS Glue alla tua fonte di dati basata sull'API REST. In genere, non sono necessarie ulteriori configurazioni sul lato dell'API REST.

## Configurazione di una connessione API REST

Per configurare un connettore API AWS Glue REST, è necessario configurare un tipo di AWS Glue connessione. Questo tipo di connessione contiene dettagli sulle proprietà del funzionamento dell'origine dati REST e interpreta elementi come autenticazione, richieste, risposte, impaginazione, convalide ed entità/metadati. Per un elenco completo delle proprietà richieste per un tipo di connessione AWS Glue REST, consulta l' RegisterConnectionTypeAPI e i passaggi per la connessione a REST. APIs

Quando si crea il connettore API REST, è necessaria la seguente politica per consentire le azioni pertinenti:

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "glue:RegisterConnectionType",
                "glue:ListConnectionTypes",
                "glue:DescribeConnectionType",
                "glue:CreateConnection",
                "secretsmanager:DescribeSecret",
                "secretsmanager:GetSecretValue",
                "secretsmanager:PutSecretValue",
                "ec2:CreateNetworkInterface",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DeleteNetworkInterface"
            ],
            "Resource": "*"
        }
    ]
}
```

## Tutorial: creazione di un'API REST ConnectionType e di una connessione

Connessione all'API REST di Foo

Creeremo un'API AWS Glue REST ConnectionType e una AWS Glue connessione corrispondente per l'API REST di Foo. Questa API ha le seguenti proprietà (che possono essere recuperate dalla documentazione dell'API REST).

- URL dell'istanza: https://foo.cloud.com/rest/ v1.
- Tipo di autenticazione: OAuth2 (credenziali del cliente).
- Metodo REST: GET.
- Tipo di impaginazione: Offset con le proprietà «limit» e «offset» inserite nel parametro di query della richiesta.
- Entità supportate:
  - Barra: percorso relativo [/bar.json].
  - Baz: percorso relativo [/baz.json].

Una volta ottenuti tutti i dettagli, possiamo iniziare a creare la AWS Glue connessione all'API REST di Foo.

Per creare una connessione API REST:

1. Crea il tipo di connessione API REST AWS Glue chiamando l' RegisterConnectionType API tramite AWS API, CLI o SDK. Questo creerà una nuova ConnectionType risorsa in. AWS Glue

```
{
    "ConnectionType": "REST-FOO-CONNECTOR",
    "IntegrationType": "REST",
    "Description": "AWS Glue Connection Type for the FOO REST API",
    "ConnectionProperties": {
        "Url": {
            "Name": "Url",
            "Required": true,
            "DefaultValue": "https://foo.cloud.com/rest/v1",
            "PropertyType": "USER_INPUT"
        }
    },
    "ConnectorAuthenticationConfiguration": {
        "AuthenticationTypes": ["OAUTH2"],
        "OAuth2Properties": {
            "OAuth2GrantType": "CLIENT_CREDENTIALS"
        }
    },
```

```
    "RestConfiguration": {
        "GlobalSourceConfiguration": {
        "RequestMethod": "GET",
        "ResponseConfiguration": {
            "ResultPath": "$.result",
            "ErrorPath": "$.error.message"
        },
        "PaginationConfiguration": {
            "OffsetConfiguration": {
                "OffsetParameter": {
                    "Key": "offset",
                    "PropertyLocation": "QUERY_PARAM"
                },
                "LimitParameter": {
                    "Key": "limit",
                    "PropertyLocation": "QUERY_PARAM",
                    "DefaultValue": "50"
                }
            }
        }
    },
    "ValidationEndpointConfiguration": {
        "RequestMethod": "GET",
        "RequestPath": "/bar.json?offset=1&limit=10"
    },
    "EntityConfigurations": {
        "bar": {
            "SourceConfiguration": {
                "RequestMethod": "GET",
                "RequestPath": "/bar.json",
                "ResponseConfiguration": {
                    "ResultPath": "$.result",
                    "ErrorPath": "$.error.message"
                }
            },
            "Schema": {
                "name": {
                    "Name": "name",
                    "FieldDataType": "STRING"
                },
                "description": {
                    "Name": "description",
                    "FieldDataType": "STRING"
                },
```

```
                "id": {
                    "Name": "id",
                    "FieldDataType": "STRING"
                },
                "status": {
                    "Name": "status",
                    "FieldDataType": "STRING"
                }
            }
        }
    }
}
}
```

2. In AWS Secrets Manager, crea un segreto. Il segreto deve contenere l'app connessa Consumer Secret con `USER_MANAGED_CLIENT_APPLICATION_CLIENT_SECRET` come chiave.

> **Note**
>
> È necessario creare un segreto per ogni connessione in AWS Glue

3. Crea la AWS Glue connessione chiamando l' CreateConnection API utilizzando l' AWS API, la CLI o l'SDK.
   a. Fai riferimento al nome del tipo di connessione REST del passaggio 1 come «»ConnectionType.
   b. Fornisci il InstanceUrl e tutti gli altri ConnectionProperties che sono stati definiti durante il processo di AWS Glue ConnectionType registrazione.
   c. Scegli tra i tipi di autenticazione configurati. L'API REST utilizzata da Foo OAuth2 con il tipo di ClientCredentials concessione.
   d. Fornisci SecretArne altri AuthenticationPropertiesche sono configurati. Ad esempio, abbiamo configurato 0AUTH2 come tale, AuthenticationType quindi imposteremo le «OAuth2Proprietà» in CreateConnectionInput. Ciò richiederà proprietà come «OAuth2GrantType», «TokenUrl» e «OAuth2ClientApplication».
4. Effettua la CreateConnection richiesta che creerà la AWS Glue connessione.

```
{
    "ConnectionInput": {
        "Name": "ConnectionFooREST",
        "ConnectionType": "REST-FOO-CONNECTOR",
        "ConnectionProperties": {},
```

```
        "ValidateCredentials": true,
        "AuthenticationConfiguration": {
            "AuthenticationType": "OAUTH2",
            "SecretArn":
 "arn:aws:secretsmanager:<region>:<accountId>:secret:<secretId>",
            "OAuth2Properties": {
                "OAuth2GrantType": "CLIENT_CREDENTIALS",
                "TokenUrl": "https://foo.cloud.com/oauth/token",
                "OAuth2ClientApplication": {
                    "UserManagedClientApplicationClientId": "your-managed-client-id"
                }
            }
        }
    }
}
```

### Limitazioni

Di seguito sono riportate le limitazioni per il connettore API REST

- Il connettore API REST è disponibile solo tramite AWS API, CLI o SDK. Non è possibile configurare i connettori REST tramite la console.
- Il AWS Glue REST ConnectionType può essere configurato solo per LEGGERE i dati dall'origine dati basata sull'API REST. La connessione può essere utilizzata solo come SOURCE nei job AWS Glue ETL.
- Il filtraggio e il partizionamento non sono supportati.
- La selezione dei campi non è supportata.

## Connessione a Salesforce

Salesforce fornisce software di gestione delle relazioni con i clienti (CRM) che aiutano nelle vendite, nell'assistenza clienti, nell'e-commerce e altro ancora. Se sei un utente Salesforce, puoi connetterti AWS Glue al tuo account Salesforce. Quindi, puoi usare Salesforce come origine o destinazione dei dati per i tuoi processi ETL. Esegui questi processi per trasferire dati tra Salesforce e AWS i servizi o altre applicazioni supportate.

Argomenti

- AWS Glue supporto per Salesforce

- Policy contenenti le operazioni API per creare e utilizzare le connessioni
- Configurazione di Salesforce
- Applicare il profilo di amministratore di sistema
- Configurazione delle connessioni di Salesforce
- Lettura da Salesforce
- Scrivere su Salesforce
- Opzioni di connessione Salesforce
- Limitazioni per il connettore Salesforce
- Configurare il flusso del codice di autorizzazione per Salesforce
- Configura il OAuth flusso portatore JWT per Salesforce


## AWS Glue supporto per Salesforce

AWS Glue supporta Salesforce come segue:

Supportato come origine?

Sì. È possibile utilizzare i job AWS Glue ETL per interrogare i dati di Salesforce.

Supportato come destinazione?

Sì. È possibile utilizzare i processi AWS Glue ETL per scrivere record in Salesforce.

Versioni di API Salesforce supportate

Sono supportate le seguenti versioni di API Salesforce

- v58.0
- v59.0
- v60.0

## Policy contenenti le operazioni API per creare e utilizzare le connessioni

La seguente policy IAM di esempio descrive le autorizzazioni necessarie per creare, gestire e utilizzare le connessioni Salesforce all'interno AWS Glue dei job ETL. Se si sta creando un nuovo ruolo, creare una policy che contenga quanto segue:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:DescribeSecret",
        "secretsmanager:GetSecretValue",
        "secretsmanager:PutSecretValue",
        "glue:ListConnectionTypes",
        "glue:DescribeConnectionType",
        "glue:RefreshOAuth2Tokens",
        "glue:ListEntities",
        "glue:DescribeEntity"
      ],
      "Resource": "*"
    }
  ]
}
```

È anche possibile utilizzare le seguenti policy IAM per autorizzare l'accesso:

- AWSGlueServiceRole— Garantisce l'accesso alle risorse che i vari AWS Glue processi richiedono per l'esecuzione per conto dell'utente. Queste risorse includono AWS Glue Amazon S3, IAM, CloudWatch Logs e Amazon. EC2 Se segui la convenzione di denominazione delle risorse specificata in questa politica, AWS Glue i processi dispongono delle autorizzazioni richieste. Questa policy è in genere collegata ai ruoli specificati quando si definiscono crawler, processi ed endpoint di sviluppo.
- AWSGlueConsoleFullAccess— Concede l'accesso completo alle AWS Glue risorse quando un'identità a cui è allegata la policy utilizza la AWS console di gestione. Se si segue la convenzione per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questa policy è in genere associata agli utenti della AWS Glue console.

Se si forniscono opzioni di rete durante la creazione di una connessione di Salesforce, nel ruolo IAM devono essere incluse anche le seguenti azioni:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DeleteNetworkInterface"
      ],
      "Resource": "*"
    }
  ]
}
```

Per le connessioni di Salesforce Zero-ETL, consultare Prerequisiti di Zero-ETL.

Per le connessioni di Salesforce Zero-ETL, consultare Prerequisiti di Zero-ETL.

## Configurazione di Salesforce

Prima di poter AWS Glue utilizzare il trasferimento di dati da o verso Salesforce, devi soddisfare questi requisiti:

Requisiti minimi

I requisiti minimi sono i seguenti:

- Avere un account Salesforce.
- L'account Salesforce è abilitato all'accesso tramite API. L'accesso alle API è abilitato per impostazione predefinita per le edizioni Enterprise, Unlimited, Developer e Performance.

Se soddisfi questi requisiti, sei pronto per connetterti AWS Glue al tuo account Salesforce. AWS Glue gestisce i requisiti rimanenti con l'app AWS connessa gestita.

L'app AWS connessa gestita per Salesforce

L'app connessa AWS gestita ti aiuta a creare una connessione Salesforce in meno passaggi. In Salesforce, un'app connessa è un framework che autorizza le applicazioni esterne, ad esempio AWS Glue, ad accedere ai dati di Salesforce utilizzando la versione 2.0. OAuth Per utilizzare l'app AWS connessa gestita, crea una connessione Salesforce utilizzando la console. AWS Glue Quando configuri la connessione, imposta il tipo di OAuth concessione su Codice di autorizzazione e lascia selezionata la casella Usa l'applicazione client AWS gestita.

Quando salvi la connessione, verrai reindirizzato a Salesforce per accedere e approvare AWS Glue l'accesso al tuo account Salesforce.

Applicare il profilo di amministratore di sistema

In Salesforce, seguire questi passaggi per applicare il profilo di amministratore di sistema:

1. In Salesforce, accedi a Impostazioni > App connesse > Utilizzo delle app connesse. OAuth
2. Nell'elenco delle app connesse, trova AWS Glue e scegli Installa. Se necessario, scegliere Sblocca.
3. Andare su Impostazioni > Gestisci app connesse e scegliere AWS Glue. In OAuth Politiche, scegli Amministratore Gli utenti approvati sono pre-autorizzati e seleziona il profilo Amministratore di sistema. Questa azione limita l'accesso AWS Glue solo agli utenti con il profilo di amministratore di sistema.

## Applicare il profilo di amministratore di sistema

In Salesforce, seguire questi passaggi per applicare il profilo di amministratore di sistema:

1. In Salesforce, accedi a Impostazioni > App connesse > Utilizzo app connesse. OAuth
2. Nell'elenco delle app connesse, trova AWS Glue e scegli Installa. Se necessario, scegliere Sblocca.
3. Andare su Impostazioni > Gestisci app connesse e scegliere AWS Glue. In OAuth Politiche, scegli Amministratore Gli utenti approvati sono pre-autorizzati e seleziona il profilo Amministratore di sistema. Questa azione limita l'accesso AWS Glue solo agli utenti con il profilo di amministratore di sistema.

## Configurazione delle connessioni di Salesforce

Per configurare una connessione Salesforce:

1. In AWS Secrets Manager, crea un segreto con i seguenti dettagli:
   a. Per il tipo di concessione JWT_TOKEN, il segreto deve contenere la chiave JWT_TOKEN con il relativo valore.
   b. Per il tipo di AuthorizationCode sovvenzione:
      i. Per un'app AWS connessa gestita, è necessario fornire un segreto vuoto o un segreto con un valore temporaneo.
      ii. Per un'app connessa gestita dal cliente, il segreto deve contenere l'app connessa `Consumer Secret` con `USER_MANAGED_CLIENT_APPLICATION_CLIENT_SECRET` come chiave.
   c. Nota: è necessario creare un segreto per la connessione in AWS Glue.
2. In AWS Glue Glue Studio, crea una connessione in Connessioni dati seguendo i passaggi seguenti:
   a. Quando si seleziona un tipo di connessione, selezionare Salesforce.
   b. Fornire INSTANCE_URL dell'istanza di Salesforce a cui connettersi.
   c. Fornire l'ambiente Salesforce.
   d. Seleziona il ruolo AWS IAM che AWS Glue può assumere e dispone delle autorizzazioni per le seguenti azioni:

      JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:DescribeSecret",
        "secretsmanager:GetSecretValue",
        "secretsmanager:PutSecretValue",
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DeleteNetworkInterface"
      ],
      "Resource": "*"
    }
```

```
  ]
}
```

e. Seleziona il tipo di OAuth2 concessione che desideri utilizzare per le connessioni. Il tipo di concessione determina il modo in cui AWS Glue comunica con Salesforce per richiedere l'accesso ai dati. La scelta influisce sui requisiti da soddisfare prima di creare la connessione. È possibile scegliere uno di questi tipi:

- Tipo di concessione JWT_BEARER: questo tipo di concessione funziona bene per gli scenari di automazione in quanto consente di creare in anticipo un JSON Web Token (JWT) con le autorizzazioni di un particolare utente nell'istanza di Salesforce. Il creatore ha il controllo sulla durata di validità del JWT. AWS Glue è in grado di utilizzare il JWT per ottenere un token di accesso che viene utilizzato per chiamare Salesforce. APIs

  Questo flusso richiede che l'utente abbia creato un'app connessa nella propria istanza di Salesforce che consenta l'emissione di token di accesso basati su JWT per gli utenti.

  Per informazioni sulla creazione di un'app connessa per il JWT bearer OAuth flow, vedi OAuth 2.0 JWT bearer flow for integration. server-to-server Per configurare il flusso del bearer JWT con l'app connessa di Salesforce, consultare Configura il OAuth flusso portatore JWT per Salesforce.
- Tipo di concessione AUTHORIZATION_CODE: Questo tipo di concessione è considerato «a tre gambe» in OAuth quanto si basa sul reindirizzamento degli utenti al server di autorizzazione di terze parti per autenticare l'utente. Viene utilizzato durante la creazione di connessioni tramite la console. AWS Glue Per impostazione predefinita, l'utente che crea una connessione può fare affidamento su un'app AWS Glue AWS Glue connessa (applicazione client gestita) in cui non è necessario fornire alcuna informazione OAuth correlata ad eccezione dell'URL dell'istanza Salesforce. La AWS Glue console reindirizzerà l'utente a Salesforce, dove l'utente deve effettuare il login e consentire le autorizzazioni richieste per accedere AWS Glue alla propria istanza Salesforce.

  Gli utenti possono comunque scegliere di creare la propria app connessa in Salesforce e fornire il proprio ID client e il segreto del client durante la creazione di connessioni tramite la console. AWS Glue In questo scenario, verranno comunque reindirizzati a Salesforce per effettuare il login e AWS Glue autorizzare l'accesso alle proprie risorse.

  Questo tipo di concessione genera un token di aggiornamento e un token di accesso. Il token di accesso dura poco tempo e può essere aggiornato automaticamente senza l'interazione dell'utente utilizzando il token di aggiornamento.

Per informazioni sulla creazione di un'app connessa per il flusso del codice di autorizzazione, consulta. OAuth the section called "Configurare il flusso del codice di autorizzazione per Salesforce"

f. Seleziona quello `secretName` che desideri utilizzare per questa connessione per AWS Glue archiviare i token OAuth 2.0.

g. Selezionare le opzioni di rete se si desidera utilizzare la propria rete.

3. Concedi il permesso di lettura `secretName` al ruolo IAM associato al tuo AWS Glue lavoro.

4. Se si offrono opzioni di rete, concedere al ruolo IAM anche le seguenti autorizzazioni:

   JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DeleteNetworkInterface"
      ],
      "Resource": "*"
    }
  ]
}
```

Configurazione delle connessioni Salesforce con la CLI AWS

Puoi creare connessioni Salesforce utilizzando la CLI: AWS

```
aws glue create-connection --connection-input \
"{\"Name\": \"salesforce-conn1\",\"ConnectionType\": \"SALESFORCE\",
\"ConnectionProperties\": {\"ROLE_ARN\": \"arn:aws:iam::123456789012:role/glue-role\",
\"INSTANCE_URL\": \"https://example.my.salesforce.com\"},\"ValidateCredentials\": true,
\"AuthenticationConfiguration\": {\"AuthenticationType\": \"OAUTH2\",\"SecretArn\":
 \"arn:aws:secretsmanager:us-east-1:123456789012:secret:salesforce-conn1-secret-IAmcdk
\",\"OAuth2Properties\": {\"OAuth2GrantType\": \"JWT_BEARER\",\"TokenUrl\": \"https://
login.salesforce.com/services/oauth2/token\"}}}" \
--endpoint-url https://glue.us-east-1.amazonaws.com \
```

```
--region us-east-1
```

## Lettura da Salesforce

### Prerequisito

Un oggetto di Salesforce da cui si desidera leggere. È necessario il nome dell'oggetto, come `Account`, `Case` o `Opportunity`.

Esempio:

```
salesforce_read = glueContext.create_dynamic_frame.from_options(
    connection_type="salesforce",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "Account",
        "API_VERSION": "v60.0"
    }
)
```

### Query di partizionamento

Puoi fornire le opzioni Spark aggiuntive `PARTITION_FIELD`, `LOWER_BOUND`, `UPPER_BOUND` e `NUM_PARTITIONS` se desideri utilizzare la concorrenza in Spark. Con questi parametri, la query originale viene suddivisa in `NUM_PARTITIONS` query secondarie che possono essere eseguite contemporaneamente dalle attività Spark.

- `PARTITION_FIELD`: il nome del campo da utilizzare per partizionare la query.
- `LOWER_BOUND`: un valore limite inferiore inclusivo del campo di partizione scelto.

  Per i campi Date o Timestamp, il connettore accetta il formato di timestamp Spark utilizzato nelle query SQL di Spark.

  Esempi di valori validi:

  ```
  "TIMESTAMP \"1707256978123\""
  "TIMESTAMP '2018-01-01 00:00:00.000 UTC'"
  "TIMESTAMP \"2018-01-01 00:00:00 Pacific/Tahiti\""
  "TIMESTAMP \"2018-01-01 00:00:00\""
  "TIMESTAMP \"-123456789\" Pacific/Tahiti"
  "TIMESTAMP \"1702600882\""
  ```

- UPPER_BOUND: un valore limite superiore esclusivo del campo di partizione scelto.
- NUM_PARTITIONS: il numero di partizioni.
- TRANSFER_MODE: supporta due modalità, SYNC e ASYNC. Il valore predefinito è "SYNC". Se impostato su ASYNC, Bulk API 2.0 Query sarà utilizzato per l'elaborazione.

Esempio:

```
salesforce_read = glueContext.create_dynamic_frame.from_options(
    connection_type="salesforce",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "Account",
        "API_VERSION": "v60.0",
        "PARTITION_FIELD": "SystemModstamp",
        "LOWER_BOUND": "TIMESTAMP '2021-01-01 00:00:00 Pacific/Tahiti'",
        "UPPER_BOUND": "TIMESTAMP '2023-01-10 00:00:00 Pacific/Tahiti'",
        "NUM_PARTITIONS": "10",
        "TRANSFER_MODE": "ASYNC"
    }
)
```

Opzione FILTER_PREDICATE

FILTER_PREDICATE: è un parametro opzionale. Questa opzione viene utilizzata per il filtro delle query.

Esempi di FILTER_PREDICATE:

```
     Case 1: FILTER_PREDICATE with single criterion
     Examples:
       LastModifiedDate >= TIMESTAMP '2025-04-01 00:00:00 Pacific/Tahiti'
       LastModifiedDate <= TIMESTAMP "2025-04-01 00:00:00"
       LastModifiedDate >= TIMESTAMP '2018-01-01 00:00:00.000 UTC'
       LastModifiedDate <= TIMESTAMP "-123456789 Pacific/Tahiti"
       LastModifiedDate <= TIMESTAMP "1702600882"

     Case 2: FILTER_PREDICATE with multiple criteria
     Examples:
       LastModifiedDate >= TIMESTAMP '2025-04-01 00:00:00 Pacific/Tahiti' AND Id =
 "0012w00001CotGiAAJ"
       LastModifiedDate >= TIMESTAMP "1702600882" AND Id = "001gL000002i26MQAQ"
```

```
    Case 3: FILTER_PREDICATE single criterion with LIMIT
    Examples:
      LastModifiedDate >= TIMESTAMP "1702600882" LIMIT 2

    Case 4: FILTER_PREDICATE with LIMIT
    Examples:
      LIMIT 2
```

## Scrivere su Salesforce

### Prerequisiti

Un oggetto di Salesforce su cui si desidera scrivere. È necessario il nome dell'oggetto, come `Account`, `Case` o `Opportunity`.

Il connettore Salesforce supporta quattro operazioni di scrittura:

- INSERT
- UPSERT
- UPDATE
- DELETE

Quando si utilizza l'operazione di scrittura `UPSERT`, è necessario indicare l'opzione `ID_FIELD_NAMES` per specificare il campo ID esterno per i record.

È inoltre possibile aggiungere le seguenti opzioni di connessione:

- `TRANSFER_MODE`: supporta due modalità, `SYNC` e `ASYNC`. Il valore predefinito è "`SYNC`". Se impostato su `ASYNC`, Bulk API 2.0 Ingest sarà utilizzato per l'elaborazione.
- `FAIL_ON_FIRST_ERROR`: Il valore predefinito è`FALSE`, il che significa che il AWS Glue processo continuerà a elaborare tutti i dati anche se ci sono alcuni record di scrittura non riusciti. Se impostato su`TRUE`, il AWS Glue processo avrà esito negativo in caso di record di scrittura non riusciti e non continuerà l'elaborazione.

### Esempio

```
salesforce_write = glueContext.write_dynamic_frame.from_options(
```

```
    frame=frameToWrite,
    connection_type="salesforce",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "Account",
        "API_VERSION": "v60.0",
        "WRITE_OPERATION": "INSERT",
        "TRANSFER_MODE": "ASYNC",
        "FAIL_ON_FIRST_ERROR": "true"
    }
)
```

## Opzioni di connessione Salesforce

Le seguenti opzioni di connessione sono supportate per il connettore Salesforce:

- `ENTITY_NAME`(Stringa): (obbligatorio) utilizzato per lettura/scrittura. Il nome del tuo oggetto in Salesforce.
- `API_VERSION`(Stringa): (obbligatorio) utilizzato per lettura/scrittura. Versione dell'API Rest di Salesforce da usare.
- `SELECTED_FIELDS`(List<String>): predefinito: vuoto (SELECT *). Utilizzato per la lettura. Colonne da selezionare per l'oggetto.
- `FILTER_PREDICATE`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Deve essere in formato Spark SQL.

  Quando si fornisce un predicato di filtro, è supportato solo l'operatore `AND`. Altri operatori come `OR` e `IN` non sono attualmente supportati.
- `QUERY`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Query Spark SQL completa.
- `PARTITION_FIELD`(Stringa): utilizzato per la lettura. Campo da utilizzare per partizionare la query.
- `LOWER_BOUND`(Stringa): utilizzato per la lettura. Un valore limite inferiore inclusivo del campo di partizione scelto.
- `UPPER_BOUND`(Stringa): utilizzato per la lettura. Un valore limite superiore esclusivo del campo di partizione scelto.
- `NUM_PARTITIONS`(Numero intero): predefinito: 1. Utilizzato per la lettura. Numero di partizioni da leggere.
- `IMPORT_DELETED_RECORDS`(Stringa) - predefinito: FALSE. Utilizzato per la lettura. Per ottenere i record eliminati durante la query.

- `WRITE_OPERATION`(Stringa): predefinito: INSERT. Utilizzato per la scrittura. Il valore deve essere INSERT, UPDATE, UPSERT, DELETE.
- `ID_FIELD_NAMES`(String): predefinito: nullo. Obbligatorio per UPDATE e UPSERT.

## Limitazioni per il connettore Salesforce

Di seguito sono riportate le limitazioni per il connettore di Salesforce:

- Supportiamo solo Spark SQL e Salesforce SOQL non è supportato.
- I segnalibri dei processi non sono supportati.
- I nomi di campo di Salesforce fanno distinzione tra maiuscole e minuscole. Quando si scrive su Salesforce, i dati devono corrispondere alle maiuscole e alle minuscole dei campi definiti all'interno di Salesforce.

## Configurare il flusso del codice di autorizzazione per Salesforce

Consulta la documentazione pubblica di Salesforce per abilitare il flusso del codice di autorizzazione OAuth 2.0.

Per configurare l'app connessa:

1. Attiva la casella di controllo Abilita OAuth impostazioni.
2. Nel campo di testo Callback URL, inserisci uno o più URLs reindirizzamenti per. AWS Glue

   I reindirizzamenti URLs hanno il seguente formato:

   https://*region*.console.aws.amazon. com/gluestudio/oauth

   In questo URL, la regione è il codice della AWS regione in cui si utilizza per AWS Glue trasferire i dati da Salesforce. Ad esempio, il codice per la regione Stati Uniti orientali (Virginia settentrionale) è `us-east-1`. Per quella regione, l'URL è il seguente:

   https://us-east-1.console.aws.amazon.com/gluestudio/oauth

   Per le AWS regioni che AWS Glue supportano e i relativi codici, consulta gli AWS Glue endpoint e le quote nel Riferimento generale.AWS
3. Attivare la casella di spunta Richiedi segreto per Web Server Flow.
4. Nell'elenco Ambiti disponibili, aggiungi i seguenti OAuth ambiti:

- Gestisci i dati degli utenti tramite APIs (api)
- Accedi alle autorizzazioni personalizzate (custom_permissions)
- Accedi al servizio di URL dell'identità (id, profile, email, address, phone)
- Accedi a identificatori utente univoci (openid)
- Esegui le richieste in qualsiasi momento (refresh_token, offline_access)

5. Impostare la policy sui token di aggiornamento per l'app connessa su Il token di aggiornamento è valido fino alla revoca. In caso contrario, i processi avranno esito negativo alla scadenza del token di aggiornamento. Per ulteriori informazioni su come controllare e modificare la politica del token di aggiornamento, consulta Gestire le politiche di OAuth accesso per un'app connessa nella documentazione di Salesforce.

### Configura il OAuth flusso portatore JWT per Salesforce

Consulta la documentazione pubblica di Salesforce per abilitare l' server-to-serverintegrazione con i token Web JSON 2.0. OAuth

Dopo aver creato un JWT e configurato l'app connessa in modo appropriato in Salesforce, è possibile creare una nuova connessione di Salesforce con la chiave `JWT_TOKEN` impostata nel segreto di Secrets Manager. Imposta il tipo di OAuth concessione su JWT Bearer Token durante la creazione della connessione.

## Connessione a Salesforce Marketing Cloud

Salesforce Marketing Cloud è un fornitore di software di automazione e analisi del marketing per e-mail, dispositivi mobili, social media e marketing online. Inoltre offre servizi di consulenza e implementazione. Come utente di Salesforce Marketing Cloud, puoi connetterti AWS Glue al tuo account Salesforce Marketing Cloud. Quindi, puoi usare Salesforce Marketing Cloud come origine o destinazione dei dati per i tuoi processi ETL. Esegui questi processi per trasferire dati tra Salesforce Marketing Cloud e AWS servizi o altre applicazioni supportate.

Argomenti

- AWS Glue supporto per Salesforce Marketing Cloud
- Policy contenenti le operazioni API per creare e utilizzare le connessioni
- Configurazione di Salesforce Marketing Cloud
- Configurazione delle connessioni di Salesforce Marketing Cloud
- Lettura dalle entità Salesforce Marketing Cloud

- Scrivere alle entità Salesforce Marketing Cloud
- Opzioni di connessione di Salesforce Marketing Cloud
- Limitazioni e note per il connettore di Salesforce Marketing Cloud

## AWS Glue supporto per Salesforce Marketing Cloud

AWS Glue supporta Salesforce Marketing Cloud come segue:

Supportato come origine?

Sì. Puoi utilizzare i job AWS Glue ETL per interrogare i dati da Salesforce Marketing Cloud.

Supportato come destinazione?

No.

Versioni API supportate di Salesforce Marketing Cloud

Le seguenti versioni dell'API Salesforce Marketing Cloud sono supportate:

- v1

## Policy contenenti le operazioni API per creare e utilizzare le connessioni

La seguente policy di esempio descrive le autorizzazioni AWS IAM richieste per la creazione e l'utilizzo delle connessioni. Se si sta creando un nuovo ruolo, creare una policy che contenga quanto segue:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:ListConnectionTypes",
        "glue:DescribeConnectionType",
        "glue:RefreshOAuth2Tokens",
        "glue:ListEntities",
        "glue:DescribeEntity"
```

```
      ],
      "Resource": "*"
    }
  ]
}
```

In alternativa, se non desideri utilizzare il metodo precedente, utilizza le seguenti policy IAM gestite:

- AWSGlueServiceRole— Garantisce l'accesso alle risorse che AWS Glue i vari processi richiedono per l'esecuzione per conto dell'utente. Queste risorse includono AWS Glue Amazon S3, IAM, CloudWatch Logs e Amazon. EC2 Se segui la convenzione di denominazione delle risorse specificata in questa politica, AWS Glue i processi dispongono delle autorizzazioni richieste. Questa policy è in genere collegata ai ruoli specificati quando si definiscono crawler, processi ed endpoint di sviluppo.
- AWSGlueConsoleFullAccess— Concede l'accesso completo alle AWS Glue risorse quando un'identità a cui è allegata la policy utilizza la AWS console di gestione. Se si segue la convenzione per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questa policy è in genere associata agli utenti della AWS Glue console.

## Configurazione di Salesforce Marketing Cloud

Prima di poter AWS Glue utilizzare il trasferimento di dati da Salesforce Marketing Cloud, devi soddisfare questi requisiti:

Requisiti minimi

I requisiti minimi sono i seguenti:

- Avere un account Salesforce Marketing Cloud. Per ulteriori informazioni, consulta Creazione di un account Salesforce Marketing Cloud.
- L'account Salesforce Marketing Cloud è abilitato all'accesso tramite API. L'accesso alle API è abilitato per impostazione predefinita per le edizioni Enterprise, Unlimited, Developer e Performance.

Se soddisfi questi requisiti, sei pronto per connetterti AWS Glue al tuo account Salesforce Marketing Cloud. Per le connessioni tipiche, non è necessario fare nient'altro in Salesforce Marketing Cloud.

### Creazione di un account Salesforce Marketing Cloud

Per Salesforce Marketing Cloud, è necessario contattare il fornitore per la creazione dell'account. Se l'utente o la sua azienda hanno un'associazione con Salesforce, contattare l'account manager Salesforce per richiedere una licenza Salesforce Marketing Cloud. Altrimenti, attenersi alla seguente procedura per richiedere di essere contattati da un rappresentante Salesforce:

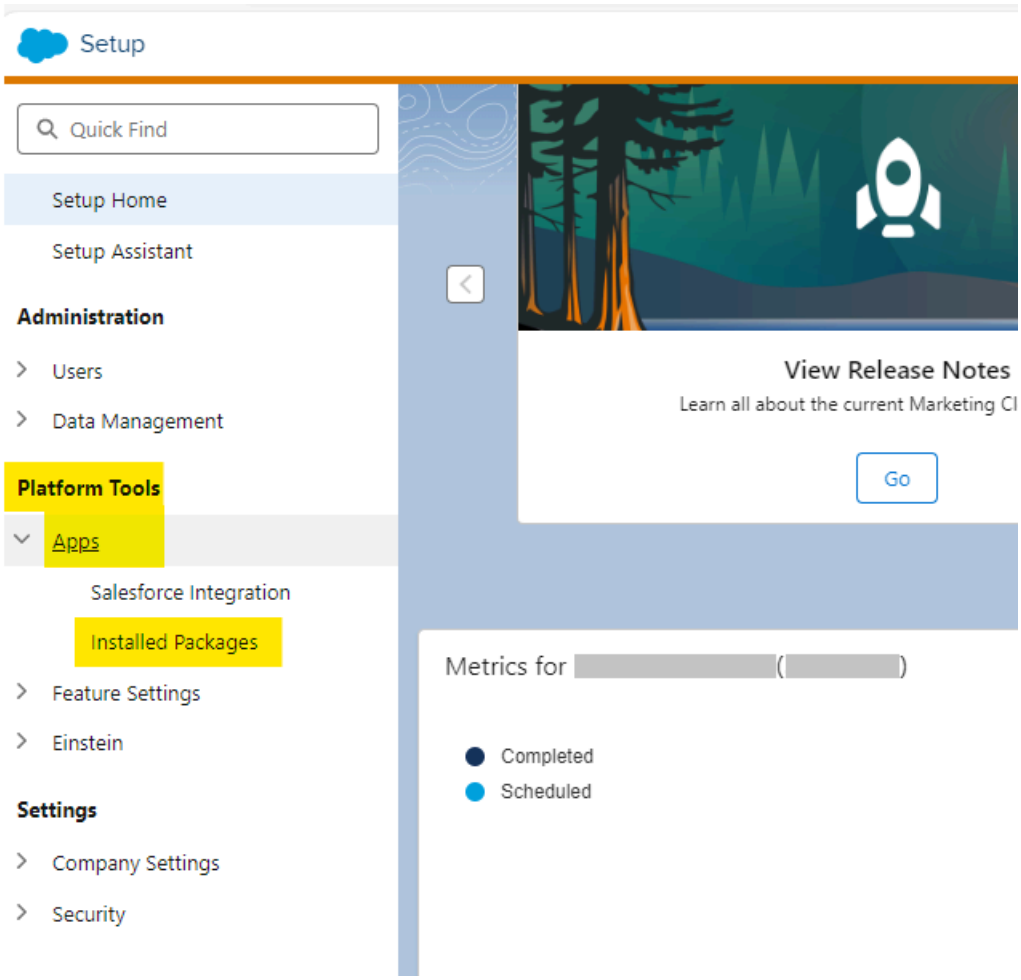
1. Vai a https://www.salesforce.com/in/products/marketing-cloud/overview/e scegli Registrati.
2. Selezionare il link Contattaci nella parte superiore destra della pagina.
3. Inserire le informazioni richieste nel modulo e scegliere Contattami.

Un rappresentante Salesforce contatterà l'utente per discutere delle sue esigenze.

### Creazione di un progetto e credenziali OAuth 2.0

Per ottenere un progetto e le credenziali OAuth 2.0:

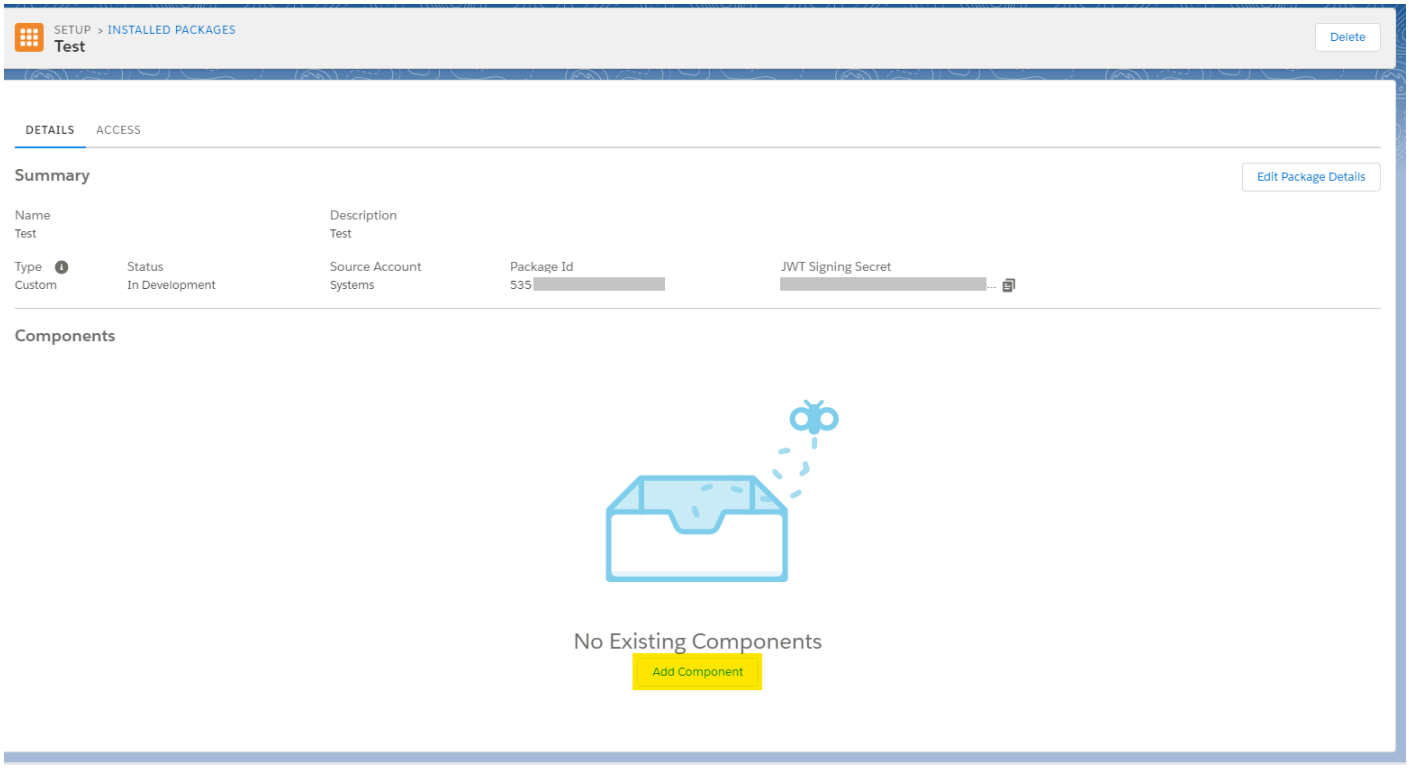
1. Accedere all'istanza di Salesforce Marketing Cloud con nome utente e password ed eseguire l'autenticazione utilizzando il numero di cellulare registrato.
2. Fare clic sul proprio profilo nell'angolo in alto a destra, quindi andare su Configurazione.
3. In Strumenti piattaforma, scegliere App, poi scegliere Pacchetti installati.




4. Nella pagina Pacchetti installati, fare clic su Nuovo nell'angolo in alto a destra. Fornire il nome e la descrizione del pacchetto.

   Salvare il pacchetto. Dopo il salvataggio del pacchetto, è possibile visualizzarne i dettagli.

5. Nella pagina Dettagli del pacchetto, nella sezione Componente, scegliere Aggiungi componente.




6. Selezionare il Tipo di componente come "Integrazione API" e fare clic su Avanti.
7. Seleziona il tipo di integrazione come «Da server a server» (che include il tipo di OAuth concessione delle credenziali del client) e fai clic su Avanti.
8. Aggiungere gli ambiti in base ai requisiti e fare clic su Salva.

## Configurazione delle connessioni di Salesforce Marketing Cloud

Salesforce Marketing Cloud supporta il tipo di concessione CLIENT CREDENTIALS per. OAuth2

- Questo tipo di concessione è considerato OAuth 2.0 a 2 gambe in quanto viene utilizzato dai client per ottenere un token di accesso al di fuori del contesto di un utente. AWS Glue è in grado di utilizzare l'ID client e il client secret per autenticare Salesforce Marketing Cloud APIs , forniti dai servizi personalizzati definiti dall'utente.
- Ogni servizio personalizzato è di proprietà di un utente che utilizza solo API e dispone di una serie di ruoli e autorizzazioni che autorizzano il servizio a eseguire azioni specifiche. Un token di accesso è associato a un singolo servizio personalizzato.
- Questo tipo di concessione determina un token di accesso di breve durata e che può essere rinnovato chiamando un endpoint dell'identità.

- Per la documentazione pubblica di Salesforce Marketing Cloud per OAuth 2.0 con le credenziali del cliente, consulta Configurazione dell'ambiente di sviluppo per pacchetti avanzati.

Per configurare una connessione di Salesforce Marketing Cloud:

1. In AWS Secrets Manager, crea un segreto con i seguenti dettagli:
   a. Per le app connesse gestite dal cliente: il segreto deve contenere l'app connessa Consumer Secret con `USER_MANAGED_CLIENT_APPLICATION_CLIENT_SECRET` come chiave.
   b. Nota: devi creare un segreto per le tue connessioni in AWS Glue.
2. In AWS Glue Glue Studio, crea una connessione in Connessioni dati seguendo i passaggi seguenti:
   a. Quando si seleziona un tipo di connessione, selezionare Salesforce Marketing Cloud.
   b. Fornire il `Subdomain Endpoint` del Salesforce Marketing Cloud a cui ci si desidera connettersi.
   c. Seleziona il ruolo AWS IAM che AWS Glue può assumere e dispone delle autorizzazioni per le seguenti azioni:

      JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:DescribeSecret",
        "secretsmanager:GetSecretValue",
        "secretsmanager:PutSecretValue",
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DeleteNetworkInterface"
      ],
      "Resource": "*"
    }
  ]
}
```

   d. Seleziona quello `secretName` che desideri utilizzare per questa connessione per AWS Glue inserire i token.

e. Selezionare le opzioni di rete se si desidera utilizzare la propria rete.

3. Concedi il permesso di lettura `secretName` al ruolo IAM associato al tuo AWS Glue lavoro.

## Lettura dalle entità Salesforce Marketing Cloud

Prerequisito

Un oggetto di Salesforce Marketing Cloud da cui si desidera leggere. È necessario il nome dell'oggetto, come `Activity` o `Campaigns`. Nella tabella seguente sono illustrate le entità supportate.

Entità supportate come origine:

| Entità | Interfaccia | Può essere Filtrato | Supporta limite | Supporta Ordina per | Supporta SELEZIONA * | Supporta il partizionamento |
|---|---|---|---|---|---|---|
| Callback di notifiche di eventi | REST | No | No | No | Sì | No |
| Elenco di seed | REST | No | Sì | No | Sì | No |
| Configurazione | REST | Sì | Sì | No | Sì | No |
| Verifica del dominio | REST | Sì | Sì | Sì | Sì | No |
| Oggetti tag nidificati | REST | Sì | No | No | Sì | No |
| Contatti | REST | No | Sì | No | Sì | No |
| Iscrizione alle notifiche di eventi | REST | No | No | No | Sì | No |

| Entità | Interfaccia | Può essere Filtrato | Supporta limite | Supporta Ordina per | Supporta SELEZIONA * | Supporta il partizionamento |
|---|---|---|---|---|---|---|
| Messaggistica | REST | No | Sì | No | Sì | No |
| Attività | SOAP | No | No | No | Sì | Sì |
| Evento di mancato recapito | SOAP | No | No | No | Sì | Sì |
| Evento di clic | SOAP | No | No | No | Sì | Sì |
| Area contenuti | SOAP | No | No | No | Sì | Sì |
| Estensione di dati | SOAP | No | Sì | No | Sì | Sì |
| Email | SOAP | No | Sì | No | Sì | Sì |
| Evento di e-mail inoltrata | SOAP | No | Sì | No | Sì | Sì |
| Inoltra email OptInEvent | SOAP | No | Sì | No | Sì | Sì |
| Link | SOAP | No | Sì | No | Sì | Sì |
| Link Invia | SOAP | No | Sì | No | Sì | Sì |
| List | SOAP | No | Sì | No | Sì | Sì |
| Abbonato all'elenco | SOAP | No | Sì | No | Sì | Sì |

| Entità | Interfaccia | Può essere Filtrato | Supporta limite | Supporta Ordina per | Supporta SELEZIONA * | Supporta il partizionamento |
|---|---|---|---|---|---|---|
| Evento non inviato | SOAP | No | Sì | No | Sì | Sì |
| Evento aperto | SOAP | No | Sì | No | Sì | Sì |
| Invia | SOAP | No | Sì | No | Sì | Sì |
| Evento inviato | SOAP | No | Sì | No | Sì | Sì |
| Abbonato | SOAP | No | Sì | No | Sì | Sì |
| Evento di sondaggio | SOAP | No | Sì | No | Sì | Sì |
| Evento di annullamento dell'iscrizione | SOAP | No | Sì | No | Sì | Sì |
| Eventi di audit | REST | No | Sì | Sì | Sì | No |
| Campagne | REST | No | Sì | Sì | Sì | No |
| Interazioni | REST | No | Sì | Sì | Sì | No |
| Risorse di contenuti | REST | No | Sì | Sì | Sì | No |

Esempio per REST:

```
salesforcemarketingcloud_read = glueContext.create_dynamic_frame.from_options(
    connection_type="salesforcemarketingcloud",
```

```
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "Campaigns",
        "API_VERSION": "v1",
        "INSTANCE_URL": "https://**********.rest.marketingcloudapis.com"
    }
)
```

Esempio per SOAP:

```
salesforcemarketingcloud_read = glueContext.create_dynamic_frame.from_options(
    connection_type="salesforcemarketingcloud",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "Activity",
        "API_VERSION": "v1",
        "INSTANCE_URL": "https://**********.soap.marketingcloudapis.com"
    }
)
```

Dettagli dell'entità e del campo di Salesforce Marketing Cloud:

Nelle tabelle seguenti sono descritte le entità di Salesforce Marketing Cloud. Sono presenti entità REST con metadati statici ed entità SOAP con metadati dinamici.

Entità REST con metadati statici:

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| Callback di notifiche di eventi | callbackId | Stringa | |
| | callbackName | Stringa | |
| | url | Stringa | |
| | maxBatchSize | Numero intero | |
| | status | Stringa | |
| | statusReason | Stringa | |
| Seed-List | id | Stringa | |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | nome | Stringa | |
| | description | Stringa | |
| | activeSeedCount | Numero intero | |
| Configurazione | customerKey | Stringa | |
| | nome | Stringa | |
| | description | Stringa | |
| | locationType | Stringa | '=' |
| | awsFileTransferLocation | Struct | |
| Verifica del dominio | enterpriseId | Numero intero | |
| | status | Stringa | '=' |
| | domainType | Stringa | '=' |
| | memberId | Numero intero | |
| | emailSendTime | DateTime | |
| | domain | Stringa | |
| | isSendable | Booleano | |
| Oggetti tag nidificati | id | Numero intero | |
| | modifiedDate | DateTime | |
| | tag | List | |
| | nome | Stringa | |
| | description | Stringa | |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | parentId | Numero intero | |
| Contatti | values | List | |
| Iscrizione alle notifiche di eventi | subscriptionName | Stringa | |
| | callbackId | Stringa | |
| | callbackName | Stringa | |
| | eventCategoryTypes | List | |
| | filtri | List | |
| | url | Stringa | |
| | maxBatchSize | Numero intero | |
| | subscriptionId | Stringa | |
| | status | Stringa | |
| | statusReason | Stringa | |
| Messaggistica | deliveryTime | DateTime | |
| | id | Stringa | |
| | messageId | Stringa | |
| | status | Stringa | |
| | in | Struct | |
| Interazioni | status | Stringa | '=' |
| | id | Stringa | |
| | key | Stringa | |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | nome | Stringa | |
| | lastPublishedDate | DateTime | |
| | description | Stringa | |
| | version | Numero intero | |
| | workflowApiVersion | Numero intero | |
| | createdDate | DateTime | |
| | modifiedDate | DateTime | |
| | goals | Struct | |
| | stats | Struct | |
| | entryMode | Stringa | |
| | defaults | Struct | |
| | executionMode | Struct | |
| | definitionId | Stringa | |
| Risorse di contenuti | id | Numero intero | |
| | customerKey | Stringa | |
| | objectId | Stringa | |
| | contentType | Stringa | |
| | assetType | Struct | |
| | nome | Stringa | |
| | description | Stringa | |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | owner | Struct | |
| | createdDate | DateTime | |
| | createdBy | Struct | |
| | modifiedDate | DateTime | |
| | modifiedBy | Struct | |
| | thumbnail | Struct | |
| | category | Struct | |
| | meta | Struct | |
| | viste | Struct | |
| | availableViews | Struct | |
| | data | Struct | |
| | legacyData | Struct | |
| | modelVersion | Numero intero | |
| | Versione | Numero intero | |
| | Locked (Bloccato) | Booleano | |
| | FileProperties | Struct | |
| | Tag | List | |
| | Contenuti | Stringa | |
| | Progettazione | Stringa | |
| | SuperContent | Stringa | |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | CustomFields | Struct | |
| | Blocchi | Struct | |
| | MinBlocks | Numero intero | |
| | MaxBlocks | Numero intero | |
| | Canali | Struct | |
| | AllowedBlocks | List | |
| | Slot | Struct | |
| | BusinessUnitAvailability | Struct | |
| | sharingProperties | Struct | |
| | sharingProperties.sharedWith | Struct | |
| | sharingProperties.sharingType | Stringa | |
| | Modello | Struct | |
| | File | Stringa | |
| | GenerateFrom | Stringa | |
| Eventi di audit | id | Numero intero | |
| | createdDate | DateTime | |
| | memberId | Numero intero | |
| | enterpriseId | Numero intero | |
| | employee | Struct | |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | objectType | Struct | |
| | operation | Struct | |
| | oggetto | Struct | |
| | transactionId | Stringa | |
| Campagne | id | Numero intero | |
| | createdDate | DateTime | |
| | modifiedDate | DateTime | |
| | nome | Stringa | |
| | description | Stringa | |
| | campaignCode | Stringa | |
| | color | Stringa | |
| | favorite | Booleano | |

Entità SOAP con metadati dinamici:

| Entità | Tipo di dati | Operatori supportati |
|---|---|---|
| Attività | Stringa | LIKE,!=,= |
| | Struct | |
| | Numero intero | !=,=,>=,<=,<,> |
| | Double | !=,=,>=,<=,<,> |
| | Booleano | !=,= |

| Entità | Tipo di dati | Operatori supportati |
|---|---|---|
| | DateTime | >=,<=,<,>,=,BETWEEN |
| Evento di mancato recapito | Numero intero | !=,=,>=,<=,<,> |
| | DateTime | >=,<=,<,>,=,BETWEEN |
| | Stringa | LIKE,!=,= |
| | Struct | |
| Evento di clic | Numero intero | !=,=,>=,<=,<,> |
| | DateTime | >=,<=,<,>,=,BETWEEN |
| | Stringa | LIKE,!=,= |
| | Struct | |
| Area contenuti | Struct | |
| | Stringa | LIKE,!=,= |
| | Numero intero | !=,=,>=,<=,<,> |
| | DateTime | >=,<=,<,>,=,BETWEEN |
| | Booleano | !=,= |
| Estensione di dati | DateTime | >=,<=,<,>,=,BETWEEN |
| | Stringa | LIKE,!=,= |
| Email | Numero intero | !=,=,>=,<=,<,> |
| | Stringa | LIKE,!=,= |
| | DateTime | >=,<=,<,>,=,BETWEEN |
| | Booleano | !=,= |

| Entità | Tipo di dati | Operatori supportati |
|---|---|---|
| | Struct | |
| Evento di e-mail inoltrata | Numero intero | !=,=,>=,<=,<,> |
| | Stringa | LIKE,!=,= |
| | DateTime | >=,<=,<,>,=,BETWEEN |
| | Struct | |
| Email inoltrata OptInEvent | Numero intero | !=,=,>=,<=,<,> |
| | Stringa | LIKE,!=,= |
| | DateTime | >=,<=,<,>,=,BETWEEN |
| | Struct | |
| Link | Numero intero | !=,=,>=,<=,<,> |
| Link invia | Numero intero | !=,=,>=,<=,<,> |
| | Stringa | LIKE,!=,= |
| | Double | !=,=,>=,<=,<,> |
| List | Numero intero | !=,=,>=,<=,<,> |
| | Stringa | LIKE,!=,= |
| | DateTime | >=,<=,<,>,=,BETWEEN |
| | Struct | |
| Abbonato all'elenco | Numero intero | !=,=,>=,<=,<,> |
| | Stringa | LIKE,!=,= |
| | DateTime | >=,<=,<,>,=,BETWEEN |

| Entità | Tipo di dati | Operatori supportati |
|---|---|---|
| | Struct | |
| Evento non inviato | Numero intero | !=,=,>=,<=,<,> |
| | Stringa | LIKE,!=,= |
| | DateTime | >=,<=,<,>,=,BETWEEN |
| | Struct | |
| Evento aperto | Numero intero | !=,=,>=,<=,<,> |
| | Stringa | LIKE,!=,= |
| | DateTime | >=,<=,<,>,=,BETWEEN |
| | Struct | |
| Invia | Numero intero | !=,=,>=,<=,<,> |
| | Stringa | LIKE,!=,= |
| | DateTime | >=,<=,<,>,=,BETWEEN |
| | Booleano | !=,= |
| | Struct | |
| Evento inviato | Numero intero | !=,=,>=,<=,<,> |
| | Stringa | LIKE,!=,= |
| | DateTime | >=,<=,<,>,=,BETWEEN |
| | Struct | |
| Abbonato | Numero intero | !=,=,>=,<=,<,> |
| | Stringa | LIKE,!=,= |

| Entità | Tipo di dati | Operatori supportati |
|---|---|---|
| | DateTime | >=,<=,<,>,=,BETWEEN |
| | Struct | |
| Evento di sondaggio | Numero intero | !=,=,>=,<=,<,> |
| | Stringa | LIKE,!=,= |
| | DateTime | >=,<=,<,>,=,BETWEEN |
| | Struct | |
| Evento di annullamento dell'iscrizione | Numero intero | !=,=,>=,<=,<,> |
| | Stringa | LIKE,!=,= |
| | DateTime | >=,<=,<,>,=,BETWEEN |
| | Booleano | !=,= |
| | Struct | |

Query di partizionamento

In Salesforce Marketing Cloud, i campi Integer e DateTime Datatype supportano il partizionamento basato sul campo.

Puoi fornire le opzioni Spark aggiuntive `PARTITION_FIELD`, `LOWER_BOUND`, `UPPER_BOUND` e `NUM_PARTITIONS` se desideri utilizzare la concorrenza in Spark. Con questi parametri, la query originale viene suddivisa in `NUM_PARTITIONS` query secondarie che possono essere eseguite contemporaneamente dalle attività Spark.

- `PARTITION_FIELD`: il nome del campo da utilizzare per partizionare la query.
- `LOWER_BOUND`: un valore limite inferiore inclusivo del campo di partizione scelto.

  Per il campo timestamp, accettiamo il formato di timestamp Spark utilizzato nelle query SQL di Spark.

Esempi di valori validi:

```
“2024-05-07T02:03:00.00Z"
```

- UPPER_BOUND: un valore limite superiore esclusivo del campo di partizione scelto.
- NUM_PARTITIONS: il numero di partizioni.

Esempio:

```
salesforcemarketingcloud_read = glueContext.create_dynamic_frame.from_options(
    connection_type="salesforcemarketingcloud",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "ListSubscriber",
        "API_VERSION": "v1",
        "PARTITION_FIELD": "CreatedDate",
        "LOWER_BOUND": "2023-09-07T02:03:00.000Z",
        "UPPER_BOUND": "2024-05-07T02:03:00.000Z",
        "NUM_PARTITIONS": "10"
    }
)
```

## Scrivere alle entità Salesforce Marketing Cloud

Prerequisiti

- Un oggetto di Salesforce Marketing a cui si desidera scrivere. È necessario specificare il nome dell'oggetto, come ad esempio `List` o `Campaigns` una qualsiasi delle altre entità descritte nella tabella seguente.
- Il connettore Salesforce Marketing Cloud supporta tre operazioni di scrittura:
  - INSERT
  - UPSERT
  - UPDATE

  Quando si utilizzano le operazioni di scrittura `UPDATE` e `UPSERT`, è necessario indicare l'opzione `ID_FIELD_NAMES` per specificare il campo ID esterno per i record.

Entità supportate come destinazione:

| Entità | Priorità | Interfaccia (REST, SOAP, ecc.) | Può essere Inserted | Può essere Updated | Può essere Upserted |
|---|---|---|---|---|---|
| Campagne | P0 | REST | Y- Singolo | Y- Singolo | N |
| Risorse di contenuti | P0 | REST | Y- Singolo, bulk | Y- Singolo | N |
| Contatti | P1 | REST | Y- Singolo | Y- Singolo | N |
| Verifica del dominio | P1 | REST | Y- Singolo | Y- Singolo, bulk | N |
| Callback di notifiche di eventi | P1 | REST | Y- Singolo | Y- Singolo | N |
| Iscrizione alle notifiche di eventi | P1 | REST | Y- Singolo | Y- Singolo | N |
| Messaggistica | P1 | REST | Y- Singolo | N | N |
| Oggetto tag nidificato | P2 | REST | Y- Singolo | Y- Singolo | N |
| Seed-List | P1 | REST | Y- Singolo | Y- Singolo | N |
| Configurazione | P1 | REST | Y- Singolo | Y- Singolo | N |
| Estensione di dati | P0 | SOAP | Y- Singolo | Y- Singolo | Y- Singolo |
| Email | P0 | SOAP | Y- Singolo | Y- Singolo | N |
| List | P0 | SOAP | Y- Singolo | Y- Singolo | N |

| Entità | Priorità | Interfacc ia (REST, SOAP, ecc.) | Può essere Inserted | Può essere Updated | Può essere Upserted |
|---|---|---|---|---|---|
| Invia | P0 | SOAP | Y- Singolo | N | N |
| Abbonato | P0 | SOAP | Y- Singolo | Y- Singolo | N |

Esempio di operazione INSERT per REST:

```
salesforcemarketingcloud_write = glueContext.write_dynamic_frame.from_options(
    connection_type="salesforcemarketingcloud",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "Campaigns",
        "API_VERSION": "v1",
        "writeOperation" : "INSERT",
        "INSTANCE_URL": "https://**********************.rest.marketingcloudapis.com"
    }
)
```

Esempio di operazione INSERT per SOAP:

```
salesforcemarketingcloud_write = glueContext.write_dynamic_frame.from_options(
    connection_type="salesforcemarketingcloud",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "List",
        "API_VERSION": "v1",
        "writeOperation" : "INSERT",
        "INSTANCE_URL": "https://**********************.rest.marketingcloudapis.com"
    }
)
```

Esempio di operazione UPDATE per REST:

```
salesforcemarketingcloud_write = glueContext.write_dynamic_frame.from_options(
    connection_type="salesforcemarketingcloud",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "Campaigns",
```

```
        "API_VERSION": "v1",
        "writeOperation" : "UPDATE",
         "ID_FIELD_NAMES": "id",
        "INSTANCE_URL": "https://**********.rest.marketingcloudapis.com"
    }
)
```

Esempio di operazione UPDATE per SOAP:

```
salesforcemarketingcloud_write = glueContext.write_dynamic_frame.from_options(
    connection_type="salesforcemarketingcloud",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "List",
        "API_VERSION": "v1",
        "writeOperation" : "UPDATE",
         "ID_FIELD_NAMES": "id",
        "INSTANCE_URL": "https://**********.rest.marketingcloudapis.com"
    }
)
```

Esempio di operazione UPSERT per SOAP:

```
salesforcemarketingcloud_write = glueContext.write_dynamic_frame.from_options(
    connection_type="salesforcemarketingcloud",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "DataExtension/Insert-***E/6*******3",
        "API_VERSION": "v1",
        "writeOperation" : "UPSERT",
        "INSTANCE_URL": "https://**********.rest.marketingcloudapis.com"
    }
)
```

## Opzioni di connessione di Salesforce Marketing Cloud

Le seguenti sono le opzioni di connessione per Salesforce Marketing Cloud:

- `ENTITY_NAME`(String): (obbligatorio) utilizzato per la lettura. Il nome dell'oggetto in Salesforce Marketing Cloud.
- `API_VERSION`(String): (obbligatorio) utilizzato per la lettura. Versione dell'API SOAP e Rest di Salesforce Marketing Cloud da usare.

- `SELECTED_FIELDS`(List<String>): predefinito: vuoto (SELECT *). Utilizzato per la lettura. Colonne da selezionare per l'oggetto.
- `FILTER_PREDICATE`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Dev'essere in formato Spark SQL.
- `QUERY`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Query Spark SQL completa.
- `PARTITION_FIELD`(Stringa): utilizzato per la lettura. Campo da utilizzare per partizionare la query.
- `LOWER_BOUND`(Stringa): utilizzato per la lettura. Un valore limite inferiore inclusivo del campo di partizione scelto.
- `UPPER_BOUND`(Stringa): utilizzato per la lettura. Un valore limite superiore esclusivo del campo di partizione scelto.
- `NUM_PARTITIONS`(Numero intero): predefinito: 1. Utilizzato per la lettura. Numero di partizioni da leggere.
- `WRITE_OPERATION`(String): predefinito: INSERT. Utilizzato per la scrittura. Il valore deve essere INSERT, UPDATE e UPSERT.
- `ID_FIELD_NAMES`(String): predefinito: nullo. Obbligatorio per UPDATE/UPSERT.

## Limitazioni e note per il connettore di Salesforce Marketing Cloud

Di seguito sono riportate le limitazioni o le note per il connettore di Salesforce Marketing Cloud:

- Quando si utilizza il filtro sul campo del DateTime tipo di dati, è necessario passare il valore nel formato "yyyy-mm-ddTHH:mm:SSZ».
- In Anteprima dati, il valore del tipo di dati booleano viene visualizzato come vuoto.
- Per le entità SOAP, è possibile definire un massimo di due filtri e per le entità REST è possibile definire un solo filtro, il che limita il test del partizionamento con i filtri.
- Sono stati osservati diversi comportamenti imprevisti dal lato SaaS: il campo `Link.Alias` nell'entità "linksend" non supporta l'operatore CONTAINS (ad esempio, `Link.Alias CONTAINS "ViewPrivacyPolicy"`) e gli operatori di filtro per le entità Estensione dati (come EQUALS e GREATER THAN) non restituiscono i risultati previsti.
- L'API ClickEvent SOAP SFMC presenta un ritardo nel riflettere i nuovi record creati, pertanto i record creati di recente potrebbero non essere immediatamente disponibili nella risposta dell'API.

  Esempio: se si creano 5 nuovi ClickEvent record alle 14:30:00 del 2025-01-10T e li si recupera immediatamente utilizzando l'API SOAP, la risposta potrebbe non includere tutti e 5 i record.

Potrebbero essere necessari fino a 5 minuti affinché i record appena creati appaiano nella risposta dell'API. Questo ritardo può influire sia sul recupero dei dati che sulle esecuzioni pianificate.

- Due DateTime formati diversi: 2025-03-11T 04:46:00 (senza millisecondi) e 2025-03-11T 04:46:00.000 Z sono supportati quando si eseguono operazioni di scrittura in (con millisecondi). AWS Glue
- Per l'entità Iscrizione alle notifiche di eventi, è possibile creare un'iscrizione solo per un URL di callback verificato e si possono avere fino a 200 iscrizioni per callback.
- Per l'entità Callback di notifiche di eventi, è possibile creare un massimo di 50 record per account.

## Connessione a Salesforce Commerce Cloud

L'API B2C Commerce è una raccolta di istanze RESTful APIs per interagire con le istanze di B2C Commerce. È conosciuta con diversi nomi: come Salesforce Commerce API, con l'acronimo SCAPI o semplicemente come Commerce API.

L'API consente agli sviluppatori di creare un'ampia gamma di applicazioni: da vetrine complete a strumenti personalizzati per aiutare i commercianti a potenziare Business Manager. L'API è disponibile gratuitamente per tutti i clienti di B2C Commerce.

L'API è suddivisa in due gruppi principali di: Shopper e Admin. APIs APIs APIs All'interno di ogni gruppo, sono suddivise in famiglie di API e in gruppi più piccoli basati sulle funzionalità correlate.

Argomenti


- AWS Glue supporto per Salesforce Commerce Cloud
- Policy contenenti le operazioni API per creare e utilizzare le connessioni
- Configurazione di Salesforce Commerce Cloud
- Configurazione delle connessioni di Salesforce Commerce Cloud
- Lettura dalle entità Salesforce Commerce Cloud
- Riferimento per le opzioni di connessione a Salesforce Commerce Cloud
- Limitazioni


### AWS Glue supporto per Salesforce Commerce Cloud

AWS Glue supporta Salesforce Commerce Cloud come segue:

Supportato come origine?

Sì. Puoi utilizzare i job AWS Glue ETL per interrogare i dati da Salesforce Commerce Cloud.

Supportato come destinazione?

No.

Versioni API supportate di Salesforce Commerce Cloud

v1.

## Policy contenenti le operazioni API per creare e utilizzare le connessioni

La seguente politica di esempio descrive le AWS autorizzazioni richieste per la creazione e l'utilizzo delle connessioni. Se si sta creando un nuovo ruolo, creare una policy che contenga quanto segue:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:ListConnectionTypes",
        "glue:DescribeConnectionType",
        "glue:RefreshOAuth2Tokens",
        "glue:ListEntities",
        "glue:DescribeEntity"
      ],
      "Resource": "*"
    }
  ]
}
```

È anche possibile utilizzare le seguenti policy IAM per autorizzare l'accesso:

- AWSGlueServiceRole— Garantisce l'accesso alle risorse che diversi AWS Glue processi richiedono per l'esecuzione per conto dell'utente. Queste risorse includono Amazon S3 AWS Glue, IAM, CloudWatch Logs e Amazon EC2. Se segui la convenzione di denominazione delle risorse

specificata in questa politica, AWS Glue i processi dispongono delle autorizzazioni necessarie. Questa policy è in genere collegata ai ruoli specificati quando si definiscono crawler, processi ed endpoint di sviluppo.

- AWSGlueConsoleFullAccess— Concede l'accesso completo alle AWS Glue risorse quando un'identità a cui è allegata la policy utilizza la AWS console di gestione. Se si segue la convenzione per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questa policy è in genere associata agli utenti della AWS Glue console.

## Configurazione di Salesforce Commerce Cloud

Prima di poter AWS Glue utilizzare il trasferimento di dati da Salesforce Commerce Cloud, devi soddisfare questi requisiti:

Requisiti minimi

- Avere un'applicazione client Salesforce Commerce Cloud con ID client e segreto del client.
- L'account Salesforce Commerce Cloud è abilitato all'accesso tramite API.

Se soddisfi questi requisiti, sei pronto per connetterti AWS Glue al tuo account Salesforce Commerce Cloud. Per le connessioni tipiche, non è necessario fare nient'altro in Salesforce Commerce Cloud.

## Configurazione delle connessioni di Salesforce Commerce Cloud

Salesforce Commerce Cloud supporta il tipo di concessione CLIENT CREDENTIALS per. OAuth2

- Questo tipo di concessione è considerato OAuth 2.0 a 2 gambe in quanto viene utilizzato dai client per ottenere un token di accesso al di fuori del contesto di un utente. AWS Glue è in grado di utilizzare l'ID e il segreto del client per autenticare Salesforce Commerce Cloud APIs , forniti dai servizi personalizzati definiti dall'utente.
- Ogni servizio personalizzato è di proprietà di un utente che utilizza solo API e dispone di una serie di ruoli e autorizzazioni che autorizzano il servizio a eseguire azioni specifiche. Un token di accesso è associato a un singolo servizio personalizzato.
- Questo tipo di concessione determina un token di accesso di breve durata e che può essere rinnovato chiamando l'endpoint dell'identità.
- Per ulteriori informazioni sulla documentazione di Salesforce Commerce Cloud relativa alla generazione delle credenziali del client, consulta la documentazione di Salesforce.

Per configurare una connessione di Salesforce Commerce Cloud:

1. Nel Gestione dei segreti AWS, crea un segreto con i seguenti dettagli. È necessario creare un segreto per ogni connessione in AWS Glue.
   a. Per l'app connessa gestita dal cliente, il segreto deve contenere l'app connessa Consumer Secret con USER_MANAGED_CLIENT_APPLICATION_CLIENT_SECRET come chiave.
2. Nel AWS Glue Studio, crea una connessione in Connessioni dati seguendo i passaggi seguenti:
   a. In Connessioni dati, scegliere Crea connessione.
   b. Quando si seleziona un'origine dati, selezionare Salesforce Commerce Cloud.
   c. Fornire il codice breve, l'ID dell'organizzazione e l'ID del sito di Salesforce Commerce Cloud.
   d. Selezionare l'URL del dominio Salesforce Commerce Cloud dell'account Salesforce Commerce Cloud.
   e. Seleziona il ruolo IAM che AWS Glue può assumere e dispone delle autorizzazioni per le seguenti azioni:

      JSON

      ```
      {
        "Version":"2012-10-17",
        "Statement": [
          {
            "Effect": "Allow",
            "Action": [
              "secretsmanager:DescribeSecret",
              "secretsmanager:GetSecretValue",
              "secretsmanager:PutSecretValue",
              "ec2:CreateNetworkInterface",
              "ec2:DescribeNetworkInterfaces",
              "ec2:DeleteNetworkInterface"
            ],
            "Resource": "*"
          }
        ]
      }
      ```

   f. Fornisci gli OAuth ambiti (opzionale, l'applicazione client gestita dagli utenti) ClientId di Salesforce Commerce Cloud a cui desideri connetterti.
   g. Seleziona quello `secretName` che desideri utilizzare per questa connessione per AWS Glue inserire i token.

h. Selezionare le opzioni di rete se si desidera utilizzare la propria rete.

3. Concedi il permesso di lettura `secretName` al ruolo IAM associato al tuo AWS Glue lavoro.
4. Nella configurazione del tuo AWS Glue lavoro, fornisci `connectionName` una connessione di rete aggiuntiva.

## Lettura dalle entità Salesforce Commerce Cloud

Prerequisiti

- Un oggetto di Salesforce Commerce Cloud da cui si desidera leggere. Consultare la tabella delle entità supportate qui di seguito per verificare quelle disponibili.

Entità supportate

| Entità | Può essere Filtrato | Supporta Limite | Supporta Ordina per | Supporta Seleziona * | Supporta Partizionamento |
|---|---|---|---|---|---|
| Incarichi | Sì | Sì | Sì | Sì | Sì |
| Campagne | Sì | Sì | Sì | Sì | Sì |
| Cataloghi | Sì | Sì | Sì | Sì | Sì |
| Categories | Sì | Sì | Sì | Sì | Sì |
| Coupon | Sì | Sì | Sì | Sì | Sì |
| Certificati regalo | Sì | Sì | Sì | Sì | Sì |
| Prodotti | Sì | Sì | Sì | Sì | Sì |
| Promozioni | Sì | Sì | Sì | Sì | Sì |
| Gruppi di codice sorgente | Sì | Sì | Sì | Sì | Sì |

Esempio

```
salesforce_commerce_cloud_read = glueContext.create_dynamic_frame.from_options(
     connection_type="SalesforceCommerceCloud",
     connection_options={
         "connectionName": "connectionName",
         "ENTITY_NAME": "campaign",
         "API_VERSION": "v1"
     }
)
```

Dettagli dell'entità e del campo di Salesforce Commerce Cloud

Elenco delle entità:

- Incarichi: https://developer.salesforce.com/docs/commerce/commerce-api/references/assignments
- Campagne: https://developer.salesforce.com/docs/commerce/commerce-api/references/campaigns
- Cataloghi: https://developer.salesforce.com/docs/commerce/commerce-api/references/catalogs
- Categorie: https://developer.salesforce.com/docs/commerce/commerce-api/references/catalogs?meta=SearchCategories
- Buoni regalo: -certificati https://developer.salesforce.com/docs/ commerce/commerce-api/references/gift
- Prodotti: https://developer.salesforce.com/docs/commerce/commerce-api/references/products
- Promozioni: https://developer.salesforce.com/docs/commerce/commerce-api/references/promotions
- Gruppi di codice sorgente: https://developer.salesforce.com/docs/commerce/commerce-api/references/source-code-groups

Query di partizionamento

Se si desidera utilizzare la simultaneità in Spark, possono essere fornite opzioni Spark `PARTITION_FIELD`, `LOWER_BOUND`, `UPPER_BOUND`, `NUM_PARTITIONS` aggiuntive. Con questi parametri, la query originale viene suddivisa in `NUM_PARTITIONS` query secondarie che possono essere eseguite contemporaneamente dalle attività Spark.

- `PARTITION_FIELD`: il nome del campo da utilizzare per partizionare la query.
- `LOWER_BOUND`: un valore limite inferiore inclusivo del campo di partizione scelto.

Per la data, accettiamo il formato di data di Spark utilizzato nelle query di Spark SQL. Esempio di valori validi: `"2024-02-06"`.

- `UPPER_BOUND`: un valore limite superiore esclusivo del campo di partizione scelto.
- `NUM_PARTITIONS`: numero di partizioni.

I dettagli di supporto del campo di partizionamento per entità sono riportati nella tabella di seguito:

| Entità | Campo di partizionamento | DataType |
|---|---|---|
| Campagne | lastModified | DateTime |
| Campagne | startDate | DateTime |
| Campagne | endDate | DateTime |
| Cataloghi | creationDate | DateTime |
| Categories | creationDate | DateTime |
| Certificati regalo | merchantId | Stringa |
| Certificati regalo | creationDate | DateTime |
| Prodotti | creationDate | DateTime |
| Prodotti | lastModified | DateTime |
| Gruppi di codice sorgente | creationDate | DateTime |
| Gruppi di codice sorgente | startTime | DateTime |
| Gruppi di codice sorgente | endTime | DateTime |

Esempio

```
salesforceCommerceCloud_read = glueContext.create_dynamic_frame.from_options(
    connection_type="SalesforceCommerceCloud",
    connection_options={
```

```
        "connectionName": "connectionName",
        "ENTITY_NAME": "coupons",
        "API_VERSION": "v1",
        "PARTITION_FIELD": "creationDate",
        "LOWER_BOUND": "2020-05-01T20:55:02.000Z",
        "UPPER_BOUND": "2024-07-11T20:55:02.000Z",
        "NUM_PARTITIONS": "10"
    }
)
```

## Riferimento per le opzioni di connessione a Salesforce Commerce Cloud

Le seguenti sono le opzioni di connessione per Salesforce Commerce Cloud:

- `ENTITY_NAME`(String): (obbligatorio) utilizzato per la lettura. Il nome dell'oggetto in Salesforce Commerce Cloud.
- `API_VERSION`(Stringa): (obbligatorio) utilizzato per lettura/scrittura. Versione dell'API Rest di Salesforce Commerce Cloud da usare. Esempio: v1.
- `SELECTED_FIELDS`(List<String>): predefinito, vuoto (SELECT *). Utilizzato per la lettura. Colonne da selezionare per l'oggetto.
- `FILTER_PREDICATE`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Dev'essere in formato Spark SQL.
- `QUERY`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Query Spark SQL completa.
- `PARTITION_FIELD`(Stringa): utilizzato per la lettura. Campo da utilizzare per partizionare la query.
- `LOWER_BOUND`(Stringa): utilizzato per la lettura. Un valore limite inferiore inclusivo del campo di partizione scelto.
- `UPPER_BOUND`(Stringa): utilizzato per la lettura. Un valore limite superiore esclusivo del campo di partizione scelto.
- `NUM_PARTITIONS`(Numero intero): predefinito: 1. Utilizzato per la lettura. Numero di partizioni da leggere.

## Limitazioni

Di seguito sono riportate le limitazioni per il connettore di Salesforce Commerce Cloud:

- Il filtro Contains non funziona come previsto durante il partizionamento.

- L'entità di CDN Zones non supporta le istanze sandbox e supporta solo i tipi di istanze di sviluppo e produzione. Per ulteriori informazioni, vedere https://help.salesforce.com/s/ArticleView?id=cc.b2c_embedded_cdn_overview.htm.
- In Salesforce Commerce Cloud, non esiste un endpoint API per recuperare i metadati dinamici. Di conseguenza, non è previsto il supporto dei campi personalizzati nell'entità Prodotto e Categoria.
- L'ID del sito è un parametro di query obbligatorio. È necessario passare il valore di ID del sito tramite l'impostazione del connettore personalizzato. Per ulteriori informazioni, consultare URL di base e formazione di richiesta.
- È possibile applicare filtri su un massimo di due campi (esclusi i livelli, se presenti) in una singola richiesta API con la combinazione di diversi operatori come indicato nella tabella seguente:

| Criteri di filtro | È supportato? |
|---|---|
| Un campo con operatore CONTAINS in una singola richiesta API. | Sì |
| Un campo con operatore Equals in una singola richiesta API. | Sì |
| Un campo con operatore BETWEEN in una singola richiesta API. | Sì |
| Due o più campi con operatore CONTAINS in una singola richiesta API. | No |
| Due o più campi con operatore Equals in una singola richiesta API. | No |
| Due o più campi con operatore BETWEEN in una singola richiesta API. | No |
| Un campo con Equals e un campo con operatore CONTAINS in una singola richiesta API. | Sì |
| Un campo con BETWEEN e un campo con operatore CONTAINS in una singola richiesta API. | Sì |

| Criteri di filtro | È supportato? |
| --- | --- |
| Un campo con BETWEEN e un campo con operatore Equals in una singola richiesta API. | Sì |
| Un campo con Equals, un campo con CONTAINS e un campo con operatore BETWEEN in una singola richiesta API. | No |
| Un campo con operatore Equals quando INCREMENTAL PULL viene applicato in una singola richiesta API. | Sì |
| Un campo con operatore CONTAINS quando INCREMENTAL PULL viene applicato in una singola richiesta API. | Sì |
| Un campo con operatore BETWEEN quando INCREMENTAL PULL viene applicato in una singola richiesta API. | No |
| Un campo con operatore Equals e un operatore CONTAINS quando INCREMENTAL PULL viene applicato in una singola richiesta API. | No |

- In alcune entità, il tipo di dati per i campi durante il recupero è diverso da quando vengono utilizzati come campi ricercabili. Di conseguenza, non è disponibile alcuna funzionalità di filtro per questi campi. La tabella seguente fornisce i dettagli su tali campi.

| Sr. No. | Nome entità | Nome campo ricercabile | Tipo di dati come ricercabile | Tipo di dati come recuperabile |
| --- | --- | --- | --- | --- |
| 1 | Catalogo | nome | Stringa | Struct |
| 2 | Catalogo | description | Stringa | Struct |
| 3 | Categoria | nome | Stringa | Struct |

| Sr. No. | Nome entità | Nome campo ricercabile | Tipo di dati come ricercabile | Tipo di dati come recuperabile |
|---|---|---|---|---|
| 4 | Categoria | description | Stringa | Struct |
| 5 | Prodotto | nome | Stringa | Struct |
| 6 | Prodotto | searchable | Booleano | Struct |
| 7 | Prodotto | validFrom | DateTime | Struct |
| 8 | Prodotto | validTo | DateTime | Struct |
| 9 | Prodotto | tipo | Stringa | Struct |
| 10 | Prodotto | onlineFlag | Booleano | Struct |
| 11 | Promotion | nome | Stringa | Struct |

## Connessione a Salesforce Marketing Cloud Account Engagement

Salesforce Marketing Cloud Account Engagement è una soluzione di automazione del marketing che aiuta le aziende a creare connessioni significative, generare più pipeline e consentire al reparto vendite di concludere più trattative. Se sei un utente di Salesforce Marketing Cloud Account Engagement, puoi connetterti AWS Glue al tuo account Salesforce Marketing Cloud Account Engagement. Puoi utilizzare Salesforce Marketing Cloud Account Engagement come origine dati per i tuoi processi ETL. Esegui questi processi per trasferire i dati da Salesforce Marketing Cloud Account Engagement ai servizi o ad AWS altre applicazioni supportate.

Argomenti

- AWS Glue supporto per Salesforce Marketing Cloud Account Engagement
- Policy contenenti le operazioni API per creare e utilizzare le connessioni
- Configurazione di Salesforce Marketing Cloud Account Engagement
- Configurazione delle connessioni di Salesforce Marketing Cloud Account Engagement
- Lettura dalle entità Salesforce Marketing Cloud Account Engagement
- Opzioni di connessione di Salesforce Marketing Cloud Account Engagement

- Limitazioni e note per il connettore di Salesforce Marketing Cloud Account Engagement

## AWS Glue supporto per Salesforce Marketing Cloud Account Engagement

AWS Glue supporta Salesforce Marketing Cloud Account Engagement come segue:

Supportato come origine?

Sì. È possibile utilizzare i job AWS Glue ETL per interrogare i dati da Salesforce Marketing Cloud Account Engagement in modalità asincrona o sincronizzata.

Supportato come destinazione?

No.

Versioni API supportate di Salesforce Marketing Cloud Account Engagement

Le seguenti versioni dell'API Salesforce Marketing Cloud Account Engagement sono supportate:

- v5

## Policy contenenti le operazioni API per creare e utilizzare le connessioni

La seguente policy di esempio descrive le autorizzazioni AWS IAM richieste per la creazione e l'utilizzo delle connessioni. Se si sta creando un nuovo ruolo, creare una policy che contenga quanto segue:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:ListConnectionTypes",
        "glue:DescribeConnectionType",
        "glue:RefreshOAuth2Tokens",
        "glue:ListEntities",
        "glue:DescribeEntity"
```

```
      ],
      "Resource": "*"
    }
  ]
}
```

In alternativa, se non desideri utilizzare il metodo precedente, utilizza le seguenti policy IAM gestite:

- AWSGlueServiceRole— Garantisce l'accesso alle risorse che AWS Glue i vari processi richiedono per l'esecuzione per conto dell'utente. Queste risorse includono AWS Glue Amazon S3, IAM, CloudWatch Logs e Amazon. EC2 Se segui la convenzione di denominazione delle risorse specificata in questa politica, AWS Glue i processi dispongono delle autorizzazioni richieste. Questa policy è in genere collegata ai ruoli specificati quando si definiscono crawler, processi ed endpoint di sviluppo.
- AWSGlueConsoleFullAccess— Concede l'accesso completo alle AWS Glue risorse quando un'identità a cui è allegata la policy utilizza la AWS console di gestione. Se si segue la convenzione per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questa policy è in genere associata agli utenti della AWS Glue console.

## Configurazione di Salesforce Marketing Cloud Account Engagement

Prima di poter AWS Glue utilizzare il trasferimento dei dati da Salesforce Marketing Cloud Account Engagement, devi soddisfare questi requisiti:

Requisiti minimi

I requisiti minimi sono i seguenti:

- Avere un account Salesforce Marketing.
- Avere un piano Account Engagement con licenza per l'account Salesforce.
- Aver sincronizzato l'utente Salesforce con l'utente Account Engagement.
- Hai creato una nuova app connessa in App Manager per ottenere OAuth le credenziali.

Se soddisfi questi requisiti, sei pronto per connetterti AWS Glue al tuo account Salesforce Marketing Cloud Account Engagement.

## Configurazione delle connessioni di Salesforce Marketing Cloud Account Engagement

Il tipo di concessione determina il modo in cui AWS Glue comunica con Salesforce Marketing Cloud Account Engagement per richiedere l'accesso ai dati. La scelta influisce sui requisiti da soddisfare prima di creare la connessione. Salesforce Marketing Cloud Account Engagement supporta solo il tipo di concessione AUTHORIZATION_CODE per la versione 2.0. OAuth

- Questo tipo di concessione è considerato «a tre gambe» in OAuth quanto si basa sul reindirizzamento degli utenti a un server di autorizzazione di terze parti per autenticare l'utente. Viene utilizzato durante la creazione di connessioni tramite la console. AWS Glue
- Gli utenti possono comunque scegliere di creare la propria app connessa in Salesforce Marketing Cloud Account Engagement e fornire il proprio ID cliente e il segreto del cliente durante la creazione di connessioni tramite la AWS Glue console. In questo scenario, verranno comunque reindirizzati a Salesforce Marketing Cloud Account Engagement per accedere e AWS Glue autorizzare l'accesso alle proprie risorse.
- Questo tipo di concessione genera un token di aggiornamento e un token di accesso. Il token di accesso dura poco tempo e può essere aggiornato automaticamente senza l'interazione dell'utente utilizzando il token di aggiornamento.
- Per la documentazione pubblica di Salesforce Marketing Cloud Account Engagement sulla creazione di un'app connessa per il flusso del codice di autorizzazione, consulta Autenticazione. OAuth

Per configurare una connessione di Salesforce Marketing Cloud Account Engagement:

1. In AWS Glue Glue Studio, crea una connessione in Connessioni dati seguendo i passaggi seguenti:
   a. Quando si seleziona un tipo di connessione, selezionare Salesforce Marketing Cloud Account Engagement.
   b. Fornire il `INSTANCE_URL` dell'istanza Salesforce Marketing Cloud Account Engagement a cui ci si desidera connettersi.
   c. Fornire il `PARDOT_BUSINESS_UNIT_ID` dell'istanza Salesforce Marketing Cloud Account Engagement a cui ci si desidera connettersi.
   d. Selezionare l'URL del codice di autorizzazione appropriato dal menu a discesa.
   e. Selezionare l'URL del token appropriato dal menu a discesa.

f. Seleziona il ruolo AWS IAM che AWS Glue può assumere e dispone delle autorizzazioni per le seguenti azioni:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:DescribeSecret",
        "secretsmanager:GetSecretValue",
        "secretsmanager:PutSecretValue",
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DeleteNetworkInterface"
      ],
      "Resource": "*"
    }
  ]
}
```

g. Fornire l'ID client dell'applicazione client gestita dall'utente (l'ID client dall'app connessa).

h. Seleziona quello `secretName` che desideri utilizzare per questa connessione per AWS Glue inserire i token. Il segreto selezionato deve avere una chiave `USER_MANAGED_CLIENT_APPLICATION_CLIENT_SECRET` il cui valore sia il segreto client dell'app connessa.

i. Seleziona le opzioni di rete se desideri utilizzare la tua rete.

2. Concedi il permesso di lettura `secretName` al ruolo IAM associato al tuo AWS Glue lavoro.
3. Nella configurazione del AWS Glue lavoro, fornisci `connectionName` una connessione di rete aggiuntiva.

## Lettura dalle entità Salesforce Marketing Cloud Account Engagement

Prerequisito

Un oggetto di Salesforce Marketing Cloud Account Engagement da cui si desidera leggere. È necessario il nome dell'oggetto.

Entità supportate come origine Sync:

| Entità | Può essere filtrato | Supporta limite | Supporta Ordina per | Supporta Seleziona * | Supporta il partizionamento |
|---|---|---|---|---|---|
| Campagna | Sì | Sì | Sì | Sì | Sì |
| Contenuti dinamici | Sì | Sì | Sì | Sì | Sì |
| Email | Sì | Sì | Sì | Sì | Sì |
| Modello e-mail | Sì | Sì | Sì | Sì | Sì |
| Engagement Studio Program | Sì | Sì | Sì | Sì | Sì |
| Contenuti della cartella | Sì | Sì | Sì | Sì | Sì |
| Pagina di destinazione | Sì | Sì | Sì | Sì | Sì |
| Cronologia del ciclo di vita | Sì | Sì | Sì | Sì | Sì |
| Stadio del ciclo di vita | Sì | Sì | Sì | Sì | Sì |
| List | Sì | Sì | Sì | Sì | Sì |
| Elenco e-mail | Sì | Sì | Sì | Sì | Sì |
| Elenco iscrizioni | Sì | Sì | Sì | Sì | Sì |

| Entità | Può essere filtrato | Supporta limite | Supporta Ordina per | Supporta Seleziona * | Supporta il partizionamento |
|---|---|---|---|---|---|
| Opportunità | Sì | Sì | Sì | Sì | Sì |
| Prospect | Sì | Sì | Sì | Sì | Sì |
| Account Prospect | Sì | Sì | Sì | Sì | Sì |
| Utente | Sì | Sì | Sì | Sì | Sì |

Esempio:

```
salesforcepardot_read = glueContext.create_dynamic_frame.from_options(
    connection_type="SalesforcePardot",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "entityName",
        "API_VERSION": "v5"
    }
   )
```

Entità supportate come origine Async:

| Entità | Può essere filtrato | Supporta limite | Supporta Ordina per | Supporta Seleziona * | Supporta il partizionamento |
|---|---|---|---|---|---|
| Campagna | Sì | No | No | Sì | No |
| Contenuti dinamici | Sì | No | No | Sì | No |
| Modello e-mail | Sì | No | No | Sì | No |

| Entità | Può essere filtrato | Supporta limite | Supporta Ordina per | Supporta Seleziona * | Supporta il partizionamento |
|---|---|---|---|---|---|
| Pagina di destinazione | Sì | No | No | Sì | No |
| Cronologia del ciclo di vita | Sì | No | No | Sì | No |
| Stadio del ciclo di vita | Sì | No | No | Sì | No |
| List | Sì | No | No | Sì | No |
| Elenco e-mail | Sì | No | No | Sì | No |
| Elenco iscrizioni | Sì | No | No | Sì | No |
| Opportunità | Sì | No | No | Sì | No |
| Prospect | Sì | No | No | Sì | No |
| Account Prospect | Sì | No | No | Sì | No |
| Utente | Sì | No | No | Sì | No |

Esempio:

```
salesforcepardot_read = glueContext.create_dynamic_frame.from_options(
    connection_type="SalesforcePardot",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "entityName",
        "API_VERSION": "v5",
        "TRANSFER_MODE": "ASYNC"
    }
```

```
)
```

Dettagli dell'entità e del campo di Salesforce Marketing Cloud Account Engagement:

Per visualizzare i dettagli dei campi per le seguenti entità, accedere all'API Salesforce Marketing Cloud Account Engagement, scegliere Guide, scorrere verso il basso fino a Wrapper API Open Source, espandere la versione 5 Docs dal menu e scegliere un'entità.

Elenco delle entità:

- Campagna
- Contenuti dinamici
- Email
- Modello e-mail
- Engagement Studio Program
- Contenuto della cartella
- Pagina di destinazione
- Cronologia del ciclo di vita
- Stadio del ciclo di vita
- List
- Elenco e-mail
- Elenco iscrizioni
- Opportunità
- Prospect
- Account Prospect
- Utente

Oltre ai campi sopra menzionati, la modalità asincrona supporta campi filtrabili specifici per ciascuna entità, come mostrato nella tabella seguente.

| Entità | Campi filtrabili aggiuntivi supportati in modalità asincrona |
| --- | --- |
| Campagna | `createdAfter` , `createdBefore` , `deleted`, `updatedAfter` , `updatedBefore` |
| Contenuti dinamici | `createdAfter` , `createdBefore` , `deleted`, `updatedAfter` , `updatedBefore` |
| Modello e-mail | `createdAfter` , `createdBefore` , `deleted`, `updatedAfter` , `updatedBefore` |
| Engagement Studio Program | - |
| Pagina di destinazione | `createdAfter` , `createdBefore` , `deleted`, `updatedAfter` , `updatedBefore` |
| Cronologia del ciclo di vita | `createdAfter` , `createdBefore` |
| Stadio del ciclo di vita | `createdAfter` , `createdBefore` , `deleted`, `updatedAfter` , `updatedBefore` |
| List | `createdAfter` , `createdBefore` , `deleted`, `updatedAfter` , `updatedBefore` |
| Elenco e-mail | `createdAfter` , `createdBefore` , `deleted`, `updatedAfter` , `updatedBefore` |
| Elenco iscrizioni | `createdAfter` , `createdBefore` , `deleted`, `updatedAfter` , `updatedBefore` |

| Entità | Campi filtrabili aggiuntivi supportati in modalità asincrona |
| --- | --- |
| Opportunità | `createdAfter` , `createdBefore` , `deleted`, `updatedAfter` , `updatedBefore` |
| Prospect | `createdAfter` , `createdBefore` , `deleted`, `updatedAfter` , `updatedBefore` |
| Account Prospect | `createdAfter` , `createdBefore` , `deleted` |
| Utente | `createdAfter` , `createdBefore` , `deleted`, `updatedAfter` , `updatedBefore` |

Per ulteriori informazioni sui campi aggiuntivi, consultare l'API Salesforce Export

Prendere nota di quanto segue per il connettore:

- Il valore del campo `delete` nelle entità può essere `false` (impostazione predefinita), `true` o `all`.

Query di partizionamento

Partizionamento basato su filtri:

È possibile fornire le opzioni Spark aggiuntive `PARTITION_FIELD`, `LOWER_BOUND`, `UPPER_BOUND` e `NUM_PARTITIONS` se si desidera utilizzare la simultaneità in Spark. Con questi parametri, la query originale viene suddivisa in `NUM_PARTITIONS` query secondarie che possono essere eseguite contemporaneamente dalle attività Spark.

- `PARTITION_FIELD`: il nome del campo da utilizzare per partizionare la query.
- `LOWER_BOUND`: un valore limite inferiore inclusivo del campo di partizione scelto.

  Per il campo Datetime, accettiamo il formato timestamp Spark utilizzato nelle query SQL. SPark

  Esempi di valori validi:

```
"2022-01-01T01:01:01.000Z"
```

- UPPER_BOUND: un valore limite superiore esclusivo del campo di partizione scelto.
- NUM_PARTITIONS: il numero di partizioni.
- PARTITION_BY: il tipo di partizionamento da eseguire. "FIELD" deve essere passato in caso di partizionamento basato sul campo.

Esempio:

```
salesforcepardot_read = glueContext.create_dynamic_frame.from_options(
    connection_type="salesforcepardot",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "entityName",
        "API_VERSION": "v5",
        "PARTITION_FIELD": "createdAt"
        "LOWER_BOUND": "2022-01-01T01:01:01.000Z"
        "UPPER_BOUND": "2024-01-01T01:01:01.000Z"
        "NUM_PARTITIONS": "10",
        "PARTITION_BY": "FIELD"
    }
   )
```

## Opzioni di connessione di Salesforce Marketing Cloud Account Engagement

Le seguenti sono le opzioni di connessione per Salesforce Marketing Cloud Account Engagement:

- ENTITY_NAME(String): (obbligatorio) utilizzato per la lettura. Il nome dell'oggetto in Salesforce Marketing Cloud Account Engagement.
- PARDOT_BUSINESS_UNIT_ID: (obbligatorio) utilizzato per creare una connessione. L'ID dell'unità aziendale dell'istanza di Salesforce Marketing Cloud Account Engagement a cui ci si desidera connettersi.
- API_VERSION(String): (obbligatorio) utilizzato per la lettura. La versione dell'API Rest di Salesforce Marketing Cloud Account Engagement che si desidera utilizzare.
- SELECTED_FIELDS(List<String>): predefinito: vuoto (SELECT *). Utilizzato per la lettura. Colonne da selezionare per l'oggetto.
- FILTER_PREDICATE(String) -

  - In modalità sincrona, predefinito, vuoto. Utilizzato per la lettura. Deve essere in formato Spark SQL.
  - In modalità asincrona: predefinito: valore corrente `DateTime` (in base al fuso orario dell'utente): 1 anno. Utilizzato per la lettura.
- `QUERY`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Query Spark SQL completa.
- `PARTITION_FIELD`(Stringa): utilizzato per la lettura. Campo da utilizzare per partizionare la query.
- `LOWER_BOUND`(Stringa): utilizzato per la lettura. Un valore limite inferiore inclusivo del campo di partizione scelto.
- `UPPER_BOUND`(Stringa): utilizzato per la lettura. Un valore limite superiore esclusivo del campo di partizione scelto.
- `NUM_PARTITIONS`(Numero intero): predefinito: 1. Utilizzato per la lettura. Numero di partizioni da leggere.
- `INSTANCE_URL`(Stringa): (obbligatorio) utilizzato per la lettura. Un URL valido dell'istanza di Salesforce Marketing Cloud Account Engagement.
- `PARTITION_BY`(String): (obbligatorio) utilizzato per la lettura. Il tipo di partizionamento da eseguire. “FIELD” deve essere passato in caso di partizionamento basato sul campo.
- `TRANSFER_MODE`(String): (facoltativo), valore da utilizzare per eseguire un processo in modalità ASYNC, se questa opzione non è fornita, il processo verrà eseguito in modalità SYNC.

## Limitazioni e note per il connettore di Salesforce Marketing Cloud Account Engagement

Si applicano le seguenti note e limitazioni:

- Quando vengono applicati sia un limite che il partizionamento, il limite ha la precedenza sul partizionamento.
- Secondo i documenti API, ha `SalesforceMarketingCloudEngagement` applicato un alle richieste RateLimit giornaliere e simultanee. Per ulteriori informazioni, consultare Limiti di frequenza.
- L'API Export è soggetta al limite giornaliero di chiamate dell'API Account Engagement e al limite di chiamate simultanee dell'API Account Engagement per l'account.
- Analogamente a una coda, le chiamate Export/Async API vengono eseguite in sequenza per ogni account. Le esportazioni meno recenti vengono elaborate prima delle esportazioni più recenti.
- La partizione non è supportata in modalità asincrona.

- Il numero di campi selezionati specificati nelle chiamate Export/Async API non può superare 150.
- L'entità Prospect supporta oltre 150 campi, ma è possibile selezionare solo 150 campi alla volta. Se viene selezionato `Select All`, alcuni campi verranno esclusi. Per recuperare i dati per questi campi esclusi, è necessario includerli nell'opzione `Selected Fields`.

  Di seguito è riportato l'elenco dei campi esclusi in `SELECT_ALL`: `updatedBy.firstName`, `updatedBy.lastName`, `updatedBy.jobTitle`, `updatedBy.roleName`, `updatedBy.salesforceId`, `updatedBy.createdAt`, `updatedBy.updatedAt`, `updatedBy.isDeleted`, `updatedBy.createdById`, `updatedBy.updatedById`, `updatedBy.tagReplacementLanguage`
- I campi di raccolta non possono essere esportati per la modalità asincrona. Ad esempio, in Elenco e-mail, i campi `senderOptions` e `replyToOptions` and non sono supportati.
- Per tutte le entità, il filtro è obbligatorio. Se non viene fornito alcun filtro, il predicato di filtro predefinito viene impostato sul campo `Created After` con un valore di data-ora attuale (adattata al fuso orario) meno un anno.
- In base alle limitazioni di Salesforce Marketing Cloud Account Engagement, in modalità asincrona, l'intervallo massimo per il recupero dei dati è di 1 anno. Se viene fornita una query per più di un anno, il processo genererà un errore.
- Attualmente, è presente un bug in Salesforce Pardot. Quando il lavoro include solo un singolo campo che non contiene dati, il valore del campo non restituisce il risultato corretto e, invece, il nome del campo viene restituito più volte. Il team di Salesforce Pardot è a conoscenza del problema e sta lavorando attivamente per risolverlo.

## Connessione a SAP HANA in AWS Glue Studio

AWS Glue fornisce il supporto integrato per SAP HANA. AWS Glue Studio fornisce un'interfaccia visiva per connettersi a SAP HANA, creare processi di integrazione dei dati ed eseguirli sul runtime Spark serverless AWS Glue Studio.

AWS Glue Studio crea una connessione unificata per SAP HANA. Per ulteriori informazioni, consulta Considerazioni.

Argomenti

- Creazione di una connessione SAP HANA
- Creazione di un nodo di origine SAP HANA
- Creazione di un nodo di destinazione SAP HANA

- Opzioni avanzate

## Creazione di una connessione SAP HANA

Per connetterti a SAP HANA da AWS Glue, dovrai creare e archiviare le tue credenziali SAP HANA in un luogo Gestione dei segreti AWS segreto, quindi associare quel segreto a una connessione SAP HANA. AWS Glue Dovrai configurare la connettività di rete tra il tuo servizio SAP HANA e AWS Glue.

Prerequisiti:

- Se il tuo servizio SAP HANA si trova in un Amazon VPC, configura Amazon VPC per consentire al tuo AWS Glue lavoro di comunicare con il servizio SAP HANA senza che il traffico attraversi la rete Internet pubblica.

  In Amazon VPC, identifica o crea un VPC, una sottorete e un gruppo di sicurezza da utilizzare durante l'esecuzione del AWS Glue lavoro. Inoltre, assicurati che Amazon VPC sia configurato per consentire il traffico di rete tra l'endpoint SAP HANA e questa posizione. Il tuo processo dovrà stabilire una connessione TCP con la tua porta SAP HANA JDBC. Per ulteriori informazioni sulle porte SAP HANA, consulta la documentazione SAP HANA. In base al layout della rete, ciò potrebbe richiedere modifiche alle regole del gruppo di sicurezza, alla rete ACLs, ai gateway NAT e alle connessioni peering.

Per configurare una connessione a SAP HANA:

1. Nel Gestione dei segreti AWS, crea un segreto utilizzando le tue credenziali SAP HANA. Per creare un segreto in Secrets Manager, segui il tutorial disponibile in Crea un Gestione dei segreti AWS segreto nella Gestione dei segreti AWS documentazione. Dopo aver creato il segreto, mantieni il nome del segreto *`secretName`* per il passaggio successivo.

   - Quando selezionate le coppie chiave/valore, create una coppia per la chiave `username/USERNAME` con il valore. *`saphanaUsername`*
   - Quando selezionate le coppie chiave/valore, create una coppia per la chiave `password/PASSWORD` con il valore. *`saphanaPassword`*
2. Nella AWS Glue console, crea una connessione seguendo la procedura riportata di seguito. the section called "Aggiungere una AWS Glue connessione" Dopo aver creato la connessione, conserva il nome della connessione*`connectionName`*, per utilizzi futuri in AWS Glue.

   - In Tipo di connessione, seleziona SAP HANA.

- Quando fornisci l'URL SAP HANA, fornisci l'URL per la tua istanza.

  SAP HANA JDBC URLs sono nel formato `jdbc:sap://saphanaHostname:saphanaPort/?databaseName=saphanaDBname,Paramete`

  AWS Glue richiede i seguenti parametri URL JDBC:

  - `databaseName`: un database predefinito in SAP HANA a cui connettersi.
- Quando selezioni un AWS segreto, fornisci. *`secretName`*

Dopo aver creato una connessione AWS Glue SAP HANA, dovrai eseguire i seguenti passaggi prima di eseguire il AWS Glue processo:

- Concedi il permesso di lettura al ruolo IAM associato al tuo AWS Glue lavoro. *`secretName`*

## Creazione di un nodo di origine SAP HANA

Prerequisiti necessari

- Una connessione AWS Glue SAP HANA, configurata con un Gestione dei segreti AWS segreto, come descritto nella sezione precedente,. the section called "Creazione di una connessione SAP HANA"
- Autorizzazioni appropriate sul processo per leggere il segreto utilizzato dalla connessione.
- Una tabella SAP HANA da leggere o interrogare. *`tableName targetQuery`*

  Una tabella può essere specificata con un nome di tabella SAP HANA e di schema, nel modulo *`schemaName.tableName`*. Il nome dello schema e il separatore "." non sono necessari se la tabella si trova nello schema predefinito, "pubblico". Chiama questo. *`tableIdentifier`* Il database viene fornito come parametro URL JDBC in `connectionName`.

Aggiunta di un'origine dati SAP HANA

Per aggiungere un nodo origine dati: SAP HANA:

1. Scegli la connessione per la tua origine dati SAP HANA. Dato che l'hai creato, dovrebbe essere disponibile nel menu a discesa. Se devi creare una connessione, scegli Crea connessione SAP HANA. Per ulteriori informazioni, consulta la sezione the section called "Creazione di una connessione SAP HANA" precedente.

Dopo aver scelto una connessione, puoi visualizzare le proprietà della connessione facendo clic su Visualizza proprietà.

2. Scegli un'opzione per il campo origine SAP HANA:

   - Scegli una singola tabella: accedi a tutti i dati da un'unica tabella.
   - Inserisci una query personalizzata: accedi a un set di dati da più tabelle in base alla tua query personalizzata.

3. Se hai scelto una sola tabella, inserisci*`tableName`*.

   Se hai scelto Inserisci una query personalizzata, inserisci una query SQL SELECT.

4. In Proprietà personalizzate di SAP HANA, inserisci i parametri e i valori necessari.

## Creazione di un nodo di destinazione SAP HANA

Prerequisiti necessari

- Una connessione AWS Glue SAP HANA, configurata con un Gestione dei segreti AWS segreto, come descritto nella sezione precedente,. the section called "Creazione di una connessione SAP HANA"
- Autorizzazioni appropriate sul processo per leggere il segreto utilizzato dalla connessione.
- Una tabella SAP HANA su cui scrivere,. *`tableName`*

  Una tabella può essere specificata con un nome di tabella SAP HANA e di schema, nel modulo *`schemaName`*`.`*`tableName`*. Il nome dello schema e il separatore "." non sono necessari se la tabella si trova nello schema predefinito, "pubblico". Chiama questo. *`tableIdentifier`* Il database viene fornito come parametro URL JDBC in `connectionName`.

Aggiunta di una destinazione di dati SAP HANA

Per aggiungere un nodo destinazione dati: SAP HANA:

1. Scegli la connessione per la tua origine dati SAP HANA. Dato che l'hai creato, dovrebbe essere disponibile nel menu a discesa. Se devi creare una connessione, scegli Crea connessione SAP HANA. Per ulteriori informazioni, consulta la sezione the section called "Creazione di una connessione SAP HANA" precedente.

Dopo aver scelto una connessione, puoi visualizzare le proprietà della connessione facendo clic su Visualizza proprietà.

2. Configura il nome della tabella fornendo*tableName*.
3. In Proprietà personalizzate di Teradata, inserisci i parametri e i valori necessari.

### Opzioni avanzate

È possibile fornire opzioni avanzate durante la creazione di un nodo SAP HANA. Queste opzioni sono le stesse disponibili durante la programmazione AWS Glue per gli script Spark.

Per informazioni, consulta the section called "Connessioni SAP HANA".

## Connessione a SAP OData

SAP OData è un protocollo Web standard utilizzato per interrogare e aggiornare i dati presenti in SAP utilizzando ABAP (Advanced Business Application Programming), applicando e sviluppando tecnologie Web come HTTP per fornire l'accesso alle informazioni da una varietà di applicazioni, piattaforme e dispositivi esterni. Questo prodotto ti consente di accedere a tutto ciò che ti serve per integrare perfettamente il tuo sistema, la tua applicazione o i tuoi dati SAP.

Argomenti


- AWS Glue supporto per SAP OData
- Creare connessioni
- Creazione di un lavoro SAP OData
- Scrivere su SAP OData
- Utilizzo dello script di gestione OData dello stato SAP
- Partizionamento per entità non ODP
- Opzioni di OData connessione SAP
- Dettagli OData dell'entità e del campo SAP


### AWS Glue supporto per SAP OData

AWS Glue supporta SAP OData come segue:

Supportato come origine?

Sì. È possibile utilizzare i job AWS Glue ETL per interrogare i dati da SAP. OData

Supportato come destinazione?

Sì. È possibile utilizzare i processi AWS Glue ETL per scrivere record in SAP. OData

Versioni API OData SAP supportate

Sono supportate le seguenti versioni OData dell'API SAP:

- 2.0

Origini supportate

Sono supportate le seguenti origini:

- Origini ODP (Operational Data Provisioning):
  - Estrattori BW () DataSources
  - Viste CDS
  - SLT
- Origini non ODP, ad esempio:
  - Servizi delle viste CDS
  - Servizi basati su RFC
  - Servizi ABAP personalizzati

Componenti SAP supportati

I requisiti minimi sono i seguenti:

- È necessario abilitare il servizio di catalogo per il rilevamento dei servizi.
  - Configurare le origini dati ODP (Operational Data Provisioning) per l'estrazione nel gateway SAP del sistema SAP.
  - OData V2.0: abilita i servizi di catalogo OData V2.0 nel tuo gateway SAP tramite transazione. `/IWFND/MAINT_SERVICE`
  - Abilita i servizi OData V2.0 nel tuo gateway SAP tramite transazione. `/IWFND/MAINT_SERVICE`

  - Il tuo OData servizio SAP deve supportare pagination/query opzioni lato client come e. `$top` `$skip` Deve inoltre supportare l'opzione di query del sistema `$count`.
  - È necessario fornire l'autorizzazione richiesta all'utente in SAP per scoprire i servizi ed estrarre i dati utilizzando i servizi OData SAP. Consultare la documentazione di sicurezza fornita da SAP.
- Se si desidera utilizzare la OAuth versione 2.0 come meccanismo di autorizzazione, è necessario abilitare la OAuth versione 2.0 per il OData servizio e registrare il OAuth client in base alla documentazione SAP.
- Per generare un OData servizio basato su fonti di dati ODP, SAP Gateway Foundation deve essere installato localmente nello ERP/BW stack o in una configurazione hub.
  - Per ERP/BW le tue applicazioni, lo stack SAP NetWeaver AS ABAP deve essere pari o superiore a 7,50 SP02.
  - Per il sistema hub (SAP Gateway), il SAP NetWeaver AS ABAP del sistema hub deve essere 7.50 SP01 o superiore per la configurazione dell'hub remoto.
- Per le fonti non ODP, la versione NetWeaver dello stack SAP deve essere 7.40 SP02 o superiore.

Metodi di autenticazione supportati

Sono supportati i seguenti metodi di autenticazione:

- Autenticazione di base
- OAuth 2.0

Prerequisiti

Prima di iniziare un AWS Glue processo per l'estrazione dei dati da SAP OData utilizzando la OData connessione SAP, completare i seguenti prerequisiti:

- Il OData servizio SAP pertinente deve essere attivato nel sistema SAP, assicurando che la fonte di dati sia disponibile per il consumo. Se il OData servizio non è attivato, il job Glue non sarà in grado di accedere o estrarre dati da SAP.
- È necessario configurare in SAP meccanismi di autenticazione appropriati come l'autenticazione di base (personalizzata) o OAuth 2.0 per garantire che il AWS Glue job possa stabilire correttamente una connessione con il servizio OData SAP.
- Configura le policy IAM per concedere al AWS Glue lavoro le autorizzazioni appropriate per accedere a SAP, Secrets Manager e ad altre AWS risorse coinvolte nel processo.

- Se il sistema SAP è ospitato all'interno di una rete privata, la connettività VPC deve essere configurata per garantire che AWS Glue il lavoro possa comunicare in modo sicuro con SAP senza esporre dati sensibili su Internet pubblico.

AWS Secrets Manager può essere utilizzato per archiviare in modo sicuro informazioni sensibili come le credenziali SAP, che il AWS Glue job può recuperare dinamicamente in fase di esecuzione. Questo approccio elimina la necessità di codificare le credenziali, migliorando la sicurezza e la flessibilità.

I seguenti prerequisiti forniscono step-by-step indicazioni su come configurare ogni componente per un'integrazione fluida tra e SAP. AWS Glue OData

Argomenti

- Attivazione SAP OData
- Policy IAM
- Connettività/Connessione VPC
- Autenticazione SAP
- AWS Secrets Manager per memorizzare il tuo segreto di autenticazione

Attivazione SAP OData

Completa i seguenti passaggi per la connessione SAP: OData

Origini ODP

Prima di poter eseguire il trasferimento da un provider ODP, è necessario soddisfare i seguenti requisiti:

- Hai un'istanza SAP NetWeaver AS ABAP.
- L' NetWeaver istanza SAP contiene un provider ODP da cui desideri trasferire i dati. I provider ODP includono:
  - SAP DataSources (codice di transazione) RSO2
  - Viste CDS ABAP di SAP Core Data Services
  - Sistemi SAP BW o SAP BW/4HANA (, Oggetto) InfoObject DataStore
  - Replica in tempo reale di tabelle e viste DB dal sistema di origine SAP tramite SAP Landscape Replication Server (SAP SLT)

  - Viste di informazioni SAP HANA in origini basate su SAP ABAP
- L'istanza SAP ha il componente NetWeaver SAP Gateway Foundation.
- Hai creato un OData servizio che estrae i dati dal tuo provider ODP. Per creare il OData servizio, si utilizza SAP Gateway Service Builder. Per accedere ai tuoi dati ODP, Amazon AppFlow chiama questo servizio utilizzando l' OData API. Per ulteriori informazioni, consulta Generazione di un servizio per l'estrazione di dati ODP OData nella documentazione SAP BW/4HANA.
- Per generare un OData servizio basato su fonti di dati ODP, SAP Gateway Foundation deve essere installato localmente nello stack o in una configurazione hub. ERP/BW
  - Per ERP/BW le tue applicazioni, lo stack SAP NetWeaver AS ABAP deve essere pari o superiore a 7,50 SP02.
  - Per il sistema hub (SAP Gateway), il SAP NetWeaver AS ABAP del sistema hub deve essere 7.50 SP01 o superiore per la configurazione dell'hub remoto.

Origini non ODP

- La versione NetWeaver dello stack SAP deve essere 7.40 SP02 o superiore.
- È necessario abilitare il servizio di catalogo per il rilevamento dei servizi.
  - OData V2.0: I servizi di catalogo OData V2.0 possono essere abilitati nel gateway SAP tramite transazione `/IWFND/MAINT_SERVICE`
- Il OData servizio SAP deve supportare pagination/query opzioni lato client come e. `$top` `$skip` Deve inoltre supportare l'opzione di query del sistema `$count`.
- Per la OAuth versione 2.0, è necessario abilitare la OAuth versione 2.0 per il OData servizio e registrare il OAuth client in base alla documentazione SAP e impostare l'URL di reindirizzamento autorizzato come segue:
  - `https://<region>.console.aws.amazon.com/gluestudio/oauth`, sostituendo `<region>` con la regione in cui AWS Glue è in esecuzione, esempio: us-east-1.
  - È necessario abilitare la configurazione sicura per la connessione tramite HTTPS.
- È necessario fornire l'autorizzazione richiesta all'utente in SAP per scoprire i servizi ed estrarre i dati utilizzando i servizi SAP. OData Consultare la documentazione di sicurezza fornita da SAP.

Policy IAM

Policy contenenti le operazioni API per creare e utilizzare le connessioni

La seguente policy di esempio descrive le autorizzazioni AWS IAM richieste per la creazione e l'utilizzo delle connessioni. Se si sta creando un nuovo ruolo, creare una policy che contenga quanto segue:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:ListConnectionTypes",
        "glue:DescribeConnectionType",
        "glue:RefreshOAuth2Tokens",
        "glue:ListEntities",
        "glue:DescribeEntity"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:DescribeSecret",
        "secretsmanager:GetSecretValue",
        "secretsmanager:PutSecretValue"
      ],
      "Resource": "*"
    }
  ]
}
```

Il ruolo deve concedere l'accesso a tutte le risorse utilizzate dal processo, ad esempio Amazon S3. In alternativa, se non si desidera utilizzare il metodo precedente, usare le seguenti policy IAM gestite.

- AWSGlueServiceRole— Garantisce l'accesso alle risorse che AWS Glue i vari processi richiedono per l'esecuzione per conto dell'utente. Queste risorse includono AWS Glue Amazon S3, IAM,

CloudWatch Logs e Amazon. EC2 Se segui la convenzione di denominazione delle risorse specificata in questa politica, AWS Glue i processi dispongono delle autorizzazioni richieste. Questa policy è in genere collegata ai ruoli specificati quando si definiscono crawler, processi ed endpoint di sviluppo.

- AWSGlueConsoleFullAccess— Concede l'accesso completo alle AWS Glue risorse quando un'identità a cui è allegata la policy utilizza la AWS console di gestione. Se si segue la convenzione per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questa policy è in genere associata agli utenti della AWS Glue console.
- SecretsManagerReadWrite— Fornisce read/write l'accesso a AWS Secrets Manager tramite la console di AWS gestione. Nota: questo esclude le azioni IAM, quindi occorre combinarlo con `IAMFullAccess` se è richiesta la configurazione di rotazione.

IAM Policies/Permissions era necessario per configurare il VPC

Le seguenti autorizzazioni IAM sono necessarie durante l'utilizzo della connessione VPC per AWS Glue la creazione di Connection. Per maggiori informazioni, consultare Creazione di una policy IAM per AWS Glue.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateNetworkInterface",
        "ec2:DeleteNetworkInterface",
        "ec2:DescribeNetworkInterfaces"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

Connettività/Connessione VPC

Passaggi per la connessione VPC:

1. Usare una connessione VPC esistente o crearne una nuova seguendo la Documentazione di Amazon VPC.
2. Assicurarsi di disporre di un gateway NAT che indirizza il traffico verso Internet.
3. Scegliere un endpoint VPC come gateway Amazon S3 per creare una connessione.
4. Abilita la risoluzione DNS e il nome host DNS per utilizzare i servizi DNS AWS forniti.
5. Vai al VPC creato e aggiungi gli endpoint necessari per diversi servizi come STS AWS Glue, Secret Managers.
   a. Scegliere Crea endpoint.
   b. Per Categoria di servizio, scegli AWS Servizi.
   c. Per Nome del servizio, scegliere il servizio a cui ci si connette.
   d. Scegliere VPC e abilitare il nome DNS.
   e. Endpoint VCP necessari per la connessione VPC:
      i. STS
      ii. AWS Glue
      iii. Secrets Manager

Configurazione del gruppo di sicurezza

Il gruppo di sicurezza deve consentire il traffico verso la sua porta di ascolto dal AWS Glue VPC per AWS Glue potersi connettere ad esso. È buona norma limitare il più possibile l'intervallo di indirizzi IP di origine.

AWS Glue richiede un gruppo di sicurezza speciale che consenta tutto il traffico in entrata da se stesso. È possibile creare una regola autoreferenziale che consente tutto il traffico proveniente dal gruppo di sicurezza. È possibile modificare un gruppo di sicurezza esistente e specificare il gruppo di sicurezza come origine.

Aprire la comunicazione dalle porte HTTPS dell'endpoint URL (istanza NLB o SAP).

Opzioni di connettività

- Connessione HTTPS con NLB interno ed esterno, certificato SSL dell'autorità di certificazione (CA), certificato SSL non autofirmato

- Connessione HTTPS con certificato SSL dell'istanza SAP dall'autorità di certificazione (CA), certificato SSL non autofirmato

### Autenticazione SAP

Il connettore SAP supporta i metodi di autenticazione CUSTOM (l'autenticazione SAP BASIC) e OAUTH.

### Autenticazione personalizzata

AWS Glue supporta Custom (autenticazione di base) come metodo per stabilire connessioni ai sistemi SAP, consentendo l'uso di un nome utente e una password per un accesso sicuro. Questo tipo di autenticazione funziona bene per gli scenari di automazione in quanto consente di utilizzare nome utente e password in anticipo con le autorizzazioni di un particolare utente nell'istanza SAP. OData AWS Glue è in grado di utilizzare il nome utente e la password per autenticare SAP. OData APIs Nel AWS Glue, l'autorizzazione di base viene implementata come autorizzazione personalizzata.

Per la OData documentazione SAP pubblica per il flusso di autenticazione di base, consulta Autenticazione di base HTTP.

### OAuth Autenticazione 2.0

AWS Glue supporta anche la OAuth versione 2.0 come meccanismo di autenticazione sicuro per stabilire connessioni ai sistemi SAP. Ciò consente un'integrazione perfetta, garantendo al contempo la conformità ai moderni standard di autenticazione e migliorando la sicurezza dell'accesso ai dati.

### Tipo di concessione AUTHORIZATION_CODE

Il tipo di concessione determina il modo in cui AWS Glue comunica con SAP OData per richiedere l'accesso ai dati. SAP OData supporta solo il tipo di concessione. `AUTHORIZATION_CODE` Questo tipo di concessione è considerato «a tre gambe» in OAuth quanto si basa sul reindirizzamento degli utenti al server di autorizzazione di terze parti per autenticare l'utente. Viene utilizzato durante la creazione di connessioni tramite la console. AWS Glue

Gli utenti possono comunque scegliere di creare la propria app connessa in SAP OData e fornire il proprio ID client e il segreto del client durante la creazione di connessioni tramite la AWS Glue console. In questo scenario, verranno comunque reindirizzati a SAP OData per effettuare il login e autorizzare l'accesso AWS Glue alle proprie risorse.

Questo tipo di concessione genera un token di aggiornamento e un token di accesso. Il token di accesso dura poco tempo e può essere aggiornato automaticamente senza l'interazione dell'utente utilizzando il token di aggiornamento.

Per la OData documentazione SAP pubblica sulla creazione di un'app connessa per il OAuth flusso del codice di autorizzazione, vedi Authentication Using 2.0. OAuth

AWS Secrets Manager per memorizzare il tuo segreto di autenticazione

Dovrai archiviare i segreti di OData connessione SAP in AWS Secrets Manager, configurare le autorizzazioni necessarie per il recupero come specificato nella Policy IAM sezione e utilizzarli durante la creazione di una connessione.

Usa la console di AWS gestione di AWS Secrets Manager per creare un segreto per la tua fonte SAP. Per ulteriori informazioni, consulta Creare un segreto di AWS Secrets Manager. I dettagli in AWS Secrets Manager devono includere gli elementi del codice seguente.

Segreto di autenticazione personalizzata

È necessario inserire il nome utente del sistema SAP al posto di **<your SAP username>** e la relativa password al posto di **<your SAP username password>** e True o False. In questo contesto, l'impostazione di `basicAuthDisableSSO` su `true` disabilita il Single Sign-On (SSO) per le richieste di autenticazione di base, richiedendo credenziali utente esplicite per ogni richiesta. Al contrario, impostarlo su `false` consente l'uso di sessioni SSO esistenti, se disponibili.

```
{
   "basicAuthUsername": "<your SAP username>",
   "basicAuthPassword": "<your SAP username password>",
   "basicAuthDisableSSO": "<True/False>",
   "customAuthenticationType": "CustomBasicAuth"
}
```

OAuth 2.0 Segreto

Nel caso in cui si utilizzi la OAuth versione 2.0 come meccanismo di autenticazione, il segreto in AWS Secrets Manager dovrebbe avere l'applicazione User Managed Client ClientId nel seguente formato. È necessario inserire il segreto del client SAP al posto di **<your client secret>**.

```
{"USER_MANAGED_CLIENT_APPLICATION_CLIENT_SECRET": "<your client secret>"
}
```

## Creare connessioni

Per configurare una OData connessione SAP:

1. Accedi alla console di AWS gestione e apri la AWS Glue console. In AWS Glue Studio, crea una connessione seguendo i passaggi seguenti:
   a. Fare clic su Connessioni dati nel pannello sinistro.
   b. Fare clic su Crea connessione.
   c. Seleziona SAP OData in Scegli l'origine dati
   d. Fornisci l'URL dell'host dell'applicazione dell' OData istanza SAP a cui desideri connetterti. L'URL dell'host dell'applicazione deve essere accessibile su Internet pubblico per una connessione non VPC.
   e. Fornisci il percorso del servizio dell'applicazione dell' OData istanza SAP a cui desideri connetterti. È identico al percorso del servizio di catalogo. Ad esempio:`/sap/opu/odata/iwfnd/catalogservice;v=2`. AWS Glue non accetta un percorso specifico dell'oggetto.
   f. Fornisci il numero client dell' OData istanza SAP a cui desideri connetterti. I valori accettabili sono [001-999]. Ad esempio: 010
   g. Fornisci il numero di porta dell' OData istanza SAP a cui desideri connetterti. Ad esempio: 443
   h. Fornisci la lingua di accesso dell' OData istanza SAP a cui desideri connetterti. Ad esempio: EN
   i. Seleziona il ruolo AWS IAM che AWS Glue può assumere e che dispone delle autorizzazioni, come indicato nella sezione. Policy IAM
   j. Seleziona il tipo di autenticazione che desideri utilizzare per questa connessione AWS Glue dall'elenco a discesa: o PERSONALIZZATO OAUTH2
      i. CUSTOM: selezionare il segreto creato come specificato nella sezione AWS Secrets Manager per memorizzare il tuo segreto di autenticazione.
      ii. OAUTH 2.0: inserisci i seguenti input solo nel caso di 2.0: OAuth
         A. In User Managed Client Application ClientId, inserisci il tuo ID cliente.
         B. `USER_MANAGED_CLIENT_APPLICATION_CLIENT_SECRET`(il segreto del tuo client) nel AWS Secrets Manager che hai creato nella AWS Secrets Manager per memorizzare il tuo segreto di autenticazione sezione.
         C. In URL del codice di autorizzazione, inserire l'URL del codice di autorizzazione.
         D. In URL del token di autorizzazione, inserire l'URL del token di autorizzazione.
         E. In OAuth Ambiti, inserisci OAuth gli ambiti separati da uno spazio. Ad esempio: `/IWFND/SG_MED_CATALOG_0002 ZAPI_SALES_ORDER_SRV_0001`

k. Selezionare le opzioni di rete se si desidera utilizzare la propria rete. Per ulteriori dettagli, consultare Connettività/Connessione VPC.

2. Concedi il permesso di lettura al ruolo IAM associato al tuo AWS Glue lavoro. `secretName` Per ulteriori dettagli, consultare Policy IAM.
3. Scegliere Test connessione e testare la connessione. Se il test di connessione ha esito positivo, fare clic su Avanti, inserire il nome della connessione e salvarla. La funzionalità di test della connessione non è disponibile se è stato scelto Opzioni di rete (VPC).

## Creazione di un lavoro SAP OData

Fai riferimento a Creazione di job ETL visivi con Studio AWS Glue

Origini ODP (Operational Data Provisioning)

Operational Data Provisioning (ODP) fornisce un'infrastruttura tecnica da utilizzare per supportare l'estrazione e la replica dei dati per varie applicazioni di destinazione e supporta i meccanismi delta in questi scenari. Nel caso di una procedura delta, i dati provenienti da un'origine (ODP Provider) vengono automaticamente scritti in una coda delta (Operational Delta Queue, ODQ) utilizzando un processo di aggiornamento o trasmessi alla coda delta utilizzando un'interfaccia di estrazione. Un provider ODP può essere un DataSource (estrattori), ABAP Core Data Services Views (ABAP CDS Views), SAP BW o SAP BW/4HANA, SAP Landscape Transformation Replication Server (SLT) e SAP HANA Information Views (visualizzazioni di calcolo). Le applicazioni di destinazione (denominate "abbonati" ODQ o più in generale "consumatori ODP") recuperano i dati dalla coda delta e continuano a elaborarli.

Caricamento completo

Nel contesto delle entità SAP OData e ODP, un Full Load si riferisce al processo di estrazione di tutti i dati disponibili da un'entità ODP in un'unica operazione. Questa operazione recupera il set di dati completo dal sistema di origine, garantendo che il sistema di destinazione disponga di una up-to-date copia completa dei dati dell'entità. I caricamenti completi vengono in genere utilizzati per origini che non supportano carichi incrementali o quando è richiesto un aggiornamento del sistema di destinazione.

Esempio

È possibile impostare esplicitamente il `ENABLE_CDC` flag su false, durante la creazione di. DynamicFrame Nota: `ENABLE_CDC` è false per impostazione predefinita, se non si vuole inizializzare

la coda delta, non bisogna inviare questo flag o impostarlo su true. La mancata impostazione di questo flag su true comporterà un'estrazione di caricamento completo.

```
sapodata_df = glueContext.create_dynamic_frame.from_options(
    connection_type="SAPOData",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "entityName",
        "ENABLE_CDC": "false"
    }, transformation_ctx=key)
```

Carico incrementale

Un carico incrementale nel contesto delle entità ODP (Operational Data Provisioning) comporta l'estrazione solo dei dati nuovi o modificati (delta) dal sistema di origine dall'ultima estrazione dei dati, evitando la pre-elaborazione dei record già elaborati. Questo approccio migliora in modo significativo l'efficienza, riduce i volumi di trasferimento dei dati, migliora le prestazioni, garantisce una sincronizzazione efficiente tra i sistemi e minimizza i tempi di elaborazione, soprattutto per set di dati di grandi dimensioni che cambiano frequentemente.

Trasferimenti incrementali basati su token Delta

Per abilitare il trasferimento incrementale utilizzando Change Data Capture (CDC) per le entità abilitate a ODP che lo supportano, seguire questi passaggi:

1. Creare il processo di trasferimento incrementale in modalità script.
2. Quando crei DataFrame o Glue DynamicFrame, devi passare l'opzione`"ENABLE_CDC": "True"`. Questa opzione garantisce la ricezione di un token Delta da SAP, che può essere utilizzato per il successivo recupero dei dati modificati.

Il token delta sarà presente nell'ultima riga del dataframe, nella colonna DELTA_TOKEN. Questo token può essere utilizzato come opzione di connettore nelle chiamate successive per recuperare in modo incrementale il set di dati successivo.

Esempio

- Abbiamo impostato la `ENABLE_CDC` bandiera su`true`, durante la creazione di DynamicFrame. Nota: `ENABLE_CDC` è `false` per impostazione predefinita, se non si vuole inizializzare la coda delta, non è necessario inviare questo flag o impostarlo su true. La mancata impostazione di questo flag su true comporterà un'estrazione di caricamento completo.

```
sapodata_df = glueContext.create_dynamic_frame.from_options(
    connection_type="SAPOData",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "entityName",
        "ENABLE_CDC": "true"
    }, transformation_ctx=key)

# Extract the delta token from the last row of the DELTA_TOKEN column
delta_token_1 = your_logic_to_extract_delta_token(sapodata_df) # e.g.,
 D20241029164449_000370000
```

- Il token Delta estratto può essere passato come opzione per recuperare nuovi eventi.

```
sapodata_df_2 = glueContext.create_dynamic_frame.from_options(
    connection_type="SAPOData",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "entityName",
        // passing the delta token retrieved in the last run
        "DELTA_TOKEN": delta_token_1
    } , transformation_ctx=key)

# Extract the new delta token for the next run
delta_token_2 = your_logic_to_extract_delta_token(sapodata_df_2)
```

L'ultimo record, in cui è presente `DELTA_TOKEN`, non è un record transazionale di origine ed è presente solo allo scopo di trasmettere il valore del token delta.

Oltre a `DELTA_TOKEN`, i seguenti campi vengono restituiti in ogni riga del dataframe.

- GLUE_FETCH_SQ: questo è un campo di sequenza, generato dalla marcatura temporale EPOC nell'ordine in cui il record è stato ricevuto, ed è unico per ogni record. Può essere utilizzato se è necessario conoscere o stabilire l'ordine delle modifiche nel sistema di origine. Questo campo sarà presente solo per le entità abilitate per ODP.
- DML_STATUS: verrà visualizzato `UPDATED` per tutti i record appena inseriti e aggiornati dall'origine e `DELETED` per i record che sono stati eliminati dall'origine.

Per maggiori informazioni su come gestire lo stato e riutilizzare il token delta per recuperare i record modificati tramite un esempio, consultare la sezione Utilizzo dello script di gestione OData dello stato SAP.

Invalidazione del token Delta

Un token Delta è associato a un utente e alla raccolta di servizi. Se `"ENABLE_CDC" : "true"` viene avviato un nuovo pull with iniziale per la stessa raccolta di servizi e per lo stesso utente, tutti i token delta precedenti emessi a seguito di una precedente inizializzazione verranno invalidati dal servizio SAP. OData L'invocazione del connettore con un token Delta scaduto porterà a un'eccezione:

```
Could not open data access via extraction API RODPS_REPL_ODP_OPEN
```

OData Servizi (fonti non ODP)

Caricamento completo

Per i sistemi non ODP (Operational Data Provisioning), un Carico completo prevede l'estrazione dell'intero set di dati dal sistema di origine e il suo caricamento nel sistema di destinazione. Poiché i sistemi non ODP non supportano intrinsecamente meccanismi avanzati di estrazione dei dati come i delta, il processo è semplice ma può richiedere molte risorse a seconda delle dimensioni dei dati.

Carico incrementale

Per i sistemi o le entità che non supportano ODP (Operational Data Provisioning), il trasferimento incrementale dei dati può essere gestito manualmente implementando un meccanismo basato su marcature temporali per tracciare ed estrarre le modifiche.

Trasferimenti incrementali basati su marcature temporali

Per le entità non abilitate per ODP (o per le entità abilitate per ODP che non utilizzano il flag ENABLE_CDC), possiamo usare un'opzione `filteringExpression` nel connettore per indicare l'intervallo `datetime` per il quale vogliamo recuperare i dati. Questo metodo si basa su un campo timestamp nei dati che indica quando ogni record è stato creato/modificato per l'ultima volta.

Esempio

Recuperare i record modificati dopo 2024-01-01T00:00:00.000

```
sapodata_df = glueContext.create_dynamic_frame.from_options(
    connection_type="SAPOData",
```

```
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "entityName",
        "filteringExpression": "LastChangeDateTime >= 2024-01-01T00:00:00.000"
    }, transformation_ctx=key)
```

Nota: in questo esempio, `LastChangeDateTime` è il campo che rappresenta l'ultima modifica di ogni record. Il nome effettivo del campo può variare a seconda dell'entità OData SAP specifica.

Per ottenere un nuovo sottoinsieme di dati nelle esecuzioni successive, è necessario aggiornare `filteringExpression` con una nuova marcatura temporale. In genere, questo sarebbe il valore massimo della marcatura temporale dei dati recuperati in precedenza.

Esempio

```
max_timestamp = get_max_timestamp(sapodata_df)  # Function to get the max timestamp
 from the previous run
next_filtering_expression = f"LastChangeDateTime > {max_timestamp}"

# Use this next_filtering_expression in your next run
```

Nella prossima sezione, forniremo un approccio automatizzato per gestire questi trasferimenti incrementali basati sulla marcatura temporale, eliminando la necessità di aggiornare manualmente l'espressione di filtraggio tra le esecuzioni.

## Scrivere su SAP OData

Questa sezione descrive come scrivere dati sul OData servizio SAP utilizzando il AWS Glue connettore per SAP. OData

Prerequisiti

- Accesso a un servizio SAP OData
- Un OData EntitySet oggetto SAP su cui desideri scrivere. È necessario il nome dell'oggetto.
- OData Credenziali SAP valide e connessione valida
- Autorizzazioni appropriate come descritto nelle Policy IAM

Il OData connettore SAP supporta due operazioni di scrittura:

- INSERT
- UPDATE

Quando si utilizza l'operazione di scrittura UPDATE, è necessario indicare l'opzione ID_FIELD_NAMES per specificare il campo ID esterno per i record.

Esempio:

```
sapodata_write = glueContext.write_dynamic_frame.from_options(
    frame=frameToWrite,
    connection_type="sapodata",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "entityName",
        "WRITE_OPERATION": "INSERT"
    }
```

## Utilizzo dello script di gestione OData dello stato SAP

Per utilizzare lo script di gestione OData dello stato SAP nel tuo AWS Glue lavoro, segui questi passaggi:

- Scaricare lo script di gestione dello stato: `s3://aws-blogs-artifacts-public/artifacts/BDB-4789/sap_odata_state_management.zip` dal bucket pubblico Amazon S3.
- Carica lo script in un bucket Amazon S3 a cui il tuo AWS Glue job dispone delle autorizzazioni di accesso.
- Fai riferimento allo script nel tuo AWS Glue job: quando crei o aggiorni il AWS Glue job, passa l'`'--extra-py-files'`opzione che fa riferimento al percorso dello script nel tuo bucket Amazon S3. Ad esempio: `--extra-py-files s3://your-bucket/path/to/sap_odata_state_management.py`
- Importa e usa la libreria di gestione dello stato nei tuoi AWS Glue script di lavoro.

Esempio di trasferimento incrementale basato su token Delta

Ecco un esempio di come utilizzare lo script di gestione dello stato per i trasferimenti incrementali basati su delta-token:

```
from sap_odata_state_management import StateManagerFactory, StateManagerType, StateType
```

```
# Initialize the state manager
state_manager = StateManagerFactory.create_manager(
    manager_type=StateManagerType.JOB_TAG,
    state_type=StateType.DELTA_TOKEN,
    options={
        "job_name": args['JOB_NAME'],
        "logger": logger
    }
)

# Get connector options (including delta token if available)
key = "SAPODataNode"
connector_options = state_manager.get_connector_options(key)

# Use the connector options in your Glue job
df = glueContext.create_dynamic_frame.from_options(
    connection_type="SAPOData",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "entityName",
        "ENABLE_CDC": "true",
        **connector_options
    }
)

# Process your data here...

# Update the state after processing
state_manager.update_state(key, sapodata_df.toDF())
```

Esempio di trasferimento incrementale basato su marcature temporali

Ecco un esempio di come utilizzare lo script di gestione dello stato per i trasferimenti incrementali basati su delta-token:

```
from sap_odata_state_management import StateManagerFactory, StateManagerType, StateType

# Initialize the state manager
state_manager = StateManagerFactory.create_manager(
    manager_type=StateManagerType.JOB_TAG,
    state_type=StateType.DELTA_TOKEN,
    options={
```

```
        "job_name": args['JOB_NAME'],
        "logger": logger
    }
)

# Get connector options (including delta token if available)
key = "SAPODataNode"
connector_options = state_manager.get_connector_options(key)

# Use the connector options in your Glue job
df = glueContext.create_dynamic_frame.from_options(
    connection_type="SAPOData",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "entityName",
        "ENABLE_CDC": "true",
        **connector_options
    }
)

# Process your data here...

# Update the state after processing
state_manager.update_state(key, sapodata_df.toDF())
```

In entrambi gli esempi, lo script di gestione dello stato gestisce la complessità della memorizzazione dello stato (token delta o marcatura temporale) tra le esecuzioni dei processi. Recupera automaticamente l'ultimo stato conosciuto quando ottiene le opzioni del connettore e aggiorna lo stato dopo l'elaborazione, assicurando che ogni esecuzione del processo elabori solo dati nuovi o modificati.

## Partizionamento per entità non ODP

In Apache Spark, il partizionamento indica il modo in cui i dati vengono divisi e distribuiti tra i nodi worker in un cluster per l'elaborazione parallela. Ogni partizione è un blocco logico di dati che può essere elaborato indipendentemente da un'attività. Il partizionamento è un concetto fondamentale in Spark che influisce direttamente sulle prestazioni, sulla scalabilità e sull'utilizzo delle risorse. AWS Glue i job utilizzano il meccanismo di partizionamento di Spark per dividere il set di dati in blocchi più piccoli (partizioni) che possono essere elaborati in parallelo tra i nodi di lavoro del cluster. Il partizionamento non è applicabile alle entità ODP.

Per maggiori dettagli, consulta Spark and jobs.AWS Glue PySpark

Prerequisiti

Un oggetto SAP OData da cui vorresti leggere. Avrai bisogno del object/EntitySet nome. Ad esempio: `/sap/opu/odata/sap/API_SALES_ORDER_SRV/A_SalesOrder` .

Esempio

```
sapodata_read = glueContext.create_dynamic_frame.from_options(
    connection_type="SAPOData",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "/sap/opu/odata/sap/API_SALES_ORDER_SRV/A_SalesOrder"
    }, transformation_ctx=key)
```

Query di partizionamento

Partizionamento basato sul campo

Puoi fornire le opzioni Spark aggiuntive `PARTITION_FIELD`, `LOWER_BOUND`, `UPPER_BOUND` e `NUM_PARTITIONS` se desideri utilizzare la concorrenza in Spark. Con questi parametri, la query originale viene suddivisa in `NUM_PARTITIONS` query secondarie che possono essere eseguite contemporaneamente dalle attività Spark. Numero intero, Data e DateTime campi supportano il partizionamento basato sul campo nel connettore SAP. OData

- `PARTITION_FIELD`: il nome del campo da utilizzare per partizionare la query.
- `LOWER_BOUND`: un valore limite inferiore inclusivo del campo di partizione scelto.

  Per qualsiasi campo il cui tipo di dati è DateTime, viene accettato il formato di timestamp Spark utilizzato nelle query SQL di Spark.

  Esempi di valori validi: `"2000-01-01T00:00:00.000Z"`
- `UPPER_BOUND`: un valore limite superiore esclusivo del campo di partizione scelto.
- `NUM_PARTITIONS`: numero di partizioni.
- `PARTITION_BY`: il tipo di partizionamento da eseguire, `FIELD` da passare in caso di partizionamento basato sul campo.

Esempio

```
sapodata= glueContext.create_dynamic_frame.from_options(
```

```
    connection_type="sapodata",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "/sap/opu/odata/sap/SEPM_HCM_SCENARIO_SRV/EmployeeSet",
        "PARTITION_FIELD": "validStartDate"
        "LOWER_BOUND": "2000-01-01T00:00:00.000Z"
        "UPPER_BOUND": "2020-01-01T00:00:00.000Z"
        "NUM_PARTITIONS": "10",
        "PARTITION_BY": "FIELD"
    }, transformation_ctx=key)
```

Partizionamento basato su record

La query originale viene suddivisa in NUM_PARTITIONS query secondarie che possono essere eseguite contemporaneamente dalle attività Spark.

Il partizionamento basato sui record è supportato solo per le entità non ODP, poiché l'impaginazione nelle entità ODP è supportata tramite il token successivo. token/skip

- PARTITION_BY: il tipo di partizionamento da eseguire. COUNT è da passare in caso di partizionamento basato sul record.

Esempio

```
sapodata= glueContext.create_dynamic_frame.from_options(
    connection_type="sapodata",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "/sap/opu/odata/sap/SEPM_HCM_SCENARIO_SRV/EmployeeSet",
        "NUM_PARTITIONS": "10",
        "PARTITION_BY": "COUNT"
    }, transformation_ctx=key)
```

Limitazioni/Callout

- Le entità ODP non sono compatibili con il partizionamento basato su record poiché l'impaginazione viene gestita utilizzando il token skip. token/delta Di conseguenza, per il partizionamento basato su record, il valore predefinito per maxConcurrency è impostato su “null” indipendentemente dall'input dell'utente.
- Quando vengono applicati sia il limite che la partizione, il limite ha la precedenza sul partizionamento.

## Opzioni di OData connessione SAP

Di seguito sono riportate le opzioni di connessione per SAP: OData

- `ENTITY_NAME`(String): (obbligatorio) utilizzato per la lettura. Il nome dell'oggetto in OData SAP.

  Ad esempio:/_ sap/opu/odata/sap/API_SALES_ORDER_SRV/A SalesOrder
- `API_VERSION`(String): (facoltativo) utilizzato per la lettura. Versione dell' OData API SAP Rest che desideri utilizzare. Esempio: 2.0.
- `SELECTED_FIELDS`(List<String>): predefinito: vuoto (SELECT *). Utilizzato per la lettura. Colonne da selezionare per l'oggetto.

  Ad esempio: SalesOrder
- `FILTER_PREDICATE`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Deve essere in formato Spark SQL.

  Ad esempio: `SalesOrder = "10"`
- `QUERY`(String): predefinito, vuoto. Utilizzato per la lettura. Query Spark SQL completa.

  Ad esempio: `SELECT * FROM /sap/opu/odata/sap/API_SALES_ORDER_SRV/A_SalesOrder`
- `PARTITION_FIELD`(String): utilizzato per la lettura. Campo da utilizzare per partizionare la query.

  Ad esempio: `ValidStartDate`
- `LOWER_BOUND`(String): utilizzato per la lettura. Un valore limite inferiore inclusivo del campo di partizione scelto.

  Ad esempio: `"2000-01-01T00:00:00.000Z"`
- `UPPER_BOUND`(String): utilizzato per la lettura. Un valore limite superiore esclusivo del campo di partizione scelto.

  Ad esempio: `"2024-01-01T00:00:00.000Z"`
- `NUM_PARTITIONS`(Integer): predefinito, 1. Utilizzato per la lettura. Numero di partizioni da leggere.
- `INSTANCE_URL`(String): l'URL dell'host dell'applicazione dell'istanza SAP.

  Ad esempio: `https://example-externaldata.sierra.aws.dev`
- `SERVICE_PATH`(String): il percorso del servizio dell'applicazione dell'istanza SAP.

Ad esempio: `/sap/opu/odata/iwfnd/catalogservice;v=2`

- `CLIENT_NUMBER`(String): il numero del client dell'applicazione dell'istanza SAP.

  Ad esempio: 100

- `PORT_NUMBER`(String): predefinito, il numero della porta dell'applicazione dell'istanza SAP.

  Ad esempio: 443

- `LOGON_LANGUAGE`(String): il linguaggio di accesso all'applicazione dell'istanza SAP.

  Ad esempio: `EN`

- `ENABLE_CDC`(String): definisce se eseguire un processo con CDC abilitato, ovvero con tracciamento delle modifiche.

  Ad esempio: `True/False`

- `DELTA_TOKEN`(String): esegue un pull di dati incrementale basato sul token Delta valido fornito.

  Ad esempio: `D20241107043437_000463000`

- `PAGE_SIZE`(Integer): definisce la dimensione della pagina per la query dei record. La dimensione della pagina predefinita è 50.000. Quando viene specificata una dimensione della pagina, SAP restituisce solo il numero definito di record per chiamata API, anziché l'intero set di dati. Il connettore continuerà a fornire il numero totale di record e gestirà l'impaginazione utilizzando la dimensione della pagina specificata. Se è necessaria una dimensione della pagina più grande, si può scegliere qualsiasi valore fino a 500.000, che è il massimo consentito. Qualsiasi dimensione della pagina specificata superiore a 500.000 verrà ignorata. Al contrario, il sistema utilizzerà la dimensione della pagina massima consentita. Puoi specificare la dimensione della pagina nell' AWS Glue Studio interfaccia utente aggiungendo un'opzione di connessione `PAGE_SIZE` con il valore desiderato.

  Ad esempio: `20000`

## Dettagli OData dell'entità e del campo SAP

| Entità | Tipo di dati | Operatori supportati |
|---|---|---|
| Tabelle (entità dinamiche) | Stringa | =, !=, >, >=, <, <=, BETWEEN, LIKE |

| Entità | Tipo di dati | Operatori supportati |
|---|---|---|
| | Numero intero | =, !=, >, >=, <, <=, BETWEEN, LIKE |
| | Long | =, !=, >, >=, <, <=, BETWEEN, LIKE |
| | Double | =, !=, >, >=, <, <=, BETWEEN, LIKE |
| | Data | =, !=, >, >=, <, <=, BETWEEN, LIKE |
| | DateTime | =, !=, >, >=, <, <=, BETWEEN, LIKE |
| | Booleano | =, != |
| | Struct | =, !=, >, >=, <, <=, BETWEEN, LIKE |

## Connessione a SendGrid

SendGrid è una piattaforma di comunicazione con i clienti per e-mail transazionali e di marketing.

- SendGrid connector aiuta a creare e gestire elenchi di contatti e creare campagne di email marketing.
- SendGrid consente alle aziende online, alle organizzazioni non profit e ad altre entità online di creare e inviare e-mail di marketing a un vasto pubblico e di monitorare il coinvolgimento con tali e-mail.

Argomenti

- AWS Glue supporto per SendGrid
- Policy contenenti le operazioni API per creare e utilizzare le connessioni
- Configurazione SendGrid
- Configurazione delle connessioni SendGrid

- Lettura da SendGrid entità
- SendGrid opzioni di connessione
- SendGrid limitazioni

## AWS Glue supporto per SendGrid

AWS Glue supporta SendGrid quanto segue:

Supportato come origine?

Sì. È possibile utilizzare i job AWS Glue ETL da SendGrid cui interrogare i dati.

Supportato come destinazione?

No.

Versioni SendGrid API supportate

Sono supportate le seguenti versioni SendGrid API:

- v3

## Policy contenenti le operazioni API per creare e utilizzare le connessioni

La seguente policy di esempio descrive le autorizzazioni AWS IAM richieste per la creazione e l'utilizzo delle connessioni. Se si sta creando un nuovo ruolo, creare una policy che contenga quanto segue:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:ListConnectionTypes",
        "glue:DescribeConnectionType",
        "glue:RefreshOAuth2Tokens",
        "glue:ListEntities",
```

```
        "glue:DescribeEntity"
      ],
      "Resource": "*"
    }
  ]
}
```

In alternativa, se non desideri utilizzare il metodo precedente, utilizza le seguenti policy IAM gestite:

- AWSGlueServiceRole— Garantisce l'accesso alle risorse che AWS Glue i vari processi richiedono per l'esecuzione per conto dell'utente. Queste risorse includono AWS Glue Amazon S3, IAM, CloudWatch Logs e Amazon. EC2 Se segui la convenzione di denominazione delle risorse specificata in questa politica, AWS Glue i processi dispongono delle autorizzazioni richieste. Questa policy è in genere collegata ai ruoli specificati quando si definiscono crawler, processi ed endpoint di sviluppo.
- AWSGlueConsoleFullAccess— Concede l'accesso completo alle AWS Glue risorse quando un'identità a cui è allegata la policy utilizza la AWS console di gestione. Se si segue la convenzione per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questa policy è in genere associata agli utenti della AWS Glue console.

## Configurazione SendGrid

Prima di poter utilizzare AWS Glue per trasferire dati da SendGrid, è necessario soddisfare i seguenti requisiti:

Requisiti minimi

I requisiti minimi sono i seguenti:

- Hai un SendGrid account con una chiave API.
- Il tuo SendGrid account ha accesso all'API con una licenza valida.

Se soddisfi questi requisiti, sei pronto per connetterti AWS Glue al tuo SendGrid account. Per le connessioni tipiche, non devi fare nient'altro SendGrid.

## Configurazione delle connessioni SendGrid

SendGrid supporta l'autenticazione personalizzata.

Per la SendGrid documentazione pubblica sulla generazione delle chiavi API richieste per l'autenticazione personalizzata, vedi Autenticazione.

Per configurare una SendGrid connessione:

1. In AWS Secrets Manager, crea un segreto con i seguenti dettagli:
   a. Per le app connesse gestite dal cliente: il segreto deve contenere l'app connessa Consumer Secret con *api_key* come chiave.
   b. Nota: devi creare un segreto per le tue connessioni in AWS Glue.

1. In AWS Glue Glue Studio, crea una connessione in Connessioni dati seguendo i passaggi seguenti:
   a. Quando si seleziona un tipo di connessione, selezionare SendGrid.
   b. Fornisci `INSTANCE_URL` l' SendGrid istanza a cui desideri connetterti.
   c. Seleziona il ruolo AWS IAM che AWS Glue può assumere e dispone delle autorizzazioni per le seguenti azioni:

      JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:DescribeSecret",
        "secretsmanager:GetSecretValue",
        "secretsmanager:PutSecretValue",
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DeleteNetworkInterface"
      ],
      "Resource": "*"
    }
  ]
}
```

   d. Seleziona quello `secretName` che desideri utilizzare per questa connessione per AWS Glue inserire i token.

e. Selezionare le opzioni di rete se si desidera utilizzare la propria rete.

2. Concedi il permesso di lettura `secretName` al ruolo IAM associato al tuo AWS Glue lavoro.

## Lettura da SendGrid entità

Prerequisito

Un SendGrid oggetto da cui vorresti leggere. È necessario il nome dell'oggetto, come `lists`, `singlesends` o `segments`.

Entità supportate come origine:

| Entità | Può essere filtrato | Supporta limite | Supporta Ordina per | Supporta Seleziona * | Supporta il partizionamento |
|---|---|---|---|---|---|
| Elenchi | No | Sì | No | Sì | No |
| Invii singoli | Sì | Sì | No | Sì | No |
| Statistiche e automazioni delle campagne di marketing | Sì | Sì | No | Sì | No |
| Invii singoli per statistiche della campagna di marketing | Sì | Sì | No | Sì | No |
| Segmenti | Sì | No | No | Sì | No |
| Contatti | Sì | No | No | Sì | No |
| Categoria | No | No | No | Sì | No |
| Statistiche | Sì | No | No | Sì | No |

| Entità | Può essere filtrato | Supporta limite | Supporta Ordina per | Supporta Seleziona * | Supporta il partizionamento |
|---|---|---|---|---|---|
| Gruppi di annullamento dell'iscrizione | Sì | No | No | Sì | No |

Esempio:

```
sendgrid_read = glueContext.create_dynamic_frame.from_options(
    connection_type="sendgrid",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "lists",
        "API_VERSION": "v3",
        "INSTANCE_URL": "instanceUrl"
    }
```

SendGrid dettagli dell'entità e del campo:

Entità con metadati statici:

| Entità | Campo | Tipo di dati | Operatori supportati | |
|---|---|---|---|---|
| Elenchi | id | Stringa | N/A | |
| | nome | Stringa | N/A | |
| | contact_count | Numero intero | N/D | |
| | _metadata | Struct | N/D | |
| Invii singoli | id | Stringa | N/A | |
| | nome | Stringa | EQUAL_TO | |
| | abtest | Struct | N/D | |

| Entità | Campo | Tipo di dati | Operatori supportati | |
|---|---|---|---|---|
| | status | Stringa | EQUAL_TO | |
| | categorie | List | EQUAL_TO | |
| | send_at | Stringa | N/A | |
| | is_abtest | Booleano | N/D | |
| | updated_at | Stringa | N/A | |
| | created_at | Stringa | N/A | |
| | canali | List | N/D | |
| Statistiche e automazioni delle campagne di marketing | id | Stringa | N/A | |
| | aggregation | Stringa | N/A | |
| | step_id | Stringa | N/A | |
| | stats | Struct | N/D | |
| | automation_ids | List | EQUAL_TO | |
| Invii singoli per statistiche della campagna di marketing | id | Stringa | N/A | |
| | ab_variation | Stringa | N/A | |
| | ab_phase | Stringa | N/A | |
| | aggregation | Stringa | N/A | |
| | stats | Struct | N/D | |
| | singlesend_ids | List | EQUAL_TO | |
| Segmenti | id | Stringa | N/A | |
| | nome | Stringa | N/A | |

| Entità | Campo | Tipo di dati | Operatori supportati | |
|---|---|---|---|---|
| | query_version | Stringa | N/A | |
| | contacts_count | Numero intero | N/D | |
| | sample_updated_at | Stringa | N/A | |
| | next_sample_update | Stringa | N/A | |
| | created_at | Stringa | N/A | |
| | updated_at | Stringa | N/A | |
| | parent_list_id | Stringa | N/A | |
| | status | Struct | N/D | |
| | parent_list_ids | Stringa | EQUAL_TO | |
| | no_parent_list_id | Booleano | EQUAL_TO | |
| Contatti | id | Stringa | N/A | |
| | first_name | Stringa | N/A | |
| | last_name | Stringa | N/A | |
| | unique_name | Stringa | N/A | |
| | e-mail | Stringa | N/A | |
| | alternate_emails | List | N/D | |
| | address_line_1 | Stringa | N/A | |
| | address_line_2 | Stringa | N/A | |
| | città | Stringa | N/A | |

| Entità | Campo | Tipo di dati | Operatori supportati | |
|---|---|---|---|---|
| | state_province_region | Stringa | N/A | |
| | paese | Stringa | N/A | |
| | postal_code | Stringa | N/A | |
| | phone_number | Stringa | N/A | |
| | whatsapp | Stringa | N/A | |
| | line | Stringa | N/A | |
| | facebook | Stringa | N/A | |
| | list_ids | List | N/D | |
| | custom_fields | Struct | N/D | |
| | created_at | Stringa | N/A | |
| | updated_at | Stringa | N/A | |
| | _metadata | Struct | N/D | |
| | event_timestamp | DateTime | BETWEEN | |
| Categoria | categorie | List | N/D | |
| Statistiche | data | Stringa | N/A | |
| | stats | List | N/D | |
| | start_date | DateTime | EQUAL_TO, BETWEEN | |
| | aggregated_by | Stringa | EQUAL_TO | |

| Entità | Campo | Tipo di dati | Operatori supportati | |
|---|---|---|---|---|
| Gruppi di annullamento dell'iscrizione | id | Numero intero | EQUAL_TO | |
| | nome | Stringa | N/A | |
| | description | Stringa | N/A | |
| | last_email_sent_at | Numero intero | N/D | |
| | is_default | Booleano | N/D | |
| | unsubscribes | Numero intero | N/D | |

> **Note**
>
> I tipi di dati Struct e List vengono convertiti in tipi di dati String e il tipo di DateTime dati viene convertito in Timestamp nella risposta dei connettori.

Query di partizionamento

SendGrid non supporta il partizionamento basato su filtri o il partizionamento basato su record.

## SendGrid opzioni di connessione

Di seguito sono elencate le opzioni di connessione per SendGrid:

- `ENTITY_NAME`(String): (obbligatorio) utilizzato per la lettura. Il nome dell'oggetto in SendGrid.
- `API_VERSION`(String) - (Obbligatorio) Usato per la lettura. SendGrid Versione dell'API Rest che desideri utilizzare.
- `INSTANCE_URL`(String): (obbligatorio) utilizzato per la lettura. Un URL di SendGrid istanza valido.
- `SELECTED_FIELDS`(List<String>): predefinito: vuoto (SELECT *). Utilizzato per la lettura. Colonne da selezionare per l'oggetto.
- `FILTER_PREDICATE`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Dev'essere in formato Spark SQL.

- `QUERY`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Query Spark SQL completa.

### SendGrid limitazioni

Di seguito sono riportate le limitazioni o le note relative a SendGrid:

- Il pull incrementale è supportato solo dall'entità Stats sul campo `start_date` e dall'entità Contact sul campo `event_timestamp`.
- L'impaginazione è supportata solo dalle entità Marketing Campaign Stats (Automations), Marketing Campaign Stats (Single Sends), Single Sends e Lists.
- Per l'entità Stats, `start_date` è un parametro di filtro obbligatorio.
- Una chiave API con accesso limitato non può supportare l'accesso in lettura per l'API Email e le entità Stats. Utilizza una chiave API con accesso completo. Per ulteriori informazioni, consultare la panoramica su API.

## Connessione a ServiceNow

ServiceNow è una piattaforma SaaS basata su cloud per l'automazione dei flussi di lavoro di gestione IT. La ServiceNow piattaforma si integra facilmente con altri strumenti, consentendo agli utenti di gestire progetti, team e interazioni con i clienti utilizzando una varietà di app e plugin. Come ServiceNow utente puoi connetterti AWS Glue al tuo account. ServiceNow Quindi, puoi utilizzarlo ServiceNow come fonte di dati nei tuoi lavori ETL. Esegui questi processi per trasferire dati tra ServiceNow AWS servizi o altre applicazioni supportate.

Argomenti

- AWS Glue supporto per ServiceNow
- Policy contenenti le operazioni API per creare e utilizzare le connessioni
- Configurazione ServiceNow
- Configurazione delle connessioni ServiceNow
- Lettura da ServiceNow entità
- ServiceNow opzioni di connessione
- Limitazioni e note per il ServiceNow connettore

## AWS Glue supporto per ServiceNow

AWS Glue supporta ServiceNow quanto segue:

Supportato come origine?

Sì. È possibile utilizzare i job AWS Glue ETL da ServiceNow cui interrogare i dati.

Supportato come destinazione?

No.

Versioni ServiceNow API supportate

Sono supportate le seguenti versioni ServiceNow API:

- v2

Per il supporto delle entità per le versioni specifiche, consulta Entità supportate per l'origine.

## Policy contenenti le operazioni API per creare e utilizzare le connessioni

La seguente policy di esempio descrive le autorizzazioni AWS IAM richieste per la creazione e l'utilizzo delle connessioni. Se si sta creando un nuovo ruolo, creare una policy che contenga quanto segue:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:ListConnectionTypes",
        "glue:DescribeConnectionType",
        "glue:RefreshOAuth2Tokens",
        "glue:ListEntities",
        "glue:DescribeEntity"
      ],
      "Resource": "*"
    }
```

```
  ]
}
```

In alternativa, se non desideri utilizzare il metodo precedente, utilizza le seguenti policy IAM gestite:

- AWSGlueServiceRole— Garantisce l'accesso alle risorse che AWS Glue i vari processi richiedono per l'esecuzione per conto dell'utente. Queste risorse includono AWS Glue Amazon S3, IAM, CloudWatch Logs e Amazon. EC2 Se segui la convenzione di denominazione delle risorse specificata in questa politica, AWS Glue i processi dispongono delle autorizzazioni richieste. Questa policy è in genere collegata ai ruoli specificati quando si definiscono crawler, processi ed endpoint di sviluppo.
- AWSGlueConsoleFullAccess— Concede l'accesso completo alle AWS Glue risorse quando un'identità a cui è allegata la policy utilizza la AWS console di gestione. Se si segue la convenzione per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questa policy è in genere associata agli utenti della AWS Glue console.

## Configurazione ServiceNow

Prima di poter AWS Glue utilizzare il trasferimento di dati da ServiceNow, è necessario soddisfare i seguenti requisiti:

Requisiti minimi

I requisiti minimi sono i seguenti:

- Hai un ServiceNow account con email e password. Per ulteriori informazioni, consulta Creare un ServiceNow account.
- Il tuo ServiceNow account è abilitato per l'accesso all'API. Tutti gli utilizzi dell' ServiceNow API sono disponibili senza costi aggiuntivi.

Se soddisfi questi requisiti, sei pronto per connetterti AWS Glue al tuo ServiceNow account.

Creare un ServiceNow account

Per creare un ServiceNow account:

1. Andare sulla pagina di registrazione su servicenow.com, inserire i propri dati e fare clic su Continua.

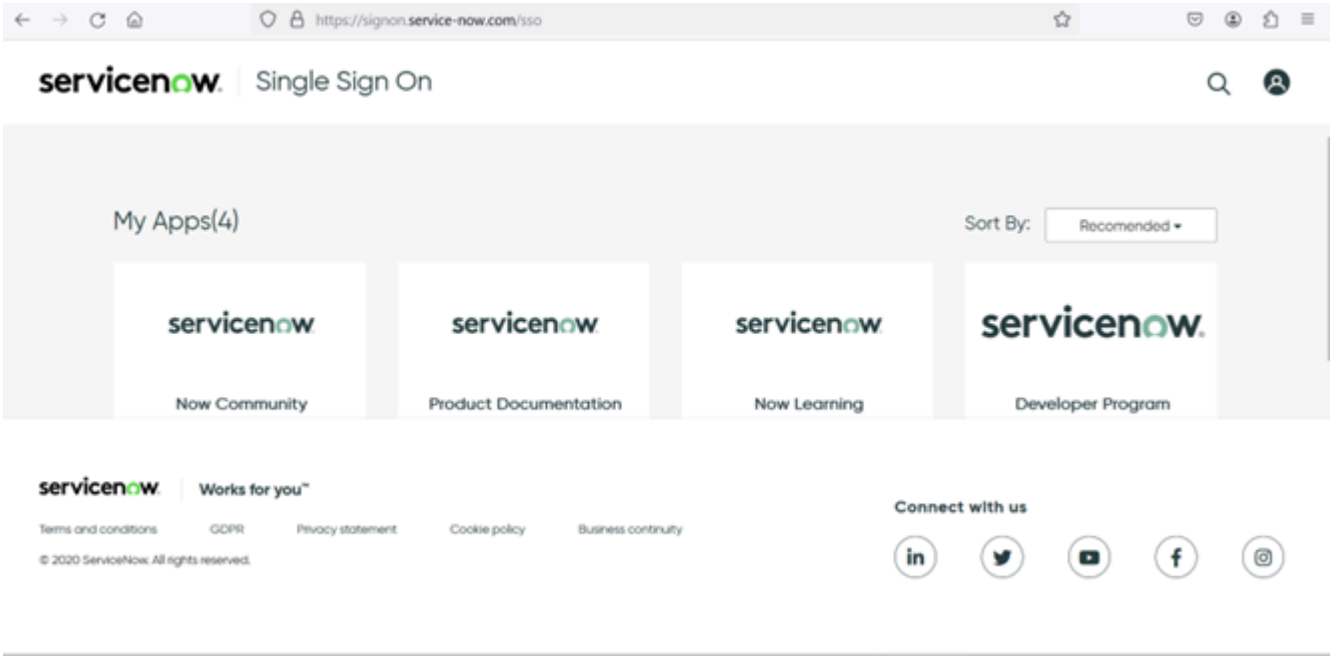
2. Quando si riceve un codice di verifica all'indirizzo e-mail registrato, inserire quel codice e scegliere Verifica.
3. Configurare l'autenticazione a più fattori o saltare l'operazione.

Il tuo account viene creato e ServiceNow mostra il tuo profilo.

Creazione di un'istanza per ServiceNow sviluppatori

Richiedi un'istanza per sviluppatori dopo aver effettuato l'accesso a ServiceNow.

1. Nella pagina di ServiceNow accesso, inserisci le credenziali del tuo account.
2. Scegli il programma per ServiceNow sviluppatori.




3. Scegliere Richiedi istanza in alto a destra.
4. Inserire le responsabilità lavorative. Indicare l'accettazione dei termini di utilizzo e scegliere Termina configurazione.
5. Dopo aver creato l'istanza, annotare le credenziali e l'URL dell'istanza.

Recupero delle credenziali BasicAuth

Per recuperare le credenziali per l'autenticazione di base per un account gratuito:

1. Nella pagina di ServiceNow accesso, inserisci le credenziali del tuo account.

2. Nella home page scegliere la sezione di modifica del profilo (in alto a destra) e scegliere Gestisci password dell'istanza.
3. Recuperare le credenziali di accesso come nome utente, password e URL dell'istanza.

> **Note**
>
> Se la MFA è abilitata per l'account, aggiungere il token MFA alla fine della password dell'utente nell'autenticazione di base: <username>:<password><MFA Token>

Per ulteriori informazioni, consulta Creazione di applicazioni nella ServiceNow documentazione.

Creazione di credenziali OAuth 2.0

Per utilizzare OAuth2 2.0 nel ServiceNow connettore, è necessario creare un client in entrata () per generare l'ID client e il segreto del client:

1. Nella pagina di ServiceNow accesso, inserisci le credenziali del tuo account.
2. Nella home page, scegliere Avviare compilazione.
3. Nella pagina App Engine Studio, cercare Registro delle applicazioni.
4. Scegliere Nuovo in alto a destra.
5. Scegli l'opzione Crea un endpoint OAuth API per client esterni.
6. Apporta le modifiche necessarie alla OAuth configurazione e scegli Aggiorna.

   Esempio di URL di reindirizzamento: https://us-east-1.console.aws.amazon.com/gluestudio/ oauth
7. Seleziona l'app OAuth client appena creata per recuperare l'ID client e il segreto del cliente.
8. Memorizzare l'ID client e il segreto del client per l'ulteriore elaborazione.

Per configurare OAuth in un account sviluppatore non di produzione:

1. Crea un profilo di autenticazione utilizzando l'argomento Crea un profilo di autenticazione nella ServiceNow documentazione.
2. Nel Profilo di autenticazione per OAuth, seleziona Digita come OAuth e seleziona il client in entrata creato sopra per impostare l'entità. OAuth
3. Se sono presenti più client, è necessario creare più profili di autenticazione per impostare l' OAuth entità richiesta nel profilo di autenticazione.

4. Se non è configurata, creare una policy di accesso a REST API per consentire l'accesso alla tabella API. Consultare Creazione di una policy di accesso alla REST API.

## Configurazione delle connessioni ServiceNow

Il tipo di concessione determina la modalità di AWS Glue comunicazione con ServiceNow cui richiedere l'accesso ai dati. La tua scelta influisce sui requisiti che devi soddisfare prima di creare la connessione. ServiceNow supporta solo il tipo di concessione AUTHORIZATION_CODE per la versione 2.0. OAuth

- Questo tipo di concessione è considerato «a tre gambe» in OAuth quanto si basa sul reindirizzamento degli utenti a un server di autorizzazione di terze parti per autenticare l'utente. Viene utilizzato durante la creazione di connessioni tramite la console. AWS Glue La AWS Glue console reindirizzerà l'utente al ServiceNow punto in cui deve effettuare il login e consentirà AWS Glue alle autorizzazioni richieste di accedere alla propria ServiceNow istanza.
- Gli utenti possono comunque scegliere di creare la propria app connessa ServiceNow e fornire il proprio ID client e il segreto del client durante la creazione di connessioni tramite la AWS Glue console. In questo scenario, verranno comunque reindirizzati all'accesso e ServiceNow all'autorizzazione ad accedere AWS Glue alle proprie risorse.
- Questo tipo di concessione genera un token di aggiornamento e un token di accesso. Il token di accesso dura poco tempo e può essere aggiornato automaticamente senza l'interazione dell'utente utilizzando il token di aggiornamento.
- Per la ServiceNow documentazione pubblica sulla creazione di un'app connessa per il OAuth flusso del codice di autorizzazione, vedi Configurazione. OAuth

Per configurare una ServiceNow connessione:

1. In AWS Secrets Manager, crea un segreto con i seguenti dettagli:
   a. Per l'autenticazione di base, il segreto deve contenere l'app connessa Consumer Secret con `USERNAME` e `PASSWORD` come chiave.
   b. Per un tipo di concessione con codice di autorizzazione, il segreto deve contenere l'app connessa Consumer Secret con `USER_MANAGED_CLIENT_APPLICATION_CLIENT_SECRET` come chiave.
   c. Nota: è necessario creare un segreto per la connessione in AWS Glue.

2. In AWS Glue Glue Studio, crea una connessione in Connessioni dati seguendo i passaggi seguenti:
   a. Quando si seleziona un tipo di connessione, selezionare ServiceNow.
   b. Fornisci l'INSTANCE_URL dell' ServiceNow istanza a cui desideri connetterti.
   c. Seleziona il ruolo AWS IAM che AWS Glue può assumere e dispone delle autorizzazioni per le seguenti azioni:

      JSON

      ```
      {
        "Version":"2012-10-17",
        "Statement": [
          {
            "Effect": "Allow",
            "Action": [
              "secretsmanager:DescribeSecret",
              "secretsmanager:GetSecretValue",
              "secretsmanager:PutSecretValue",
              "ec2:CreateNetworkInterface",
              "ec2:DescribeNetworkInterfaces",
              "ec2:DeleteNetworkInterface"
            ],
            "Resource": "*"
          }
        ]
      }
      ```

   d. Selezionare il tipo di autenticazione da utilizzare per questa connessione in AWS Glue.
      i. Autenticazione di base: questo tipo di autenticazione funziona bene per gli scenari di automazione in quanto consente di utilizzare nome utente e password in anticipo con le autorizzazioni di un particolare utente nell'istanza. ServiceNow AWS Glue è in grado di utilizzare il nome utente e la password per l'autenticazione. ServiceNow APIs Inserire i seguenti input solo in caso di autenticazione di base: `Username` e `Password`.
      ii. OAuth2: inserisci i seguenti input solo in caso di OAuth2:`ClientId`,,`ClientSecret`,`Authorization URL`. `Authorization Token URL`
   e. Seleziona quello `secretName` che vuoi usare per questa connessione per AWS Glue inserire i token.
   f. Selezionare le opzioni di rete se si desidera utilizzare la propria rete.
3. Concedi il permesso di lettura `secretName` al ruolo IAM associato al tuo AWS Glue lavoro.

## Lettura da ServiceNow entità

Prerequisito

Un oggetto ServiceNow Tables da cui desideri leggere. È necessario il nome dell'oggetto, come pa_bucket o incident.

Esempio:

```
servicenow_read = glueContext.create_dynamic_frame.from_options(
    connection_type="servicenow",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "pa_buckets",
        "API_VERSION": "v2"
        "instanceUrl": "https://<instance-name>.service-now.com"
    }
)
```

ServiceNow dettagli dell'entità e del campo:

Per le seguenti entità, ServiceNow fornisce endpoint per recuperare i metadati in modo dinamico, in modo che il supporto dell'operatore venga acquisito a livello di tipo di dati per ciascuna entità.

| Entità | Tipo di dati | Operatori supportati |
|---|---|---|
| Tabelle (entità dinamiche) | Numero intero | =, !=, <, <=, >, >=, BETWEEN |
| | BigDecimal | =, !=, <, <=, >, >=, BETWEEN |
| | Float | =, !=, <, <=, >, >=, BETWEEN |
| | Long | =, !=, <, <=, >, >=, BETWEEN |
| | Data | =, !=, <, <=, >, >=, BETWEEN |
| | DateTime | =, !=, <, <=, >, >=, BETWEEN |
| | Booleano | =, != |
| | Stringa | =, !=, <, <=, >, >=, BETWEEN, LIKE |

| Entità | Tipo di dati | Operatori supportati |
|---|---|---|
| | Struct | N/D |

> **Note**
>
> Il tipo di dati Struct viene convertito nel tipo di dati String nella risposta del connettore.

> **Note**
>
> `DML_STATUS`è un attributo aggiuntivo definito dall'utente utilizzato per tenere traccia dei record. CREATED/UPDATED

Query di partizionamento

Partizionamento basato sul campo:

Puoi fornire le opzioni Spark aggiuntive `PARTITION_FIELD`, `LOWER_BOUND`, `UPPER_BOUND` e `NUM_PARTITIONS` se desideri utilizzare la concorrenza in Spark. Con questi parametri, la query originale viene suddivisa in `NUM_PARTITIONS` query secondarie che possono essere eseguite contemporaneamente dalle attività Spark.

| Nome dell'entità | Campi di partizionamento | Tipo di dati |
|---|---|---|
| Entità dinamica | sys_mod_count | Numero intero |
| | sys_created_on, sys_updated_on | DateTime |

- `PARTITION_FIELD`: il nome del campo da utilizzare per partizionare la query.
- `LOWER_BOUND`: un valore limite inferiore inclusivo del campo di partizione scelto.

  Per il campo Datetime, accettiamo il formato di timestamp Spark utilizzato nelle query SQL. SPark

  Esempi di valori validi:

```
"2024-01-30T06:47:51.000Z"
```

- UPPER_BOUND: un valore limite superiore esclusivo del campo di partizione scelto.
- NUM_PARTITIONS: il numero di partizioni.

La seguente tabella descrive i dettagli del campo di supporto del partizionamento delle entità:

Esempio:

```
servicenow_read = glueContext.create_dynamic_frame.from_options(
    connection_type="servicenow",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "pa_buckets",
        "API_VERSION": "v2",
        "instanceUrl": "https://<instance-name>.service-now.com"
        "PARTITION_FIELD": "sys_created_on"
        "LOWER_BOUND": "2024-01-30T06:47:51.000Z"
        "UPPER_BOUND": "2024-06-30T06:47:51.000Z"
        "NUM_PARTITIONS": "10"
    }
```

Partizionamento basato su record:

Puoi fornire l'opzione Spark aggiuntiva NUM_PARTITIONS se desideri utilizzare la concorrenza in Spark. Con questo parametro, la query originale viene suddivisa in NUM_PARTITIONS query secondarie che possono essere eseguite contemporaneamente dalle attività Spark.

Nel partizionamento basato sui record, il numero totale di record presenti viene richiesto dall'API e diviso per il numero fornito. ServiceNow NUM_PARTITIONS Il numero di record risultante viene quindi recuperato contemporaneamente da ciascuna query secondaria.

- NUM_PARTITIONS: il numero di partizioni.

Esempio:

```
servicenow_read = glueContext.create_dynamic_frame.from_options(
    connection_type="servicenow",
    connection_options={
```

```
        "connectionName": "connectionName",
        "ENTITY_NAME": "pa_buckets",
        "API_VERSION": "v2",
        "instanceUrl": "https://<instance-name>.service-now.com"
        "NUM_PARTITIONS": "2"
    }
```

## ServiceNow opzioni di connessione

Di seguito sono elencate le opzioni di connessione per ServiceNow:

- `ENTITY_NAME`(String): (obbligatorio) utilizzato per la lettura. Il nome dell'oggetto in ServiceNow.
- `API_VERSION`(String) - (Obbligatorio) Usato per la lettura. ServiceNow Versione dell'API Rest che desideri utilizzare. Ad esempio: v1, v2, v3, v4.
- `SELECTED_FIELDS`(Elenco<Stringa>): predefinito: vuoto (SELECT *). Utilizzato per la lettura. Colonne da selezionare per l'oggetto.
- `FILTER_PREDICATE`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Dev'essere in formato Spark SQL.
- `QUERY`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Query Spark SQL completa.
- `PARTITION_FIELD`(Stringa): utilizzato per la lettura. Campo da utilizzare per partizionare la query.
- `LOWER_BOUND`(Stringa): utilizzato per la lettura. Un valore limite inferiore inclusivo del campo di partizione scelto. Ad esempio: 2024-01-30T06:47:51.000Z.
- `UPPER_BOUND`(Stringa): utilizzato per la lettura. Un valore limite superiore esclusivo del campo di partizione scelto. Ad esempio: 2024-06-30T06:47:51.000Z.
- `NUM_PARTITIONS`(Numero intero): predefinito: 1. Utilizzato per la lettura. Numero di partizioni da leggere. Ad esempio: 10.
- `INSTANCE_URL`(String) - (Obbligatorio) Un URL di ServiceNow istanza valido con formato https://<instance-name>.service-now.com.
- `PAGE_SIZE`(Integer): definisce la dimensione della pagina per la query dei record. La dimensione della pagina predefinita è 1.000. Quando viene specificata una dimensione di pagina, ServiceNow restituisce solo il numero definito di record per chiamata API, anziché l'intero set di dati. Il connettore continuerà a fornire il numero totale di record e gestirà l'impaginazione utilizzando la dimensione della pagina specificata. Se è necessaria una dimensione della pagina più grande, si può scegliere qualsiasi valore fino a 10.000, che è il massimo consentito. Qualsiasi dimensione della pagina specificata superiore a 10.000 verrà ignorata. Al contrario, il sistema utilizzerà la

dimensione della pagina massima consentita. È possibile specificare la dimensione della pagina nell'interfaccia utente di AWS Glue Studio UI aggiungendo un'opzione di connessione `PAGE_SIZE` con il valore desiderato. Ad esempio: 5.000.

### Limitazioni e note per il ServiceNow connettore

Di seguito sono riportate le limitazioni o le note relative al ServiceNow connettore:

- Secondo la documentazione SaaS, `sys_created_on`, `sys_updated_on` e `sys_mod_count` sono campi generati dal sistema. Il connettore si affida a APIs SaaS per fornire questi campi nel corpo della risposta.
  - Se SaaS non genera questi campi per nessuna entità, il partizionamento basato su filtri non può essere supportato.
- Se SaaS APIs non restituisce `sys_created_on` e `sys_updated_on` i campi della risposta `DML_STATUS` non possono essere calcolati.
- Migliorare le prestazioni e l'efficienza di lettura
  - Il ServiceNow connettore ora ordina automaticamente i record in ordine crescente in base al `sys_id` campo (deve essere presente nei metadati) quando l'utente non ha specificato alcuna clausola ORDER BY. In questo caso, i record verranno impaginati mediante la nuova impaginazione ottimizzata basata su un set di chiavi.
  - Se viene specificata la clausola ORDER BY, la nuova ottimizzazione non verrà utilizzata e i record verranno recuperati utilizzando il metodo esistente (impaginazione Order By definita dall'utente e basata su Offset-Limit).

## Connessione a Slack in AWS Glue Studio

Slack è un'app per la comunicazione aziendale che consente agli utenti di inviare messaggi e allegati tramite diversi canali pubblici e privati. Se sei un utente Slack, puoi connetterti AWS Glue al tuo account Slack. Quindi, puoi usare Slack come origine dati nei tuoi processi ETL. Esegui questi processi per trasferire dati tra Slack e i AWS servizi o altre applicazioni supportate.

Argomenti

- AWS Glue supporto per Slack
- Policy contenenti le operazioni API per creare e utilizzare le connessioni
- Configurazione di Slack

- Configurazione delle connessioni Slack
- Lettura da entità di Slack
- Opzioni di connessione di Slack
- Limitazioni
- Creazione di un nuovo account Slack e configurazione dell'app client

## AWS Glue supporto per Slack

AWS Glue supporta Slack come segue:

Supportato come origine?

Sì. Puoi utilizzare i job AWS Glue ETL per interrogare i dati da Slack.

Supportato come destinazione?

No.

Versioni dell'API Slack supportate

API Slack v2.

## Policy contenenti le operazioni API per creare e utilizzare le connessioni

Nella seguente policy di esempio sono descritte le autorizzazioni IAM per creare e utilizzare le connessioni. Se si sta creando un nuovo ruolo, creare una policy che contenga quanto segue:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:ListConnectionTypes",
        "glue:DescribeConnectionType",
        "glue:RefreshOAuth2Tokens",
        "glue:ListEntities",
```

```
        "glue:DescribeEntity"
      ],
      "Resource": "*"
    }
  ]
}
```

È anche possibile utilizzare le seguenti policy IAM per autorizzare l'accesso:

- AWSGlueServiceRole— Garantisce l'accesso alle risorse che AWS Glue i vari processi richiedono per l'esecuzione per conto dell'utente. Queste risorse includono AWS Glue Amazon S3 Amazon CloudWatch Logs, IAM e Amazon EC2. Se segui la convenzione di denominazione delle risorse specificata in questa politica, AWS Glue i processi dispongono delle autorizzazioni necessarie. Questa policy è in genere collegata ai ruoli specificati quando si definiscono crawler, processi ed endpoint di sviluppo.
- AWSGlueConsoleFullAccess— Concede l'accesso completo alle AWS Glue risorse quando un'identità a cui è allegata la policy utilizza la AWS console di gestione. Se si segue la convenzione per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questa policy è in genere associata agli utenti della AWS Glue console.

## Configurazione di Slack

Prima di poter AWS Glue utilizzare il trasferimento di dati da o verso Slack, devi soddisfare questi requisiti:

Requisiti minimi

- Avere un account Slack. Per ulteriori informazioni, consulta Creazione di un nuovo account Slack e configurazione dell'app client.

Se soddisfi questi requisiti, sei pronto per connetterti AWS Glue al tuo account Slack.

## Configurazione delle connessioni Slack

Slack supporta il tipo di `AUTHORIZATION_CODE` concessione per OAuth 2.

Questo tipo di concessione è considerato «a tre gambe» in OAuth quanto si basa sul reindirizzamento degli utenti al server di autorizzazione di terze parti per autenticare l'utente.

Viene utilizzato durante la creazione di connessioni tramite la console. AWS Glue La AWS Glue Console reindirizzerà l'utente a Slack, dove l'utente deve effettuare il login e consentire AWS Glue le autorizzazioni richieste per accedere alla propria istanza Slack.

Gli utenti possono comunque scegliere di creare la propria app connessa in Slack e fornire il proprio ID client e il segreto del client quando creano connessioni tramite la Console. AWS Glue In questo scenario, verranno comunque reindirizzati a Slack per effettuare il login e autorizzare l'accesso AWS Glue alle proprie risorse.

Questo tipo di concessione genera un token di aggiornamento e uno di accesso. Il token di accesso scade 1 ora dopo la creazione. È possibile recuperare un nuovo token di accesso utilizzando il token di aggiornamento.

Per ulteriori informazioni sulla creazione di un'app connessa per il OAuth flusso del codice di autorizzazione, consulta l'API Slack.

Per configurare una connessione Slack:

1. In AWS Secrets Manager, crea un segreto con i seguenti dettagli. È necessario creare un segreto per la connessione in AWS Glue.
   a. Per le app connesse gestite dal cliente: il segreto deve contenere l'app connessa Consumer Secret con `USER_MANAGED_CLIENT_APPLICATION_CLIENT_SECRET` come chiave.
2. In AWS Glue Glue Studio, crea una connessione in Connessioni dati seguendo i passaggi seguenti:
   a. In Tipo di connessione, selezionare Slack.
   b. Fornire l'ambiente Slack.
   c. Seleziona il ruolo IAM che AWS Glue può assumere e dispone delle autorizzazioni per le seguenti azioni:

      JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:DescribeSecret",
        "secretsmanager:GetSecretValue",
```

```
            "secretsmanager:PutSecretValue",
            "ec2:CreateNetworkInterface",
            "ec2:DescribeNetworkInterfaces",
            "ec2:DeleteNetworkInterface"
        ],
        "Resource": "*"
    }
  ]
}
```

d. Seleziona quello `secretName` che desideri utilizzare per questa connessione per AWS Glue inserire i token.

e. Selezionare le opzioni di rete se si desidera utilizzare la propria rete.

3. Concedi il permesso di lettura `secretName` al ruolo IAM associato al tuo AWS Glue lavoro.

## Lettura da entità di Slack

### Prerequisiti

- Un oggetto di Slack da cui si desidera leggere.

### Entità supportate

| Entità | Può essere Filtrato | Supporta Limite | Supporta Ordina per | Supporta Seleziona * | Supporta Partizionamento |
|---|---|---|---|---|---|
| conversations | Sì | Sì | No | Sì | Sì |

### Esempio

```
slack_read = glueContext.create_dynamic_frame.from_options(
    connection_type="slack",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "conversations/C058W38R5J8"
    }
)
```

Dettagli dell'entità e del campo di Slack

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| conversations | attachments | List | N/A |
| conversations | bot_id | Stringa | N/A |
| conversations | blocks | List | N/A |
| conversations | client_msg_id | Stringa | N/A |
| conversations | is_starred | Booleano | N/A |
| conversations | last_read | Stringa | N/A |
| conversations | latest_reply | Stringa | N/A |
| conversations | reactions | List | N/A |
| conversations | replies | List | N/A |
| conversations | reply_count | Numero intero | N/A |
| conversations | reply_users | List | N/A |
| conversations | reply_users_count | Numero intero | N/A |
| conversations | subscribed | Booleano | N/A |
| conversations | subtype | Stringa | N/A |
| conversations | testo | Stringa | N/A |
| conversations | team | Stringa | N/A |
| conversations | thread_ts | Stringa | N/A |
| conversations | ts | Stringa | EQUAL_TO, BETWEEN, LESS_THAN , LESS_THAN |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | | | _OR_EQUAL_TO, GREATER_T HAN, GREATER_T HAN_OR_EQUAL_TO |
| conversations | tipo | Stringa | N/A |
| conversations | user | Stringa | N/A |
| conversations | inviter | Stringa | N/A |
| conversations | root | Struct | N/A |
| conversations | is_locked | Booleano | N/A |
| conversations | files | List | N/A |
| conversations | stanza | Struct | N/A |
| conversations | caricamento | Booleano | N/A |
| conversations | display_as_bot | Booleano | N/A |
| conversations | channel | Stringa | N/A |
| conversations | no_notifications | Booleano | N/A |
| conversations | permalink | Stringa | N/A |
| conversations | pinned_to | List | N/A |
| conversations | pinned_info | Struct | N/A |
| conversations | edited | Struct | N/A |
| conversations | app_id | Stringa | N/A |
| conversations | bot_profile | Struct | N/A |
| conversations | metadati | Struct | N/A |

### Query di partizionamento

Se si desidera utilizzare la simultaneità in Spark, possono essere fornite opzioni Spark `PARTITION_FIELD`, `LOWER_BOUND`, `UPPER_BOUND`, `NUM_PARTITIONS` aggiuntive. Con questi parametri, la query originale viene suddivisa in `NUM_PARTITIONS` query secondarie che possono essere eseguite contemporaneamente dalle attività Spark.

- `PARTITION_FIELD`: il nome del campo da utilizzare per partizionare la query.
- `LOWER_BOUND`: un valore limite inferiore inclusivo del campo di partizione scelto.

  Per la data, accettiamo il formato di data di Spark utilizzato nelle query di Spark SQL. Esempio di valore valido: `"2024-07-01T00:00:00.000Z"`.
- `UPPER_BOUND`: un valore limite superiore esclusivo del campo di partizione scelto.
- `NUM_PARTITIONS`: numero di partizioni.

I dettagli di supporto del campo di partizionamento per entità sono riportati nella tabella di seguito.

| Nome entità | Campo di partizionamento | Tipo di dati |
|---|---|---|
| conversations | ts | Stringa |

### Esempio

```
slack_read = glueContext.create_dynamic_frame.from_options(
    connection_type="slack",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "conversations/C058W38R5J8",
        "PARTITION_FIELD": "ts"
        "LOWER_BOUND": "2022-12-01T00:00:00.000Z"
        "UPPER_BOUND": "2024-09-23T15:00:00.000Z"
        "NUM_PARTITIONS": "2"
    }
)
```

## Opzioni di connessione di Slack

Di seguito sono riportate le opzioni di connessione per Slack:

- `ENTITY_NAME`(String): (obbligatorio) utilizzato per la lettura. Il nome dell'entità supportata. Esempio: `conversations/C058W38R5J8`.
- `SELECTED_FIELDS`(List<String>): predefinito: vuoto (SELECT *). Utilizzato per la lettura. Campi da selezionare per l'oggetto.
- `FILTER_PREDICATE`(String): predefinito, vuoto. Utilizzato per la lettura. Dev'essere in formato Spark SQL.
- `QUERY`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Query Spark SQL completa.
- `NUM_PARTITIONS`(Integer): predefinito: 1. Utilizzato per la lettura. Numero di partizioni da leggere.

## Limitazioni

Di seguito sono riportate le limitazioni per il connettore di Slack:

- Il partizionamento basato sui record non è supportato in quanto il connettore non fornisce alcun mezzo per recuperare il numero totale di record (messaggi) disponibili in una determinata conversazione.

## Creazione di un nuovo account Slack e configurazione dell'app client

Creazione di un account Slack

1. Aprire la home page di Slack per creare un account.
2. Scegliere ISCRIVITI CON INDIRIZZO E-MAIL. Immettere l'ID e-mail e scegliere Continua.
3. Immettere il codice di 6 caratteri inviato all'indirizzo e-mail, che ti permetterà di un'area di lavoro o di unirsi a un'area di lavoro esistente.
4. Scegliere Crea un'area di lavoro per creare una nuova area di lavoro. Sarà necessario rispondere ad alcune domande nell'ambito del processo di configurazione.
   - Nome dell'azienda
   - Nome
   - Per aggiungere colleghi tramite e-mail
   - Su cosa sta lavorando il team? (Questo sarà il nome del canale)
5. Compilare i campi per queste domande e continuare. L'account è ora pronto per l'uso.

Creazione di un'app per sviluppatori Slack

1. Accedere all'account Slack e all'area di lavoro.
2. Dal menu dell'area di lavoro, selezionare Strumenti e impostazioni, quindi selezionare Gestisci app.
3. Dal menu Slack App Directory, selezionare Crea.
4. Nella pagina Le tue app, selezionare Crea un'app.
5. Nella pagina Crea un'app, selezionare Da zero.
6. Nella finestra di dialogo Dai un nome all'app e scegli l'area di lavoro che si apre, aggiungere un nome per l'app e Scegli un'area di lavoro in cui distribuire l'app. Poi selezionare Crea app.
7. Annotare l'ID cliente e il segreto visualizzati nelle credenziali dell'app
8. Nella barra laterale OAuth & Autorizzazioni, vai su Ambiti e scegli Aggiungi un ambito. OAuth Puoi aggiungere il reindirizzamento URLs alla tua app per configurare la generazione automatica del pulsante «Aggiungi a Slack» o per distribuire l'app. Scorri verso l'alto fino alla URLs sezione Reindirizzamento e scegli Aggiungi nuovo URL di reindirizzamento e salva.
9. Quindi, scorri fino alla sezione OAuth Tokens for Your Workspace e scegli Installa in Workspace.
10. Nella finestra di dialogo che si apre per informare che l'app creata richiede l'autorizzazione per accedere all'area di lavoro Slack a cui si desidera connetterla, selezionare Consenti.
11. Una volta completato con successo, la console mostrerà la schermata OAuth Tokens for Your Workspace.
12. Dalla schermata OAuth Tokens for Your Workspace, copia e salva il OAuth token che utilizzerai per connetterti a AWS Glue
13. Successivamente, si recupera l'ID del team Slack. Dal menu dell'area di lavoro di Slack, selezionare Strumenti e impostazioni, quindi selezionare Gestisci app. Apparirà l'ID del team nell'URL della pagina che si apre.
14. Per distribuire pubblicamente l'app, è possibile attivarla andando sul pulsante Gestisci distribuzione nella barra laterale. Scorrere verso il basso fino alla sezione Condividi l'app con altre aree di lavoro e scegliere Rimuovi informazioni codificate. Fornire il consenso e scegliere Distribuzione pubblica attiva.
15. L'app è ora distribuita pubblicamente. Per accedere all'entità APIs, l'app deve essere aggiunta a ogni canale di workspace da cui l'utente desidera accedere.
16. Accedere all'account Slack e aprire l'area di lavoro con il canale a cui deve essere effettuato l'accesso.

17. Nell'area di lavoro, aprire il canale a cui l'app desidera accedere e scegliere il titolo del canale. Selezionare la scheda Integrazioni dal pop-up e aggiungere l'app. In questo modo, l'app è integrata con il canale per avere accesso all'API.

    L'ID client OAuth 2.0 deve avere uno o più URLs reindirizzamenti autorizzati. I reindirizzamenti URLs hanno il seguente formato:

    > **Note**
    >
    > I reindirizzamenti di Appflow URLs sono soggetti a modifiche, i reindirizzamenti postali URLs per AWS Glue la piattaforma sono disponibili. Client ID e Client Secret provengono dalle impostazioni del tuo ID client OAuth 2.0.

| L'URL di reindirizzamento può essere uno dei seguenti | |
|---|---|
| URL di reindirizzamento per l'ambiente Gamma | |
| https://us-west-2.console.aws.amazon.com/appflow/oauth | https://us-east-1.awsc-integ.aws.amazon.com/appflow/giuramento |
| https://us-east-2.console.aws.amazon.com/appflow/giuramento | |
| https://us-west-1.console.aws.amazon.com/appflow/giuramento | |
| https://ap-south-1.console.aws.amazon.com/appflow/giuramento | |
| https://ap-southeast-1.console.aws.amazon.com/appflow/giuramento | |
| https://ap-southeast-2.console.aws.amazon.com/appflow/giuramento | |

| L'URL di reindirizzamento può essere uno dei seguenti | |
|---|---|
| URL di reindirizzamento per l'ambiente Gamma | |
| https://ap-northeast-1.console.aws.amazon.com/appflow/giuramento | |
| https://ap-northeast-2.console.aws.amazon.com/appflow/giuramento | |
| https://ca-central-1.console.aws.amazon.com/appflow/giuramento | |
| https://eu-central-1.console.aws.amazon.com/appflow/giuramento | |
| https://eu.west-1.console.aws.amazon.com/appflow/giuramento | |
| https://eu-west-2.console.aws.amazon.com/appflow/giuramento | |
| https://eu-west-3.console.aws.amazon.com/appflow/giuramento | |
| https://sa-east-1.console.aws.amazon.com/appflow/giuramento | |
| https://us-west-2.awsc-integ.aws.amazon.com/appflow/giuramento | |
| https://af-south-1.console.aws.amazon.com/appflow/giuramento | |

# Connessione a Smartsheet

Smartsheet è un prodotto SaaS per la gestione del lavoro e la collaborazione. In sostanza, Smartsheet consente agli utenti di usare oggetti simili a fogli di calcolo per creare, archiviare e utilizzare i dati aziendali.

Argomenti


- AWS Glue supporto per Smartsheet
- Policy contenenti le operazioni API per creare e utilizzare le connessioni
- Configurazione di Smartsheet
- Configurazione delle connessioni di Smartsheet
- Lettura da entità Smartsheet
- Opzioni di connessione di Smartsheet
- Creazione di un account Smartsheet
- Limitazioni


## AWS Glue supporto per Smartsheet

AWS Glue supporta Smartsheet come segue:

Supportato come origine?

Sì. Puoi utilizzare i lavori AWS Glue ETL per interrogare i dati da Smartsheet.

Supportato come destinazione?

No.

Versioni dell'API Smartsheet supportate

v2.0

## Policy contenenti le operazioni API per creare e utilizzare le connessioni

La seguente politica di esempio descrive le AWS autorizzazioni richieste per la creazione e l'utilizzo delle connessioni. Se si sta creando un nuovo ruolo, creare una policy che contenga quanto segue:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:ListConnectionTypes",
        "glue:DescribeConnectionType",
        "glue:RefreshOAuth2Tokens",
        "glue:ListEntities",
        "glue:DescribeEntity"
      ],
      "Resource": "*"
    }
  ]
}
```

In alternativa, se non desideri utilizzare il metodo precedente, utilizza le seguenti policy IAM:

- AWSGlueServiceRole — Garantisce l'accesso alle risorse che diversi AWS Glue processi richiedono per l'esecuzione per conto dell'utente. Queste risorse includono Amazon S3 AWS Glue, IAM, CloudWatch Logs e Amazon EC2. Se segui la convenzione di denominazione delle risorse specificata in questa politica, AWS Glue i processi dispongono delle autorizzazioni necessarie. Questa policy è in genere collegata ai ruoli specificati quando si definiscono crawler, processi ed endpoint di sviluppo.
- AWSGlueConsoleFullAccess — Concede l'accesso completo alle AWS Glue risorse quando un'identità a cui è allegata la politica utilizza la AWS console di gestione. Se si segue la convenzione per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questa policy è in genere associata agli utenti della AWS Glue console.

## Configurazione di Smartsheet

Prima di poter AWS Glue utilizzare il trasferimento da Smartsheet, devi soddisfare i seguenti requisiti:

Requisiti minimi

- Avere un account Smartsheet con e-mail e password. Per ulteriori informazioni sulla creazione di account, consultare Creazione di un account Smartsheet.
- L'account Smartsheet dispone dell'accesso all'API con una licenza valida.
- L'account Smartsheet ha un piano tariffario Pro per entità `Sheets` e un piano tariffario Enterprise con un componente di segnalazione di eventi per entità `Events`.

Se soddisfi questi requisiti, sei pronto per connetterti AWS Glue al tuo account Smartsheet. Per le connessioni tipiche, non è necessario fare nient'altro in Smartsheet.

## Configurazione delle connessioni di Smartsheet

Smartsheet supporta il tipo di `AUTHORIZATION_CODE` concessione per OAuth2.

Questo tipo di concessione è considerato `OAuth` "a tre gambe" in quanto si basa sul reindirizzamento degli utenti al server di autorizzazione di terzi per l'autenticazione. Gli utenti possono scegliere di creare la propria app connessa in Smartsheet e fornire il proprio ID cliente e il segreto del client quando creano connessioni tramite la AWS Glue console. In questo scenario, verranno comunque reindirizzati a Smartsheet per effettuare il login e autorizzare l'accesso AWS Glue alle proprie risorse.

Questo tipo di concessione genera un token di aggiornamento e un token di accesso. Il token di accesso dura poco tempo e può essere aggiornato automaticamente senza l'interazione dell'utente utilizzando il token di aggiornamento.

Per la documentazione pubblica di Smartsheet sulla creazione di un'app connessa per il flusso OAuth AUTHORIZATION_CODE, consulta Smartsheet. APIs

Per configurare una connessione a Smartsheet:

1. In AWS Secrets Manager, crea un segreto con i seguenti dettagli:

   Per le app connesse gestite dal cliente: il segreto deve contenere l'app connessa Consumer Secret con `USER_MANAGED_CLIENT_APPLICATION_CLIENT_SECRET` come chiave.

   > **Note**
   >
   > È necessario creare un segreto per ogni connessione in AWS Glue.

2. In AWS Glue Studio, crea una connessione in Connessioni dati seguendo i passaggi seguenti:
   a. In Tipo di connessione, selezionare Smartsheet.
   b. Fornire il `instanceUrl` di Smartsheet a cui si desidera connettersi.
   c. Seleziona il ruolo IAM per il quale AWS Glue può assumere e dispone delle autorizzazioni per le seguenti azioni:

      JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:DescribeSecret",
        "secretsmanager:GetSecretValue",
        "secretsmanager:PutSecretValue",
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DeleteNetworkInterface"
      ],
      "Resource": "*"
    }
  ]
}
```

   d. Seleziona quello `secretName` che desideri utilizzare per questa connessione per AWS Glue inserire i token.
   e. Selezionare le opzioni di rete se si desidera utilizzare la propria rete.
3. Concedi il permesso di lettura `secretName` al ruolo IAM associato al tuo AWS Glue lavoro.

## Lettura da entità Smartsheet

### Prerequisiti

Un oggetto di `Smartsheet` da cui si desidera leggere. Consultare la tabella delle entità supportate qui di seguito per verificare quelle disponibili.

### Entità supportate

| Entità | Può essere Filtrato | Supporta Limite | Supporta Ordina per | Supporta Seleziona * | Supporta Partizionamento |
|---|---|---|---|---|---|
| Foglio di elenco | Sì | Sì | No | Sì | No |
| Metadati di righe | Sì | Sì | No | Sì | No |
| Metadati del foglio | No | No | No | Sì | No |
| Dati nel foglio | Sì | Sì | Sì | Sì | No |
| Event | Sì | Sì | No | Sì | No |

Esempio

```
Smartsheet_read = glueContext.create_dynamic_frame.from_options(
    connection_type="smartsheet",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "list-sheets",
        "API_VERSION": "2.0",
        "INSTANCE_URL": "https://api.smartsheet.com"
    })
```

Dettagli dell'entità e del campo di Smartsheet

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| Fogli di elenco | id | Long | N/A |
| | accessLevel | Stringa | N/A |
| | createdAt | DateTime | N/A |
| | modifiedAt | DateTime | N/A |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | nome | Stringa | N/A |
| | permalink | Stringa | N/A |
| | modifiedSince | DateTime | >= |
| | version | Numero intero | N/A |
| | origine | Struct | N/A |
| Metadati di righe | id | Long | N/A |
| | sheetId | Long | N/A |
| | accessLevel | Stringa | N/A |
| | attachments | List | N/A |
| | columns | List | N/A |
| | conditionalFormat | Stringa | N/A |
| | createdAt | DateTime | N/A |
| | createdBy | Struct | N/A |
| | discussions | List | N/A |
| | proofs | Struct | N/A |
| | expanded | Booleano | N/A |
| | filteredOut | Booleano | N/A |
| | format | Stringa | N/A |
| | inCriticalPath | Booleano | N/A |
| | locked | Booleano | N/A |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | lockedForUser | Booleano | N/A |
| | modifiedAt | DateTime | N/A |
| | modifiedBy | Struct | N/A |
| | permalink | Stringa | N/A |
| | rowNumber | Numero intero | N/A |
| | version | Numero intero | N/A |
| | totalRowCount | Numero intero | N/A |
| | rowsModifiedSince | DateTime | > |
| | filterId | Long | "=" |
| | siblingId | Long | N/A |
| | parentId | Long | N/A |
| Metadati del foglio | id | Long | N/A |
| | fromId | Long | N/A |
| | ownerId | Long | N/A |
| | accessLevel | Stringa | N/A |
| | attachments | List | N/A |
| | columns | List | N/A |
| | createdAt | DateTime | N/A |
| | crossSheetReferences | List | N/A |
| | dependenciesEnabled | Booleano | N/A |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | discussions | List | N/A |
| | effectiveAttachmentOptions | List | N/A |
| | favorite | Booleano | N/A |
| | ganttEnabled | Booleano | N/A |
| | hasSummaryFields | Booleano | N/A |
| | modifiedAt | DateTime | N/A |
| | nome | Stringa | N/A |
| | owner | Stringa | N/A |
| | permalink | Stringa | N/A |
| | projectSettings | Struct | N/A |
| | readOnly | Booleano | N/A |
| | resourceManagementEnabled | Booleano | N/A |
| | showParentRowsForFilters | Booleano | N/A |
| | origine | Struct | N/A |
| | riepilogo | Struct | N/A |
| | totalRowCount | Numero intero | N/A |
| | userPermissions | Struct | N/A |
| | userSettings | Struct | N/A |
| | version | Numero intero | N/A |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | workspace | Struct | N/A |
| | filtri | List | N/A |
| | ganttConfig | Struct | N/A |
| | resourceManagementType | Stringa | N/A |
| | cellImageUploadAbilitato | Booleano | N/A |
| | isMultiPicklistAbilitato | Booleano | N/A |
| Eventi | eventId | Stringa | N/A |
| | objectType | Stringa | N/A |
| | operazione | Stringa | N/A |
| | objectId | Long | N/A |
| | eventTimestamp | DateTime | N/A |
| | userId | Long | N/A |
| | requestUserId | Long | N/A |
| | accessTokenName | Stringa | N/A |
| | origine | Stringa | N/A |
| | additionalDetails | Struct | N/A |
| | since | DateTime | >= |

Entità con metadati dinamici:

Per l'entità seguente, Smartsheet fornisce un endpoint per recuperare dinamicamente i metadati, consentendo di acquisire il supporto dell'operatore a livello di tipo di dati.

| Entità | Tipo di dati | Operatori supportati |
|---|---|---|
| Dati nel foglio | Stringa | N/A |
| Dati nel foglio | Long | "=" |
| Dati nel foglio | Numero intero | N/A |
| Dati nel foglio | DateTime | > |

Esempio

```
Smartsheet_read = glueContext.create_dynamic_frame.from_options(
    connection_type="smartsheet",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "list-sheets",
        "API_VERSION": "2.0",
        "INSTANCE_URL": "https://api.smartsheet.com"
    }
```

## Opzioni di connessione di Smartsheet

Di seguito sono riportate le opzioni di connessione per Smartsheet:

- `ENTITY_NAME` (stringa): (obbligatorio) utilizzato per lettura/scrittura. Il nome dell'oggetto in Smartsheet.
- `API_VERSION`(String): (obbligatorio) utilizzato per la lettura/scrittura. Versione dell'API Rest di Smartsheet da utilizzare. Ad esempio: v2.0.
- `INSTANCE_URL`(String): (obbligatorio) utilizzato per la lettura. URL dell'istanza Smartsheet.
- `SELECTED_FIELDS` (List<String>): predefinito, vuoto (SELECT *). Utilizzato per la lettura. Colonne da selezionare per l'oggetto.
- `FILTER_PREDICATE`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Dev'essere in formato Spark SQL.
- `QUERY`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Query Spark SQL completa.

## Creazione di un account Smartsheet

1. Registrarsi per creare un account Smartsheet accedendo alla Pagina di iscrizione a Smartsheet.
2. Scegliere Creane uno per creare un nuovo account o accedere utilizzando l'account Google, Microsoft o Apple registrato.
3. 
4. 
5. Aprire l'e-mail di conferma di Smartsheet e scegliere il link di conferma per verificare l'account.

   Per impostazione predefinita, si attiverà l'abbonamento al piano di prova.
6. Nell'angolo in basso a sinistra, scegliere l'icona Account e scegliere Aggiungi licenze/Aggiorna per aggiornare il piano tariffario.

   > **Note**
   >
   > È obbligatorio per accedere a Segnalazione eventi, che è un componente aggiuntivo del piano Enterprise.
7. Nel piano Enterprise, scegliere Contattaci per richiedere un upgrade dell'account al team di supporto.
8. Nel modulo di richiesta di assistenza, fornire i dettagli richiesti e i requisiti per aggiornare il piano.

   Questo completa l'aggiornamento al piano Enterprise.

### Creazione delle credenziali di **`OAuth2.0`**

1. Dopo aver aggiornato il piano tariffario dell'account per accedere agli Strumenti per sviluppatori, accedere a Sviluppatori Smartsheet.

   L'utente riceverà un'e-mail di attivazione.
2. Aprire l'e-mail di attivazione da Smartsheet e scegliere il link di attivazione per attivare gli strumenti per sviluppatori sull'account.

   Lo strumento per sviluppatori consente di creare l'app.
3. Aprire la home page dell'account Smartsheet e scegliere Account per verificare l'accesso.
4. Scegliere Strumenti per sviluppatori dall'elenco dei servizi e inserire i dettagli del Profilo dello sviluppatore.

5. Scegliere Crea nuova app.
6. Inserire i seguenti dettagli nel modulo di registrazione dell'app:
   - Nome: il nome dell'app.
   - Descrizione: descrizione dell'app.
   - URL: URL che consente di avviare l'app o l'URL della pagina di destinazione.
   - Contatto/Supporto: informazioni di contatto per il team di assistenza.
   - URL di reindirizzamento: URL (noto anche come URL di callback) all'interno dell'applicazione che riceverà le credenziali OAuth 2.0.
7. Scegli Save (Salva).

   Smartsheet assegna un ID client e un segreto del client all'app. Registrare questi valori per le fasi successive. È anche possibile cercarli di nuovo più avanti nella sezione Strumenti per sviluppatori.

### Limitazioni

Smartsheet non supporta il partizionamento basato su campi o su record.

## Connessione a Snapchat Ads in AWS Glue Studio

Snapchat è un'applicazione e un servizio di messaggistica istantanea multimediale sviluppato da Snap Inc., originariamente Snapchat Inc. Una delle caratteristiche principali di Snapchat è che le immagini e i messaggi sono solitamente disponibili solo per un breve periodo di tempo prima di diventare inaccessibili per i destinatari. Il marketing su Snapchat prevede che le aziende possano pagare per pubblicare post destinati agli utenti di Snapchat.

Argomenti

- AWS Glue supporto per Snapchat Ads
- Policy contenenti le operazioni API per creare e utilizzare le connessioni
- Configurazione di Snapchat Ads
- Configurazione delle connessioni di Snapchat Ads
- Lettura dalle entità di Snapchat Ads
- Opzioni di connessione di Snapchat Ads
- Creazione di un account di Snapchat Ad e configurazione dell'app client

- Creazione di un'app nel proprio account Snapchat Ads

## AWS Glue supporto per Snapchat Ads

AWS Glue supporta Snapchat Ads come segue:

Supportato come origine?

Sì. Puoi utilizzare i job AWS Glue ETL per interrogare i dati da Snapchat Ads.

Supportato come destinazione?

No.

Versioni dell'API di Snapchat Ads supportate

v1.

## Policy contenenti le operazioni API per creare e utilizzare le connessioni

La seguente politica di esempio descrive le AWS autorizzazioni richieste per la creazione e l'utilizzo delle connessioni. Se si sta creando un nuovo ruolo, creare una policy che contenga quanto segue:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:ListConnectionTypes",
        "glue:DescribeConnectionType",
        "glue:RefreshOAuth2Tokens",
        "glue:ListEntities",
        "glue:DescribeEntity"
      ],
      "Resource": "*"
    }
  ]
}
```

È anche possibile utilizzare le seguenti policy IAM per autorizzare l'accesso:

- AWSGlueServiceRole— Garantisce l'accesso alle risorse che diversi AWS Glue processi richiedono per l'esecuzione per conto dell'utente. Queste risorse includono AWS Glue Amazon S3 Amazon CloudWatch Logs, IAM e Amazon EC2. Se segui la convenzione di denominazione delle risorse specificata in questa politica, AWS Glue i processi dispongono delle autorizzazioni necessarie. Questa policy è in genere collegata ai ruoli specificati quando si definiscono crawler, processi ed endpoint di sviluppo.
- AWSGlueConsoleFullAccess— Concede l'accesso completo alle AWS Glue risorse quando un'identità a cui è allegata la policy utilizza la AWS console di gestione. Se si segue la convenzione per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questa policy è in genere associata agli utenti della AWS Glue console.

## Configurazione di Snapchat Ads

Prima di poter AWS Glue utilizzare il trasferimento da Snapchat Ads, devi soddisfare questi requisiti:

Requisiti minimi

- Avere un account Snapchat Ads. Per ulteriori informazioni sulla creazione di un account, consultare Creazione di un account di Snapchat Ad e configurazione dell'app client.
- Hai creato un' OAuth2 app nel tuo account Snapchat Ads. Questa integrazione fornisce le credenziali che AWS Glue utilizza per accedere in modo sicuro ai dati quando si effettuano chiamate autenticate al proprio account: Per ulteriori informazioni, consulta Creazione di un'app nel proprio account Snapchat Ads.

Se soddisfi questi requisiti, sei pronto per connetterti AWS Glue al tuo account Snapchat Ads. In Snapchat Ads, un'app connessa è un framework che autorizza applicazioni esterne, ad esempio AWS Glue, ad accedere ai dati di Snapchat Ads.

## Configurazione delle connessioni di Snapchat Ads

Snapchat Ads supporta solo il tipo di concessione `AUTHORIZATION_CODE`.

Questo tipo di concessione è considerato «a tre gambe» in OAuth quanto si basa sul reindirizzamento degli utenti al server di autorizzazione di terze parti per autenticare l'utente. Viene utilizzato durante la creazione di connessioni tramite la console. AWS Glue Per impostazione predefinita, l'utente che crea una connessione può affidarsi a un'app connessa di AWS Glue proprietà

(applicazione client AWS Glue gestita) in cui non deve fornire alcuna informazione OAuth correlata ad eccezione dell'URL dell'istanza di Snapchat Ads. La AWS Glue Console reindirizzerà l'utente a Snapchat Ads, dove l'utente deve effettuare il login e concedere AWS Glue le autorizzazioni richieste per accedere alla propria istanza di Snapchat Ads.

Gli utenti possono comunque scegliere di creare la propria app connessa in Snapchat Ads e fornire il proprio ID cliente e il segreto del client quando creano connessioni tramite la AWS Glue Console. In questo scenario, verranno comunque reindirizzati a Snapchat Ads per effettuare il login e autorizzare l'accesso AWS Glue alle proprie risorse.

Questo tipo di concessione genera un token di aggiornamento e uno di accesso. Il token di accesso scade 1 ora dopo la creazione. È possibile recuperare un nuovo token di accesso utilizzando il token di aggiornamento.

Per ulteriori informazioni sulla creazione di un'app connessa per il OAuth flusso del codice di autorizzazione, consulta Ads API.

Per configurare una connessione a Snapchat Ads:

1. In AWS Secrets Manager, crea un segreto con i seguenti dettagli. È necessario creare un segreto per ogni connessione in AWS Glue.
   a. Per le app connesse gestite dal cliente: il segreto deve contenere l'app connessa Consumer Secret con `USER_MANAGED_CLIENT_APPLICATION_CLIENT_SECRET` come chiave.
2. In AWS Glue Glue Studio, crea una connessione in Connessioni dati seguendo i passaggi seguenti:
   a. In Tipo di connessione, selezionare Snapchat Ads.
   b. Fornire l'ambiente di Snapchat Ads.
   c. Seleziona il ruolo IAM che AWS Glue può assumere e dispone delle autorizzazioni per le seguenti azioni:

      JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:DescribeSecret",
        "secretsmanager:GetSecretValue",
```

```
        "secretsmanager:PutSecretValue",
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DeleteNetworkInterface"
      ],
      "Resource": "*"
    }
  ]
}
```

d. Seleziona quello `secretName` che desideri utilizzare per questa connessione per AWS Glue inserire i token.

e. Selezionare le opzioni di rete se si desidera utilizzare la propria rete.

3. Concedi il permesso di lettura `secretName` al ruolo IAM associato al tuo AWS Glue lavoro.

## Lettura dalle entità di Snapchat Ads

Prerequisiti

- Un oggetto di Snapchat Ads da cui si desidera leggere. Consultare la tabella delle entità supportate qui di seguito per verificare quelle disponibili.

Entità supportate

| Entità | Può essere Filtrato | Supporta Limite | Supporta Ordina per | Supporta Seleziona * | Supporta Partizionamento |
|---|---|---|---|---|---|
| Organizzazione | No | No | No | Sì | No |
| Account pubblicitario | No | No | No | Sì | No |
| Creativo | No | No | No | Sì | No |
| Media | No | No | No | Sì | No |
| Campagna | Sì | No | No | Sì | No |

| Entità | Può essere Filtrato | Supporta Limite | Supporta Ordina per | Supporta Seleziona * | Supporta Partizionamento |
|---|---|---|---|---|---|
| Annuncio sotto Account pubblicitario | Sì | No | No | Sì | No |
| Annuncio sotto campagna | No | No | No | Sì | No |
| Annuncio Squadra | Sì | No | No | Sì | No |
| Segment | No | No | No | Sì | No |

Esempio

```
snapchatads_read = glueContext.create_dynamic_frame.from_options(
    connection_type="snapchatAds",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "organization",
        "API_VERSION": "v1"
    }
)
```

Dettagli sull'entità e sul campo di Snapchat Ads

Snapchat Ads carica dinamicamente i campi disponibili nell'entità selezionata. A seconda del tipo di dati di campo, supporta i seguenti operatori di filtro.

| Tipo di dati dei campi | Operatori di filtro supportati |
|---|---|
| Booleano | = |

Query di partizionamento

- Partizionamento basato sul campo: non supportato.
- Partizionamento basato su record: non supportato.

## Opzioni di connessione di Snapchat Ads

Di seguito sono riportate le opzioni di connessione per Snapchat Ads:

- `ENTITY_NAME`(String): (obbligatorio) utilizzato per la lettura. Il nome dell'entità di Snapchat Ads. Esempio: `campaign` .
- `API_VERSION`(String): (obbligatorio) utilizzato per la lettura. La versione dell'API Rest di Snapchat Ads che si desidera utilizzare. Il valore sarà v1, poiché Snapchat Ads attualmente supporta solo la versione v1.
- `SELECTED_FIELDS`(List<String>): predefinito: vuoto (SELECT *). Utilizzato per la lettura. Elenco di colonne separate da virgole che si desidera selezionare per l'entità selezionata.
- `FILTER_PREDICATE`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Dev'essere in formato Spark SQL.
- `QUERY`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Query Spark SQL completa.

## Creazione di un account di Snapchat Ad e configurazione dell'app client

Argomenti

- Registrarsi a Snapchat Ads
- Passaggi per creare un account Snapchat Ad

### Registrarsi a Snapchat Ads

Per registrarti a Snapchat Ads:

1. Vai su Snapchat Ads Manager. Seleziona Registrati accanto a È la prima volta che usi Snapchat?.
2. Nella schermata Crea account, segui le istruzioni per inserire il nome della tua attività, l'indirizzo e-mail, la password, ecc. Scegli Next (Successivo).
3. Nella schermata Crea profilo, inserisci Nome utente e Sito Web (facoltativo). Poi seleziona Crea account. A questo punto avrai la possibilità di aggiungere una foto profilo e una biografia nella schermata Modifica profilo. Scegli Conferma.

4. Nella schermata Informazioni aziendali, compila i campi obbligatori come Paese, Valuta, Numero di telefono, GSTIN ecc. e completa il processo di creazione dell'account selezionando Avanti.

Passaggi per creare un account Snapchat Ad

Per creare un account Snapchat Ad:

1. Accedi ad Ads Manager. Poi fai clic sulla barra di navigazione nell'angolo superiore e seleziona Account pubblicitario.
2. Scegli + Nuovo Account pubblicitario. Inserisci i tuoi dati in quanto inserzionista:
   - Indica se sei un'agenzia che acquista annunci per conto di un inserzionista. Se selezioni "Sì", il tuo annuncio potrebbe essere rifiutato qualora utilizzi parametri di targeting che potrebbero includere l'età, il sesso o il codice postale. Il targeting basato sull'età può essere applicato a un minimo di 21 anni.
   - Indica se il tuo account pubblicitario pubblicherà annunci immobiliari, annunci relativi a crediti o annunci di lavoro. Se selezioni "Sì", il tuo annuncio potrebbe essere rifiutato qualora utilizzi parametri di targeting che potrebbero includere l'età, il sesso o il codice postale. Il targeting basato sull'età può essere applicato a un minimo di 21 anni.
   - Indica se utilizzerai l'account pubblicitario per annunci politici. Se pubblichi un annuncio politico, inserisci l'organizzazione politica o il gruppo di sostegno che lo sponsorizza e che ne sostiene i costi. Se non inserisci l'organizzazione politica in modo corretto, i tuoi annunci potrebbero essere rifiutati. Prima di inviare gli annunci, dovrai inoltre compilare obbligatoriamente il "Modulo di revisione degli annunci politici" collegato.
3. Seleziona Dettagli account e inserisci le informazioni relative al tuo account pubblicitario:

| Campo | Description |
| --- | --- |
| Nome | Il nome del tuo account pubblicitario. |
| Tipo di account | Seleziona il tipo di account, se non viene compilato automaticamente. |
| Tipo di fatturazione | Se vuoi avere accesso continuo alla tua fonte di finanziamento, seleziona "Revolvin g". Questa opzione ti consente di ricostituire la tua linea di credito e spesso viene scelta |

| Campo | Description |
| --- | --- |
| | per pubblicare annunci in Ads Manager. Se preferisci lanciare una campagna una tantum con un limite di spesa, puoi scegliere l'opzione Ordine di inserzione.<br><br>Se crei un account con un ordine di inserzion e ma desideri impostare un limite di spesa in un secondo momento, puoi lasciare il limite di spesa a 0 $. In questo modo l'account avrà un limite di spesa pari a 0$, ma potrai applicarne uno in un secondo momento. |
| Organizzazione pubblicitaria | L'organizzazione che acquista gli annunci. |
| Centro di fatturazione | Scegli il centro di fatturazione presso il quale desideri ricevere le fatture. Puoi usare quello creato automaticamente quando hai rivendicato la tua attività commerciale per la prima volta o aggiungerne un altro seguendo i passaggi del Centro assistenza per le attività commerciali. |
| Valuta | Scegli la valuta. |
| Limite di spesa | Se hai scelto "Ordine di inserzione" come tipo di fatturazione, inserisci un budget per l'account pubblicitario. |
| Fuso orario | Seleziona il tuo fuso orario. |

4. Seleziona Crea account. Il tuo account pubblicitario verrà creato e potrai trovarlo nella sezione Account pubblicitari di Ads Manager. Per iniziare a pubblicare annunci, dovrai inserire un metodo di pagamento. Puoi anche aggiungere dei membri al tuo account pubblicitario.
5. Scegli se preferisci usare un metodo di pagamento esistente o crearne uno nuovo. Quindi, seleziona Salva metodo di pagamento.
6. Seleziona tutti i membri che hai invitato nella tua attività commerciale e aggiungili all'account pubblicitario. Per maggiori informazioni sui ruoli e sulle autorizzazioni che si possono assegnare,

vedi Panoramica su ruoli e autorizzazioni. I membri aggiunti potranno quindi accedere ad Ads Manager e a questo account pubblicitario. Quando hai finito, salva i membri.

Per ulteriori informazioni sugli account pubblicitari, consulta https://businesshelp.snapchat.com/s/article/roles-permissions?language=en_US https://businesshelp.snapchat.com/s/ article/roles-permissions? language=it_US

### Creazione di un'app nel proprio account Snapchat Ads

Per attivare l'accesso all'API di marketing di Snapchat, assicurati di aver configurato un account aziendale. Quindi segui la procedura riportata di seguito.

1. Accedi ad Ads Manager. Quindi seleziona il menu nell'angolo in alto a sinistra e scegli Business Dashboard, quindi seleziona Dettagli aziendali.
2. Scegli + OAuth App.
3. Inserisci il nome dell'app e aggiungi il seguente URL come URI di reindirizzamento Snap: `https://<aws-region>.console.aws.amazon.com/gluestudio/oauth`. Ad esempio, se utilizzi la regione us-west-1, l'URL sarebbe: `https://us-west-1.console.aws.amazon.com/gluestudio/oauth) and choose **Create OAuth App**`. Scegli Crea OAuth app.
4. Verranno visualizzate le credenziali dell'app (ID Client e Segreto client). Salvateli perché saranno necessari per creare una connessione.

## Connessione a Snowflake in AWS Glue Studio

> **Note**
>
> È possibile utilizzare AWS Glue per Spark per leggere e scrivere su tabelle in Snowflake in AWS Glue 4.0 e versioni successive. Per configurare una connessione Snowflake con i processi AWS Glue a livello di programmazione, consulta la pagina Connessioni Redshift.

AWS Glue fornisce il supporto integrato per Snowflake. AWS Glue Studio fornisce un'interfaccia visiva per connettersi a Snowflake, creare processi di integrazione dei dati ed eseguirli sul runtime Spark serverless AWS Glue Studio.

AWS Glue Studio crea una connessione unificata per Snowflake. Per ulteriori informazioni, consulta Considerazioni.

Argomenti

- Creazione di una connessione Snowflake
- Creazione di un nodo di origine Snowflake
- Creazione di un nodo di destinazione Snowflake
- Configurare il flusso del codice di autorizzazione per Snowflake
- Opzioni avanzate

## Creazione di una connessione Snowflake

> **Note**
>
> Le connessioni unificate (connessione v2) standardizzano tutte le connessioni per utilizzare le chiavi `USERNAME` e `PASSWORD` per le credenziali di autenticazione di base. È comunque possibile creare una connessione v1 tramite API con segreti contenenti `sfUser` e `sfPassword`.

Quando aggiungi un'origine dati - nodo Snowflake inAWS Glue Studio, puoi scegliere una connessione AWS Glue Snowflake esistente o creare una nuova connessione. È necessario scegliere un tipo di connessione `SNOWFLAKE` e non un tipo di connessione `JDBC`configurato per la connessione a Snowflake. Segui la seguente procedura per creare una connessione Snowflake: AWS Glue

Creazione di una connessione Snowflake

1. In Snowflake, genera un utente e una password,*`snowflakeUser`*. *`snowflakePassword`*
2. Determina con quale magazzino Snowflake interagirà questo utente,. *`snowflakeWarehouse`* Puoi impostarlo come modulo *`snowflakeUser`* in Snowflake o ricordarlo per il passaggio successivo. `DEFAULT_WAREHOUSE`
3. Nel Gestione dei segreti AWS, crea un segreto usando le tue credenziali Snowflake. Per creare un segreto in Secrets Manager, segui il tutorial disponibile in Crea un Gestione dei segreti AWS segreto nella Gestione dei segreti AWS documentazione. Dopo aver creato il segreto, mantieni il nome del segreto *`secretName`* per il passaggio successivo.

- Quando selezionate coppie chiave/valore, create una coppia per *snowflakeUser* con la chiave. sfUser
- Quando selezionate coppie chiave/valore, create una coppia per *snowflakePassword* con la chiave. sfPassword
- Quando selezionate coppie chiave/valore, create una coppia per *snowflakeWarehouse* con la chiave. sfWarehouse Questo non è necessario se in Snowflake è impostato un valore predefinito.

4. Nel AWS Glue Data Catalog, crea una connessione seguendo i passaggi descritti in Aggiungere una AWS Glue connessione. Dopo aver creato la connessione, mantieni il nome della connessione per il passaggio successivo. *connectionName*

   - In Tipo di connessione, seleziona Snowflake.
   - In URL Snowflake, fornisci il nome host dell'istanza Snowflake. L'URL utilizzerà un nome host nel modulo *account_identifier*.snowflakecomputing.com.
   - Quando selezioni un AWS segreto, fornisci*secretName*.

## Creazione di un nodo di origine Snowflake

Autorizzazioni necessarie

AWS Glue Studio i lavori che utilizzano origini dati Snowflake richiedono autorizzazioni aggiuntive. Per ulteriori informazioni su come aggiungere autorizzazioni ai processi ETL, consulta la pagina Review IAM permissions needed for ETL jobs.

SNOWFLAKE AWS Glue le connessioni utilizzano un Gestione dei segreti AWS segreto per fornire informazioni sulle credenziali. I tuoi ruoli di lavoro e di anteprima dei dati AWS Glue Studio devono essere autorizzati a leggere questo segreto.

Aggiunta di un'origine dati Snowflake

Prerequisiti:

- Un Gestione dei segreti AWS segreto per le tue credenziali Snowflake
- Una connessione Data Catalog di tipo Snowflake AWS Glue

Per aggiungere un nodo Origine dati: Snowflake:

1. Scegli la connessione per la tua origine dati Snowflake. Ciò presuppone che la connessione esista già e che sia possibile effettuare una selezione tra le connessioni esistenti. Se hai bisogno di creare una connessione, scegli Crea connessione Snowflake. Per ulteriori informazioni, consulta la pagina Overview of using connectors and connections.

   Dopo aver scelto una connessione, puoi visualizzare le proprietà della connessione facendo clic su Visualizza proprietà. Le informazioni sulla connessione sono visibili, tra cui URL, gruppi di sicurezza, sottorete, zona di disponibilità, descrizione, nonché timestamp di creazione (UTC) e ultimo aggiornamento (UTC).
2. Scegli un'opzione di origine Snowflake:

   - Scegli una singola tabella: questa è la tabella che contiene i dati a cui desideri accedere da una singola tabella Snowflake.
   - Inserisci una query personalizzata: ti consente di accedere a un set di dati da più tabelle Snowflake in base alla tua query personalizzata.
3. Se hai scelto una singola tabella, inserisci il nome di uno schema Snowflake.

   In alternativa, scegli Inserisci query personalizzata. Scegli questa opzione per accedere a un set di dati personalizzato da più tabelle Snowflake. Se scegli questa opzione, inserisci la query Snowflake.
4. In Prestazioni e sicurezza (facoltativo),

   - Abilita il push down delle query: scegli se vuoi trasferire il processo sull'istanza Snowflake.
5. In Proprietà personalizzate di Snowflake (facoltativo), inserisci i parametri e i valori necessari.

## Creazione di un nodo di destinazione Snowflake

### Autorizzazioni necessarie

AWS Glue Studio i lavori che utilizzano sorgenti dati Snowflake richiedono autorizzazioni aggiuntive. Per ulteriori informazioni su come aggiungere autorizzazioni ai processi ETL, consulta la pagina Review IAM permissions needed for ETL jobs.

`SNOWFLAKE` AWS Glue le connessioni utilizzano un Gestione dei segreti AWS segreto per fornire informazioni sulle credenziali. I tuoi ruoli di lavoro e di anteprima dei dati AWS Glue Studio devono essere autorizzati a leggere questo segreto.

### Aggiunta di una destinazione dati Snowflake

Per creare un nodo di destinazione Snowflake:

1. Scegli una tabella Snowflake esistente come destinazione o inserisci un nuovo nome per la tabella.
2. Quando utilizzi il nodo di destinazione Destinazione dati - Snowflake, puoi scegliere tra le seguenti opzioni:

   - AGGIUNGI: se esiste già una tabella, scarica tutti i nuovi dati nella tabella come inserto. Se la tabella non esiste, procedi alla sua creazione e quindi inserisci tutti i nuovi dati.
   - MERGE: AWS Glue aggiorna o aggiunge i dati alla tabella di destinazione in base alle condizioni specificate.

     Scegli le opzioni:

     - Scegli chiavi e operazioni semplici: scegli le colonne da utilizzare come chiavi di corrispondenza tra i dati di origine e il set di dati di destinazione.

       Specifica le seguenti opzioni in caso di corrispondenza:

       - Aggiorna il record nel set di dati di destinazione con i dati dell'origine.
       - Elimina il record nel set di dati di destinazione.

       Specifica le seguenti opzioni in caso di mancata corrispondenza:

       - Inserisci i dati di origine come nuova riga nel set di dati di destinazione.
       - Non fare nulla.
     - Inserisci un'istruzione MERGE personalizzata: puoi quindi scegliere Convalida l'istruzione MERGE per verificare che l'istruzione sia valida o non valida.
   - TRUNCATE: se esiste già una tabella, tronca i dati della tabella cancellando prima il contenuto della tabella di destinazione. Se il troncamento ha esito positivo, inserisci tutti i dati. Se la tabella non esiste, procedi alla sua creazione e quindi inserisci tutti i dati. Se il troncamento non va a buon fine, l'operazione non andrà a buon fine.
   - DROP: se esiste già una tabella, elimina i metadati e i dati della tabella. Se l'eliminazione ha esito positivo, inserisci tutti i dati. Se la tabella non esiste, procedi alla sua creazione e quindi inserisci tutti i dati. Se l'eliminazione non va a buon fine, l'operazione non andrà a buon fine.

### Configurare il flusso del codice di autorizzazione per Snowflake

Per utilizzare il metodo di OAuth autenticazione, assicuratevi che la seguente configurazione sia completa:

- Configura Snowflake OAuth per un client personalizzato seguendo la documentazione ufficiale di Snowflake: Configura Snowflake per client personalizzati. OAuth
- Imposta l'URI di reindirizzamento corretto durante la creazione dell'integrazione di sicurezza Snowflake. Ad esempio: se state creando la connessione nella regione DUB (eu-west-1), l'URI di reindirizzamento dovrebbe essere: `https://eu-west-1.console.aws.amazon.com/gluestudio/oauth`
- Dopo aver creato l'integrazione di sicurezza, conservate le seguenti informazioni da utilizzare durante la creazione della connessione Glue:
  - OAUTH_CLIENT_ID: questo valore deve essere fornito come ID client dell'applicazione User Managed Client nella pagina di creazione della connessione Glue.
  - OAUTH_CLIENT_SECRET: questo valore deve essere memorizzato nel Secret utilizzato per la connessione, sotto la chiave AWS USER_MANAGED_CLIENT_APPLICATION_CLIENT_SECRET.

### Opzioni avanzate

Vedi Snowflake connections nella guida per AWS Glue sviluppatori.

## Connessione a Stripe in AWS Glue Studio

Stripe è una piattaforma online per l'elaborazione dei pagamenti e delle carte di credito destinata alle aziende. La piattaforma Stripe consente alle aziende di accettare pagamenti online, creare abbonamenti (fatturazione ricorrente) per il proprio e-commerce e configurare un conto secondario per ricevere i pagamenti. Stripe supporta anche i pagamenti tra più parti, consentendo alle aziende di configurare il proprio marketplace e di riscuotere i pagamenti per poi versarli ai venditori o ai fornitori di servizi tramite un conto "connesso".

Argomenti

- AWS Glue supporto per Stripe
- Policy contenenti le operazioni API per creare e utilizzare le connessioni
- Configurazione di Stripe

- Configurazione delle connessioni di Stripe
- Lettura da entità di Stripe
- Opzioni di connessione di Stripe
- Limitazioni
- Creazione di un nuovo account Stripe e configurazione dell'app client

## AWS Glue supporto per Stripe

AWS Glue supporta Stripe come segue:

Supportato come origine?

Sì. Puoi utilizzare i job AWS Glue ETL per interrogare i dati da Stripe.

Supportato come destinazione?

No.

Versioni dell'API Slack supportate

v1

## Policy contenenti le operazioni API per creare e utilizzare le connessioni

Nella seguente policy di esempio sono descritte le autorizzazioni IAM per creare e utilizzare le connessioni. Se si sta creando un nuovo ruolo, creare una policy che contenga quanto segue:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:ListConnectionTypes",
        "glue:DescribeConnectionType",
        "glue:RefreshOAuth2Tokens",
        "glue:ListEntities",
        "glue:DescribeEntity"
      ],
```

```
      "Resource": "*"
    }
  ]
}
```

È anche possibile utilizzare le seguenti policy IAM per autorizzare l'accesso:

- AWSGlueServiceRole— Garantisce l'accesso alle risorse che AWS Glue i vari processi richiedono per l'esecuzione per conto dell'utente. Queste risorse includono AWS Glue Amazon S3 Amazon CloudWatch Logs, IAM e Amazon EC2. Se segui la convenzione di denominazione delle risorse specificata in questa politica, AWS Glue i processi dispongono delle autorizzazioni necessarie. Questa policy è in genere collegata ai ruoli specificati quando si definiscono crawler, processi ed endpoint di sviluppo.
- AWSGlueConsoleFullAccess— Concede l'accesso completo alle AWS Glue risorse quando un'identità a cui è allegata la policy utilizza la AWS console di gestione. Se si segue la convenzione per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questa policy è in genere associata agli utenti della AWS Glue console.

## Configurazione di Stripe

Prima di poter AWS Glue utilizzare il trasferimento di dati da Stripe, devi soddisfare questi requisiti:

Requisiti minimi

- Avere un account Stripe con e-mail e password. Per ulteriori informazioni, consultare Creazione di un nuovo account Stripe e configurazione dell'app client.
- L'account Stripe è abilitato per l'accesso alle API. Tutti gli utilizzi dell'API di Stripe sono disponibili senza costi aggiuntivi.

Se soddisfi questi requisiti, sei pronto per connetterti AWS Glue al tuo account Stripe.

## Configurazione delle connessioni di Stripe

Stripe supporta l'autenticazione personalizzata. Per ulteriori informazioni sulla generazione delle chiavi API richieste per l'autorizzazione personalizzata, consultare la documentazione dell'API REST STRIPE.

Per configurare una connessione di Stripe:

1. In AWS Secrets Manager, crea un segreto con i seguenti dettagli. È necessario creare un segreto per ogni connessione in AWS Glue.
   a. Per le app connesse gestite dal cliente: il segreto deve contenere l'app connessa Consumer Secret con `USER_MANAGED_CLIENT_APPLICATION_CLIENT_SECRET` come chiave.
2. In AWS Glue Glue Studio, crea una connessione in Connessioni dati seguendo i passaggi seguenti:
   a. In Tipo di connessione, selezionare Stripe.
   b. Seleziona il ruolo IAM che AWS Glue può assumere e dispone delle autorizzazioni per le seguenti azioni:

      JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:DescribeSecret",
        "secretsmanager:GetSecretValue",
        "secretsmanager:PutSecretValue",
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DeleteNetworkInterface"
      ],
      "Resource": "*"
    }
  ]
}
```

   c. Seleziona quello `secretName` che desideri utilizzare per questa connessione per AWS Glue inserire i token.
   d. Selezionare le opzioni di rete se si desidera utilizzare la propria rete.
3. Concedi il permesso di lettura `secretName` al ruolo IAM associato al tuo AWS Glue lavoro.

## Lettura da entità di Stripe

### Prerequisiti

- Un oggetto di Stripe da cui si desidera leggere.

Entità supportate

| Entità | Può essere Filtrato | Supporta Limite | Supporta Ordina per | Supporta Seleziona * | Supporta Partizionamento |
|---|---|---|---|---|---|
| Saldo | No | No | No | Sì | No |
| Transazioni di saldo | Sì | Sì | No | Sì | Sì |
| Costi | Sì | Sì | No | Sì | Sì |
| Controversie | Sì | Sì | No | Sì | Sì |
| LInk ai file | Sì | Sì | No | Sì | Sì |
| PaymentIntents | Sì | Sì | No | Sì | Sì |
| SetupIntents | Sì | Sì | No | Sì | Sì |
| Pagamenti | Sì | Sì | No | Sì | Sì |
| Rimborsi | Sì | Sì | No | Sì | Sì |
| Prodotti | Sì | Sì | No | Sì | Sì |
| Prezzi | Sì | Sì | No | Sì | Sì |
| Coupon | Sì | Sì | No | Sì | Sì |
| Codici promozionali | Sì | Sì | No | Sì | Sì |
| Codici fiscali | No | Sì | No | Sì | No |

| Entità | Può essere Filtrato | Supporta Limite | Supporta Ordina per | Supporta Seleziona * | Supporta Partizionamento |
|---|---|---|---|---|---|
| Aliquote fiscali | Sì | Sì | No | Sì | Sì |
| Tariffe di spedizione | Sì | Sì | No | Sì | Sì |
| Sessioni | Sì | Sì | No | Sì | Sì |
| Note di credito | Sì | Sì | No | Sì | Sì |
| Cliente | Sì | Sì | No | Sì | Sì |
| Fatture | Sì | Sì | No | Sì | Sì |
| Unità di fatturazione | Sì | Sì | No | Sì | No |
| Piani | Sì | Sì | No | Sì | Sì |
| Preventivi | Sì | Sì | No | Sì | No |
| Sottoscrizioni | Sì | Sì | No | Sì | |
| Articoli di abbonamento | No | Sì | No | Sì | No |
| Programmi di abbonamento | Sì | Sì | No | Sì | Sì |
| Account | No | Sì | No | Sì | Sì |
| Tariffe di applicazione | Sì | Sì | No | Sì | Sì |

| Entità | Può essere Filtrato | Supporta Limite | Supporta Ordina per | Supporta Seleziona * | Supporta Partizionamento |
|---|---|---|---|---|---|
| Specifiche del Paese | No | Sì | No | Sì | No |
| Trasferimenti | Sì | Sì | No | Sì | Sì |
| Avvisi anticipati di frode | Sì | Sì | No | Sì | Sì |
| Tipi di report | No | No | No | Sì | No |

Esempio

```
stripe_read = glueContext.create_dynamic_frame.from_options(
    connection_type="stripe",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "coupons",
        "API_VERSION": "v1"
    }
)
```

Dettagli dell'entità e del campo di Stripe

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| Saldo | | | |
| | disponibile | List | |
| | connect_reserved | List | |
| | in sospeso | List | |
| | livemode | Booleano | |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | oggetto | Stringa | |
| | instant_available | List | |
| | issuing | Struct | |
| Transazioni di saldo | | | |
| | id | Stringa | |
| | oggetto | Stringa | |
| | quantità | Numero intero | |
| | available_on | DateTime | =, >=, <=,<,> |
| | created | DateTime | =, >=, <=,<,> |
| | currency | Stringa | |
| | description | Stringa | |
| | exchange_rate | BigDecimal | |
| | fee | Numero intero | |
| | fee_details | List | |
| | net | Numero intero | |
| | reporting_category | Stringa | |
| | origine | Stringa | = |
| | status | Stringa | |
| | tipo | Stringa | = |
| | cross_border_classification | Stringa | |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| Costi | | | |
| | id | Stringa | |
| | oggetto | Stringa | |
| | quantità | Numero intero | =, <, > |
| | amount_captured | Numero intero | |
| | amount_refunded | Numero intero | |
| | applicazione | Stringa | |
| | application_fee | Stringa | |
| | application_fee_amount | Numero intero | |
| | balance_transaction | Stringa | |
| | billing_details | Struct | |
| | calculated_statement_descriptor | Stringa | |
| | captured | Booleano | |
| | created | DateTime | =, >=, <=,<,> |
| | currency | Stringa | |
| | customer | Stringa | = |
| | description | Stringa | |
| | destinazione | Stringa | |
| | dispute | Stringa | |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | disputed | Booleano | = |
| | failure_balance_transaction | Stringa | |
| | failure_code | Stringa | |
| | failure_message | Stringa | |
| | fraud_details | Struct | |
| | fattura | Stringa | |
| | livemode | Booleano | |
| | metadati | Struct | |
| | on_behalf_of | Stringa | |
| | order | Stringa | |
| | outcome | Struct | |
| | paid | Booleano | |
| | payment_intent | Stringa | = |
| | payment_method | Stringa | |
| | payment_method_details | Struct | |
| | receipt_email | Stringa | |
| | receipt_number | Stringa | |
| | receipt_url | Stringa | |
| | refunded | Booleano | = |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | refunds | Struct | |
| | review | Stringa | |
| | shipping | Struct | |
| | origine | Struct | |
| | source_transfer | Stringa | |
| | statement_descriptor | Stringa | |
| | statement_descriptor_suffix | Stringa | |
| | status | Stringa | |
| | transfer | Stringa | |
| | transfer_data | Struct | |
| | transfer_group | Stringa | = |
| Controversie | | | |
| | id | Stringa | |
| | oggetto | Stringa | |
| | quantità | Numero intero | =, <, > |
| | balance_transaction | Stringa | |
| | balance_transactions | List | |
| | charge | Stringa | = |
| | created | DateTime | =, >=, <=,<,> |
| | currency | Stringa | |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | evidence | Struct | |
| | evidence_details | Struct | |
| | is_charge_refundable | Booleano | |
| | livemode | Booleano | |
| | metadati | Struct | |
| | payment_intent | Stringa | = |
| | motivo | Stringa | = |
| | status | Stringa | |
| | payment_method_details | Struct | |
| Link ai file | | | |
| | id | Stringa | |
| | oggetto | Stringa | |
| | created | DateTime | =, >=, <=,<,> |
| | expired | Booleano | = |
| | expires_at | DateTime | |
| | file | Stringa | = |
| | livemode | Booleano | |
| | metadati | Struct | |
| | url | Stringa | |
| PaymentIntents | | | |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | id | Stringa | |
| | oggetto | Stringa | |
| | quantità | Numero intero | |
| | amount_capturable | Numero intero | |
| | amount_details | Struct | |
| | amount_received | Numero intero | |
| | applicazione | Stringa | |
| | application_fee_amount | Numero intero | |
| | automatic_payment_methods | Struct | |
| | canceled_at | DateTime | |
| | cancellation_reason | Stringa | |
| | capture_method | Stringa | |
| | client_secret | Stringa | |
| | confirmation_method | Stringa | |
| | created | DateTime | =, >=, <=,<,> |
| | currency | Stringa | |
| | customer | Stringa | = |
| | description | Stringa | |
| | fattura | Stringa | |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | last_payment_error | Struct | |
| | latest_charge | Stringa | |
| | livemode | Booleano | |
| | metadati | Struct | |
| | next_action | Struct | |
| | on_behalf_of | Stringa | |
| | payment_method | Stringa | |
| | payment_method_options | Struct | |
| | payment_method_types | List | |
| | payment_method_configuration_details | Struct | |
| | elaborazione | Struct | |
| | receipt_email | Stringa | |
| | review | Stringa | |
| | setup_future_usage | Stringa | |
| | shipping | Struct | |
| | origine | Stringa | |
| | statement_descriptor | Stringa | |
| | statement_descriptor_suffix | Stringa | |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | status | Stringa | |
| | transfer_data | Struct | |
| | transfer_group | Stringa | |
| SetupIntents | | | |
| | id | Stringa | |
| | oggetto | Stringa | |
| | applicazione | Stringa | |
| | cancellation_reason | Stringa | |
| | client_secret | Stringa | |
| | created | DateTime | =, >=, <=,<,> |
| | customer | Stringa | = |
| | description | Stringa | |
| | flow_directions | List | |
| | last_setup_error | Struct | |
| | latest_attempt | Stringa | |
| | livemode | Booleano | |
| | mandate | Stringa | |
| | metadati | Struct | |
| | next_action | Struct | |
| | on_behalf_of | Stringa | |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | payment_method | Stringa | |
| | payment_method_options | Struct | |
| | payment_method_types | List | |
| | single_use_mandate | Stringa | |
| | status | Stringa | |
| | utilizzo | Stringa | |
| | automatic_payment_methods | Struct | |
| Pagamenti | | | |
| | id | Stringa | |
| | oggetto | Stringa | |
| | quantità | Numero intero | =, <, > |
| | arrival_date | DateTime | =, >=, <=,<,> |
| | automatic | Booleano | |
| | balance_transaction | Stringa | |
| | created | DateTime | =, >=, <=,<,> |
| | currency | Stringa | |
| | description | Stringa | = |
| | destinazione | Stringa | |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | failure_balance_transaction | Stringa | |
| | failure_code | Stringa | |
| | failure_message | Stringa | |
| | livemode | Booleano | |
| | metadati | Struct | |
| | metodo | Stringa | |
| | original_payout | Stringa | |
| | reversed_by | Stringa | |
| | reconciliation_status | Stringa | |
| | source_type | Stringa | |
| | statement_descriptor | Stringa | |
| | status | Stringa | |
| | tipo | Stringa | |
| | application_fee | Stringa | |
| | application_fee_amount | Numero intero | |
| Rimborsi | | | |
| | id | Stringa | |
| | oggetto | Stringa | |
| | quantità | Numero intero | |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | balance_transaction | Stringa | |
| | charge | Stringa | = |
| | created | DateTime | =, >=, <=,<,> |
| | currency | Stringa | |
| | metadati | Struct | |
| | destination_details | Struct | |
| | payment_intent | Stringa | = |
| | motivo | Stringa | |
| | receipt_number | Stringa | |
| | source_transfer_reversal | Stringa | |
| | status | Stringa | |
| | transfer_reversal | Stringa | |
| Prodotti | | | |
| | id | Stringa | |
| | oggetto | Stringa | |
| | attiva | Booleano | = |
| | attributes | List | |
| | created | DateTime | =, >=, <=,<,> |
| | default_price | Stringa | |
| | description | Stringa | |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | images | List | |
| | livemode | Booleano | |
| | metadati | Struct | |
| | nome | Stringa | |
| | package_dimensions | Struct | |
| | shippable | Booleano | |
| | statement_descriptor | Stringa | |
| | tax_code | Stringa | |
| | tipo | Stringa | = |
| | unit_label | Stringa | |
| | updated | DateTime | |
| | url | Stringa | |
| | caratteristiche | List | |
| Prezzi | | | |
| | id | Stringa | |
| | oggetto | Stringa | |
| | attiva | Booleano | = |
| | billing_scheme | Stringa | |
| | created | DateTime | =, >=, <=,<,> |
| | currency | Stringa | = |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | custom_unit_amount | Struct | |
| | livemode | Booleano | |
| | lookup_key | Stringa | |
| | metadati | Struct | |
| | nickname | Stringa | |
| | product | Stringa | = |
| | recurring | Struct | |
| | tax_behavior | Stringa | |
| | tiers_mode | Stringa | |
| | transform_quantity | Struct | |
| | tipo | Stringa | = |
| | unit_amount | Numero intero | |
| | unit_amount_decimal | Stringa | |
| Coupon | | | |
| | Id | Stringa | |
| | oggetto | Stringa | |
| | amount_off | Numero intero | |
| | created | DateTime | =, >=, <=,<,> |
| | currency | Stringa | = |
| | durata | Stringa | = |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | duration_in_months | Numero intero | =,<,> |
| | livemode | Booleano | |
| | max_redemptions | Numero intero | =, <, > |
| | metadati | Struct | |
| | nome | Stringa | |
| | percent_off | Double | = |
| | redeem_by | DateTime | =, >=, <=, <, > |
| | times_redeemed | Numero intero | |
| | valid | Booleano | |
| Codici promozionali | | | |
| | Id | Stringa | |
| | oggetto | Stringa | |
| | attiva | Booleano | = |
| | code | Stringa | = |
| | coupon | Struct | |
| | created | DateTime | =,>=,<=,<,> |
| | customer | Stringa | |
| | expires_at | DateTime | |
| | livemode | Booleano | |
| | max_redemptions | Numero intero | |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | metadati | Struct | |
| | restrizioni | Struct | |
| | times_redeemed | Numero intero | |
| Codici fiscali | | | |
| | Id | Stringa | |
| | oggetto | Stringa | |
| | description | Stringa | |
| | nome | Stringa | |
| Aliquote fiscali | | | |
| | Id | Stringa | |
| | oggetto | Stringa | |
| | attiva | Booleano | = |
| | paese | Stringa | |
| | created | DateTime | =, >=, <=, <, > |
| | description | Stringa | |
| | display_name | Stringa | |
| | inclusive | Booleano | = |
| | jurisdiction | Stringa | |
| | jurisdiction_level | Stringa | |
| | livemode | Booleano | |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | metadati | Struct | |
| | percentage | Double | |
| | effective_percentage | Double | |
| | stato | Stringa | |
| | tax_type | Stringa | |
| Tariffe di spedizione | | | |
| | Id | Stringa | |
| | oggetto | Stringa | |
| | attiva | Booleano | = |
| | created | DateTime | =, >=, <=, <, > |
| | delivery_estimate | Struct | |
| | display_name | Stringa | |
| | fixed_amount | Struct | |
| | livemode | Booleano | |
| | metadati | Struct | |
| | tax_behavior | Stringa | |
| | tax_code | Stringa | |
| | tipo | Stringa | |
| Sessioni | | | |
| | id | Stringa | |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | oggetto | Stringa | |
| | after_expiration | Struct | |
| | allow_promotion_codes | Booleano | |
| | amount_subtotal | Numero intero | |
| | amount_total | Numero intero | |
| | automatic_tax | Struct | |
| | billing_address_collection | Stringa | |
| | cancel_url | Stringa | |
| | client_reference_id | Stringa | |
| | consent | Struct | |
| | consent_collection | Struct | |
| | created | DateTime | =, >=, <=, <, > |
| | currency | Stringa | |
| | custom_text | Struct | |
| | customer | Stringa | |
| | customer_creation | Stringa | |
| | customer_details | Struct | |
| | customer_email | Stringa | |
| | expires_at | DateTime | |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | fattura | Stringa | |
| | invoice_creation | Struct | |
| | livemode | Booleano | |
| | locale | Stringa | |
| | metadati | Struct | |
| | mode | Stringa | |
| | payment_intent | Stringa | = |
| | payment_link | Stringa | |
| | payment_method_collection | Stringa | |
| | payment_method_options | Struct | |
| | payment_method_types | List | |
| | payment_status | Stringa | |
| | phone_number_collection | Struct | |
| | recovered_from | Stringa | |
| | setup_intent | Stringa | |
| | shipping_address_collection | Struct | |
| | shipping_cost | Struct | |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | shipping_details | Struct | |
| | shipping_options | List | |
| | status | Stringa | |
| | submit_type | Stringa | |
| | sottoscrizione | Stringa | |
| | success_url | Stringa | |
| | tax_id_collection | Struct | |
| | total_details | Struct | |
| | url | Stringa | |
| | ui_mode | Stringa | |
| Note di credito | | | |
| | id | Stringa | |
| | oggetto | Stringa | |
| | quantità | Numero intero | |
| | created | DateTime | =, >=, <=, <, > |
| | currency | Stringa | |
| | customer | Stringa | = |
| | customer_balance_transaction | Stringa | |
| | discount_amount | Numero intero | |
| | discount_amounts | List | |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | fattura | Stringa | = |
| | lines | Struct | |
| | livemode | Booleano | |
| | memo | Stringa | |
| | metadati | Struct | |
| | numero | Stringa | |
| | out_of_band_amount | Numero intero | |
| | pdf | Stringa | |
| | motivo | Stringa | |
| | refund | Stringa | |
| | status | Stringa | |
| | subtotal | Numero intero | |
| | subtotal_excluding_tax | Numero intero | |
| | tax_amounts | List | |
| | total | Numero intero | |
| | total_excluding_tax | Numero intero | |
| | tipo | Stringa | |
| | voided_at | DateTime | |
| | amount_shipping | Numero intero | |
| | effective_at | DateTime | |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | shipping_cost | Struct | |
| Cliente | | | |
| | id | Stringa | |
| | oggetto | Stringa | |
| | address | Struct | |
| | balance | Numero intero | |
| | created | DateTime | |
| | currency | Stringa | =, >=, <=, <, > |
| | default_source | Stringa | |
| | delinquent | Booleano | = |
| | description | Stringa | |
| | discount | Struct | |
| | e-mail | Stringa | = |
| | invoice_prefix | Stringa | |
| | invoice_settings | Struct | |
| | livemode | Booleano | |
| | metadati | Struct | |
| | nome | Stringa | |
| | next_invoice_sequence | Numero intero | |
| | telefono | Stringa | |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | preferred_locales | List | |
| | shipping | Struct | |
| | tax_exempt | Stringa | |
| | test_clock | Stringa | |
| Fatture | | | |
| | id | Stringa | |
| | oggetto | Stringa | |
| | account_country | Stringa | |
| | account_name | Stringa | |
| | account_tax_ids | List | |
| | amount_due | Numero intero | |
| | amount_paid | Numero intero | |
| | amount_remaining | Numero intero | |
| | applicazione | Stringa | |
| | application_fee_amount | Numero intero | |
| | attempt_count | Numero intero | |
| | attempted | Booleano | = |
| | auto_advance | Booleano | = |
| | automatic_tax | Struct | |
| | billing_reason | Stringa | |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | charge | Stringa | |
| | collection_method | Stringa | = |
| | created | DateTime | =, >=, <=, <, > |
| | currency | Stringa | |
| | custom_fields | List | |
| | customer | Stringa | = |
| | customer_address | Struct | |
| | customer_email | Stringa | |
| | customer_name | Stringa | |
| | customer_phone | Stringa | |
| | customer_shipping | Struct | |
| | customer_tax_exempt | Stringa | |
| | customer_tax_ids | List | |
| | default_payment_method | Stringa | |
| | default_source | Stringa | |
| | default_tax_rates | List | |
| | description | Stringa | |
| | discount | Struct | |
| | discounts | List | |
| | due_date | DateTime | =, >=, <=, <, > |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | ending_balance | Numero intero | |
| | footer | Stringa | |
| | from_invoice | Struct | |
| | hosted_invoice_url | Stringa | |
| | invoice_pdf | Stringa | |
| | last_finalization_error | Struct | |
| | latest_revision | Stringa | |
| | lines | Struct | |
| | livemode | Booleano | |
| | metadati | Struct | |
| | next_payment_attempt | DateTime | |
| | numero | Stringa | |
| | on_behalf_of | Stringa | |
| | paid | Booleano | = |
| | paid_out_of_band | Booleano | |
| | payment_intent | Stringa | |
| | payment_settings | Struct | |
| | period_end | DateTime | =, >=, <=, <, > |
| | period_start | DateTime | =, >=, <=, <, > |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | post_payment_credit_notes_amount | Numero intero | |
| | pre_payment_credit_notes_amount | Numero intero | |
| | quote | Stringa | |
| | receipt_number | Stringa | |
| | rendering | Struct | |
| | rendering_options | Struct | |
| | starting_balance | Numero intero | |
| | statement_descriptor | Stringa | |
| | status | Stringa | = |
| | status_transitions | Struct | |
| | sottoscrizione | Stringa | |
| | subscription_details | Struct | |
| | subtotal | Numero intero | =, <, > |
| | subtotal_excluding_tax | Numero intero | |
| | tax | Numero intero | |
| | test_clock | Stringa | |
| | total | Numero intero | =, <, > |
| | total_discount_amounts | List | |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | total_excluding_tax | Numero intero | |
| | total_tax_amounts | List | |
| | transfer_data | Struct | |
| | webhooks_delivered_at | DateTime | |
| | automatically_finalizes_at | DateTime | |
| | effective_at | DateTime | |
| | approvatore | Struct | |
| Unità di fatturazione | | | |
| | id | Stringa | |
| | oggetto | Stringa | |
| | quantità | Numero intero | =, <, > |
| | currency | Stringa | |
| | customer | Stringa | = |
| | data | DateTime | |
| | description | Stringa | |
| | discountable | Booleano | |
| | discounts | List | |
| | fattura | Stringa | = |
| | livemode | Booleano | |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | metadati | Struct | |
| | punto | Struct | |
| | plan | Struct | |
| | price | Struct | |
| | proration | Booleano | = |
| | quantity | Numero intero | |
| | sottoscrizione | Stringa | |
| | subscription_item | Stringa | |
| | tax_rates | List | |
| | test_clock | Stringa | |
| | unit_amount | Numero intero | |
| | unit_amount_decimal | Stringa | |
| Piani | | | |
| | id | Stringa | |
| | oggetto | Stringa | |
| | attiva | Booleano | = |
| | aggregate_usage | Stringa | |
| | quantità | Numero intero | |
| | amount_decimal | Stringa | |
| | billing_scheme | Stringa | |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | created | DateTime | =, >=, <=, <, > |
| | currency | Stringa | = |
| | intervallo | Stringa | = |
| | interval_count | Numero intero | |
| | livemode | Booleano | |
| | metadati | Struct | |
| | nickname | Stringa | |
| | product | Stringa | = |
| | tiers_mode | Stringa | |
| | transform_usage | Struct | |
| | trial_period_days | Numero intero | =, <, > |
| | usage_type | Stringa | |
| | meter | Stringa | |
| Preventivi | | | |
| | id | Stringa | |
| | oggetto | Stringa | |
| | amount_subtotal | Numero intero | |
| | amount_total | Numero intero | |
| | applicazione | Stringa | |
| | application_fee_amount | Numero intero | |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | application_fee_percent | Double | |
| | automatic_tax | Struct | |
| | collection_method | Stringa | |
| | calcolato | Struct | |
| | created | DateTime | |
| | currency | Stringa | |
| | customer | Stringa | = |
| | default_tax_rates | List | |
| | description | Stringa | |
| | discounts | List | |
| | expires_at | DateTime | |
| | footer | Stringa | |
| | from_quote | Struct | |
| | intestazione | Stringa | |
| | fattura | Stringa | |
| | invoice_settings | Struct | |
| | livemode | Booleano | |
| | metadati | Struct | |
| | numero | Stringa | |
| | on_behalf_of | Stringa | |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | status | Stringa | = |
| | status_transitions | Struct | |
| | sottoscrizione | Stringa | |
| | subscription_data | Struct | |
| | subscription_schedule | Stringa | |
| | test_clock | Stringa | |
| | total_details | Struct | |
| | transfer_data | Struct | |
| Sottoscrizioni | | | |
| | id | Stringa | |
| | oggetto | Stringa | |
| | applicazione | Stringa | |
| | application_fee_percent | Double | |
| | automatic_tax | Struct | |
| | billing_cycle_anchor | DateTime | |
| | billing_thresholds | Struct | |
| | cancel_at | DateTime | |
| | cancel_at_period_end | Booleano | |
| | canceled_at | DateTime | |
| | collection_method | Stringa | = |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | created | DateTime | =, >=, <=,<,> |
| | currency | Stringa | |
| | current_period_end | DateTime | =, >=, <= |
| | current_period_start | DateTime | =, >=, <= |
| | customer | Stringa | = |
| | days_until_due | Numero intero | |
| | default_payment_method | Stringa | |
| | default_source | Stringa | |
| | default_tax_rates | List | |
| | description | Stringa | |
| | discount | Struct | |
| | ended_at | DateTime | |
| | items | Struct | |
| | latest_invoice | Stringa | |
| | livemode | Booleano | |
| | metadati | Struct | |
| | next_pending_invoice_item_invoice | DateTime | |
| | pause_collection | Struct | |
| | payment_settings | Struct | |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | pending_invoice_item_interval | Struct | |
| | pending_setup_intent | Stringa | |
| | pending_update | Struct | |
| | plan | Struct | |
| | quantity | Numero intero | |
| | schedule | Stringa | |
| | start_date | DateTime | |
| | status | Stringa | = |
| | test_clock | Stringa | |
| | transfer_data | Struct | |
| | trial_end | DateTime | |
| | trial_start | DateTime | |
| Articoli di abbonamento | | | |
| | Id | Stringa | |
| | oggetto | Stringa | |
| | billing_thresholds | Struct | |
| | created | DateTime | =, >=, <=, <, > |
| | metadati | Struct | |
| | plan | Struct | |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | price | Struct | |
| | sottoscrizione | Stringa | |
| | tax_rates | List | |
| | discounts | List | |
| Programmi di abbonamento | | | |
| | oggetto | Stringa | |
| | applicazione | Stringa | |
| | canceled_at | DateTime | |
| | completed_at | DateTime | |
| | created | DateTime | |
| | current_phase | Struct | |
| | customer | Stringa | = |
| | default_settings | Struct | |
| | end_behavior | Stringa | |
| | livemode | Booleano | |
| | metadati | Struct | |
| | phases | List | |
| | released_at | DateTime | |
| | released_subscription | Stringa | |
| | renewal_interval | Stringa | |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | status | Stringa | |
| | sottoscrizione | Stringa | |
| | test_clock | Stringa | |
| Account | | | |
| | details_submitted | Booleano | |
| | tos_acceptance | Struct | |
| | tipo | Stringa | |
| | metadati | Struct | |
| | id | Stringa | |
| | oggetto | Stringa | |
| | default_currency | Stringa | |
| | capabilities | Struct | |
| | charges_enabled | Booleano | |
| | impostazioni | Struct | |
| | requirements | Struct | |
| | payouts_enabled | Booleano | |
| | future_requirements | Struct | |
| | external_accounts | Struct | |
| | controller | Struct | |
| | paese | Stringa | |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | e-mail | Stringa | |
| | created | DateTime | =, >=, <=, <, > |
| | business_profile | Struct | |
| | business_type | Stringa | |
| | company | Struct | |
| Tariffe di applicazione | | | |
| | id | Stringa | |
| | oggetto | Stringa | |
| | account | Stringa | |
| | quantità | Numero intero | =, <, > |
| | amount_refunded | Numero intero | =, <, > |
| | applicazione | Stringa | |
| | balance_transaction | Stringa | |
| | charge | Stringa | = |
| | created | DateTime | |
| | currency | Stringa | |
| | livemode | Booleano | |
| | originating_transaction | Stringa | |
| | refunded | Booleano | = |
| | refunds | Struct | |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | fee_source | Struct | |
| Specifiche del Paese | | | |
| | id | Stringa | |
| | oggetto | Stringa | |
| | default_currency | Stringa | |
| | supported_bank_account_currencies | Struct | |
| | supported_payment_currencies | List | |
| | supported_payment_methods | List | |
| | supported_transfer_countries | List | |
| | verification_fields | Struct | |
| Trasferimenti | | | |
| | id | Stringa | |
| | oggetto | Stringa | |
| | quantità | Numero intero | =, <, > |
| | amount_reversed | Numero intero | |
| | balance_transaction | Stringa | |
| | created | DateTime | =, >=, <=, <, > |
| | currency | Stringa | = |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | description | Stringa | |
| | destinazione | Stringa | = |
| | destination_payment | Stringa | |
| | livemode | Booleano | |
| | metadati | Struct | |
| | reversals | Struct | |
| | reversed | Booleano | |
| | source_transaction | Stringa | |
| | source_type | Stringa | |
| | transfer_group | Stringa | = |
| Avvisi anticipati di frode | | | |
| | id | Stringa | |
| | oggetto | Stringa | |
| | actionable | Booleano | |
| | charge | Stringa | = |
| | created | DateTime | =, >=, <=, <, > |
| | fraud_type | Stringa | |
| | livemode | Booleano | |
| | payment_intent | Stringa | = |
| Tipi di report | | | |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | id | Stringa | |
| | oggetto | Stringa | |
| | data_available_end | DateTime | |
| | data_available_start | DateTime | |
| | default_columns | List | |
| | livemode | Booleano | |
| | nome | Stringa | |
| | updated | DateTime | |
| | version | Numero intero | |

Query di partizionamento

Se si desidera utilizzare la simultaneità in Spark, possono essere fornite opzioni Spark `PARTITION_FIELD`, `LOWER_BOUND`, `UPPER_BOUND`, `NUM_PARTITIONS` aggiuntive. Con questi parametri, la query originale viene suddivisa in `NUM_PARTITIONS` query secondarie che possono essere eseguite contemporaneamente dalle attività Spark.

- `PARTITION_FIELD`: il nome del campo da utilizzare per partizionare la query.
- `LOWER_BOUND`: un valore limite inferiore inclusivo del campo di partizione scelto.

  Per la data, accettiamo il formato di data di Spark utilizzato nelle query di Spark SQL. Esempio di valore valido: `"2024-07-01T00:00:00.000Z"`.
- `UPPER_BOUND`: un valore limite superiore esclusivo del campo di partizione scelto.
- `NUM_PARTITIONS`: numero di partizioni.

I dettagli di supporto del campo di partizionamento per entità sono riportati nella tabella di seguito.

| Nome entità | Campo di partizionamento | Tipo di dati |
|---|---|---|
| Transazioni di saldo | created | DateTime |
| Costi | created | DateTime |
| Controversie | created | DateTime |
| Link ai file | created | DateTime |
| PaymentIntents | created | DateTime |
| SetupIntents | created | DateTime |
| Pagamenti | created | DateTime |
| Rimborsi | created | DateTime |
| Prodotti | created | DateTime |
| Prezzi | created | DateTime |
| Coupon | created | DateTime |
| Codici promozionali | created | DateTime |
| Aliquote fiscali | created | DateTime |
| Tariffe di spedizione | created | DateTime |
| Sessioni | created | DateTime |
| Note di credito | created | DateTime |
| Cliente | created | DateTime |
| Fatture | created | DateTime |
| Piani | created | DateTime |
| Sottoscrizioni | created | DateTime |

| Nome entità | Campo di partizionamento | Tipo di dati |
|---|---|---|
| Programmi di abbonamento | created | DateTime |
| Account | created | DateTime |
| Tariffe di applicazione | created | DateTime |
| Trasferimenti | created | DateTime |
| Avvisi anticipati di frode | created | DateTime |

Esempio

```
stripe_read = glueContext.create_dynamic_frame.from_options(
    connection_type="stripe",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "coupons",
        "API_VERSION": "v1",
        "PARTITION_FIELD": "created"
        "LOWER_BOUND": "2024-05-01T20:55:02.000Z"
        "UPPER_BOUND": "2024-07-11T20:55:02.000Z"
        "NUM_PARTITIONS": "10"
    }
)
```

## Opzioni di connessione di Stripe

Di seguito sono riportate le opzioni di connessione per Stripe:

- `ENTITY_NAME`(Stringa): (obbligatorio) utilizzato per lettura/scrittura. Il nome dell'oggetto in Stripe.
- `API_VERSION`(Stringa): (obbligatorio) utilizzato per lettura/scrittura. Versione di Rest API di Stripe da utilizzare. Esempio: v1.
- `SELECTED_FIELDS`(List<String>): predefinito, vuoto (SELECT *). Utilizzato per la lettura. Colonne da selezionare per l'oggetto.
- `FILTER_PREDICATE`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Dev'essere in formato Spark SQL.
- `QUERY`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Query Spark SQL completa.

- `PARTITION_FIELD`(Stringa): utilizzato per la lettura. Campo da utilizzare per partizionare la query.
- `LOWER_BOUND`(Stringa): utilizzato per la lettura. Un valore limite inferiore inclusivo del campo di partizione scelto.
- `UPPER_BOUND`(Stringa): utilizzato per la lettura. Un valore limite superiore esclusivo del campo di partizione scelto.
- `NUM_PARTITIONS`(Numero intero): predefinito: 1. Utilizzato per la lettura. Numero di partizioni da leggere.

## Limitazioni

Di seguito sono riportate le limitazioni per il connettore di Stripe:

- Solo il partizionamento basato sul campo è supportato dal connettore.
- Il partizionamento basato sui record non è supportato dal connettore, nessuna disposizione per recuperare il conteggio totale dei record.
- Il tipo di dati della chiave primaria è String, quindi il partizionamento basato su ID non è supportato per connettore.

## Creazione di un nuovo account Stripe e configurazione dell'app client

Creazione di un account Stripe

1. Scegli sul link https://dashboard.stripe.com/register.
2. Inserire e-mail, nome completo, password e scegliere Crea account.
3. Dopo aver effettuato l'accesso con l'account, verificare l'account scegliendo Apri Gmail.
4. Verificare l'account facendo clic sul link di verifica ricevuto via e-mail.
5. Dopo aver fatto clic su Verifica indirizzo e-mail, si passerà a un'altra pagina
6. Dopo aver fatto clic su Attiva pagamenti per attivare l'account, verrà reindirizzato alla pagina Attiva pagamenti (https://dashboard.stripe.com/benvenuto) e assicurati di inserire tutti i dati validi, dopodiché scegli il pulsante Continua.

Creazione di un'app per sviluppatori Slack

1. Accedere a Stripe.

2. Scegliere Sviluppatori come mostrato nella parte superiore dell'immagine qui sotto.
3. Scegliere Chiavi API in Sviluppatori.
4. Scegliere Rivela chiave di test per ottenere le chiavi API.

## Connessione a Teradata Vantage in AWS Glue Studio

AWS Glue fornisce il supporto integrato per Teradata Vantage. AWS Glue Studio fornisce un'interfaccia visiva per connettersi a Teradata, creare processi di integrazione dei dati ed eseguirli sul runtime Spark serverless AWS Glue Studio.

AWS Glue Studio crea una connessione unificata per Teradata Vantage. Per ulteriori informazioni, consulta Considerazioni.

Argomenti

- Creazione di una connessione Teradata Vantage
- Creazione di un nodo di origine Teradata
- Creazione di un nodo di destinazione Teradata
- Opzioni avanzate

### Creazione di una connessione Teradata Vantage

Per connettersi a Teradata Vantage da AWS Glue, è necessario creare e archiviare le credenziali Teradata in modo Gestione dei segreti AWS segreto, quindi associare tale segreto a una connessione Teradata. AWS Glue

Prerequisiti:

- Se accedi al tuo ambiente Teradata tramite Amazon VPC, configura Amazon VPC per consentire al tuo AWS Glue job di comunicare con l'ambiente Teradata. Sconsigliamo l'accesso all'ambiente Teradata tramite la rete Internet pubblica.

  In Amazon VPC, identifica o crea un VPC, una sottorete e un gruppo di sicurezza da utilizzare durante l'esecuzione del AWS Glue lavoro. Inoltre, assicurati che Amazon VPC sia configurato per consentire il traffico di rete tra l'istanza Teradata e questa posizione. Il tuo processo dovrà stabilire una connessione TCP con la tua porta del client Teradata. Per ulteriori informazioni sulle porte Teradata, consulta la documentazione di Teradata.

In base al layout di rete, la connettività VPC sicura potrebbe richiedere modifiche ad Amazon VPC e ad altri servizi di rete. Per ulteriori informazioni sulla AWS connettività, consulta le opzioni di AWS connettività nella documentazione di Teradata.

Per configurare una connessione AWS Glue Teradata:

1. Nella configurazione Teradata, identifica o crea un utente e la password si AWS Glue connetterà a, e. *teradataUser* *teradataPassword* Per ulteriori informazioni, consulta Vantage Security Overview nella documentazione di Teradata.
2. Nel Gestione dei segreti AWS, crea un segreto utilizzando le tue credenziali Teradata. Per creare un segreto in Secrets Manager, segui il tutorial disponibile in Crea un Gestione dei segreti AWS segreto nella Gestione dei segreti AWS documentazione. Dopo aver creato il segreto, mantieni il nome del segreto *secretName* per il passaggio successivo.
   - Quando selezionate le coppie chiave/valore, create una coppia per la chiave `user` con il valore. *teradataUsername*
   - Quando selezionate le coppie chiave/valore, create una coppia per la chiave `password` con il valore. *teradataPassword*
3. Nella AWS Glue console, crea una connessione seguendo i passaggi riportati di seguito. the section called "Aggiungere una AWS Glue connessione" Dopo aver creato la connessione, mantieni il nome della connessione per il passaggio successivo. *connectionName*
   - In Tipo di connessione, seleziona Snowflake.
   - Quando fornisci JDBC URL, fornisci l'URL per la tua istanza. Puoi anche codificare determinati parametri di connessione, separati da virgole, nel tuo URL JDBC. L'URL deve rispettare il seguente formato: `jdbc:teradata://`*teradataHostname*`/`*ParameterName*`=`*ParameterValue*`,`*ParameterName*

     I parametri URL supportati includono:
     - `DATABASE`: nome del database sull'host a cui accedere per impostazione predefinita.
     - `DBS_PORT`: la porta del database, utilizzata con una porta non standard.
   - Quando selezioni un tipo di credenziale, seleziona Gestione dei segreti AWS, quindi imposta AWS Segreto *secretName* su.
4. Nelle seguenti situazioni, potresti aver bisogno di una configurazione aggiuntiva:

- Per le istanze Teradata ospitate su AWS un Amazon VPC
  - Dovrai fornire le informazioni di connessione Amazon VPC alla AWS Glue connessione che definisce le tue credenziali di sicurezza Teradata. Durante la creazione o l'aggiornamento della connessione, imposta VPC, sottorete e Gruppi di sicurezza nelle opzioni di rete.

## Creazione di un nodo di origine Teradata

Prerequisiti necessari

- Una connessione AWS Glue Teradata Vantage, configurata con un Gestione dei segreti AWS segreto, come descritto nella sezione precedente,. the section called "Creazione di una connessione Teradata Vantage"
- Autorizzazioni appropriate sul processo per leggere il segreto utilizzato dalla connessione.
- Una tabella Teradata da cui si desidera leggere o interrogare. *`tableName`* *`targetQuery`*

Aggiunta di un'origine dati Teradata

Per aggiungere un nodo origine dati: Teradata:

1. Scegli la connessione per la tua origine dati Teradata. Dato che l'hai creato, dovrebbe essere disponibile nel menu a discesa. Se devi creare una connessione, scegli Crea una nuova connessione. Per ulteriori informazioni, consulta la sezione the section called "Creazione di una connessione Teradata Vantage" precedente.

   Dopo aver scelto una connessione, puoi visualizzare le proprietà della connessione facendo clic su Visualizza proprietà.
2. Scegli un'opzione Origine Teradata:
   - Scegli una singola tabella: accedi a tutti i dati da un'unica tabella.
   - Inserisci una query personalizzata: accedi a un set di dati da più tabelle in base alla tua query personalizzata.
3. Se hai scelto una singola tabella, inserisci. *`tableName`*

   Se hai scelto Inserisci una query personalizzata, inserisci una query SQL SELECT.
4. In Proprietà personalizzate di Teradata, inserisci i parametri e i valori necessari.

### Creazione di un nodo di destinazione Teradata

Prerequisiti necessari

- Una connessione AWS Glue Teradata Vantage, configurata con un Gestione dei segreti AWS segreto, come descritto nella sezione precedente,. the section called "Creazione di una connessione Teradata Vantage"
- Autorizzazioni appropriate sul processo per leggere il segreto utilizzato dalla connessione.
- Una tabella Teradata su cui scrivere,. *tableName*

Aggiunta di una destinazione dati Teradata

Per aggiungere un nodo destinazione dati: Teradata:

1. Scegli la connessione per la tua origine dati Teradata. Dato che l'hai creato, dovrebbe essere disponibile nel menu a discesa. Se devi creare una connessione, scegli Crea connessione Teradata. Per ulteriori informazioni, consulta la pagina Overview of using connectors and connections.

   Dopo aver scelto una connessione, puoi visualizzare le proprietà della connessione facendo clic su Visualizza proprietà.
2. Configura il nome della tabella fornendo. *tableName*
3. In Proprietà personalizzate di Teradata, inserisci i parametri e i valori necessari.

### Opzioni avanzate

È possibile fornire opzioni avanzate durante la creazione di un nodo Teradata. Queste opzioni sono le stesse disponibili durante la programmazione AWS Glue per gli script Spark.

Per informazioni, consulta the section called "Connessioni Teradata Vantage".

## Connessione a Twilio

Twilio fornisce strumenti di comunicazione programmabili per effettuare e ricevere telefonate, inviare e ricevere messaggi di testo ed eseguire altre funzioni di comunicazione utilizzando il proprio servizio web. APIs Twilio APIs alimenta la sua piattaforma per le comunicazioni. Alla base di tutto ciò c' APIs è un livello software che collega e ottimizza le reti di comunicazione in tutto il mondo per consentire agli utenti di chiamare e inviare messaggi a chiunque, a livello globale. Come utente Twilio, puoi

connetterti AWS Glue al tuo account Twilio. Quindi, puoi usare Twilio come origine dati nei tuoi processi ETL. Esegui questi lavori per trasferire dati tra Twilio e i AWS servizi o altre applicazioni supportate.

Argomenti


- AWS Glue supporto per Twilio
- Policy contenenti le operazioni API per creare e utilizzare le connessioni
- Configurazione di Twilio
- Configurazione delle connessioni Twilio
- Lettura da entità Twilio
- Opzioni di connessione di Twilio
- Limitazioni e note per il connettore di Twilio


## AWS Glue supporto per Twilio

AWS Glue supporta Twilio come segue:

Supportato come origine?

Sì. È possibile utilizzare i job AWS Glue ETL per interrogare i dati di Twilio.

Supportato come destinazione?

No.

Versioni API di Twilio supportate

Sono supportate le seguenti versioni API di Twilio:

- v1
- 2010-04-01

## Policy contenenti le operazioni API per creare e utilizzare le connessioni

La seguente policy di esempio descrive le autorizzazioni AWS IAM richieste per la creazione e l'utilizzo delle connessioni. Se si sta creando un nuovo ruolo, creare una policy che contenga quanto segue:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:ListConnectionTypes",
        "glue:DescribeConnectionType",
        "glue:RefreshOAuth2Tokens",
        "glue:ListEntities",
        "glue:DescribeEntity"
      ],
      "Resource": "*"
    }
  ]
}
```

In alternativa, se non desideri utilizzare il metodo precedente, utilizza le seguenti policy IAM gestite:

- AWSGlueServiceRole— Garantisce l'accesso alle risorse che AWS Glue i vari processi richiedono per l'esecuzione per conto dell'utente. Queste risorse includono AWS Glue Amazon S3, IAM, CloudWatch Logs e Amazon. EC2 Se segui la convenzione di denominazione delle risorse specificata in questa politica, AWS Glue i processi dispongono delle autorizzazioni richieste. Questa policy è in genere collegata ai ruoli specificati quando si definiscono crawler, processi ed endpoint di sviluppo.
- AWSGlueConsoleFullAccess— Concede l'accesso completo alle AWS Glue risorse quando un'identità a cui è allegata la policy utilizza la AWS console di gestione. Se si segue la convenzione per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questa policy è in genere associata agli utenti della AWS Glue console.

## Configurazione di Twilio

Prima di poter AWS Glue utilizzare il trasferimento di dati da Twilio, devi soddisfare questi requisiti:

Requisiti minimi

I requisiti minimi sono i seguenti:

- Avere un account Twilio con nome utente e password.
- L'account Twilio è abilitato per l'accesso alle API.

Se soddisfi questi requisiti, sei pronto per connetterti AWS Glue al tuo account Twilio. Per le connessioni tipiche, non è necessario fare nient'altro in Twilio.

## Configurazione delle connessioni Twilio

Twilio supporta nome utente e password per l'autenticazione di base. L'autenticazione di base è un metodo di autenticazione semplice in cui i client forniscono direttamente le credenziali per accedere alle risorse protette. AWS Glue è in grado di utilizzare il nome utente (Account SID) e la password (Auth Token) per autenticare Twilio. APIs

Per la documentazione pubblica di Twilio per il flusso di autenticazione di base, consultare Autenticazione di base | Twilio.

Per configurare una connessione di Twilio:

1. In AWS Secrets Manager, crea un segreto con i seguenti dettagli:
   - Per l'autenticazione di base: il segreto deve contenere l'app connessa Consumer Secret con il SID dell'account (nome utente) e il token di autenticazione (password).

     > **Note**
     >
     > Devi creare un segreto per le tue connessioni in AWS Glue.

2. In AWS Glue Glue Studio, crea una connessione in Connessioni dati seguendo i passaggi seguenti:
   a. In Tipo di connessione, selezionare Twilio.
   b. Fornire `Edge_Location` dell'istanza Twilio a cui connettersi.
   c. Seleziona il ruolo AWS IAM che AWS Glue può assumere e dispone delle autorizzazioni per le seguenti azioni:

      JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
```

```
      "Effect": "Allow",
      "Action": [
        "secretsmanager:DescribeSecret",
        "secretsmanager:GetSecretValue",
        "secretsmanager:PutSecretValue",
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DeleteNetworkInterface"
      ],
      "Resource": "*"
    }
  ]
}
```

d. Seleziona quello `secretName` che desideri utilizzare per questa connessione per AWS Glue inserire i token.

e. Selezionare le opzioni di rete se si desidera utilizzare la propria rete.

3. Concedi il permesso di lettura `secretName` al ruolo IAM associato al tuo AWS Glue lavoro.

## Lettura da entità Twilio

Prerequisito

Un oggetto di Twilio da cui si desidera leggere. È necessario il nome dell'oggetto, come `SMS-Message` o `SMS-CountryPricing`.

Entità supportate come origine:

| Entità | Interfaccia | Può essere Filtrato | Supporta limite | Supporta Ordina per | Supporta Seleziona * | Supporta il partizionamento |
|---|---|---|---|---|---|---|
| SMS-Message | REST | Sì | Sì | No | Sì | Sì |
| SMS-CountryPricing | REST | No | No | No | Sì | No |
| Voice-Call | REST | Sì | Sì | No | Sì | No |

| Entità | Interfaccia | Può essere Filtrato | Supporta limite | Supporta Ordina per | Supporta Seleziona * | Supporta il partizionamento |
|---|---|---|---|---|---|---|
| Voice-Application | REST | Sì | Sì | No | Sì | No |
| OutgoingCallerIdentificazione vocale | REST | Sì | Sì | No | Sì | No |
| Voice-Queue | REST | Sì | Sì | No | Sì | No |
| Conversations-Conversation | REST | Sì | Sì | No | Sì | No |
| Conversations-User | REST | No | Sì | No | Sì | No |
| Conversations-Role | REST | No | Sì | No | Sì | No |
| Conversations-Configuration | REST | No | No | No | Sì | No |
| Conversazioni-AddressConfiguration | REST | Sì | Sì | No | Sì | No |
| Conversazioni-WebhookConfiguration | REST | No | No | No | Sì | No |

| Entità | Interfaccia | Può essere Filtrato | Supporta limite | Supporta Ordina per | Supporta Seleziona * | Supporta il partizionamento |
|---|---|---|---|---|---|---|
| Conversazioni-ParticipantConversation | REST | No | No | No | Sì | No |
| Conversations-Credential | REST | No | Sì | No | Sì | No |
| Conversazioni-ConversationService | REST | No | Sì | No | Sì | No |

Esempio:

```
twilio_read = glueContext.create_dynamic_frame.from_options(
    connection_type="twilio",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "sms-message",
        "API_VERSION": "2010-04-01",
        "Edge_Location": "sydney.us1"
    }
```

Dettagli dell'entità e del campo di Twilio

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| SMS-Message | account_sid | Stringa | N/A |
| | api_version | Stringa | N/A |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | body | Stringa | N/A |
| | date_created | Datetime | N/D |
| | date_sent | Datetime | >=, <=, = |
| | date_updated | Datetime | N/D |
| | direzione | Stringa | N/A |
| | error_code | Numero intero | N/D |
| | error_message | Stringa | N/A |
| | from | Numero intero | = |
| | messaging_service_sid | Stringa | N/A |
| | num_media | Stringa | N/A |
| | num_segments | Stringa | N/A |
| | price | Stringa | N/A |
| | price_unit | Struct | N/D |
| | sid | Numero intero | N/D |
| | status | Stringa | N/A |
| | subresource_uris | Eseguire la mappatura | N/D |
| | in | Numero intero | = |
| | uri | Datetime | N/D |
| SMS- CountryPricing | paese | Stringa | N/A |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | iso_country | Stringa | N/A |
| | url | Stringa | N/A |
| | outbound_sms_prices | List | N/D |
| | inbound_sms_prices | List | N/D |
| | price_unit | Stringa | N/A |
| Voice-Call | account_sid | Stringa | N/A |
| | annotazione | Stringa | N/A |
| | answered_by | Stringa | N/A |
| | api_version | Stringa | N/A |
| | caller_name | Stringa | N/A |
| | date_created | Datetime | N/D |
| | date_updated | Datetime | N/D |
| | direzione | Stringa | N/A |
| | durata | Stringa | N/A |
| | end_time | Datetime | >=, <=, = |
| | forwarded_from | Stringa | N/A |
| | from | Stringa | = |
| | from_formatted | Stringa | N/A |
| | group_sid | Stringa | N/A |
| | parent_call_sid | Stringa | N/A |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | phone_number_sid | Stringa | N/A |
| | price | Stringa | N/A |
| | price_unit | Stringa | N/A |
| | sid | Stringa | N/A |
| | start_time | Datetime | >=, <=, = |
| | status | Stringa | = |
| | subresource_uris | Stringa | N/A |
| | in | Stringa | = |
| | to_formatted | Stringa | N/A |
| | trunk_sid | Stringa | N/A |
| | uri | Stringa | N/A |
| | queue_time | Stringa | N/A |
| Voice-Application | account_sid | Stringa | N/A |
| | api_version | Stringa | N/A |
| | date_created | Datetime | N/D |
| | date_updated | Datetime | N/D |
| | friendly_name | Stringa | = |
| | message_status_callback | Stringa | N/A |
| | sid | Stringa | N/A |
| | sms_fallback_method | Stringa | N/A |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | sms_fallback_url | Stringa | N/A |
| | sms_method | Stringa | N/A |
| | sms_status_callback | Stringa | N/A |
| | sms_url | Stringa | N/A |
| | status_callback | Stringa | N/A |
| | status_callback_method | Stringa | N/A |
| | uri | Stringa | N/A |
| | voice_caller_id_lookup | Booleano | N/D |
| | voice_fallback_method | Stringa | N/A |
| | voice_fallback_url | Stringa | N/A |
| | voice_method | Stringa | N/A |
| | voice_url | Stringa | N/A |
| public_application_connect_enabled | Booleano | N/D | |
| OutgoingCallerIdentificazione vocale | sid | Stringa | N/A |
| | date_created | Datetime | N/D |
| | date_updated | Datetime | N/D |
| | account_sid | Stringa | N/A |
| | friendly_name | Stringa | = |
| | phone_number | Stringa | = |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | uri | Stringa | N/A |
| Voice-Queue | date_created | Datetime | N/D |
| | date_updated | Datetime | N/D |
| | current_size | Numero intero | N/D |
| | friendly_name | Stringa | N/A |
| | uri | Stringa | N/A |
| | account_sid | Stringa | N/A |
| | average_wait_time | Numero intero | N/D |
| | sid | Stringa | N/A |
| | max_size | Numero intero | N/D |
| Conversations-Conversation | account_sid | Stringa | N/A |
| | chat_service_sid | Stringa | N/A |
| | messaging_service_sid | Stringa | N/A |
| | sid | Stringa | N/A |
| | friendly_name | Stringa | N/A |
| | unique_name | Stringa | N/A |
| | attributes | Stringa | N/A |
| | stato | Stringa | = |
| | date_created | Datetime | N/D |
| | date_updated | Datetime | N/D |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | timers | Struct | N/D |
| | url | Stringa | N/A |
| | links | Struct | N/D |
| | bindings | Struct | N/D |
| | start_date | Datetime | = |
| | end_date | Datetime | = |
| | Timer. DateInactive | Stringa | N/A |
| | Timer. DateClosed | Stringa | N/A |
| Conversations-User | sid | Stringa | N/A |
| | account_sid | Stringa | N/A |
| | chat_service_sid | Stringa | N/A |
| | role_sid | Stringa | N/A |
| | identità | Stringa | N/A |
| | friendly_name | Stringa | N/A |
| | attributes | Stringa | N/A |
| | is_online | Booleano | N/D |
| | is_notifiable | Booleano | N/D |
| | date_created | Datetime | N/D |
| | date_updated | Datetime | N/D |
| | url | Stringa | N/A |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | links | Struct | N/D |
| Conversations-Role | sid | Stringa | N/A |
| | account_sid | Stringa | N/A |
| | chat_service_sid | Stringa | N/A |
| | friendly_name | Stringa | N/A |
| | tipo | Stringa | N/A |
| | autorizzazioni | Stringa | N/A |
| | date_created | Datetime | N/D |
| | date_updated | Datetime | N/D |
| | url | Stringa | N/A |
| Conversations-Configuration | account_sid | Long | N/D |
| | default_chat_service_sid | Stringa | N/A |
| | default_messaging_service_sid | Stringa | N/A |
| | default_inactive_timer | Stringa | N/A |
| | default_closed_timer | Stringa | N/A |
| | url | Stringa | N/A |
| | links | Eseguire la mappatura | N/D |
| Conversazioni-AddressConfiguration | sid | Stringa | N/A |
| | account_sid | Stringa | N/A |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | tipo | Stringa | N/A |
| | address | Stringa | N/A |
| | friendly_name | Stringa | N/A |
| | auto_creation | Struct | N/D |
| | date_created | Datetime | N/D |
| | date_updated | Datetime | N/D |
| | url | Stringa | N/A |
| | address_country | Stringa | N/A |
| | AutoCreation.Abilitato | Booleano | N/D |
| | AutoCreation.Tipo | Stringa | N/A |
| | AutoCreation.ConversationServiceSid | Stringa | N/A |
| | AutoCreation.WebhookUrl | Stringa | N/A |
| | AutoCreation.WebhookMethod | Stringa | N/A |
| | AutoCreation.WebhookFilters | List | N/D |
| | AutoCreation.StudioFlowSid | Stringa | N/A |
| | AutoCreation.StudioRetryCount | Numero intero | N/D |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| Conversazioni-WebhookConfiguration | account_sid | Stringa | N/A |
| | metodo | Stringa | N/A |
| | filtri | List | N/D |
| | pre_webhook_url | Stringa | N/A |
| | post_webhook_url | Stringa | N/A |
| | target | Stringa | N/A |
| | url | Stringa | N/A |
| Conversazioni-ParticipantConversation | account_sid | Stringa | N/A |
| | chat_service_sid | Stringa | N/A |
| | participant_sid | Stringa | N/A |
| | participant_user_sid | Stringa | N/A |
| | participant_identity | Stringa | N/A |
| | participant_messaging_binding | Struct | N/D |
| | Conversation_sid | Stringa | N/A |
| | conversation_unique_name | Stringa | N/A |
| | conversation_friendly_name | Stringa | N/A |
| | conversation_attributes | Stringa | N/A |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | conversation_date_created | Datetime | N/D |
| | conversation_date_updated | Datetime | N/D |
| | conversation_created_by | Stringa | N/A |
| | conversation_state | Stringa | N/A |
| | conversation_timers | Struct | N/D |
| | links | Eseguire la mappatura | N/D |
| | address | Stringa | = |
| | identità | Stringa | = |
| Conversation-Credentials | sid | Stringa | N/A |
| | account_sid | Stringa | N/A |
| | friendly_name | Stringa | N/A |
| | tipo | Stringa | N/A |
| | sandbox | Stringa | N/A |
| | date_created | Datetime | N/D |
| | dated_updated | Datetime | N/D |
| | url | Stringa | N/A |
| | certificato | Stringa | N/A |
| | private_key | Stringa | N/A |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | api_key | Stringa | N/A |
| | Secret | Stringa | N/A |
| Conversazioni-ConversationService | sid | Stringa | N/A |
| | account_sid | Stringa | N/A |
| | friendly_name | Stringa | N/A |
| | date_created | Datetime | N/D |
| | date_updated | Datetime | N/D |
| | url | Stringa | N/A |
| | links | Eseguire la mappatura | N/D |

Query di partizionamento

Campi che supportano il partizionamento:

In Twilio, i campi del DateTime tipo di dati supportano il partizionamento basato sul campo.

Puoi fornire le opzioni Spark aggiuntive `PARTITION_FIELD`, `LOWER_BOUND`, `UPPER_BOUND` e `NUM_PARTITIONS` se desideri utilizzare la concorrenza in Spark. Con questi parametri, la query originale viene suddivisa in `NUM_PARTITIONS` query secondarie che possono essere eseguite contemporaneamente dalle attività Spark.

- `PARTITION_FIELD`: il nome del campo da utilizzare per partizionare la query.
- `LOWER_BOUND`: un valore limite inferiore inclusivo del campo di partizione scelto.

  Per il campo Datetime accettiamo il formato di timestamp Spark utilizzato nelle query SQL di Spark.

  Esempi di valori validi:

```
"2024-05-01T20:55:02.000Z"
```

- UPPER_BOUND: un valore limite superiore esclusivo del campo di partizione scelto.
- NUM_PARTITIONS: il numero di partizioni.

Esempio:

```
twilio_read = glueContext.create_dynamic_frame.from_options(
    connection_type="twilio",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "sms-message",
        "API_VERSION": "2010-04-01",
        "PARTITION_FIELD": "date_sent"
        "LOWER_BOUND": "2024-05-01T20:55:02.000Z"
        "UPPER_BOUND": "2024-06-01T20:55:02.000Z"
        "NUM_PARTITIONS": "10"
    }
```

## Opzioni di connessione di Twilio

Di seguito sono riportate le opzioni di connessione per Twilio:

- ENTITY_NAME(String): (obbligatorio) utilizzato per la lettura. Il nome dell'oggetto in Twilio.
- EDGE_LOCATION(String): (obbligatorio) una posizione edge valida di Twilio.
- API_VERSION(String): (obbligatorio) utilizzato per la lettura. Versione dell'API Rest di Twilio da utilizzare. Twilio supporta due versioni dell'API: "v1" e "2010-04-01".
- SELECTED_FIELDS(List<String>): predefinito: vuoto (SELECT *). Utilizzato per la lettura. Colonne da selezionare per l'oggetto.
- FILTER_PREDICATE(Stringa): predefinito: vuoto. Utilizzato per la lettura. Dev'essere in formato Spark SQL.
- QUERY(Stringa): predefinito: vuoto. Utilizzato per la lettura. Query Spark SQL completa.
- PARTITION_FIELD(Stringa): utilizzato per la lettura. Campo da utilizzare per partizionare la query.
- LOWER_BOUND(Stringa): utilizzato per la lettura. Un valore limite inferiore inclusivo del campo di partizione scelto.
- UPPER_BOUND(Stringa): utilizzato per la lettura. Un valore limite superiore esclusivo del campo di partizione scelto.
- NUM_PARTITIONS(Numero intero): predefinito: 1. Utilizzato per la lettura. Numero di partizioni da leggere.

- `INSTANCE_URL`(Stringa): (obbligatorio) utilizzato per la lettura. Un URL di istanza valido di Twilio.

### Limitazioni e note per il connettore di Twilio

Di seguito sono riportate le limitazioni o le note per il connettore di Twilio:

- Il partizionamento basato su record non è supportato, in quanto non è previsto il recupero del numero totale di record da Twilio.
- I campi `date_sent`, `start_time` e `end_time` sono del tipo di dati Datetime, ma durante il filtraggio supportano solo i valori di data (i componenti temporali non vengono considerati).
- Il filtraggio dei campi "da" o "a" funziona solo se i valori non includono alcun prefisso (ad esempio, un protocollo o un'etichetta). Se è presente un prefisso, il filtraggio per il rispettivo campo non funziona. Ad esempio, se si passa "a": "whatsapp: +14xxxxxxxxxx" come filtro, Twilio non restituirà una risposta. Occorre passarlo come "to": "+14xxxxxxxx", quindi restituirà i record se esistono.
- Il filtro del campo "identity" è obbligatorio quando si effettua la query dell'entità `conversation-participant-conversation`.

## Connessione a Vertica in AWS Glue Studio

AWS Glue fornisce il supporto integrato per Vertica. AWS Glue Studiofornisce un'interfaccia visiva per connettersi a Vertica, creare processi di integrazione dei dati ed eseguirli sul runtime Spark serverless AWS Glue Studio.

AWS Glue Studio crea una connessione unificata per Vertica. Per ulteriori informazioni, consulta Considerazioni.

Argomenti

- Creazione di una connessione Vertica
- Creazione di un nodo di origine Vertica
- Creazione di un nodo di destinazione Vertica
- Opzioni avanzate

### Creazione di una connessione Vertica

Prerequisiti:

- Un bucket o una cartella Amazon S3 da utilizzare per l'archiviazione temporanea durante la lettura e la scrittura sul database, a cui si fa riferimento da. *tempS3Path*

> **Note**
>
> Quando si utilizza Vertica nelle anteprime dei dati di AWS Glue lavoro, i file temporanei potrebbero non essere rimossi automaticamente da. *tempS3Path* Per garantire la rimozione dei file temporanei, interrompi direttamente la sessione di anteprima dei dati scegliendo Termina sessione nel riquadro Anteprima dei dati.
> Se non sei in grado di terminare direttamente la sessione di anteprima dei dati, valuta la possibilità di impostare la configurazione del ciclo di vita di Amazon S3 per rimuovere i dati obsoleti. Consigliamo di rimuovere i dati più vecchi di 49 ore, in base al runtime massimo del processo in aggiunta a un margine. Per ulteriori informazioni sulla configurazione del ciclo di vita di Amazon S3, consulta Gestione del ciclo di vita dello storage nella documentazione di Amazon S3.

- Una policy IAM con autorizzazioni appropriate per il tuo percorso Amazon S3 che puoi associare al AWS Glue tuo ruolo lavorativo.
- Se la tua istanza Vertica si trova in un Amazon VPC, configura Amazon VPC per consentire AWS Glue al processo di comunicare con l'istanza Vertica senza che il traffico attraversi la rete Internet pubblica.

  In Amazon VPC, identifica o crea un VPC, una sottorete e un gruppo di sicurezza da utilizzare durante l'esecuzione del AWS Glue lavoro. Inoltre, assicurati che Amazon VPC sia configurato per consentire il traffico di rete tra l'istanza Vertica e questa posizione. Il tuo processo dovrà stabilire una connessione TCP con la tua porta del client Vertica, (per impostazione predefinita, 5433). In base al layout della rete, ciò potrebbe richiedere modifiche alle regole del gruppo di sicurezza, alla rete ACLs, ai gateway NAT e alle connessioni peering.

Per configurare una connessione a Vertica:

1. Nel Gestione dei segreti AWS, crea un segreto utilizzando le tue credenziali Vertica, e. *verticaUsername verticaPassword* Per creare un segreto in Secrets Manager, segui il tutorial disponibile in Crea un Gestione dei segreti AWS segreto nella Gestione dei segreti AWS documentazione. Dopo aver creato il segreto, mantieni il nome del segreto *secretName* per il passaggio successivo.

- Quando selezionate le coppie chiave/valore, create una coppia per la chiave `user` con il valore. *verticaUsername*
- Quando selezionate le coppie chiave/valore, create una coppia per la chiave `password` con il valore. *verticaPassword*

2. Nella AWS Glue console, crea una connessione seguendo la procedura riportata di seguito. the section called "Aggiungere una AWS Glue connessione" Dopo aver creato la connessione, mantieni il nome della connessione per il passaggio successivo. *connectionName*

   - In Tipo di connessione, seleziona Vertica.
   - In Host Vertica, fornisci il nome host dell'installazione Vertica.
   - In Porta Vertica, indica la porta tramite cui è disponibile l'installazione di Vertica.
   - Quando selezioni un AWS segreto, fornisci*secretName*.

3. Nelle seguenti situazioni, potresti aver bisogno di una configurazione aggiuntiva:

   - Per le istanze Vertica ospitate su un AWS Amazon VPC
     - Fornisci le informazioni di connessione Amazon VPC alla AWS Glue connessione che definisce le tue credenziali di sicurezza Vertica. Durante la creazione o l'aggiornamento della connessione, imposta VPC, sottorete e Gruppi di sicurezza nelle opzioni di rete.

Prima di eseguire il lavoro, dovrai eseguire i seguenti passaggi: AWS Glue

- Concedi il ruolo IAM associato alle tue autorizzazioni AWS Glue lavorative a*tempS3Path*.
- Concedi al ruolo IAM associato al tuo AWS Glue lavoro il permesso di lettura*secretName*.

## Creazione di un nodo di origine Vertica

Prerequisiti necessari

- Una connessione AWS Glue Data Catalog di tipo Vertica *connectionName* e una posizione Amazon S3 temporanea*tempS3Path*, come descritto nella sezione precedente,. the section called "Creazione di una connessione Vertica"
- Una tabella Vertica da cui desideri leggere o interrogare*tableName*. *targetQuery*

### Aggiunta di un'origine dati Vertica

Per aggiungere un nodo origine dati: Vertica:

1. Scegli la connessione per la tua origine dati Vertica. Dato che l'hai creato, dovrebbe essere disponibile nel menu a discesa. Se devi creare una connessione, scegli Crea connessione Vertica. Per ulteriori informazioni, consulta la sezione the section called "Creazione di una connessione Vertica" precedente.

   Dopo aver scelto una connessione, puoi visualizzare le proprietà della connessione facendo clic su Visualizza proprietà.
2. Scegli il Database contenente la tabella.
3. Scegli l'area di staging in Amazon S3, inserisci un URI S3A in. *`tempS3Path`*
4. Scegli Origine Vertica.
   - Scegli una singola tabella: accedi a tutti i dati da un'unica tabella.
   - Inserisci una query personalizzata: accedi a un set di dati da più tabelle in base alla tua query personalizzata.
5. Se hai scelto una singola tabella, inserisci ***`tableName`*** e, facoltativamente, seleziona uno schema.

   Se hai scelto Inserisci una query personalizzata, inserisci una query SQL SELECT e, facoltativamente, seleziona uno schema.
6. In Proprietà personalizzate di Vertica, inserisci i parametri e i valori necessari.

## Creazione di un nodo di destinazione Vertica

### Prerequisiti necessari

- Una connessione AWS Glue Data Catalog di tipo Vertica *`connectionName`* e una posizione Amazon S3 temporanea*`tempS3Path`*, come descritto nella sezione precedente,. the section called "Creazione di una connessione Vertica"

Aggiunta di una destinazione dati Vertica

Per aggiungere un nodo destinazione dati: Vertica:

1. Scegli la connessione per la tua origine dati Vertica. Dato che l'hai creato, dovrebbe essere disponibile nel menu a discesa. Se devi creare una connessione, scegli Crea connessione Vertica. Per ulteriori informazioni, consulta la sezione the section called "Creazione di una connessione Vertica" precedente.

   Dopo aver scelto una connessione, puoi visualizzare le proprietà della connessione facendo clic su Visualizza proprietà.
2. Scegli il Database contenente la tabella.
3. Scegli l'area di staging in Amazon S3, inserisci un URI S3A in. *`tempS3Path`*
4. Inserisci ***`tableName`*** e, facoltativamente, seleziona uno schema.
5. In Proprietà personalizzate di Vertica, inserisci i parametri e i valori necessari.

### Opzioni avanzate

È possibile fornire opzioni avanzate durante la creazione di un nodo Vertica. Queste opzioni sono le stesse disponibili durante la programmazione AWS Glue per gli script Spark.

Per informazioni, consulta the section called "Connessioni Vertica".

## Connessione a WooCommerce

WooCommerce è una soluzione software flessibile open source creata per siti Web WordPress basati. Viene comunemente utilizzata per creare negozi di e-commerce online. Con questa soluzione software, chiunque può trasformare il proprio sito web tradizionale in un negozio online perfettamente funzionante.

Argomenti

- AWS Glue supporto per WooCommerce
- Policy contenenti le operazioni API per creare e utilizzare le connessioni
- Configurazione WooCommerce
- Configurazione delle connessioni WooCommerce
- Lettura da WooCommerce entità
- WooCommerce opzioni di connessione

## AWS Glue supporto per WooCommerce

AWS Glue supporta WooCommerce quanto segue:

Supportato come origine?

Sì. È possibile utilizzare i job AWS Glue ETL da WooCommerce cui interrogare i dati.

Supportato come destinazione?

No.

Versioni WooCommerce API supportate

Sono supportate le seguenti versioni WooCommerce API:

- v3

## Policy contenenti le operazioni API per creare e utilizzare le connessioni

La seguente policy di esempio descrive le autorizzazioni AWS IAM richieste per la creazione e l'utilizzo delle connessioni. Se si sta creando un nuovo ruolo, creare una policy che contenga quanto segue:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:ListConnectionTypes",
        "glue:DescribeConnectionType",
        "glue:RefreshOAuth2Tokens",
        "glue:ListEntities",
        "glue:DescribeEntity"
      ],
      "Resource": "*"
    }
  ]
```

```
}
```

In alternativa, se non desideri utilizzare il metodo precedente, utilizza le seguenti policy IAM gestite:

- AWSGlueServiceRole— Garantisce l'accesso alle risorse che AWS Glue i vari processi richiedono per l'esecuzione per conto dell'utente. Queste risorse includono AWS Glue Amazon S3, IAM, CloudWatch Logs e Amazon. EC2 Se segui la convenzione di denominazione delle risorse specificata in questa politica, AWS Glue i processi dispongono delle autorizzazioni richieste. Questa policy è in genere collegata ai ruoli specificati quando si definiscono crawler, processi ed endpoint di sviluppo.
- AWSGlueConsoleFullAccess— Concede l'accesso completo alle AWS Glue risorse quando un'identità a cui è allegata la policy utilizza la AWS console di gestione. Se si segue la convenzione per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questa policy è in genere associata agli utenti della AWS Glue console.

## Configurazione WooCommerce

Prima di poter utilizzare AWS Glue per trasferire dati da WooCommerce, è necessario soddisfare i seguenti requisiti:

Requisiti minimi

I requisiti minimi sono i seguenti:

- Hai un WooCommerce account con a `consumerKey` e `aconsumerSecret`.
- Il tuo WooCommerce account ha accesso all'API con una licenza valida.

Se soddisfi questi requisiti, sei pronto per connetterti AWS Glue al tuo WooCommerce account. Per le connessioni tipiche, non devi fare nient'altro WooCommerce.

## Configurazione delle connessioni WooCommerce

WooCommerce supporta l'autenticazione personalizzata. Per la WooCommerce documentazione pubblica sulla generazione delle chiavi API richieste per l'autorizzazione personalizzata, consulta Authentication — WooCommerce REST API Documentation.

Per configurare una WooCommerce connessione:

1. In AWS Secrets Manager, crea un segreto con i seguenti dettagli:
   - Per un'app connessa gestita dal cliente, il segreto deve contenere l'app connessa Consumer Secret con `consumerKey` e `consumerSecret` come chiavi. Nota: è necessario creare un segreto per la connessione in AWS Glue.

1. In AWS Glue Glue Studio, crea una connessione in Connessioni dati seguendo i passaggi seguenti:
   a. Quando si seleziona un tipo di connessione, selezionare WooCommerce.
   b. Fornisci `INSTANCE_URL` l' WooCommerce istanza a cui desideri connetterti.
   c. Seleziona il ruolo AWS IAM che AWS Glue può assumere e dispone delle autorizzazioni per le seguenti azioni:

      JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:DescribeSecret",
        "secretsmanager:GetSecretValue",
        "secretsmanager:PutSecretValue",
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DeleteNetworkInterface"
      ],
      "Resource": "*"
    }
  ]
}
```

   d. Seleziona quello `secretName` che desideri utilizzare per questa connessione per AWS Glue inserire i token.
   e. Selezionare le opzioni di rete se si desidera utilizzare la propria rete.
2. Concedi il permesso di lettura `secretName` al ruolo IAM associato al tuo AWS Glue lavoro.

## Lettura da WooCommerce entità

Prerequisito

Un WooCommerce oggetto da cui vorresti leggere. È necessario il nome dell'oggetto come coupon, ordine, prodotto, ecc.

Entità supportate come origine:

| Entità | Può essere filtrato | Supporta limite | Supporta Ordina per | Supporta Seleziona * | Supporta partizionamento |
|---|---|---|---|---|---|
| Coupon | Sì | Sì | Sì | Sì | Sì |
| Totale coupon | No | No | No | Sì | No |
| Totale clienti | No | No | No | Sì | No |
| Ordine | Sì | Sì | Sì | Sì | Sì |
| Totale ordini | No | No | No | Sì | No |
| Gateway di pagamento | No | No | No | Sì | No |
| Prodotto | Sì | Sì | Sì | Sì | Sì |
| Attributo del prodotto | Sì | Sì | Sì | Sì | Sì |
| Categoria del prodotto | Sì | Sì | Sì | Sì | Sì |
| Recensione del prodotto | Sì | Sì | Sì | Sì | Sì |
| Classe di spedizione del prodotto | Sì | Sì | Sì | Sì | Sì |

| Entità | Può essere filtrato | Supporta limite | Supporta Ordina per | Supporta Seleziona * | Supporta partizionamento |
|---|---|---|---|---|---|
| Tag del prodotto | Sì | Sì | Sì | Sì | Sì |
| Variante del prodotto | Sì | Sì | Sì | Sì | Sì |
| Totale prodotti | No | No | No | Sì | No |
| Rapporto (elenco) | No | No | No | Sì | No |
| Totale recensioni | No | No | No | Sì | No |
| Rapporto di vendita | Sì | No | No | Sì | No |
| Metodi di spedizione | No | No | No | Sì | No |
| Zona di spedizione | No | No | No | Sì | No |
| Posizione della zona di spedizione | No | No | No | Sì | No |
| Metodo della zona di spedizione | No | No | No | Sì | No |
| Aliquota | Sì | Sì | Sì | Sì | Sì |
| Classe fiscale | No | No | No | Sì | No |

| Entità | Può essere filtrato | Supporta limite | Supporta Ordina per | Supporta Seleziona * | Supporta partizionamento |
|---|---|---|---|---|---|
| Rapporto sui più venduti | Sì | No | No | Sì | No |

Esempio:

```
woocommerce_read = glueContext.create_dynamic_frame.from_options(
    connection_type="glue.spark.woocommerce",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "coupon",
        "API_VERSION": "v3",
        "INSTANCE_URL": "instanceUrl"
    }
```

WooCommerce dettagli dell'entità e del campo:

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| coupon | id | Numero intero | N/D |
| | code | Stringa | EQUAL_TO |
| | quantità | Stringa | N/A |
| | status | Stringa | N/A |
| | date_created | DateTime | N/D |
| | date_created_gmt | DateTime | N/D |
| | date_modified | DateTime | N/D |
| | date_modified_gmt | DateTime | N/D |
| | discount_type | Stringa | N/A |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | description | Stringa | N/A |
| | date_expires | Stringa | N/A |
| | date_expires_gmt | Stringa | N/A |
| | usage_count | Numero intero | N/D |
| | individual_use | Booleano | N/D |
| | product_ids | List | N/D |
| | excluded_product_ids | List | N/D |
| | usage_limit | Numero intero | N/D |
| | usage_limit_per_user | Numero intero | N/D |
| | limit_usage_to_x_items | Numero intero | N/D |
| | free_shipping | Booleano | N/D |
| | product_categories | List | N/D |
| | excluded_product_categories | List | N/D |
| | exclude_sale_items | Booleano | N/D |
| | minimum_amount | Stringa | N/A |
| | maximum_amount | Stringa | N/A |
| | email_restrictions | List | N/D |
| | used_by | List | N/D |
| | meta_data | List | N/D |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | context | Stringa | EQUAL_TO |
| | cerca | Stringa | EQUAL_TO |
| | after | DateTime | EQUAL_TO |
| | before | DateTime | EQUAL_TO |
| | order | Stringa | EQUAL_TO |
| | orderby | Stringa | EQUAL_TO |
| | modified_after | DateTime | EQUAL_TO |
| | modified_before | DateTime | EQUAL_TO |
| | dates_are_gmt | Booleano | EQUAL_TO |
| coupon-total | slug | Stringa | N/A |
| | nome | Stringa | N/A |
| | total | Numero intero | N/D |
| customer-total | slug | Stringa | N/A |
| | nome | Stringa | N/A |
| | total | Numero intero | N/D |
| order | id | Numero intero | N/D |
| | parent_id | Numero intero | N/D |
| | numero | Stringa | N/A |
| | order_key | Stringa | N/A |
| | created_via | Stringa | N/A |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | status | Stringa | N/A |
| | currency | Stringa | N/A |
| | version | Stringa | N/A |
| | date_created | DateTime | N/D |
| | date_modified | DateTime | N/D |
| | discount_total | Stringa | N/A |
| | discount_tax | Stringa | N/A |
| | shipping_total | Stringa | N/A |
| | shipping_tax | Stringa | N/A |
| | cart_tax | Stringa | N/A |
| | total | Stringa | N/A |
| | total_tax | Stringa | N/A |
| | prices_include_tax | Booleano | N/D |
| | customer_id | Numero intero | N/D |
| | customer_ip_address | Stringa | N/A |
| | customer_user_agent | Stringa | N/A |
| | customer_note | Stringa | N/A |
| | fatturazione | Struct | N/D |
| | shipping | Struct | N/D |
| | payment_method | Stringa | N/A |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | payment_method_title | Stringa | N/A |
| | transaction_id | Stringa | N/A |
| | date_paid | DateTime | N/D |
| | date_completed | DateTime | N/D |
| | cart_hash | Stringa | N/A |
| | meta_data | List | N/D |
| | line_items | List | N/D |
| | tax_lines | List | N/D |
| | shipping_lines | List | N/D |
| | fee_lines | List | N/D |
| | coupon_lines | List | N/D |
| | refunds | List | N/D |
| | payment_url | Stringa | N/A |
| | is_editable | Booleano | N/D |
| | needs_payment | Booleano | N/D |
| | needs_processing | Booleano | N/D |
| | date_created_gmt | DateTime | N/D |
| | date_modified_gmt | DateTime | N/D |
| | date_completed_gmt | DateTime | N/D |
| | date_paid_gmt | DateTime | N/D |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | currency_symbol | Stringa | N/A |
| | set_paid | Booleano | N/D |
| | context | Stringa | EQUAL_TO |
| | cerca | Stringa | EQUAL_TO |
| | after | DateTime | EQUAL_TO |
| | before | DateTime | EQUAL_TO |
| | order | Stringa | EQUAL_TO |
| | orderby | Stringa | EQUAL_TO |
| | customer | Numero intero | EQUAL_TO |
| | product | Numero intero | EQUAL_TO |
| | dp | Numero intero | EQUAL_TO |
| | modified_before | DateTime | EQUAL_TO |
| | modified_after | DateTime | EQUAL_TO |
| | dates_are_gmt | Booleano | EQUAL_TO |
| order-total | slug | Stringa | N/A |
| | nome | Stringa | N/A |
| | total | Numero intero | N/D |
| payment-gateway | titolo | Stringa | N/A |
| | description | Stringa | N/A |
| | order | Stringa | N/A |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | enabled | Booleano | N/D |
| | method_title | Stringa | N/A |
| | method_description | Stringa | N/A |
| | method_supports | List | N/D |
| | impostazioni | Stringa | N/A |
| | needs_setup | Booleano | N/D |
| | post_install_scripts | List | N/D |
| | settings_url | Stringa | N/A |
| | connection_url | Stringa | N/A |
| | setup_help_text | Stringa | N/A |
| | required_settings_keys | List | N/D |
| product | id | Numero intero | N/D |
| | nome | Stringa | N/A |
| | tipo | Stringa | EQUAL_TO |
| | permalink | Stringa | N/A |
| | date_created | DateTime | N/D |
| | date_created_gmt | DateTime | N/D |
| | date_modified | DateTime | N/D |
| | date_modified_gmt | DateTime | N/D |
| | catalog_visibility | Stringa | N/A |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | description | Stringa | N/A |
| | short_description | Stringa | N/A |
| | price | Stringa | N/A |
| | regular_price | Stringa | N/A |
| | sale_price | Stringa | N/A |
| | date_on_sale_from | DateTime | N/D |
| | date_on_sale_from_gmt | DateTime | N/D |
| | date_on_sale_to | DateTime | N/D |
| | date_on_sale_to_gmt | DateTime | N/D |
| | price_html | Stringa | N/A |
| | purchasable | Booleano | N/D |
| | total_sales | Numero intero | N/D |
| | virtual | Booleano | N/D |
| | downloadable | Booleano | N/D |
| | downloads | List | N/D |
| | download_limit | Numero intero | N/D |
| | download_expiry | Numero intero | N/D |
| | external_url | Stringa | N/A |
| | button_text | Stringa | N/A |
| | tax_status | Stringa | N/A |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | manage_stock | Booleano | N/D |
| | stock_quantity | Numero intero | N/D |
| | backorders | Stringa | N/A |
| | backorders_allowed | Booleano | N/D |
| | backordered | Booleano | N/D |
| | sold_individually | Booleano | N/D |
| | weight | Stringa | N/A |
| | dimensioni | Struct | N/D |
| | shipping_required | Booleano | N/D |
| | shipping_taxable | Booleano | N/D |
| | shipping_class_id | Numero intero | N/D |
| | reviews_allowed | Booleano | N/D |
| | average_rating | Stringa | N/A |
| | rating_count | Numero intero | N/D |
| | related_ids | List | N/D |
| | upsell_ids | List | N/D |
| | cross_sell_ids | List | N/D |
| | parent_id | Numero intero | N/D |
| | purchase_note | Stringa | N/A |
| | categorie | List | N/D |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | tag | List | N/D |
| | images | List | N/D |
| | attributes | List | N/D |
| | default_attributes | List | N/D |
| | variations | List | N/D |
| | grouped_products | List | N/D |
| | menu_order | Numero intero | N/D |
| | meta_data | List | N/D |
| | low_stock_amount | Numero intero | N/D |
| | jetpack_publicize_connections | List | N/D |
| | jetpack-related-posts | List | N/D |
| | jetpack_likes_enabled | Booleano | N/D |
| | jetpack_sharing_enabled | Booleano | N/D |
| | context | Stringa | EQUAL_TO |
| | cerca | Stringa | EQUAL_TO |
| | after | DateTime | EQUAL_TO |
| | before | DateTime | EQUAL_TO |
| | order | Stringa | EQUAL_TO |
| | orderby | Stringa | EQUAL_TO |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | slug | Stringa | EQUAL_TO |
| | status | Stringa | EQUAL_TO |
| | sku | Stringa | EQUAL_TO |
| | featured | Booleano | EQUAL_TO |
| | tag | Stringa | EQUAL_TO |
| | shipping_class | Stringa | EQUAL_TO |
| | tax_class | Stringa | EQUAL_TO |
| | on_sale | Booleano | EQUAL_TO |
| | stock_status | Stringa | EQUAL_TO |
| | has_options | Booleano | N/D |
| | modified_after | DateTime | EQUAL_TO |
| | modified_before | DateTime | EQUAL_TO |
| | dates_are_gmt | Booleano | EQUAL_TO |
| | category | Stringa | EQUAL_TO |
| | attributo | Stringa | EQUAL_TO |
| | min_price | Stringa | EQUAL_TO |
| | max_price | Stringa | EQUAL_TO |
| product-attribute | id | Numero intero | N/D |
| | nome | Stringa | N/A |
| | slug | Stringa | N/A |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | tipo | Stringa | N/A |
| | order_by | Stringa | N/A |
| | has_archives | Booleano | N/D |
| | context | Stringa | EQUAL_TO |
| product-attribute-term | id | Numero intero | N/D |
| | nome | Stringa | N/A |
| | slug | Stringa | N/A |
| | description | Stringa | N/A |
| | menu_order | Numero intero | N/D |
| | count | Numero intero | N/D |
| | context | Stringa | EQUAL_TO |
| | cerca | Stringa | EQUAL_TO |
| | order | Stringa | EQUAL_TO |
| | orderby | Stringa | EQUAL_TO |
| | hide_empty | Booleano | EQUAL_TO |
| | parent | Numero intero | EQUAL_TO |
| | product | Numero intero | EQUAL_TO |
| product-category | id | Numero intero | N/D |
| | nome | Stringa | N/A |
| | slug | Stringa | EQUAL_TO |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | description | Stringa | N/A |
| | display | Stringa | N/A |
| | image | Struct | N/D |
| | menu_order | Numero intero | N/D |
| | count | Numero intero | N/D |
| | context | Stringa | EQUAL_TO |
| | cerca | Stringa | EQUAL_TO |
| | order | Stringa | EQUAL_TO |
| | orderby | Stringa | EQUAL_TO |
| | hide_empty | Booleano | EQUAL_TO |
| | parent | Numero intero | EQUAL_TO |
| | product | Numero intero | EQUAL_TO |
| product-review | id | Numero intero | N/D |
| | date_created | DateTime | N/D |
| | date_created_gmt | DateTime | N/D |
| | product_id | Numero intero | N/D |
| | product_name | Stringa | N/A |
| | product_permalink | Stringa | N/A |
| | review | Stringa | N/A |
| | rating | Numero intero | N/D |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | verified | Booleano | N/D |
| | reviewer | Stringa | N/A |
| | reviewer_email | Stringa | N/A |
| | reviewer_avatar_urls | Struct | N/D |
| | context | Stringa | EQUAL_TO |
| | cerca | Stringa | EQUAL_TO |
| | after | DateTime | EQUAL_TO |
| | before | DateTime | EQUAL_TO |
| | order | Stringa | EQUAL_TO |
| | orderby | Stringa | EQUAL_TO |
| | status | Stringa | EQUAL_TO |
| product-shipping-class | id | Numero intero | N/D |
| | nome | Stringa | N/A |
| | slug | Stringa | EQUAL_TO |
| | description | Stringa | N/A |
| | count | Numero intero | N/D |
| | context | Stringa | EQUAL_TO |
| | cerca | Stringa | EQUAL_TO |
| | order | Stringa | EQUAL_TO |
| | orderby | Stringa | EQUAL_TO |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | hide_empty | Stringa | EQUAL_TO |
| | product | Numero intero | EQUAL_TO |
| product-tag | id | Numero intero | N/D |
| | nome | Stringa | N/A |
| | slug | Stringa | EQUAL_TO |
| | description | Stringa | N/A |
| | count | Numero intero | N/D |
| | context | Stringa | EQUAL_TO |
| | cerca | Stringa | EQUAL_TO |
| | order | Stringa | EQUAL_TO |
| | orderby | Stringa | EQUAL_TO |
| | hide_empty | Booleano | EQUAL_TO |
| | product | Numero intero | EQUAL_TO |
| product-total | slug | Stringa | N/A |
| | nome | Stringa | N/A |
| | total | Numero intero | N/D |
| product-variation | id | Numero intero | N/D |
| | date_created | DateTime | N/D |
| | date_created_gmt | DateTime | N/D |
| | date_modified | DateTime | N/D |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | date_modified_gmt | DateTime | N/D |
| | description | Stringa | N/A |
| | permalink | Stringa | N/A |
| | price | Stringa | N/A |
| | regular_price | Stringa | N/A |
| | sale_price | Stringa | N/A |
| | date_on_sale_from | DateTime | N/D |
| | date_on_sale_from_gmt | DateTime | N/D |
| | date_on_sale_to | DateTime | N/D |
| | date_on_sale_to_gmt | DateTime | N/D |
| | purchasable | Booleano | N/D |
| | virtual | Booleano | N/D |
| | downloadable | Booleano | N/D |
| | downloads | List | N/D |
| | download_limit | Numero intero | N/D |
| | download_expiry | Numero intero | N/D |
| | tax_status | Stringa | N/A |
| | manage_stock | Booleano | N/D |
| | stock_quantity | Numero intero | N/D |
| | backorders | Stringa | N/A |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | backorders_allowed | Booleano | N/D |
| | backordered | Booleano | N/D |
| | low_stock_amount | Numero intero | N/D |
| | weight | Stringa | N/A |
| | dimensioni | Struct | N/D |
| | shipping_class | Stringa | N/A |
| | shipping_class_id | Numero intero | N/D |
| | image | Struct | N/D |
| | attributes | List | N/D |
| | menu_order | Numero intero | N/D |
| | meta_data | List | N/D |
| | context | Stringa | EQUAL_TO |
| | cerca | Stringa | EQUAL_TO |
| | after | DateTime | EQUAL_TO |
| | before | DateTime | EQUAL_TO |
| | order | Stringa | EQUAL_TO |
| | orderby | Stringa | EQUAL_TO |
| | slug | Stringa | EQUAL_TO |
| | status | Stringa | EQUAL_TO |
| | sku | Stringa | EQUAL_TO |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | tax_class | Stringa | EQUAL_TO |
| | on_sale | Booleano | EQUAL_TO |
| | min_price | Stringa | EQUAL_TO |
| | max_price | Stringa | EQUAL_TO |
| | stock_status | Stringa | EQUAL_TO |
| report | slug | Stringa | N/A |
| | description | Stringa | N/A |
| review-total | slug | Stringa | N/A |
| | nome | Stringa | N/A |
| | total | Numero intero | N/D |
| sales-report | total_sales | Stringa | N/A |
| | net_sales | Stringa | N/A |
| | average_sales | Stringa | N/A |
| | total_orders | Numero intero | N/D |
| | total_items | Numero intero | N/D |
| | total_tax | Stringa | N/A |
| | total_shipping | Stringa | N/A |
| | total_refunds | Numero intero | N/D |
| | total_discount | Stringa | N/A |
| | totals_grouped_by | Stringa | N/A |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | totals | Struct | N/D |
| | total_customers | Numero intero | N/D |
| | context | Stringa | EQUAL_TO |
| | punto | Stringa | EQUAL_TO |
| | date_min | Data | EQUAL_TO |
| | date_max | Data | EQUAL_TO |
| shipping-method | id | Stringa | N/A |
| | titolo | Stringa | N/A |
| | description | Stringa | N/A |
| shipping-zone | id | Numero intero | EQUAL_TO |
| | nome | Stringa | N/A |
| | order | Numero intero | N/D |
| shipping-zone-location | code | Stringa | N/A |
| | tipo | Stringa | N/A |
| shipping-zone-method | instance_id | Numero intero | N/D |
| | id | Numero intero | EQUAL_TO |
| | titolo | Stringa | N/A |
| | order | Numero intero | N/D |
| | enabled | Booleano | N/D |
| | method_id | Stringa | N/A |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | method_title | Stringa | N/A |
| | method_description | Stringa | N/A |
| | impostazioni | Struct | N/D |
| tax-class | slug | Stringa | N/A |
| | nome | Stringa | N/A |
| tax-rate | id | Numero intero | N/D |
| | paese | Stringa | N/A |
| | stato | Stringa | N/A |
| | postcode | Stringa | N/A |
| | città | Stringa | N/A |
| | postcodes | List | N/D |
| | cities | List | N/D |
| | rate | Stringa | N/A |
| | nome | Stringa | N/A |
| | priority | Numero intero | N/D |
| | compound | Booleano | N/D |
| | shipping | Booleano | N/D |
| | context | Stringa | EQUAL_TO |
| | order | Stringa | EQUAL_TO |
| | orderby | Stringa | EQUAL_TO |

| Entità | Campo | Tipo di dati | Operatori supportati |
|---|---|---|---|
| | classe | Stringa | EQUAL_TO |
| top-seller-report | nome | Stringa | N/A |
| | product_id | Numero intero | N/D |
| | quantity | Numero intero | N/D |
| | context | Stringa | EQUAL_TO |
| | punto | Stringa | EQUAL_TO |
| | date_min | Data | EQUAL_TO |
| | date_max | Data | EQUAL_TO |

> **Note**
>
> I tipi di dati Struct e List vengono convertiti in tipo di dati String e il tipo di DateTime dati viene convertito in Timestamp nella risposta dei connettori.

Query di partizionamento

Partizionamento basato su record:

Puoi fornire l'opzione Spark aggiuntiva `NUM_PARTITIONS` se desideri utilizzare la concorrenza in Spark. Con questi parametri, la query originale viene suddivisa in `NUM_PARTITIONS` query secondarie che possono essere eseguite contemporaneamente dalle attività Spark.

Nel partizionamento basato sui record, il numero totale di record presenti viene interrogato dall' WooCommerce API e diviso per un numero fornito. `NUM_PARTITIONS` Il numero di record risultante viene quindi recuperato contemporaneamente da ciascuna query secondaria.

- `NUM_PARTITIONS`: il numero di partizioni.

Le seguenti entità supportano il partizionamento basato sui record:

- coupon
- order
- product
- product-attribute
- product-attribute-term
- product-category
- product-review
- product-shipping-class
- product-tag
- product-variation
- tax-rate

Esempio:

```
woocommerce_read = glueContext.create_dynamic_frame.from_options(
    connection_type="glue.spark.woocommerce",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "coupon",
        "API_VERSION": "v3",
        "INSTANCE_URL": "instanceUrl"
        "NUM_PARTITIONS": "10"
    }
```

Partizionamento basato su record:

La query originale viene suddivisa in `NUM_PARTITIONS` query secondarie che possono essere eseguite contemporaneamente dalle attività Spark.

- `NUM_PARTITIONS`: il numero di partizioni.

Esempio:

```
WooCommerce_read = glueContext.create_dynamic_frame.from_options(
    connection_type="WooCommerce",
```

```
    connection_options={
        "connectionName": "connectionName",
        "REALMID": "1234567890123456789",
        "ENTITY_NAME": "Bill",
        "API_VERSION": "v3",
        "NUM_PARTITIONS": "10"
    }
```

### WooCommerce opzioni di connessione

Di seguito sono elencate le opzioni di connessione per WooCommerce:

- `ENTITY_NAME`(String): (obbligatorio) utilizzato per la lettura. Il nome dell'oggetto in WooCommerce.
- `API_VERSION`(String) - (Obbligatorio) Usato per la lettura. WooCommerce Versione dell'API Rest che desideri utilizzare.
- `REALM_ID`(Stringa): un ID che identifica una singola società WooCommerce online a cui si inviano richieste.
- `SELECTED_FIELDS`(List<String>): predefinito: vuoto (SELECT *). Utilizzato per la lettura. Colonne da selezionare per l'oggetto.
- `FILTER_PREDICATE`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Dev'essere in formato Spark SQL.
- `QUERY`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Query Spark SQL completa.
- `INSTANCE_URL`<instance>(String) - (Obbligatorio) Un URL di WooCommerce istanza valido con il formato: https://.wpcomstaging.com
- `NUM_PARTITIONS`(Numero intero): predefinito: 1. Utilizzato per la lettura. Numero di partizioni da leggere.

## Connessione a Zendesk

Zendesk è una soluzione di gestione dell'help desk basata su cloud che offre strumenti personalizzabili per creare un portale di assistenza clienti, una knowledge base e una comunità online.

Argomenti

- AWS Glue supporto per Zendesk

- Policy contenenti le operazioni API per creare e utilizzare le connessioni
- Configurazione Zendesk
- Configurazione delle connessioni di Zendesk
- Lettura da entità Zendesk
- Opzioni di connessione Zendesk
- Limitazioni


## AWS Glue supporto per Zendesk

AWS Glue supporta Zendesk come segue:

Supportato come origine?

Sì. È possibile utilizzare i job AWS Glue ETL per interrogare i dati da Zendesk.

Supportato come destinazione?

No.

Versioni API Zendesk supportate

Sono supportate le seguenti versioni dell'API Zendesk

- v2

## Policy contenenti le operazioni API per creare e utilizzare le connessioni

La seguente policy di esempio descrive le autorizzazioni AWS IAM richieste per la creazione e l'utilizzo delle connessioni. Se si sta creando un nuovo ruolo, creare una policy che contenga quanto segue:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
```

```
      "Action": [
        "glue:ListConnectionTypes",
        "glue:DescribeConnectionType",
        "glue:RefreshOAuth2Tokens",
        "glue:ListEntities",
        "glue:DescribeEntity"
      ],
      "Resource": "*"
    }
  ]
}
```

In alternativa, se non si desidera utilizzare il metodo precedente, usare le seguenti policy IAM gestite:

- AWSGlueServiceRole— Garantisce l'accesso alle risorse che AWS Glue i vari processi richiedono per l'esecuzione per conto dell'utente. Queste risorse includono AWS Glue Amazon S3, IAM, CloudWatch Logs e Amazon. EC2 Se segui la convenzione di denominazione delle risorse specificata in questa politica, AWS Glue i processi dispongono delle autorizzazioni richieste. Questa policy è in genere collegata ai ruoli specificati quando si definiscono crawler, processi ed endpoint di sviluppo.
- AWSGlueConsoleFullAccess— Concede l'accesso completo alle AWS Glue risorse quando un'identità a cui è allegata la policy utilizza la AWS console di gestione. Se si segue la convenzione per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questa policy è in genere associata agli utenti della AWS Glue console.

## Configurazione Zendesk

Prima di poter AWS Glue utilizzare il trasferimento di dati da Zendesk, è necessario soddisfare i seguenti requisiti:

Requisiti minimi

Di seguito sono riportati i requisiti minimi:

- Possedere un account Zendesk. Per ulteriori informazioni, consulta Creazione di un account Zendesk.
- L'account Zendesk è abilitato all'accesso alle API.
- L'account Zendesk consente di installare app connesse.

Se soddisfi questi requisiti, sei pronto per connetterti AWS Glue al tuo account Zendesk.

Creazione di un account Zendesk

Per creare un account Zendesk:

1. Vai alla registrazione/ https://www.zendesk.com/in/
2. Inserire i dettagli come l'email di lavoro, il nome, il cognome, il numero di telefono, la qualifica, il nome dell'azienda, il numero di dipendenti dell'azienda, la password e la lingua preferita. Quindi scegliere Completa la registrazione alla versione di prova.
3. Una volta creato l'account, completare il link di verifica che è stato ricevuto per verificare il proprio indirizzo email.
4. Una volta verificato l'indirizzo email di lavoro, si verrà reindirizzati al proprio account Zendesk. Scegliere l'opzione Acquista Zendesk per il proprio piano preferito. Nota: per il connettore Zendesk si consiglia di acquistare il piano Suite Enterprise.

Creazione di un'app client e OAuth credenziali 2.0

Per creare un'app client e credenziali OAuth 2.0:

1. Accedi al tuo account Zendesk dove desideri che venga creata l'app OAuth 2.0 login/ https://www.zendesk.com/in/
2. Fare clic sull'icona Salva. Scegliere il link Vai al centro di amministrazione per aprire la pagina dell'interfaccia di amministrazione.
3. Scegli App e integrazioni nella barra laterale sinistra, quindi seleziona > API Zendesk. APIs
4. Nella pagina API Zendesk, scegli la scheda Client. OAuth
5. Scegliere Aggiungi client Oauth sul lato destro.
6. Completare i seguenti campi per creare un cliente:
   a. Nome client - Inserire un nome per l'app. Questo è il nome che gli utenti vedranno quando verrà chiesto loro di concedere l'accesso all'applicazione e quando controlleranno l'elenco delle app di terze parti che hanno accesso al loro Zendesk.
   b. Descrizione - Facoltativa. Una breve descrizione dell'app che gli utenti vedranno quando verrà chiesto loro di concedere l'accesso.
   c. Azienda - Facoltativo. Il nome dell'azienda che gli utenti vedranno quando verrà chiesto loro di concedere l'accesso all'applicazione. Le informazioni possono aiutarli a capire a chi concedono l'accesso.

d. Logo - Facoltativo. Questo è il logo che gli utenti vedranno quando verrà chiesto loro di concedere l'accesso all'applicazione. L'immagine può essere JPG, GIF o PNG. Per ottimizzare i risultati, caricare un'immagine quadrata. Verrà ridimensionata per la pagina di autorizzazione.

e. Identificatore univoco - Il campo viene compilato automaticamente con una versione riformattata del nome che è stato inserito per l'app. È possibile modificarlo, se necessario.

f. Reindirizzamento URLs : inserisci l'URL o l'URL URLs che Zendesk deve utilizzare per inviare la decisione dell'utente di concedere l'accesso all'applicazione.

   Ad esempio: oauth https://us-east-1.console.aws.amazon.com/gluestudio/

7. Fare clic su Salva.
8. Dopo l'aggiornamento della pagina, nella parte inferiore viene visualizzato un nuovo campo Segreto precompilato. Questo è il valore «client_secret» specificato nella specifica. OAuth2 Copiare il valore Segreto negli appunti e salvarlo in un posto sicuro. Nota: I caratteri possono estendersi oltre la larghezza della casella di testo, quindi assicurarsi di selezionare tutto prima di copiarli.
9. Fare clic su Salva.

## Configurazione delle connessioni di Zendesk

Il connettore Zendesk supporta il tipo di concessione Authorization Code.

- Questo tipo di concessione è considerato «a tre gambe» in OAuth quanto si basa sul reindirizzamento degli utenti a un server di autorizzazione di terze parti per autenticare l'utente. Viene utilizzato durante la creazione di connessioni tramite la console. AWS Glue Per impostazione predefinita, l'utente che crea una connessione può affidarsi a un'app connessa di AWS Glue proprietà (AWS Glue-applicazione client gestita) in cui non deve fornire alcuna informazione OAuth correlata ad eccezione dell'URL dell'istanza Zendesk. La AWS Glue console reindirizzerà l'utente a Zendesk, dove l'utente deve effettuare il login e concedere le autorizzazioni richieste per accedere AWS Glue alla propria istanza Zendesk.
- Puoi comunque scegliere di creare la tua app connessa in Zendesk e fornire il tuo ID cliente e il segreto del client quando crei connessioni tramite la console. AWS Glue In questo scenario, verrai comunque reindirizzato a Zendesk per effettuare il login e autorizzare l'accesso AWS Glue alle tue risorse.
- Questo tipo di concessione genera un token di accesso. Il token di accesso non ha scadenza.

Per la documentazione pubblica di Zendesk sulla creazione di un'app connessa per il OAuth flusso del codice di autorizzazione, consulta OAuth Tokens for Grant Types.

Per configurare una connessione Zendesk:

1. In AWS Secrets Manager, crea un segreto con i seguenti dettagli:
   a. Per il tipo di AuthorizationCode concessione: per un'app connessa gestita dal cliente, il segreto deve contenere l'app connessa Consumer Secret con `USER_MANAGED_CLIENT_APPLICATION_CLIENT_SECRET` come chiave.
   b. Nota: è necessario creare un segreto per la connessione in AWS Glue.
2. In AWS Glue Glue Studio, crea una connessione in Connessioni dati seguendo i passaggi seguenti:
   a. In Tipo di connessione, selezionare Zendesk.
   b. Fornire INSTANCE_URL dell'istanza Zendesk a cui connettersi.
   c. Fornire l'ambiente Zendesk.
   d. Seleziona il ruolo AWS IAM che AWS Glue può assumere e dispone delle autorizzazioni per le seguenti azioni:

      JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:DescribeSecret",
        "secretsmanager:GetSecretValue",
        "secretsmanager:PutSecretValue",
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DeleteNetworkInterface"
      ],
      "Resource": "*"
    }
  ]
}
```

   e. Seleziona quello `secretName` che desideri utilizzare per questa connessione per AWS Glue inserire i token.

f. Selezionare le opzioni di rete se si desidera utilizzare la propria rete.

3. Concedi il permesso di lettura `secretName` al ruolo IAM associato al tuo AWS Glue lavoro.

## Lettura da entità Zendesk

### Prerequisito

Un oggetto Zendesk da cui si desidera leggere. Sarà necessario il nome dell'oggetto, ad esempio ticket, utente o articolo, come indicato nella tabella seguente.

| Entità | Può essere filtrato | Supporta limite | Supporta Ordina per | Supporta Seleziona * | Supporta il partizionamento |
|---|---|---|---|---|---|
| Ticket | Y | Y | Y | Y | N |
| Utente | Y | Y | Y | Y | N |
| Organizzazione | Y | Y | Y | Y | N |
| Articolo | Y | Y | N | Y | N |
| Evento Ticket | Y | Y | N | Y | N |
| Evento Parametri Ticket | Y | Y | N | Y | N |
| Commenti sui Ticket | Y | Y | Y | Y | N |
| Campo Ticket | Y | Y | N | Y | N |
| Parametri Ticket | Y | Y | N | Y | N |
| Attività Ticket | Y | Y | N | Y | N |

| Entità | Può essere filtrato | Supporta limite | Supporta Ordina per | Supporta Seleziona * | Supporta il partizionamento |
|---|---|---|---|---|---|
| Ignora Ticket | N | Y | N | Y | N |
| Gruppo | Y | Y | Y | Y | N |
| Appartenenza a un gruppo | N | Y | Y | Y | N |
| Indice di soddisfazione | Y | Y | N | Y | N |
| Vista | Y | Y | Y | Y | N |
| Trigger | Y | Y | Y | Y | N |
| Categoria Trigger | N | Y | Y | Y | N |
| Macro | Y | Y | Y | Y | N |
| Automazione | N | Y | Y | Y | N |

Esempio:

```
Zendesk_read = glueContext.create_dynamic_frame.from_options(
    connection_type="Zendesk",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "Account",
        "API_VERSION": "v2"
    }
```

Entità Zendesk e dettagli sui campi:

| Entità | Campo | Tipo di dati | Operatori supportati | Commenti |
|---|---|---|---|---|
| articoli | url | Stringa | | |
| | id | Long | | |
| | author_id | Long | | |
| | body | Stringa | | |
| | comments_ disabled | Booleano | | |
| | bozza | Booleano | | |
| | edited_at | DateTime | | |
| | html_url | Stringa | | |
| | label_names | List | | |
| | locale | Stringa | EQUAL_TO | |
| | scaduto | Booleano | | |
| | outdated_locales | List | | |
| | permissio n_group_id | Long | | |
| | posizione | Numero intero | | |
| | promosso | Booleano | | |
| | section_id | Long | | |
| | source_locale | Stringa | | |
| | nome | Stringa | | |

| Entità | Campo | Tipo di dati | Operatori supportati | Commenti |
|---|---|---|---|---|
| | titolo | Stringa | | |
| | user_segment_id | Long | | |
| | content_tags_id | List | | |
| | vote_count | Numero intero | | |
| | vote_sum | Numero intero | | |
| | created_at | DateTime | | |
| | updated_at | DateTime | EQUAL_TO | |
| | label_name | Stringa | EQUAL_TO | |
| gruppo | url | Stringa | | |
| | id | Long | | |
| | is_public | Booleano | | |
| | nome | Stringa | | |
| | description | Stringa | | |
| | default | Booleano | | |
| | deleted (eliminato) | Booleano | | |
| | created_at | DateTime | | |
| | updated_at | DateTime | | |
| | exclude_deleted | Booleano | EQUAL_TO | |
| automazione | url | Stringa | | |

| Entità | Campo | Tipo di dati | Operatori supportati | Commenti |
|---|---|---|---|---|
| | id | Long | | |
| | titolo | Stringa | | |
| | attiva | Booleano | | |
| | created_at | DateTime | | |
| | updated_at | DateTime | | |
| | default | Booleano | | |
| | actions | List | | |
| | posizioni | Numero intero | | |
| | condizioni | Struct | | |
| | raw_title | Stringa | | |
| group-membership | url | Stringa | | |
| | id | Long | | |
| | user_id | Long | | |
| | group_id | Long | | |
| | default | Booleano | | |
| | created_at | DateTime | | |
| | updated_at | DateTime | | |
| macro | url | Stringa | | |
| | id | Long | | |
| | titolo | Stringa | | |

| Entità | Campo | Tipo di dati | Operatori supportati | Commenti |
|---|---|---|---|---|
| | attiva | Booleano | EQUAL_TO | |
| | created_at | DateTime | | |
| | updated_at | DateTime | | |
| | default | Booleano | | |
| | actions | List | | |
| | posizione | Numero intero | | |
| | description | Stringa | | |
| | raw_title | Stringa | | |
| | limitazione | Struct | | |
| | accedi | Stringa | EQUAL_TO | |
| | category | Numero intero | EQUAL_TO | |
| | group_id | Long | EQUAL_TO | |
| | only_viewable | Booleano | EQUAL_TO | |
| organizations | url | Stringa | | |
| | id | Long | | |
| | external_id | Stringa | | |
| | nome | Stringa | | |
| | domain_names | List | | |
| | details | Stringa | | |
| | notes | Stringa | | |

| Entità | Campo | Tipo di dati | Operatori supportati | Commenti |
|---|---|---|---|---|
| | group_id | Long | | |
| | shared_tickets | Booleano | | |
| | shared_comments | Booleano | | |
| | tag | List | | |
| | organization_fields | Struct | | |
| | created_at | DateTime | | |
| | updated_at | DateTime | EQUAL_TO | |
| | DML_STATUS | Stringa | | Un campo definito dall'utente utilizzato per tenere traccia dello stato creato, aggiornato ed eliminato del record. |
| satisfaction-rating | url | Stringa | | |
| | id | Long | | |
| | assignee_id | Long | | |
| | comment | Stringa | | |
| | group_id | Long | | |
| | motivo | Stringa | | |

| Entità | Campo | Tipo di dati | Operatori supportati | Commenti |
|---|---|---|---|---|
| | reason_code | Numero intero | | |
| | reason_id | Long | | |
| | requester_id | Long | | |
| | score | Stringa | EQUAL_TO | |
| | ticket_id | Numero intero | | |
| | created_at | DateTime | | |
| | updated_at | DateTime | EQUAL_TO | |
| | start_time | DateTime | EQUAL_TO | |
| | end_time | DateTime | EQUAL_TO | |
| | DML_STATUS | Stringa | | Un campo definito dall'utente utilizzato per tenere traccia dello stato creato, aggiornato ed eliminato del record. |
| ticket-activity | attore | Struct | | |
| | actor_id | Long | | |
| | created_at | DateTime | | |
| | id | Long | | |
| | oggetto | Struct | | |
| | target | Struct | | |

| Entità | Campo | Tipo di dati | Operatori supportati | Commenti |
|---|---|---|---|---|
| | titolo | Stringa | | |
| | updated_at | DateTime | | |
| | url | Stringa | | |
| | user | Struct | | |
| | user_id | Long | | |
| | verb | Stringa | | |
| | since | DateTime | EQUAL_TO | |
| ticket-comment | id | Long | | |
| | tipo | Stringa | | |
| | author_id | Long | | |
| | body | Stringa | | |
| | html_body | Stringa | | |
| | plain_body | Stringa | | |
| | pubblico | Booleano | | |
| | allegati | List | | |
| | audit_id | Long | | |
| | via | Struct | | |
| | created_at | DateTime | | |
| | metadati | Struct | | |
| | ticket_id | Numero intero | EQUAL_TO | |

| Entità | Campo | Tipo di dati | Operatori supportati | Commenti |
|---|---|---|---|---|
| | include_inline_images | Booleano | EQUAL_TO | |
| ticket-events | id | Long | | |
| | ticket_id | Long | | |
| | timestamp | Long | | |
| | created_at | DateTime | | |
| | updater_id | Long | | |
| | child_events | List | | |
| | via | Stringa | | |
| | sistema | Struct | | |
| | event_type | Stringa | | |
| | comment_present | Booleano | | |
| | comment_public | Booleano | | |
| | via_reference_id | Long | | |
| | created_at | DateTime | EQUAL_TO | |

| Entità | Campo | Tipo di dati | Operatori supportati | Commenti |
|---|---|---|---|---|
| | DML_STATUS | Stringa | | Un campo definito dall'utente utilizzato per tenere traccia dello stato creato, aggiornato ed eliminato del record. |
| ticket-field | url | Stringa | | |
| | id | Long | | |
| | tipo | Stringa | | |
| | titolo | Stringa | | |
| | raw_title | Stringa | | |
| | description | Stringa | | |
| | raw_description | Stringa | | |
| | posizione | Numero intero | | |
| | attiva | Booleano | | |
| | obbligatorio | Booleano | | |
| | collapsed_for_agents | Booleano | | |
| | regexp_for_validation | Stringa | | |
| | title_in_portal | Stringa | | |

| Entità | Campo | Tipo di dati | Operatori supportati | Commenti |
|---|---|---|---|---|
| | raw_title _in_portal | Stringa | | |
| | visible_in_portal | Booleano | | |
| | editable_ on_portal | Booleano | | |
| | required_ in_portal | Booleano | | |
| | tag | Stringa | | |
| | created_at | DateTime | | |
| | updated_at | DateTime | | |
| | removibile | Booleano | | |
| | agent_des cription | Stringa | | |
| | custom_fi eld_options | List | | |
| | custom_statuses | List | | |
| | relationship_filter | Struct | | |
| | relations hip_target_type | Stringa | | |
| | sub_type_id | Numero intero | | |
| | system_fi eld_options | List | | |
| | locale | Stringa | EQUAL_TO | |

| Entità | Campo | Tipo di dati | Operatori supportati | Commenti |
|---|---|---|---|---|
| ticket-metric-events | id | Long | | |
| | time | DateTime | EQUAL_TO | |
| | ticket_id | Numero intero | | |
| | parametro | Stringa | | |
| | instance_id | Numero intero | | |
| | tipo | Stringa | | |
| | DML_STATUS | Stringa | EQUAL_TO | Un campo definito dall'utente utilizzato per tenere traccia dello stato creato, aggiornato ed eliminato del record. |
| ticket-metric | url | Stringa | | |
| | id | Long | | |
| | ticket_id | Numero intero | | |
| | created_at | DateTime | | |
| | updated_at | DateTime | | |
| | group_stations | Numero intero | | |
| | assignee_stations | Numero intero | | |
| | riaperture | Numero intero | | |

| Entità | Campo | Tipo di dati | Operatori supportati | Commenti |
|---|---|---|---|---|
| | risposte | Numero intero | | |
| | assignee_updated_at | DateTime | | |
| | requester_updated_at | DateTime | | |
| | initially_assigned_at | DateTime | | |
| | assigned_at | DateTime | | |
| | solved_at | DateTime | | |
| | last_comment_added_at | DateTime | | |
| | reply_time_in_minutes | Struct | | |
| | first_resolution_time_in_minutes | Struct | | |
| | full_resolution_time_in_minutes | Struct | | |
| | agent_wait_time_in_minutes | Struct | | |
| | requester_wait_time_in_minutes | Struct | | |
| | on_hold_time_in_seconds | Struct | | |

| Entità | Campo | Tipo di dati | Operatori supportati | Commenti |
|---|---|---|---|---|
| | reply_time_in_seconds | Struct | | |
| | custom_status_updated_at | DateTime | | |
| ticket-skip | created_at | DateTime | | |
| | id | Long | | |
| | motivo | Stringa | | |
| | ticket | Struct | | |
| | ticket_id | Numero intero | | |
| | updated_at | DateTime | | |
| | user_id | Long | | |
| biglietti | url | Stringa | | |
| | id | Long | | |
| | external_id | Stringa | EQUAL_TO | |
| | tipo | Stringa | | |
| | subject | Stringa | | |
| | raw_subject | Stringa | | |
| | description | Stringa | | |
| | priority | Stringa | | |
| | status | Stringa | | |
| | recipient | Stringa | | |

| Entità | Campo | Tipo di dati | Operatori supportati | Commenti |
|---|---|---|---|---|
| | richiedente | Struct | | |
| | requester_id | Long | | |
| | submitter_id | Long | | |
| | assignee_id | Long | | |
| | organization_id | Long | | |
| | group_id | Long | | |
| | collaborator_ids | List | | |
| | emails_cc_ids | List | | |
| | follower_ids | List | | |
| | forum_topic_id | Ling | | |
| | problem_id | Long | | |
| | has_incidents | Booleano | | |
| | due_at | DateTime | | |
| | tag | List | | |
| | via | Struct | | |
| | custom_fields | List | | |
| | satisfaction_rating | Struct | | |
| | sharing_agreement_ids | List | | |
| | followup_ids | List | | |

| Entità | Campo | Tipo di dati | Operatori supportati | Commenti |
|---|---|---|---|---|
| | via_followup_source_id | Long | | |
| | ticket_form_id | Long | | |
| | brand_id | Long | | |
| | allow_channelback | Booleano | | |
| | allow_attachments | Booleano | | |
| | is_public | Booleano | | |
| | from_messaging_channel | Booleano | | |
| | created_at | DateTime | | |
| | updated_at | DateTime | EQUAL_TO | |
| | assignee_email | Stringa | | |
| | attribute_value_ids | List | | |
| | collaboratori | List | | |
| | comment | Struct | | |
| | custom_status_id | Long | | |
| | email_ccs | Struct | | |
| | follower | Struct | | |
| | macro_id | Long | | |

| Entità | Campo | Tipo di dati | Operatori supportati | Commenti |
|---|---|---|---|---|
| | macros_ids | List | | |
| | metadati | Struct | | |
| | safe_update | Booleano | | |
| | updated_stamp | DateTime | | |
| | via_id | Long | | |
| | voice_comment | Struct | | |
| | DML_STATUS | Stringa | | Un campo definito dall'utente utilizzato per tenere traccia dello stato creato, aggiornato ed eliminato del record. |
| trigger-category | url | Stringa | | |
| | id | Stringa | | |
| | nome | Stringa | | |
| | updated_at | DateTime | | |
| | created_at | DateTime | | |
| | posizione | Numero intero | | |
| Trigger | url | Stringa | | |
| | id | Long | | |
| | titolo | Stringa | | |

| Entità | Campo | Tipo di dati | Operatori supportati | Commenti |
|---|---|---|---|---|
| | attiva | Booleano | EQUAL_TO | |
| | updated_at | DateTime | | |
| | created_at | DateTime | | |
| | default | Booleano | | |
| | actions | List | | |
| | condizioni | Struct | | |
| | description | Stringa | | |
| | posizione | Numero intero | | |
| | raw_title | Stringa | | |
| | category_id | Stringa | EQUAL_TO | |
| utenti | url | Stringa | | |
| | id | Long | | |
| | external_id | Stringa | EQUAL_TO | |
| | e-mail | Stringa | | |
| | attiva | Booleano | | |
| | alias | Stringa | | |
| | chat_only | Booleano | | |
| | custom_roll_id | Long | | |
| | roll_type | Numero intero | | |
| | details | Stringa | | |

| Entità | Campo | Tipo di dati | Operatori supportati | Commenti |
|---|---|---|---|---|
| | last_login_at | DateTime | | |
| | locale | Stringa | | |
| | locale_id | Numero intero | | |
| | moderator | Booleano | | |
| | notes | Stringa | | |
| | nome | Stringa | | |
| | only_private_comments | Booleano | | |
| | organization_id | Long | | |
| | default_group_id | Long | | |
| | telefono | Stringa | | |
| | photo | Struct | | |
| | remote_photo_url | Stringa | | |
| | restricted_agent | Booleano | | |
| | role | Stringa | EQUAL_TO | |
| | shared | Booleano | | |
| | shared_agent | Booleano | | |
| | tag | List | | |
| | signature | Stringa | | |
| | suspended | Booleano | | |

| Entità | Campo | Tipo di dati | Operatori supportati | Commenti |
|---|---|---|---|---|
| | ticket_restriction | Stringa | | |
| | time_zone | Stringa | | |
| | iana_time_zone | | | |
| | two_factor_auth_enabled | | | |
| | user_fields | | | |
| | verified | Booleano | | |
| | report_csv | Booleano | | |
| | created_at | DateTime | | |
| | updated_at | DateTime | EQUAL_TO | |
| | permission_set | Long | EQUAL_TO | |
| | shared_phone_number | Booleano | | |
| | DML_STATUS | Stringa | | Un campo definito dall'utente utilizzato per tenere traccia dello stato creato, aggiornato ed eliminato del record. |
| visualizzazione | url | Stringa | | |
| | id | Long | | |

| Entità | Campo | Tipo di dati | Operatori supportati | Commenti |
|---|---|---|---|---|
| | titolo | Stringa | | |
| | attiva | Booleano | EQUAL_TO | |
| | updated_at | DateTime | | |
| | created_at | DateTime | | |
| | default | Booleano | | |
| | posizione | Numero intero | | |
| | description | Stringa | | |
| | execution | Struct | | |
| | limitazione | Struct | | |
| | raw_title | Stringa | | |
| | condizioni | Struct | | |
| | accedi | Stringa | EQUAL_TO | |
| | group_id | Long | EQUAL_TO | |

> **Note**
>
> I tipi di dati Struct e List vengono convertiti in tipo di dati Stringa nella risposta del connettore.

Query di partizionamento

Le partizioni non sono supportate in Zendesk.

## Opzioni di connessione Zendesk

Le seguenti sono le opzioni di connessione per Zendesk:

- ENTITY_NAME(String): (obbligatorio) utilizzato per la lettura. Il nome dell'oggetto in Zendesk.
- API_VERSION(String): (obbligatorio) utilizzato per la lettura. Versione di Rest API di Zendesk da utilizzare. Ad esempio: v2.
- SELECTED_FIELDS(List<String>): predefinito: vuoto (SELECT *). Utilizzato per la lettura. Colonne che si desidera selezionare per l'oggetto. Ad esempio: id, nome, url, created_at
- FILTER_PREDICATE(Stringa): predefinito: vuoto. Utilizzato per la lettura. Deve essere in formato Spark SQL. Ad esempio: group_id = 100
- QUERY(Stringa): predefinito: vuoto. Utilizzato per la lettura. Query SQL Spark completa. Ad esempio: "SELECT id,url FROM users WHERE role=\"end-user\""
- PARTITION_FIELD(Stringa): utilizzato per la lettura. Campo da utilizzare per partizionare la query. Il campo predefinito è `update_at` per le entità che supportano l'API di esportazione incrementale (`created_at` per `ticket-events` e `time` for `ticket-metric-events`).
- LOWER_BOUND(Stringa): utilizzato per la lettura. Un valore limite inferiore inclusivo del campo di partizione scelto.
- UPPER_BOUND(Stringa): utilizzato per la lettura. Un valore limite superiore esclusivo del campo di partizione scelto. Facoltativo; questa opzione verrà gestita dal connettore se non fornita nell'opzione di lavoro. Valore predefinito: "2024-05-01T20:55:02.000Z
- NUM_PARTITIONS(Intero): predefinito: 1. Utilizzato per la lettura. Numero di partizioni da leggere. Facoltativo; questa opzione verrà gestita dal connettore se non è fornita nell'opzione job. Valore predefinito: 1.
- IMPORT_DELETED_RECORDS(Stringa): predefinito: FALSE. Utilizzato per la lettura. Per ottenere i record eliminati durante l'interrogazione.
- ACCESS_TOKEN - Token di accesso da utilizzare nella richiesta.
- INSTANCE_URL - URL dell'istanza in cui l'utente desidera eseguire le operazioni. Ad esempio: https://{subdomain}/.zendesk.com

## Limitazioni

Di seguito sono riportate le limitazioni del connettore Zendesk:

- L'impaginazione basata su offset limita a 100 il numero di pagine che possono essere recuperate, ma non è consigliata in quanto il numero totale di record che è possibile recuperare è 10.000. Tuttavia, l'impaginazione basata sul cursore implementata per il connettore Zendesk supera questa limitazione. Solo l'operatore di filtro EQUAL_TO è supportato dall'API Zendesk.

A causa di questa limitazione, per il connettore Zendesk non è supportato il partizionamento.

- Per l'entità "Ticket Event" il limite di frequenza è di 10 richieste al minuto. Durante l'esecuzione di un processo AWS Glue ETL è possibile che venga visualizzato un errore 429 (troppe richieste).

# Connessione a Zoho CRM

Zoho CRM funge da archivio unico per riunire le attività di vendita, marketing e assistenza clienti e per ottimizzare processi, policy e risorse umane in un'unica piattaforma. Zoho CRM può essere facilmente personalizzato per soddisfare le specifiche esigenze delle aziende di qualsiasi tipo e dimensione.

La piattaforma per sviluppatori di Zoho CRM offre il giusto mix di strumenti low-code e pro-code per businesses/enterprises automatizzare il lavoro, integrare i dati nello stack aziendale e creare soluzioni personalizzate per web e dispositivi mobili.

Argomenti


- AWS Glue supporto per Zoho CRM
- Policy contenenti le operazioni API per creare e utilizzare le connessioni
- Configurazione di Zoho CRM
- Configurazione delle connessioni Zoho CRM
- Lettura dalle entità Zoho CRM
- Opzioni di connessione Zoho CRM
- Limitazioni e note per il connettore Zoho CRM


## AWS Glue supporto per Zoho CRM

AWS Glue supporta Zoho CRM come segue:

Supportato come origine?

Sì, sincrona e asincrona. Puoi utilizzare i job AWS Glue ETL per interrogare i dati da Zoho CRM.

Supportato come destinazione?

No.

Versioni dell'API Zoho CRM supportate

Sono supportate le seguenti versioni delle API di Zoho CRM:

- v7

## Policy contenenti le operazioni API per creare e utilizzare le connessioni

La seguente policy di esempio descrive le autorizzazioni AWS IAM richieste per la creazione e l'utilizzo delle connessioni. Se si sta creando un nuovo ruolo, creare una policy che contenga quanto segue:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:ListConnectionTypes",
        "glue:DescribeConnectionType",
        "glue:RefreshOAuth2Tokens",
        "glue:ListEntities",
        "glue:DescribeEntity"
      ],
      "Resource": "*"
    }
  ]
}
```

In alternativa, se non desideri utilizzare il metodo precedente, utilizza le seguenti policy IAM gestite:

- AWSGlueServiceRole— Garantisce l'accesso alle risorse che AWS Glue i vari processi richiedono per l'esecuzione per conto dell'utente. Queste risorse includono AWS Glue Amazon S3, IAM, CloudWatch Logs e Amazon. EC2 Se segui la convenzione di denominazione delle risorse specificata in questa politica, AWS Glue i processi dispongono delle autorizzazioni richieste. Questa policy è in genere collegata ai ruoli specificati quando si definiscono crawler, processi ed endpoint di sviluppo.

- AWSGlueConsoleFullAccess— Concede l'accesso completo alle AWS Glue risorse quando un'identità a cui è allegata la policy utilizza la AWS console di gestione. Se si segue la convenzione per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questa policy è in genere associata agli utenti della AWS Glue console.

## Configurazione di Zoho CRM

Prima di poterli utilizzare AWS Glue per trasferire dati da Zoho CRM, devi soddisfare questi requisiti:

Requisiti minimi

I requisiti minimi sono i seguenti:

- Possedere un account Zoho CRM.
- L'account Zoho CRM è abilitato per l'accesso all'API.
- Hai un client API registrato nella console API per ottenere OAuth le credenziali.

## Configurazione delle connessioni Zoho CRM

Il tipo di concessione determina il modo in cui AWS Glue comunica con Zoho CRM per richiedere l'accesso ai tuoi dati. La scelta influisce sui requisiti da soddisfare prima di creare la connessione. Zoho CRM supporta solo il tipo di concessione AUTHORIZATION_CODE per 2.0. OAuth

- Questo tipo di concessione è considerato «a tre gambe» in OAuth quanto si basa sul reindirizzamento degli utenti a un server di autorizzazione di terze parti per autenticare l'utente. Viene utilizzato durante la creazione di connessioni tramite la console. AWS Glue La AWS Glue console reindirizzerà l'utente a Zoho CRM, dove l'utente deve effettuare il login e consentire a Glue le autorizzazioni richieste per accedere alla propria istanza Zoho CRM.
- Gli utenti possono comunque scegliere di creare la propria app connessa in Zoho CRM e fornire il proprio ID client, URL di autenticazione, URL del token e URL dell'istanza durante la creazione di connessioni tramite la console. AWS Glue In questo scenario, verranno comunque reindirizzati a Zoho CRM per accedere e autorizzare l'accesso alle proprie risorse. AWS Glue
- Questo tipo di concessione genera un token di aggiornamento e uno di accesso. Il token di accesso rimarrà valido per un'ora e potrebbe essere aggiornato automaticamente senza l'interazione dell'utente utilizzando il token di aggiornamento.
- Per la documentazione pubblica di Zoho CRM sulla creazione di un'app connessa per il flusso del codice di autorizzazione, consulta Autenticazione. OAuth

Per configurare una connessione Zoho CRM:

1. In AWS Glue Glue Studio, crea una connessione in Connessioni dati seguendo i passaggi seguenti:
   a. In Tipo di connessione selezionare Zoho CRM.
   b. Specificare l'`INSTANCE_URL` dell'istanza di Zoho CRM a cui si desidera connettersi.
   c. Specificare l'ID del client dell'applicazione dell'utente.
   d. Selezionare l'URL Auth appropriato dal menu a discesa.
   e. Selezionare l'URL del token appropriato dal menu a discesa.
   f. Seleziona il ruolo AWS IAM che AWS Glue può assumere e dispone delle autorizzazioni per le seguenti azioni:

      JSON

      ```
      {
        "Version":"2012-10-17",
        "Statement": [
          {
            "Effect": "Allow",
            "Action": [
              "secretsmanager:DescribeSecret",
              "secretsmanager:GetSecretValue",
              "secretsmanager:PutSecretValue",
              "ec2:CreateNetworkInterface",
              "ec2:DescribeNetworkInterfaces",
              "ec2:DeleteNetworkInterface"
            ],
            "Resource": "*"
          }
        ]
      }
      ```

   g. Seleziona quello `secretName` che desideri utilizzare per questa connessione per AWS Glue inserire i token.
   h. Selezionare le opzioni di rete se si desidera utilizzare la propria rete.
2. Concedi il permesso di lettura `secretName` al ruolo IAM associato al tuo AWS Glue lavoro.
3. Nella configurazione del tuo AWS Glue lavoro, fornisci `connectionName` una connessione di rete aggiuntiva.

## Lettura dalle entità Zoho CRM

### Prerequisito

Oggetti Zoho CRM da cui si desidera leggere. Sarà necessario il nome dell'oggetto.

Entità supportate come origine Sync:

| Entità | Può essere filtrato | Supporta limite | Supporta Ordina per | Supporta Seleziona * | Supporta il partizionamento |
|---|---|---|---|---|---|
| Prodotto | Sì | Sì | Sì | Sì | Sì |
| Quote | Sì | Sì | Sì | Sì | Sì |
| Ordine di acquisto | Sì | Sì | Sì | Sì | Sì |
| Soluzione | Sì | Sì | Sì | Sì | Sì |
| Esegui una chiamata a | Sì | Sì | Sì | Sì | Sì |
| Operazione | Sì | Sì | Sì | Sì | Sì |
| Event | Sì | Sì | Sì | Sì | Sì |
| Fattura | Sì | Sì | Sì | Sì | Sì |
| Account | Sì | Sì | Sì | Sì | Sì |
| Contatti | Sì | Sì | Sì | Sì | Sì |
| Fornitori | Sì | Sì | Sì | Sì | Sì |
| Campagna | Sì | Sì | Sì | Sì | Sì |
| Offerta | Sì | Sì | Sì | Sì | Sì |
| Lead | Sì | Sì | Sì | Sì | Sì |

| Entità | Può essere filtrato | Supporta limite | Supporta Ordina per | Supporta Seleziona * | Supporta il partizionamento |
|---|---|---|---|---|---|
| Modulo personalizzato | Sì | Sì | Sì | Sì | Sì |
| Ordine di vendita | Sì | Sì | Sì | Sì | Sì |
| Prezzari | Sì | Sì | Sì | Sì | Sì |
| Caso | Sì | Sì | Sì | Sì | Sì |

Esempio:

```
zoho_read = glueContext.create_dynamic_frame.from_options(
    connection_type="ZOHO",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "entityName",
        "API_VERSION": "v7",
        "INSTANCE_URL": "https://www.zohoapis.in/"
    }
```

Entità supportate come origine Async:

| Entità | Può essere filtrato | Supporta limite | Supporta Ordina per | Supporta Seleziona * | Supporta il partizionamento |
|---|---|---|---|---|---|
| Prodotto | Sì | No | No | Sì | No |
| Quote | Sì | No | No | Sì | No |
| Ordine di acquisto | Sì | No | No | Sì | No |

| Entità | Può essere filtrato | Supporta limite | Supporta Ordina per | Supporta Seleziona * | Supporta il partizionamento |
|---|---|---|---|---|---|
| Soluzione | Sì | No | No | Sì | No |
| Esegui una chiamata a | Sì | No | No | Sì | No |
| Operazione | Sì | No | No | Sì | No |
| Event | Sì | No | No | Sì | No |
| Fattura | Sì | No | No | Sì | No |
| Account | Sì | No | No | Sì | No |
| Contatti | Sì | No | No | Sì | No |
| Fornitori | Sì | No | No | Sì | No |
| Campagna | Sì | No | No | Sì | No |
| Offerta | Sì | No | No | Sì | No |
| Lead | Sì | No | No | Sì | No |
| Modulo personalizzato | Sì | No | No | Sì | No |
| Ordine di vendita | Sì | No | No | Sì | No |
| Prezzari | Sì | No | No | Sì | No |
| Caso | Sì | No | No | Sì | No |

Esempio:

```
zoho_read = glueContext.create_dynamic_frame.from_options(
    connection_type="ZOHO",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "entityName",
        "API_VERSION": "v7",
        "INSTANCE_URL": "https://www.zohoapis.in/",
        "TRANSFER_MODE": "ASYNC"
    }
```

Dettagli del campo Zoho CRM:

Zoho CRM fornisce endpoint per recuperare i metadati in modo dinamico per le entità supportate. Pertanto, il supporto dell'operatore viene acquisito a livello di tipo di dati.

| Entità | Tipo di dati | Operatori supportati |
|---|---|---|
| Entità Zoho (tutte le entità) | Numero intero | !=, =, <, <=, >, >=, BETWEEN |
| | Stringa | Like, =, != |
| | BigInteger | !=, =, <, <=, >, >=, BETWEEN |
| | Booleano | = |
| | Double | !=, =, <, <=, >, >=, BETWEEN |
| | BigDecimal | !=, =, <, <=, >, >=, BETWEEN |
| | Data | !=, =, <, <=, >, >=, BETWEEN |
| | DateTime | !=, =, <, <=, >, >=, BETWEEN |
| | Struct | N/D |
| | List | N/D |

Query di partizionamento

Il partizionamento non è supportato in modalità asincrona.

Partizionamento basato su filtri (modalità sincrona):

È possibile fornire le opzioni Spark aggiuntive PARTITION_FIELD, LOWER_BOUND, UPPER_BOUND e NUM_PARTITIONS se si desidera utilizzare la concorrenza in Spark. Con questi parametri, la query originale viene suddivisa in NUM_PARTITIONS query secondarie che possono essere eseguite contemporaneamente dalle attività Spark.

- PARTITION_FIELD: il nome del campo da utilizzare per partizionare la query.
- LOWER_BOUND: un valore limite inferiore inclusivo del campo di partizione scelto.

  Per il campo Datetime accettiamo il formato di timestamp Spark utilizzato nelle query SQL di Spark.

  Esempi di valori validi:

  ```
  "2024-09-30T01:01:01.000Z"
  ```

- UPPER_BOUND: un valore limite superiore esclusivo del campo di partizione scelto.
- NUM_PARTITIONS: il numero di partizioni.

Esempio:

```
zoho_read = glueContext.create_dynamic_frame.from_options(
    connection_type="zohocrm",
    connection_options={
        "connectionName": "connectionName",
        "ENTITY_NAME": "entityName",
        "API_VERSION": "v7",
        "PARTITION_FIELD": "Created_Time"
        "LOWER_BOUND": "2022-01-01T01:01:01.000Z"
        "UPPER_BOUND": "2024-01-01T01:01:01.000Z"
        "NUM_PARTITIONS": "10"
    }
```

## Opzioni di connessione Zoho CRM

Le seguenti sono le opzioni di connessione per Zoho CRM:

- ENTITY_NAME(String): (obbligatorio) utilizzato per la lettura. Il nome dell'oggetto in Zoho CRM.
- API_VERSION(String): (obbligatorio) utilizzato per la lettura. La versione dell'API Zoho CRM Rest che si desidera utilizzare.

- `SELECTED_FIELDS`(Elenco<Stringa>): predefinito: vuoto (SELEZIONA *). Utilizzato per la lettura. Colonne da selezionare per l'oggetto.
- `FILTER_PREDICATE`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Dev'essere in formato Spark SQL.
- `QUERY`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Query Spark SQL completa.
- `PARTITION_FIELD`(Stringa): utilizzato per la lettura. Campo da utilizzare per partizionare la query.
- `LOWER_BOUND`(Stringa): utilizzato per la lettura. Un valore limite inferiore inclusivo del campo di partizione scelto.
- `UPPER_BOUND`(Stringa): utilizzato per la lettura. Un valore limite superiore esclusivo del campo di partizione scelto.
- `NUM_PARTITIONS`(Numero intero): predefinito: 1. Utilizzato per la lettura. Numero di partizioni da leggere.
- `INSTANCE_URL`(Stringa): (obbligatorio) utilizzato per la lettura. Un URL di istanza Zoho CRM valido.
- `TRANSFER_MODE`(Stringa): utilizzato per indicare se la query deve essere eseguita in modalità asincrona.

## Limitazioni e note per il connettore Zoho CRM

Di seguito sono riportate le limitazioni o le note per il connettore Zoho CRM:

- Con la versione API v7 è possibile recuperare un massimo di 100.000 record. Consultare la documentazione di Zoho.
- Per l'entità Evento, l'etichetta “Riunione” viene visualizzata come indicato nella documentazione di Zoho.
- Per la funzionalità Seleziona tutto:
  - È possibile recuperare un massimo di 50 campi da SaaS sia per la chiamata GET che per quella POST.
  - Se si desidera avere dati per un campo specifico che non appartiene ai primi 50 campi, sarà necessario fornire manualmente l'elenco dei campi selezionati.
  - Se vengono selezionati più di 50 campi, tutti i campi oltre i 50 verranno tagliati e conterranno dati nulli in Amazon S3.

  - Nel caso di un'espressione di filtro, se l'elenco di 50 campi fornito dall'utente non include "id" e "Created_Time", verrà presentata un'eccezione personalizzata per richiedere all'utente di includere questi campi.
- Gli operatori di filtro possono variare field-to-field nonostante abbiano lo stesso tipo di dati. Pertanto, è necessario specificare manualmente un operatore diverso per ciascun campo che genera un errore nella piattaforma SaaS.
- Per la funzionalità Ordina per:
  - I dati possono essere ordinati solo in base a un singolo campo senza un'espressione di filtro, mentre possono essere ordinati in base a più campi quando viene applicata un'espressione di filtro.
  - Se non viene specificato alcun tipo di ordinamento per il campo selezionato, i dati verranno recuperati in ordine crescente per impostazione predefinita.
- Le aree supportate per il connettore Zoho CRM sono Stati Uniti, Europa, India, Australia e Giappone.
- Limitazioni della funzionalità di lettura asincrona:
  - Il limite Ordina per e il partizionamento non sono supportati nella modalità asincrona.
  - In modalità asincrona possiamo trasferire dati fino a 500 pagine con 200.000 record per pagina.
  - Per un intervallo di un minuto è consentito il download di solo 10 richieste. Quando si supera il limite di download, il sistema restituisce un errore HTTP 429 e sospende tutte le richieste di download per un minuto prima che l'elaborazione possa riprendere.
  - Dopo aver completato il processo in blocco, è possibile accedere al file scaricabile solo per un periodo di un giorno. Dopodiché, non è possibile accedere al file tramite endpoint.
  - È possibile fornire un massimo di 200 campi di selezione tramite un endpoint. Se si specificano più di 200 campi di selezione in un endpoint, il sistema esporterà automaticamente tutti i campi disponibili per quel modulo.
  - I campi esterni creati in qualsiasi modulo non sono supportati in Bulk Read APIs.
  - L'ordinamento e le clausole `Group_by` non sono supportati tramite questo endpoint API.
  - I valori dei campi con dati sanitari sensibili verranno recuperati solo quando l'opzione Limita l'accesso ai dati tramite API nelle impostazioni di conformità è disabilitata. Se l'opzione è abilitata, il valore sarà vuoto nel risultato.
  - Limiti di Filtri/Criteri
    - Il numero massimo di criteri che possono essere utilizzati in una query è 25.
    - I Filtri/Criteri sui campi di testo multiriga non sono supportati.

# Connessione a Zoom Meetings

Zoom Meetings è una piattaforma di videoconferenza basata su cloud che può essere usata per videoconferenze, audioconferenze, webinar, registrazioni delle riunioni e chat dal vivo.

Argomenti


- AWS Glue supporto per Zoom Meetings
- Policy contenenti le operazioni API per creare e utilizzare le connessioni
- Configurazione di Zoom Meetings
- Configurazione dell'app client Zoom Meetings
- Configurazione delle connessioni di Zoom Meetings
- Lettura da entità Zoom Meetings
- Opzioni di connessione Zoom Meetings
- Limitazioni Zoom Meetings


## AWS Glue supporto per Zoom Meetings

AWS Glue supporta Zoom Meetings come segue:

Supportato come origine?

Sì. Puoi utilizzare i job AWS Glue ETL per interrogare i dati da Zoom Meetings.

Supportato come destinazione?

No.

Versioni dell'API Zoom Meetings supportate

Sono supportate le seguenti versioni dell'API Zoom Meetings:

- v2

## Policy contenenti le operazioni API per creare e utilizzare le connessioni

La seguente policy di esempio descrive le autorizzazioni AWS IAM richieste per la creazione e l'utilizzo delle connessioni. Se si sta creando un nuovo ruolo, creare una policy che contenga quanto segue:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:ListConnectionTypes",
        "glue:DescribeConnectionType",
        "glue:RefreshOAuth2Tokens",
        "glue:ListEntities",
        "glue:DescribeEntity"
      ],
      "Resource": "*"
    }
  ]
}
```

In alternativa, se non desideri utilizzare il metodo precedente, utilizza le seguenti policy IAM gestite:

- AWSGlueServiceRole— Garantisce l'accesso alle risorse che AWS Glue i vari processi richiedono per l'esecuzione per conto dell'utente. Queste risorse includono AWS Glue Amazon S3, IAM, CloudWatch Logs e Amazon. EC2 Se segui la convenzione di denominazione delle risorse specificata in questa politica, AWS Glue i processi dispongono delle autorizzazioni richieste. Questa policy è in genere collegata ai ruoli specificati quando si definiscono crawler, processi ed endpoint di sviluppo.
- AWSGlueConsoleFullAccess— Concede l'accesso completo alle AWS Glue risorse quando un'identità a cui è allegata la policy utilizza la AWS console di gestione. Se si segue la convenzione per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questa policy è in genere associata agli utenti della AWS Glue console.

## Configurazione di Zoom Meetings

Prima di poter AWS Glue utilizzare il trasferimento di dati da Zoom Meetings, devi soddisfare questi requisiti:

Requisiti minimi

I requisiti minimi sono i seguenti:

- Possedere un account Zoom Meetings.
- L'account Zoom è abilitato all'accesso tramite API.
- Hai creato un' OAuth2 app nel tuo account Zoom Meetings. Questa integrazione fornisce le credenziali AWS Glue utilizzate per accedere ai dati in modo sicuro quando effettua chiamate autenticate al vostro account. Per ulteriori informazioni, consulta the section called "Configurazione dell'app client Zoom Meetings".

Se soddisfi questi requisiti, sei pronto per connetterti AWS Glue al tuo account Zoom Meetings. Per le connessioni tipiche non sono necessarie altre azioni in Zoom Meetings.

## Configurazione dell'app client Zoom Meetings

1. Accedere a Zoom App Marketplace.
2. Scegliere Sviluppo > Crea app.
3. Scegli General App per un'app basata su OAuth 2.0.
4. Nella pagina Informazioni di base, aggiungi o aggiorna le informazioni sull'app, come il nome dell'app, la modalità di gestione dell'app, le credenziali dell'app e le OAuth informazioni.
5. Nella sezione Seleziona la modalità di gestione dell'app, confermare come si desidera che venga gestita l'app:
   a. Gestita dall'amministratore: gli amministratori dell'account aggiungono e gestiscono l'app
   b. Gestita dall'utente: i singoli utenti aggiungono e gestiscono l'app. L'app ha accesso solo ai dati autorizzati dell'utente.
6. Credenziali dell'app: il flusso di compilazione genera automaticamente le credenziali dell'app (ID client e segreto del client) per l'app.
7. Nella sezione OAuth Informazioni, configura OAuth la tua app.
   a. OAuth URL di reindirizzamento (obbligatorio): inserisci l'URL o l'endpoint di reindirizzamento da configurare OAuth tra l'app e Zoom.
   b. Usare l'URL in modalità rigorosa (opzionale)
   c. Controllo del sottodominio (opzionale)
   d. OAuth elenchi consentiti (obbligatorio): aggiungi eventuali reindirizzamenti univoci URLs che Zoom dovrebbe consentire come reindirizzamenti validi per i tuoi flussi. OAuth

8. Nella pagina Ambiti, selezionare i metodi dell'API Zoom che l'app è autorizzata a chiamare. Gli ambiti definiscono quali informazioni e funzionalità sono disponibili per l'utente. Selezionare i seguenti ambiti granulari:
   - user:read:list_users:admin
   - zoom_rooms:read:list_rooms:admin
   - group:read:list_members:admin
   - group:read:administrator:admin
   - group:read:list_groups:admin
   - report:read:admin
   - role:read:list_roles, role:read:list_roles:admin

   Una volta aggiunti gli ambiti, scegliere Continua e l'app è pronta per l'uso.

Per ulteriori informazioni sulla configurazione OAuth 2.0, consulta Integrazioni (app) OAuth .

## Configurazione delle connessioni di Zoom Meetings

Zoom Meetings supporta il tipo di concessione AUTHORIZATION_CODE per. OAuth2 Il tipo di concessione determina il modo in cui AWS Glue comunica con Zoom Meetings per richiedere l'accesso ai tuoi dati.

- Questo tipo di concessione è considerato «a tre gambe» in OAuth quanto si basa sul reindirizzamento degli utenti a un server di autorizzazione di terze parti per autenticare l'utente. Viene utilizzato durante la creazione di connessioni tramite la console. AWS Glue L'utente che crea una connessione deve fornire informazioni OAuth correlate come Client ID e Client Secret per la propria applicazione client Zoom Meetings. La AWS Glue console reindirizzerà l'utente a Zoom, dove l'utente deve effettuare il login e consentire AWS Glue le autorizzazioni richieste per accedere alla propria istanza di Zoom Meetings.
- Gli utenti possono comunque scegliere di creare la propria app connessa in Zoom Meetings e fornire il proprio ID cliente e il segreto del client durante la creazione di connessioni tramite la AWS Glue console. In questo scenario, verranno comunque reindirizzati a Zoom Meetings per accedere e autorizzare l'accesso AWS Glue alle proprie risorse.
- Questo tipo di concessione genera un token di aggiornamento e un token di accesso. Il token di accesso dura poco tempo e può essere aggiornato automaticamente senza l'interazione dell'utente utilizzando il token di aggiornamento.

- Per la documentazione pubblica di Zoom Meetings sulla creazione di un'app connessa per il OAuth flusso del codice di autorizzazione, consulta Using OAuth 2.0.

Per configurare una connessione di Zoom Meetings:

1. In AWS Secrets Manager, crea un segreto con i seguenti dettagli:
   a. Per le app connesse gestite dal cliente: il segreto deve contenere l'app connessa Consumer Secret con `USER_MANAGED_CLIENT_APPLICATION_CLIENT_SECRET` come chiave.
   b. Nota: devi creare un segreto per le tue connessioni in AWS Glue.

1. In AWS Glue Glue Studio, crea una connessione in Connessioni dati seguendo i passaggi seguenti:
   a. In Tipo di connessione, selezionare Zoom Meetings.
   b. Fornire l'ambiente Zoom Meetings a cui ci si desidera connettere.
   c. Seleziona il ruolo AWS IAM che AWS Glue può assumere e dispone delle autorizzazioni per le seguenti azioni:

      JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:DescribeSecret",
        "secretsmanager:GetSecretValue",
        "secretsmanager:PutSecretValue",
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DeleteNetworkInterface"
      ],
      "Resource": "*"
    }
  ]
}
```

   d. Seleziona quello `secretName` che desideri utilizzare per questa connessione per AWS Glue inserire i token.

e. Selezionare le opzioni di rete se si desidera utilizzare la propria rete.

2. Concedi il permesso di lettura `secretName` al ruolo IAM associato al tuo AWS Glue lavoro.

## Lettura da entità Zoom Meetings

Prerequisito

Un oggetto di Zoom Meetings da cui si desidera leggere. È necessario il nome dell'oggetto, come `Group` o `Zoom Rooms`.

Entità supportate come origine:

| Entità | Può essere filtrato | Supporta limite | Supporta Ordina per | Supporta Seleziona * | Supporta il partizionamento |
|---|---|---|---|---|---|
| Zoom Rooms | No | Sì | No | Sì | No |
| Gruppo | No | No | No | Sì | No |
| Membro del gruppo | Sì | Sì | No | Sì | No |
| Amministratore del gruppo | No | Sì | No | Sì | No |
| Rapporto (giornaliero) | Sì | No | No | Sì | No |
| Roles | No | No | No | Sì | No |
| Utenti | Sì | Sì | No | Sì | No |

Esempio:

```
zoom_read = glueContext.create_dynamic_frame.from_options(
    connection_type="zoom",
    connection_options={
```

```
        "connectionName": "connectionName",
        "ENTITY_NAME": "organization",
        "API_VERSION": "v2"
    }
)
```

Dettagli dell'entità e del campo di Zoom Meetings:

Zoom Meetings carica dinamicamente i campi disponibili nell'entità selezionata. A seconda del tipo di dati di campo, supporta i seguenti operatori di filtro.

| Entità | Campo | Tipo di dati | Operatori supportati | |
|---|---|---|---|---|
| Stanza Zoom | status | Stringa | = | |
| | tipo | Stringa | = | |
| | unassigned_rooms | Booleano | = | |
| | location_id | Stringa | = | |
| | room_id | Stringa | N/A | |
| | activation_code | Stringa | N/A | |
| | id | Stringa | N/A | |
| | nome | Stringa | N/A | |
| | tag_ids | Stringa | N/A | |
| | query_name | Stringa | N/A | |
| Rapporto giornaliero | mese | Data | = | |
| | data | Data | N/D | |
| | meeting_minutes | Numero intero | N/D | |
| | meetings | Numero intero | N/D | |

| Entità | Campo | Tipo di dati | Operatori supportati | |
|---|---|---|---|---|
| | new_users | Numero intero | N/D | |
| | participants | Numero intero | N/D | |
| | group_id | Stringa | N/A | |
| Utente | created_at | DateTime | N/D | |
| | dept | Stringa | N/A | |
| | e-mail | Stringa | N/A | |
| | employee_ unique_id | Stringa | N/A | |
| | first_name | Stringa | N/A | |
| | group_ids | List | N/D | |
| | host_key | Stringa | N/A | |
| | id | Stringa | N/A | |
| | im_group_ids | Stringa | N/A | |
| | last_client_versio n | Stringa | N/A | |
| | tempo_di_ultimo accesso | DateTime | N/D | |
| | last_name | Stringa | N/A | |
| | plan_united_type | Stringa | N/A | |
| | custom_at tributes | List | N/D | |

| Entità | Campo | Tipo di dati | Operatori supportati | |
|---|---|---|---|---|
| | pmi | BigInteger | N/D | |
| | role_id | Stringa | = | |
| | status | Stringa | = | |
| | timezone | Stringa | N/A | |
| | tipo | Numero intero | N/D | |
| | verified | Numero intero | N/D | |
| | user_created_at | DateTime | N/D | |
| | display_name | Stringa | N/A | |
| | phone_number | Stringa | N/A | |
| | linguaggio | Stringa | N/A | |
| | license | Stringa | = | |
| Gruppo | id | Stringa | N/A | |
| | nome | Stringa | N/A | |
| | total_members | Numero intero | N/D | |
| Membro del gruppo | e-mail | Stringa | N/A | |
| | first_name | Stringa | N/A | |
| | id | Stringa | N/A | |
| | last_name | Stringa | N/A | |
| | tipo | Numero intero | N/D | |
| | primary_group | Booleano | N/D | |

| Entità | Campo | Tipo di dati | Operatori supportati | |
|---|---|---|---|---|
| | member_id | Stringa | N/A | |
| Amministratore del gruppo | id | Stringa | N/A | |
| | e-mail | Stringa | N/A | |
| | nome | Stringa | N/A | |
| role | description | Stringa | N/A | |
| | id | Stringa | N/A | |
| | nome | Stringa | N/A | |
| | total_members | Numero intero | N/D | |
| | tipo | Stringa | = | |

Query di partizionamento

Zoom Meetings non supporta il partizionamento basato su filtri ov quello basato su record.

## Opzioni di connessione Zoom Meetings

Le seguenti sono le opzioni di connessione per Zoom Meetings:

- `ENTITY_NAME`(String): (obbligatorio) utilizzato per la lettura. Il nome dell'entità Zoom Meetings. Ad esempio, `group`.
- `API_VERSION`(String): (obbligatorio) utilizzato per la lettura. La versione di Zoom Meetings Rest API che si desidera utilizzare. Il valore sarà `v2`, poiché Zoom Meetings attualmente supporta solo la versione v2.
- `SELECTED_FIELDS`(List<String>): predefinito: vuoto (SELECT *). Utilizzato per la lettura. Un elenco di colonne separate da virgole che si desidera selezionare per l'entità selezionata.
- `FILTER_PREDICATE`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Dev'essere in formato Spark SQL.
- `QUERY`(Stringa): predefinito: vuoto. Utilizzato per la lettura. Query Spark SQL completa.

### Limitazioni Zoom Meetings

Di seguito sono riportate le limitazioni o le note per Zoom Meetings:

- Zoom Meetings non supporta orderby.
- Zoom Meetings non supporta il partizionamento basato su filtri perché non esiste un campo in grado di soddisfare i criteri richiesti.
- Zoom Meetings non supporta il partizionamento basato su record perché il limite di impaginazione e l'impaginazione basata sull'offset non sono supportati.

## Aggiunta di una connessione JDBC utilizzando i propri driver JDBC

Quando si utilizza una connessione JDBC è possibile utilizzare il proprio driver JDBC. Quando il driver predefinito utilizzato dal AWS Glue crawler non è in grado di connettersi a un database, è possibile utilizzare il proprio driver JDBC. Ad esempio, se desideri utilizzare SHA-256 con il tuo database Postgres e i driver Postgres precedenti non lo supportano, puoi utilizzare il tuo driver JDBC.

### Origini dati supportate

| Origini dati supportate | Origini dati non supportate |
|---|---|
| MySQL | Snowflake |
| Postgres | |
| Oracle | |
| Redshift | |
| SQL Server | |
| Aurora* | |

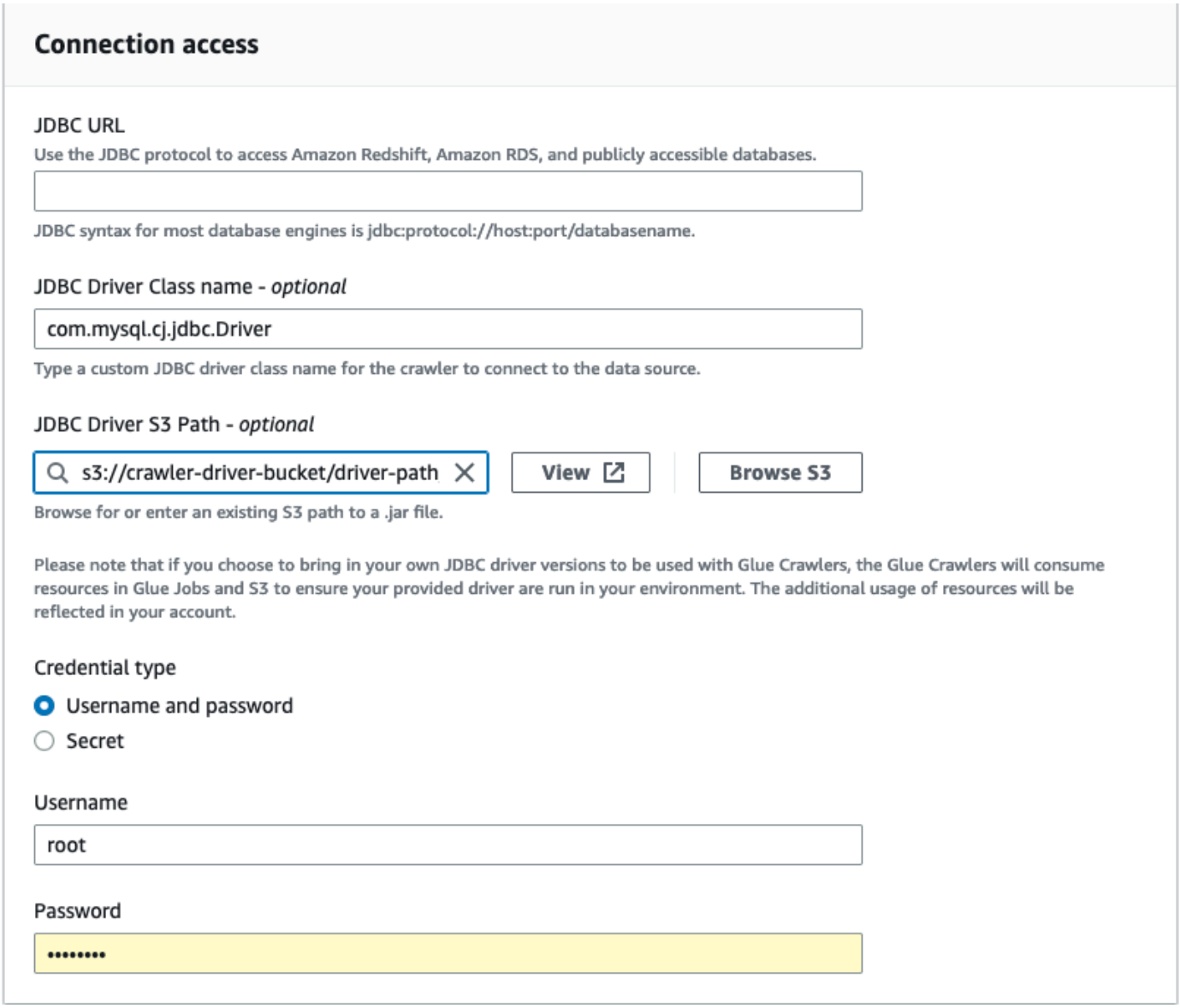
* Supportato se si utilizza il driver JDBC nativo. Non è possibile avvalersi di tutte le funzionalità del driver.

# Aggiunta del driver JDBC a una connessione JDBC

> **Note**
>
> Se scegli di importare le tue versioni dei driver JDBC, AWS Glue i crawler consumeranno risorse nei job e nei bucket AWS Glue Amazon S3 per garantire che il driver fornito venga eseguito nel tuo ambiente. L'utilizzo aggiuntivo delle risorse si rifletterà nel tuo account. Il costo dei AWS Glue crawler e dei job rientra nella categoria della fatturazione. AWS Glue Inoltre, è importante sottolineare che anche se si fornisce il proprio driver JDBC, ciò non implica automaticamente che il crawler possa sfruttare tutte le funzionalità offerte da tale driver.

Per aggiungere il proprio driver JDBC a una connessione JDBC:

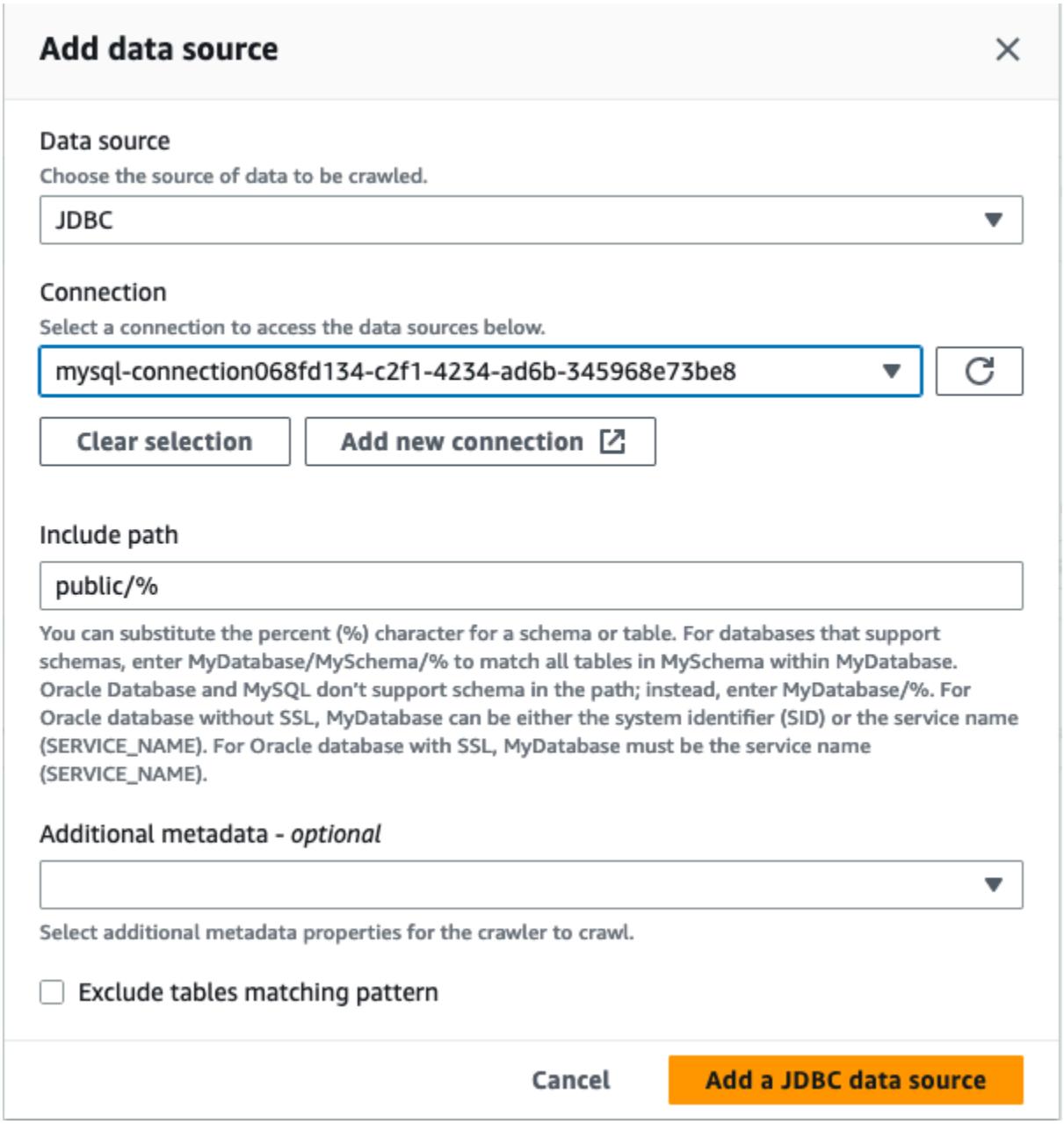
1. Aggiungi il file del driver JDBC a una posizione Amazon S3. È possibile creare una cartella bucket o utilizzare una and/or cartella bucket esistente. and/or
2. Nella AWS Glue console, scegli Connessioni nel menu a sinistra sotto Data Catalog, quindi crea una nuova connessione.
3. Completa i campi per le Proprietà di connessione e scegli JDBC per il Tipo di connessione.
4. In Accesso alla connessione, inserisci l'URL JDBC e il Nome della classe del driver JDBC - facoltativo. Il nome della classe del driver deve riferirsi a un'origine dati supportata dai crawler. AWS Glue




5. Scegli il percorso Amazon S3 in cui si trova il driver JDBC nel campo Percorso del driver JDBC Amazon S3 - facoltativo.
6. Se inserisci un nome utente e una password o un segreto, completa i campi per Tipo di credenziale. Al termine, scegli Crea connessione.

> **Note**
>
> Il test delle connessioni personalizzate non è attualmente supportato. Quando esegui il crawling dell'origine dati con un driver JDBC fornito da te, il crawler salta questo passaggio.

7. Aggiungi la connessione appena creata a un crawler. Nella AWS Glue console, scegli Crawler nel menu a sinistra sotto Data Catalog, quindi crea un nuovo crawler.
8. Nella procedura guidata Aggiungi crawler, nel passaggio 2 scegli Aggiungi un'origine dati.




9. Scegli JDBC come origine dati e scegli la connessione creata nei passaggi precedenti. Completa
10. Per utilizzare il tuo driver JDBC con un AWS Glue crawler, aggiungi le seguenti autorizzazioni al ruolo utilizzato dal crawler:

    - Concedi le autorizzazioni per le seguenti operazioni di processo: `CreateJob`, `DeleteJob`, `GetJob`, `GetJobRun`, `StartJobRun`.

- Concedi le autorizzazioni per le operazioni IAM: `iam:PassRole`
- Concedi le autorizzazioni per le operazioni di Amazon S3: `s3:DeleteObjects`, `s3:GetObject`, `s3:ListBucket`, `s3:PutObject`.
- Concedi l'accesso principale al servizio nella policy IAM. bucket/folder

Policy IAM di esempio:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:ListBucket",
        "s3:DeleteObject"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket/driver-parent-folder/driver.jar",
        "arn:aws:s3:::amzn-s3-demo-bucket"
      ]
    }
  ]
}
```

Il AWS Glue crawler crea due cartelle: _glue_job_crawler e _crawler.

Se il file jar del driver si trova nella cartella `s3://amzn-s3-demo-bucket/driver.jar`", aggiungi le seguenti risorse:

```
"Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket/_glue_job_crawler/*",
    "arn:aws:s3:::amzn-s3-demo-bucket/_crawler/*"
            ]
```

Se il file jar del driver si trova nella cartella `s3://amzn-s3-demo-bucket/tmp/driver/subfolder/driver.jar"`, aggiungi le seguenti risorse:

```
"Resource": [
               "arn:aws:s3:::amzn-s3-demo-bucket/tmp/_glue_job_crawler/*",
    "arn:aws:s3:::amzn-s3-demo-bucket/tmp/_crawler/*"
            ]
```

11. Se si utilizza un VPC, è necessario consentire l'accesso all' AWS Glue endpoint creando l'endpoint dell'interfaccia e aggiungendolo alla tabella di routing. Per ulteriori informazioni, consulta Creazione di un endpoint VPC di interfaccia per AWS Glue
12. Se utilizzi la crittografia nel tuo Data Catalog, crea l'endpoint di AWS KMS interfaccia e aggiungilo alla tabella di routing. Per ulteriori informazioni, consulta la pagina Creating a VPC endpoint for AWS KMS.

# Utilizzo di connettori e connessioni personalizzati con AWS Glue Studio

AWS Glue fornisce supporto integrato per gli archivi dati più comuni (ad esempio Amazon Redshift, Amazon Aurora, Microsoft SQL Server, MySQL, MongoDB e PostgreSQL) utilizzando le connessioni JDBC. AWS Glue consente inoltre di usare driver JDBC personalizzati nei processi di estrazione, trasformazione e caricamento (ETL). Per gli archivi dati non supportati in modo nativo, ad esempio le applicazioni SaaS, è possibile utilizzare i connettori.

Un connettore è un pacchetto di codice opzionale che aiuta con l'accesso ai datastore in AWS Glue Studio. Puoi iscriverti a diversi connettori offerti in Marketplace AWS.

Quando si creano lavori ETL, è possibile utilizzare un archivio dati supportato in modo nativo, un connettore da Marketplace AWS o connettori personalizzati. Se utilizzi un connettore, è innanzitutto necessario creare una connessione. Una connessione contiene le proprietà necessarie per connettersi a un particolare datastore. È possibile utilizzare la connessione con le tue origini dati e destinazioni dati nel processo ETL. Connettori e connessioni funzionano insieme per facilitare l'accesso ai datastore.

Le seguenti connessioni sono disponibili durante la creazione di connessioni per connettori:

- Amazon Aurora— un motore di database relazionale scalabile e ad alte prestazioni con sicurezza, backup e ripristino integrati e accelerazione in memoria.
- Amazon DocumentDB: un servizio di database di documenti scalabile, altamente disponibile e completamente gestito che supporta MongoDB e SQL. APIs
- Amazon Redshift— un servizio di database di documenti scalabile, altamente disponibile e completamente gestito che supporta MongoDB e SQL. APIs
- Azure SQL: un servizio di database relazionale basato su cloud di Microsoft Azure che offre funzionalità di archiviazione e gestione dei dati scalabili, affidabili e sicure.
- Cosmos DB: un servizio di database cloud distribuito a livello globale di Microsoft Azure che offre funzionalità di archiviazione e interrogazione di dati scalabili e ad alte prestazioni.
- Google BigQuery: un data warehouse cloud senza server per l'esecuzione di query SQL veloci su set di dati di grandi dimensioni.
- JDBC: un sistema di gestione di database relazionale (RDBMS) che utilizza un'API Java per connettersi e interagire con le connessioni dati.
- Kafka: una piattaforma di elaborazione di flussi open-source, utilizzata per lo streaming e la messaggistica di dati in tempo reale.
- MariaDB: un fork di MySQL sviluppato dalla comunità che offre prestazioni, scalabilità e funzionalità migliorate.
- MongoDB: un database orientato ai documenti multipiattaforma che offre scalabilità, flessibilità e prestazioni elevate.
- MongoDB Atlas: un'offerta di database as a service (DBaaS) basata su cloud di MongoDB che semplifica la gestione e la scalabilità delle implementazioni di MongoDB.
- Microsoft SQL Server: un sistema di gestione di database relazionale (RDBMS) di Microsoft che offre solide funzionalità di archiviazione, analisi e reporting dei dati.
- Mixpanel: una piattaforma di analisi che aiuta le aziende ad analizzare il modo in cui gli utenti interagiscono con i loro siti Web, con le loro applicazioni mobili e con altri prodotti digitali.
- MySQL: un sistema di gestione di database relazionale (RDBMS) open-source ampiamente utilizzato nelle applicazioni Web e noto per la sua affidabilità e scalabilità.
- Rete: un'origine dati di rete rappresenta una risorsa o un servizio accessibile in rete a cui è possibile accedere tramite una piattaforma di integrazione dei dati.
- OpenSearch— una fonte di OpenSearch dati è un'applicazione in grado di connettersi e importare dati da. OpenSearch

- Oracle: un sistema di gestione di database relazionale (RDBMS) di Oracle Corporation che offre solide funzionalità di archiviazione, analisi e reporting dei dati.
- PostgreSQL: un sistema di gestione di database relazionale (RDBMS) open-source che offre solide funzionalità di archiviazione, analisi e reporting dei dati.
- Salesforce: Salesforce fornisce software di gestione delle relazioni con i clienti (CRM) che aiutano nelle vendite, nell'assistenza clienti, nell'e-commerce e altro ancora. Se sei un utente Salesforce, puoi connetterti AWS Glue al tuo account Salesforce. Quindi, puoi usare Salesforce come origine o destinazione dei dati per i tuoi processi ETL. Esegui questi processi per trasferire dati tra Salesforce e AWS i servizi o altre applicazioni supportate.
- SAP HANA: un database in memoria e una piattaforma di analisi che fornisce elaborazione rapida dei dati, analisi avanzate e integrazione dei dati in tempo reale.
- Snowflake: un data warehouse basato su cloud che fornisce servizi di archiviazione e analisi dei dati scalabili e ad alte prestazioni.
- Teradata: un sistema di gestione di database relazionale (RDBMS) che offre funzionalità di archiviazione, analisi e reporting dei dati ad alte prestazioni.
- Vertica: un data warehouse analitico orientato alle colonne progettato per l'analisi di big data che offre prestazioni di query rapide, analisi avanzate e scalabilità.

## Creazione di connettori personalizzati

Puoi anche costruire un connettore personalizzato e caricare il codice del connettore su AWS Glue Studio.

I connettori personalizzati sono integrati in AWS Glue Studio attraverso l'API del runtime di Spark AWS Glue. l'API del runtime di Spark AWS Glue consente di collegare qualsiasi connettore compatibile con l'interfaccia Spark, Athena o JDBC. Consente di aggiungere qualsiasi opzione di connessione disponibile per il connettore personalizzato.

È possibile incapsulare tutte le proprietà di connessione con le connessioni AWS Glue e fornire il nome della connessione al processo ETL. L'integrazione con le connessioni del catalogo dati consente di utilizzare le stesse proprietà di connessione per più chiamate in una singola applicazione Spark o in applicazioni diverse.

Puoi specificare ulteriori opzioni per la connessione. Lo script di processo che AWS Glue Studio genera contiene una voce `Datasource` che utilizza la connessione per collegare il connettore con le opzioni di connessione specificate. Ad esempio:

```
Datasource = glueContext.create_dynamic_frame.from_options(connection_type =
"custom.jdbc", connection_options = {"dbTable":"Account","connectionName":"my-custom-
jdbc-
connection"}, transformation_ctx = "DataSource0")
```

Per aggiungere un connettore ad AWS Glue Studio

1. Crea il codice per il connettore personalizzato. Per ulteriori informazioni, consulta Sviluppo di connettori personalizzati.
2. Aggiungi il supporto per le caratteristiche AWS Glue al connettore. Di seguito sono riportati alcuni esempi di queste funzionalità e di come vengono utilizzate all'interno dello script di processo generato da AWS Glue Studio:

   - Mappatura dei dati: il connettore può eseguire il typecast le colonne durante la lettura dal datastore sottostante. Ad esempio, una `dataTypeMapping` di `{"INTEGER":"STRING"}` converte tutte le colonne di tipo `Integer` in colonne di tipo `String` durante l'analisi dei record e la costruzione del `DynamicFrame`. In questo modo gli utenti possono eseguire il cast delle colonne nel tipo desiderato.

     ```
     DataSource0 = glueContext.create_dynamic_frame.from_options(connection_type
     = "custom.jdbc", connection_options = {"dataTypeMapping":{"INTEGER":"STRING"}",
     connectionName":"test-connection-jdbc"}, transformation_ctx = "DataSource0")
     ```

   - Partizionamento per letture parallele: AWS Glue consente la lettura dei dati paralleli dal datastore partizionando i dati su una colonna. È necessario specificare la colonna, il limite inferiore e il limite superiore della partizione e il numero di partizioni. Questa funzione consente di utilizzare il parallelismo dei dati e più executor Spark allocati per l'applicazione Spark.

     ```
     DataSource0 = glueContext.create_dynamic_frame.from_options(connection_type
     = "custom.jdbc", connection_options = {"upperBound":"200","numPartitions":"4",
     "partitionColumn":"id","lowerBound":"0","connectionName":"test-connection-jdbc"},
     transformation_ctx = "DataSource0")
     ```

   - Utilizzo Gestione dei segreti AWS per l'archiviazione delle credenziali: la connessione Data Catalog può contenere anche un `secretId` file segreto memorizzato in. Gestione dei segreti AWS Il AWS segreto può archiviare in modo sicuro le informazioni di autenticazione e credenziali e fornirle in fase di esecuzione. AWS Glue In alternativa, è possibile specificare l'`secretId` dallo script Spark come segue:

```
DataSource = glueContext.create_dynamic_frame.from_options(connection_type
= "custom.jdbc", connection_options = {"connectionName":"test-connection-jdbc",
 "secretId"-> "my-secret-id"}, transformation_ctx = "DataSource0")
```

- Filtraggio dei dati di origine con predicati di riga e proiezioni di colonna: il runtime di Spark AWS Glue consente inoltre agli utenti di trasferire le query SQL per filtrare i dati all'origine con predicati di riga e proiezioni di colonna. Ciò permette al processo ETL di caricare i dati filtrati più velocemente dagli archivi dati che supportano push-down. Un esempio di query SQL trasferita in un'origine dati JDBC è: `SELECT id, name, department FROM department WHERE id < 200.`

```
DataSource = glueContext.create_dynamic_frame.from_options(connection_type =
"custom.jdbc", connection_options = {"query":"SELECT id, name, department FROM
 department
WHERE id < 200","connectionName":"test-connection-jdbc"}, transformation_ctx =
"DataSource0")
```

- Segnalibri di processo: AWS Glue supporta il caricamento incrementale dei dati da origini JDBC. AWS Glue tiene traccia dell'ultimo record elaborato dal datastore ed elabora nuovi record di dati nelle successive esecuzioni del processo ETL. I segnalibri di processo utilizzano la chiave primaria come colonna predefinita per il tasto segnalibro, a condizione che questa colonna aumenti o diminuisca in sequenza. Per ulteriori informazioni sui segnalibri di processo, consulta Segnalibri di processo nella Guida per sviluppatori di AWS Glue .

```
DataSource0 = glueContext.create_dynamic_frame.from_options(connection_type =
"custom.jdbc", connection_options = {"jobBookmarkKeys":["empno"],
 "jobBookmarkKeysSortOrder"
:"asc", "connectionName":"test-connection-jdbc"}, transformation_ctx =
 "DataSource0")
```

3. Impacchetta il connettore personalizzato come file JAR e carica il file su Amazon S3.
4. Testa il connettore personalizzato. Per ulteriori informazioni, consulta le istruzioni su Glue Custom Connectors: GitHub Local Validation Tests Guide.
5. Nella console AWS Glue Studio, scegli Connectors (Connettori) nel pannello di navigazione della console.
6. Nella pagina Connectors (Connettori), seleziona Create custom connector (Crea connettore personalizzato).

7. Nella pagina Create custom connecto (Crea connettore personalizzato), immetti le seguenti informazioni:
   - Il percorso della posizione del file JAR di codice personalizzato in Amazon S3.
   - Un nome per il connettore che verrà utilizzato da AWS Glue Studio.
   - Il tipo di connettore, che può essere uno tra JDBC, Spark o Athena.
   - Il nome del punto di ingresso all'interno del codice personalizzato che AWS Glue Studio chiama per utilizzare il connettore.
     - Per i connettori JDBC, questo campo deve essere il nome della classe del driver JDBC.
     - Per i connettori Spark, questo campo deve essere il nome completo della classe dell'origine dati, o il relativo alias, che si utilizza quando si carica l'origine dati Spark con l'operatore `format`.
   - (Solo JDBC) L'URL di base utilizzato dalla connessione JDBC per l'archivio dati.
   - (Facoltativo) Una descrizione del connettore personalizzato.
8. Scegli Create connector (Crea connettore).
9. Dalla pagina Connectors (Connettori), crea una connessione che utilizza questo connettore, come descritto in Creazione di connessioni per i connettori.

## Aggiunta di connettori a AWS Glue Studio

Un connettore è una parte di codice che facilita la comunicazione tra il datastore e AWS Glue. Puoi abbonarti a un connettore disponibile in Marketplace AWS oppure puoi creare il tuo connettore personalizzato.

Iscrizione ai Marketplace AWS connettori

AWS Glue Studiosemplifica l'aggiunta di connettori da Marketplace AWS.

Per aggiungere un connettore da Marketplace AWS a AWS Glue Studio

1. Nella console AWS Glue Studio, scegli Connectors (Connettori) nel pannello di navigazione della console.
2. Nella pagina Connectors (Connectors), scegli Go to Marketplace AWS (Vai su Marketplace AWS).

3. In Marketplace AWS, in Prodotti in evidenza, scegli il connettore che desideri utilizzare. Puoi scegliere uno dei connettori in evidenza o utilizzare la ricerca. Puoi eseguire la ricerca in base al nome o al tipo di connettore e utilizzare le opzioni per perfezionare i risultati della ricerca.

   Se desideri utilizzare uno dei connettori in evidenza, scegli View product (Visualizza prodotto). Se hai usato la ricerca per trovare un connettore, scegli il nome del connettore.
4. Nella pagina prodotto del connettore, utilizza le schede per visualizzare le informazioni sul connettore. Se decidi di acquistare il connettore, scegli Continue to Subscribe (Continua con la sottoscrizione).
5. Inserisci le informazioni di pagamento, quindi scegli Continue to Configure (Continua con la configurazione).
6. Nella pagina Configure this software (Configurazione software), scegli il metodo di implementazione e la versione del connettore da utilizzare. Quindi, scegli Continue to Launch (Continua con l'avvio).
7. Nella pagina Launch this software (Avvia software), puoi rivedere le istruzioni di utilizzo fornite dal provider del connettore. Quando vuoi continuare, scegli Attiva connessione in AWS Glue Studio.

   Dopo un breve periodo di tempo, la console mostra la pagina Create marketplace connection (Crea connessione al marketplace) in AWS Glue Studio.
8. Crea una connessione che utilizza questo connettore, come descritto in Creazione di connessioni per i connettori.

   In alternativa, ora puoi scegliere Activate connector only (Attiva solo connettore) per ignorare la creazione di una connessione. Devi creare una connessione in un secondo momento prima di poter utilizzare il connettore.

## Creazione di connessioni per i connettori

Una AWS Glue connessione è un oggetto del catalogo dati che memorizza le informazioni di connessione per un particolare data store. Le connessioni memorizzano le credenziali di accesso, le stringhe URI, le informazioni sul cloud privato virtuale (VPC) e altro ancora. La creazione di connessioni nel catalogo dati consente di evitare di dover specificare tutti i dettagli della connessione ogni volta che si crea un processo.

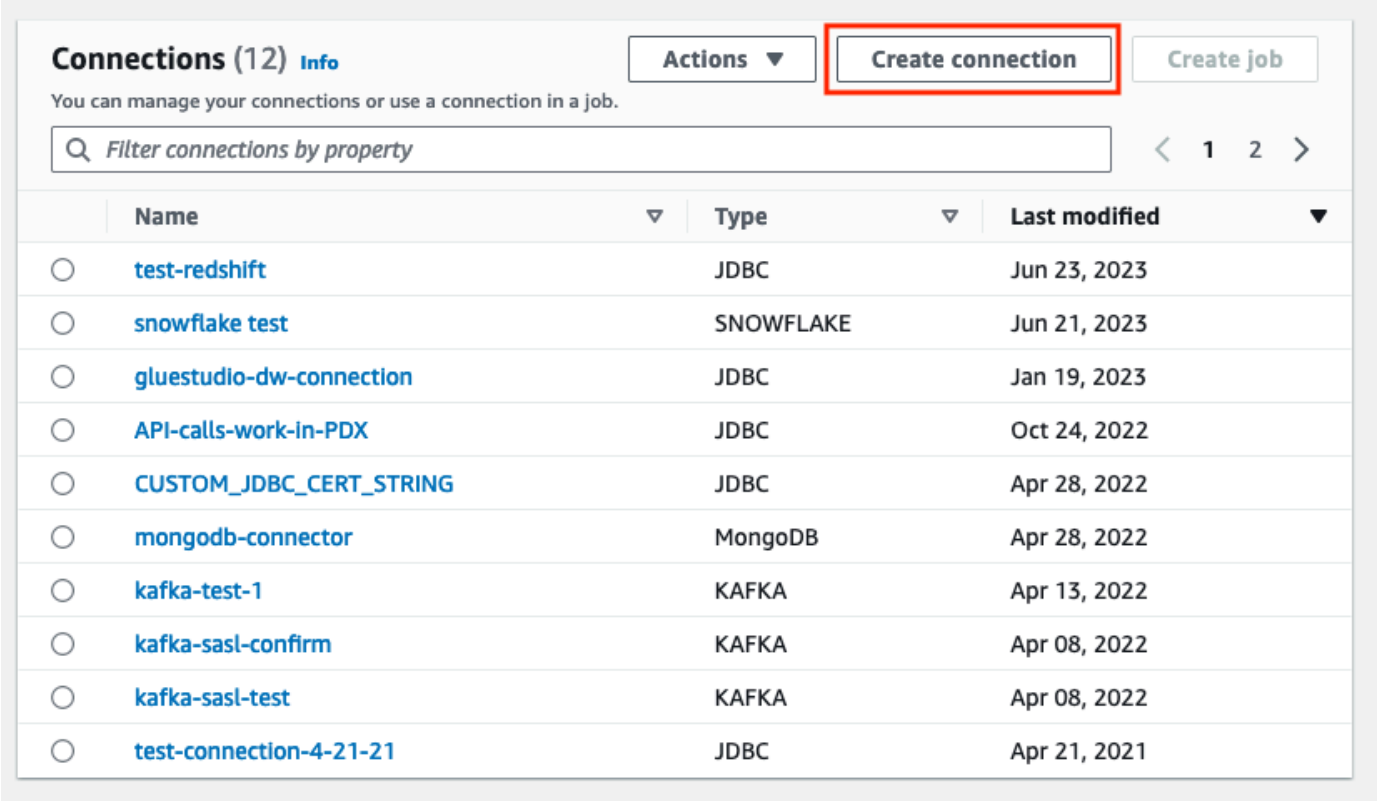
Per creare una connessione per un connettore

1. Nella AWS Glue Studio console, scegli Connettori nel pannello di navigazione della console. Nella sezione Connessioni, scegli Crea connessione.
2. Scegli l'origine dati per la quale desideri creare una connessione nel passaggio 1 della procedura guidata Crea connessione dati. Sono disponibili diversi modi per visualizzare le origini dati disponibili, inclusi i seguenti:

   - Filtra le origini dati disponibili scegliendo una scheda. Per impostazione predefinita, è selezionata l'opzione Tutti i connettori.
   - Attiva Elenco per visualizzare le origini dati sotto forma di elenco o torna a Griglia per visualizzare i connettori disponibili nel layout a griglia.
   - Utilizza la barra di ricerca per restringere l'elenco delle origini dati. Durante la digitazione, i risultati di ricerca vengono visualizzati e le fonti non corrispondenti vengono rimosse dalla visualizzazione.

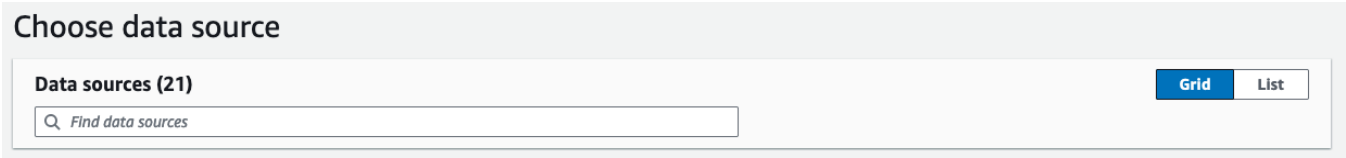
   Dopo aver scelto l'origine dati, scegli Avanti.
3. Configura la connessione nel passaggio 2 della procedura guidata.

   Inserisci i dettagli della connessione. A seconda del tipo di connettore selezionato, viene richiesto di inserire ulteriori informazioni:




4. Scegli l'origine dati per la quale desideri creare una connessione nel passaggio 1 della procedura guidata Crea connessione dati. Sono disponibili diversi modi per visualizzare le origini dati disponibili. Per impostazione predefinita, tutte le origini dati disponibili vengono visualizzate in un layout a griglia. Puoi anche:

   - Attiva Elenco per visualizzare le origini dati sotto forma di elenco o torna a Griglia per visualizzare i connettori disponibili nel layout a griglia.
   - Utilizza la barra di ricerca per restringere l'elenco delle origini dati. Durante la digitazione, i risultati di ricerca vengono visualizzati e le fonti non corrispondenti vengono rimosse dalla visualizzazione.

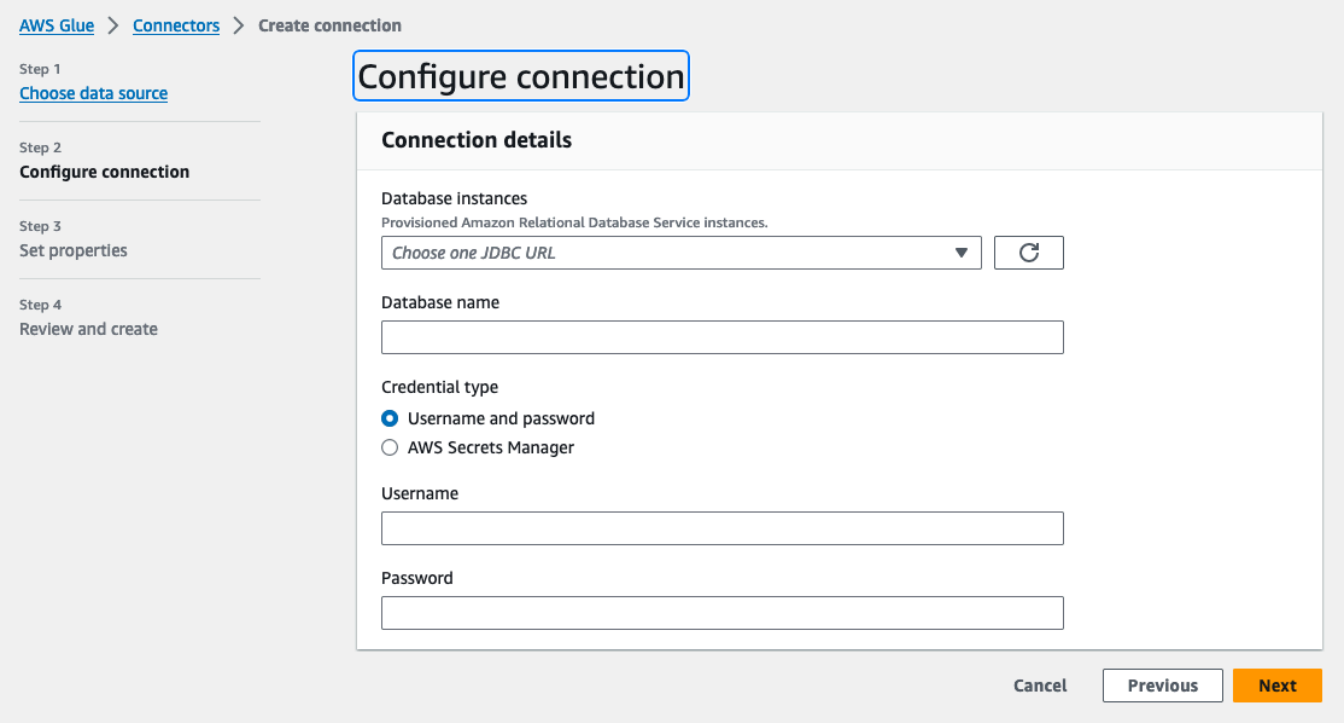
   
   

   Dopo aver scelto l'origine dati, scegli Avanti.

5. Configura la connessione nel passaggio 2 della procedura guidata.

Inserisci i dettagli della connessione. A seconda del tipo di connettore selezionato, potrebbe essere richiesto di inserire ulteriori informazioni di connessione: Sono inclusi:

- Dettagli di connessione: questi campi cambieranno a seconda dell'origine dati a cui ti stai connettendo. Ad esempio, se ti connetti a database Amazon DocumentDB, inserirai l'URL di Amazon DocumentDB. Se ti connetti a Amazon Aurora, sceglierai l'istanza del database e inserirai il nome del database. Di seguito sono riportati i dettagli di connessione necessari per Amazon Aurora:




- Tipo di credenziale: scegli tra nome utente e password o  Gestione dei segreti AWS. Inserisci le informazioni di autenticazione richieste.
- Per i connettori che utilizzano JDBC, inserisci le informazioni necessarie per creare l'URL JDBC per il datastore.
- Se usi un cloud privato virtuale (VPC), inserisci le relative informazioni di rete.

6. Impostare le proprietà di connessione nel passaggio 3 della procedura guidata. È possibile aggiungere descrizione e tag come parte opzionale di questo passaggio. Il nome è obbligatorio ed è precompilato con un valore predefinito. Scegli Next (Successivo).
7. Controlla l'origine, i dettagli e le proprietà della connessione. Per apportare modifiche, scegli Modifica per il passaggio della procedura guidata. Quando è tutto pronto, scegli Crea connessione.

Scegli Create connection (Crea connessione).

Vieni reindirizzato alla pagina Connectors (Connettori) e il banner informativo indica la connessione creata. Ora puoi utilizzare la connessione nei tuoi processi AWS Glue Studio .

## Creazione di una connessione Kafka

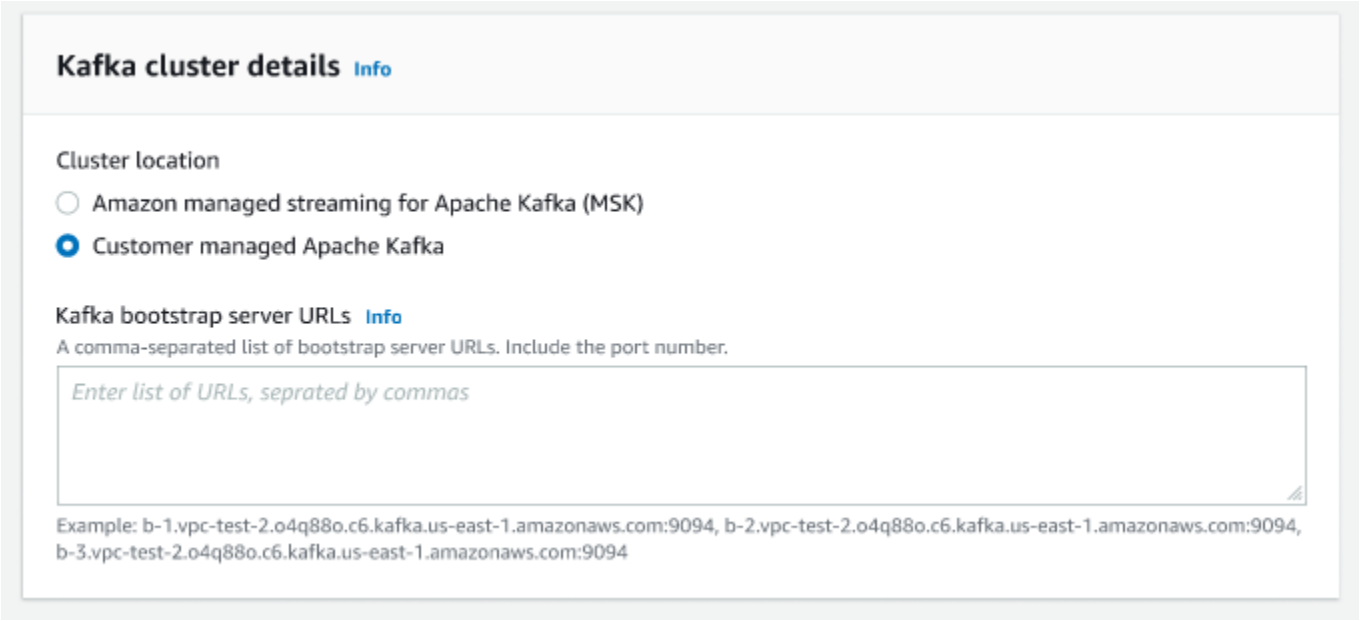
Quando crei una connessione Kafka, selezionando Kafka dal menu a discesa visualizzerai impostazioni aggiuntive da configurare:

- Dettagli del cluster Kafka
- Autenticazione
- Encryption (Crittografia)
- Opzioni di rete

Configurazione dei dettagli del cluster Kafka

1. Scegli la posizione del cluster. Puoi scegliere tra un cluster Amazon Managed Streaming for Apache Kafka (MSK) o un cluster Apache Kafka gestito dal cliente. Per ulteriori informazioni sullo streaming Amazon Managed for Apache Kafka, consulta Amazon Managed Streaming for Apache Kafka (MSK).

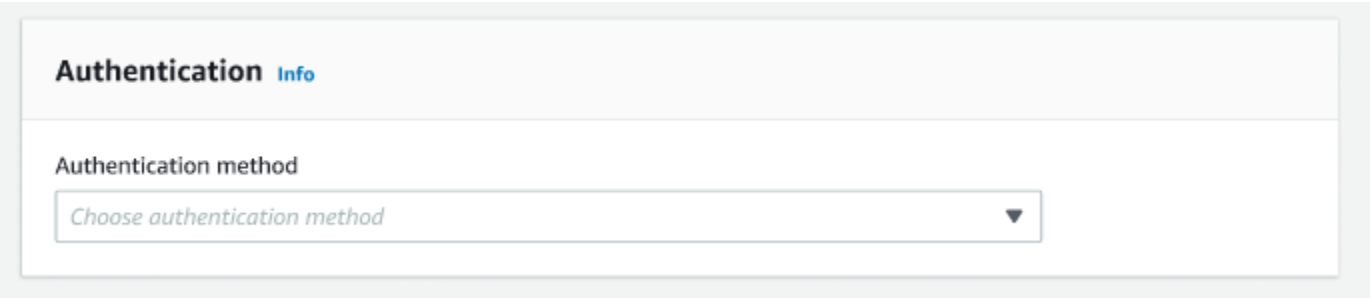
   > **Note**
   >
   > Amazon Streaming gestito per Apache Kafka supporta solo i metodi di autenticazione TLS e SASL/SCRAM-SHA-512.




2. Inserisci il file per i tuoi server di bootstrap URLs Kafka. È possibile inserirne più di uno separando ciascun server con una virgola. Includi il numero della porta alla fine dell'URL aggiungendo `:<port number>`.

   Ad esempio: `b-1.vpc-test-2.034a88o.kafka-us-east-1.amazonaws.com:9094`

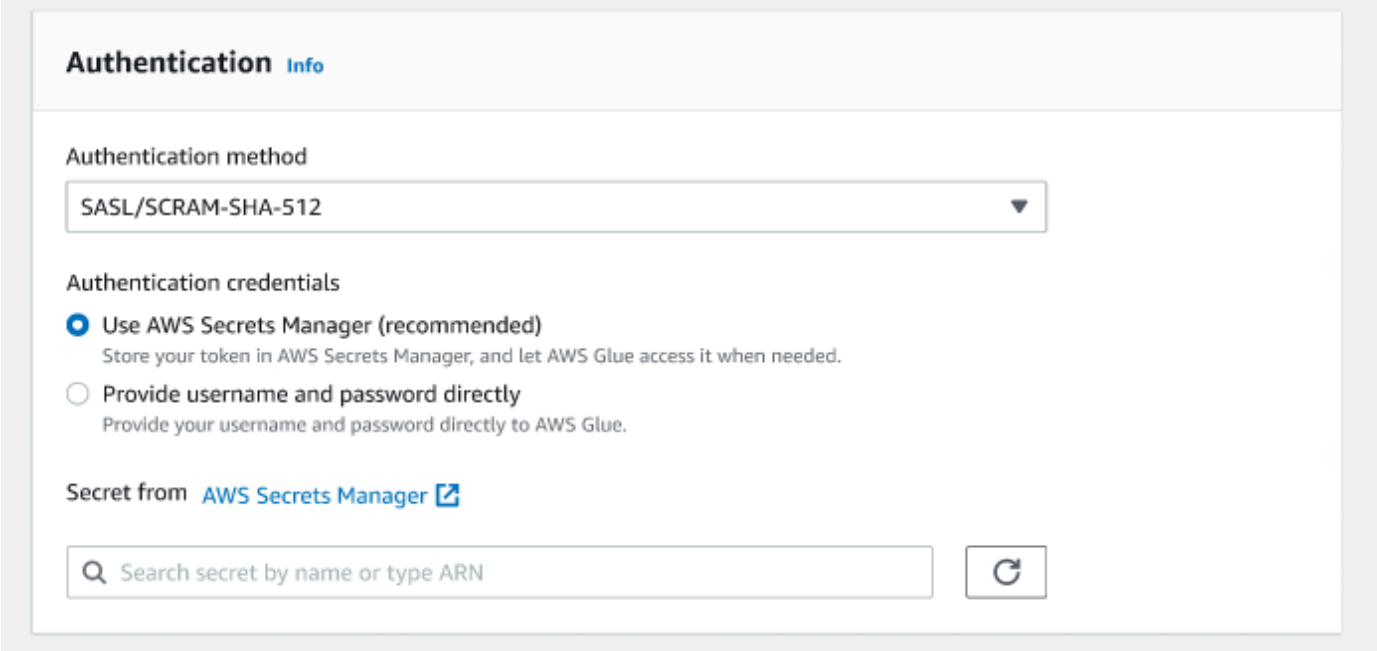
Selezionare i metodi di autenticazione




AWS Glue supporta il framework SASL (Simple Authentication and Security Layer) per l'autenticazione. Il framework SASL supporta vari meccanismi di autenticazione e AWS Glue offre i protocolli SCRAM (nome utente e password), GSSAPI (protocollo Kerberos) e PLAIN (nome utente e password).

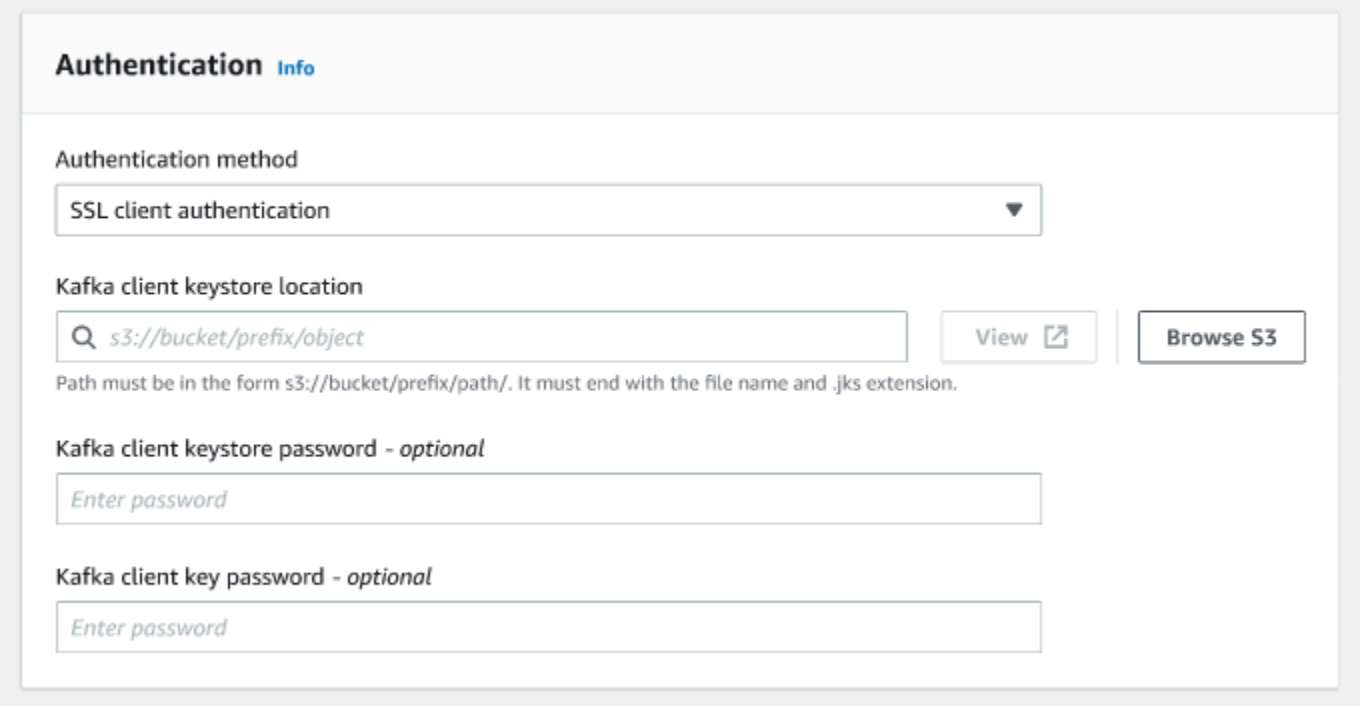
Quando si sceglie un metodo di autenticazione dal menu a discesa, è possibile selezionare i seguenti metodi di autenticazione client:

- Nessuno: nessuna autenticazione. Questo è utile se si crea una connessione a scopo di test.
- SASL/SCRAM-SHA-512: scegli questo metodo di autenticazione per specificare le credenziali di autenticazione. Sono disponibili due opzioni:
  - Usa AWS Secrets Manager (consigliato): se selezioni questa opzione, puoi memorizzare le tue credenziali in AWS Secrets Manager e consentire AWS Glue l'accesso alle informazioni quando necessario. Specifica il segreto che memorizza le credenziali di autenticazione SSL o SASL.




  - Fornisci direttamente nome utente e password.
- SASL/GSSAPI (Kerberos): se si seleziona questa opzione, è possibile selezionare la posizione del file keytab, il file krb5.conf e immettere il nome principale Kerberos e il nome del servizio Kerberos. Le posizioni per il file keytab e il file krb5.conf devono trovarsi in una posizione Amazon S3. Poiché MSK non supporta ancora SASL/GSSAPI, questa opzione è disponibile solo per i cluster Apache Kafka gestiti dal cliente. Per ulteriori informazioni, consulta la Documentazione di MIT Kerberos: keytab.
- SASL/PLAIN - Scegliete questo metodo di autenticazione per specificare le credenziali di autenticazione. Sono disponibili due opzioni:
  - Usa AWS Secrets Manager (consigliato): se selezioni questa opzione, puoi memorizzare le tue credenziali in AWS Secrets Manager e consentire AWS Glue l'accesso alle informazioni quando necessario. Specifica il segreto che memorizza le credenziali di autenticazione SSL o SASL.
  - Fornisci direttamente nome utente e password.

- Autenticazione client SSL: selezionando questa opzione, è possibile selezionare la posizione del keystore client Kafka navigando su Amazon S3. Facoltativamente, è possibile inserire la password del keystore del client Kafka e la password della chiave del client Kafka.



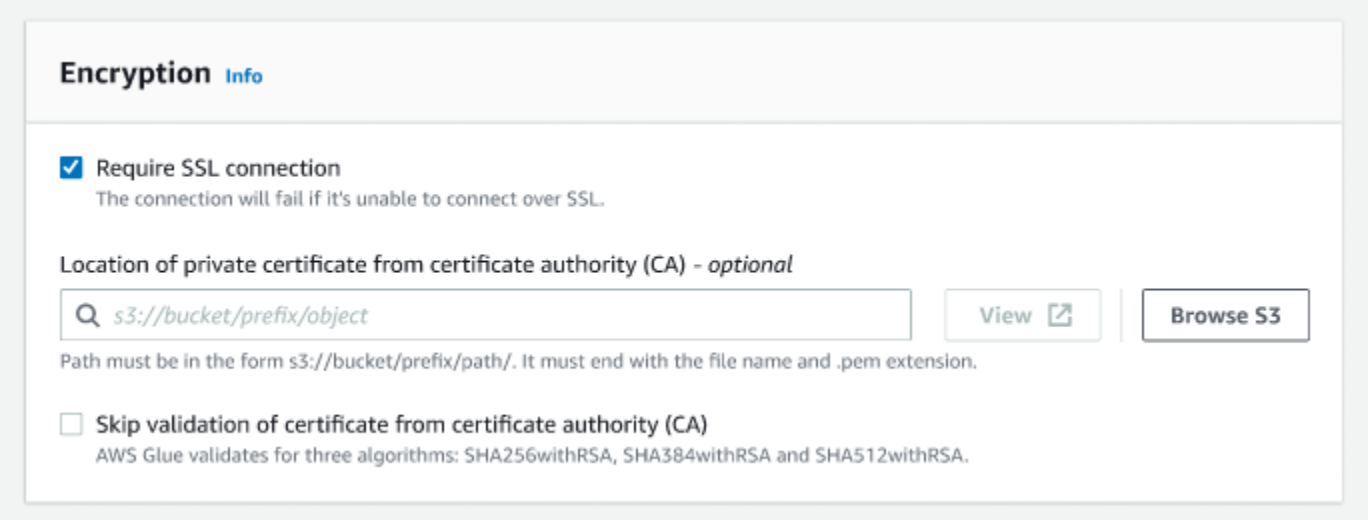

Configurazione delle impostazioni di crittografia

1. Se la connessione Kafka richiede una connessione SSL, seleziona la casella di controllo per Require SSL connection (Connessione SSL necessaria). Tieni presente che la connessione non riesce se non può connettersi tramite SSL. SSL per la crittografia può essere utilizzato con qualsiasi metodo di autenticazione (SASL/SCRAM-SHA-512, SASL/GSSAPI, SASL/PLAINo SSL Client Authentication) ed è facoltativo.

   Se il metodo di autenticazione è impostato su Autenticazione client SSL, questa opzione verrà selezionata automaticamente e verrà disattivata per evitare eventuali modifiche.
2. (Facoltativo). Scegli la posizione del certificato privato dell'autorità di certificazione (CA). Tieni presente che la posizione della certificazione deve trovarsi in una sede S3. Scegli Browse (Sfoglia) per scegliere il file da un bucket S3 collegato. Il percorso deve essere nel formato `s3://bucket/prefix/filename.pem`. Deve terminare con il nome del file e l'estensione .pem.
3. È possibile scegliere di saltare la convalida del certificato da un'autorità di certificazione (CA). Scegli la casella di controlloSkip validation of certificate from certificate authority (CA) (Salta

la convalida del certificato da un'autorità di certificazione [CA]). Se questa casella non è selezionata, AWS Glue convalida i certificati per tre algoritmi:

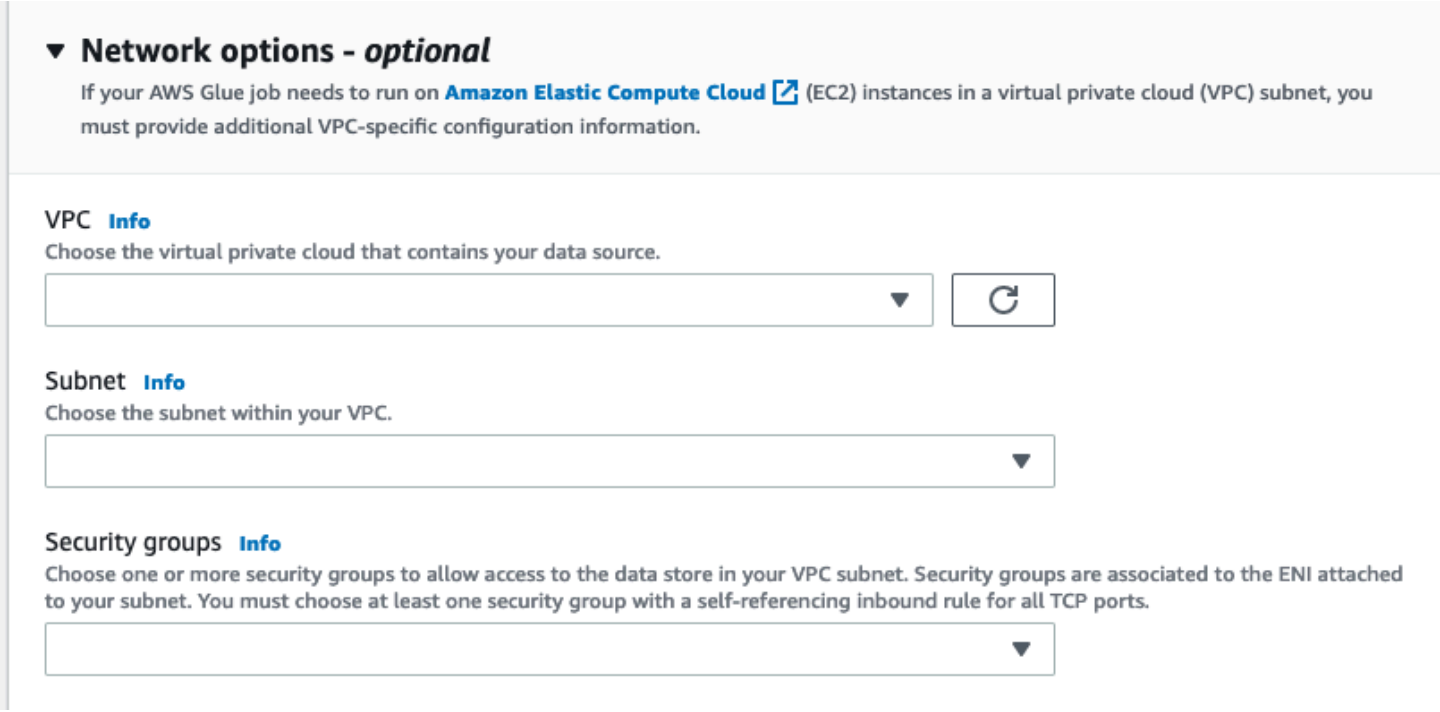
- SHA256withRSA
- SHA384withRSA
- SHA512withRSA




(Facoltativo) Opzioni di rete

Di seguito sono riportate i passaggi opzionali per configurare VPC, sottorete e gruppi di sicurezza. Se il tuo AWS Glue processo deve essere eseguito su istanze Amazon EC2 in una sottorete di cloud privato virtuale (VPC), devi fornire ulteriori informazioni di configurazione specifiche per VPC.

1. Scegli il VPC (cloud privato virtuale) che contiene l'origine dati.
2. Scegli la sottorete nel VPC.
3. Scegli uno o più gruppi di sicurezza per consentire l'accesso all'archivio dati nella sottorete VPC. I gruppi di sicurezza sono associati all'ENI collegata alla sottorete. È necessario scegliere almeno un gruppo di sicurezza con una regola in entrata autoreferenziale per tutte le porte TCP.




# Creazione di processi con connettori personalizzati

Puoi utilizzare connettori e connessioni sia per i nodi di origine dati che per i nodi di destinazione dati in AWS Glue Studio.

Argomenti

- Creare processi che utilizzano un connettore per l'origine dati
- Configurare le proprietà di origine per i nodi che utilizzano connettori
- Configurare le proprietà di destinazione per i nodi che utilizzano connettori

## Creare processi che utilizzano un connettore per l'origine dati

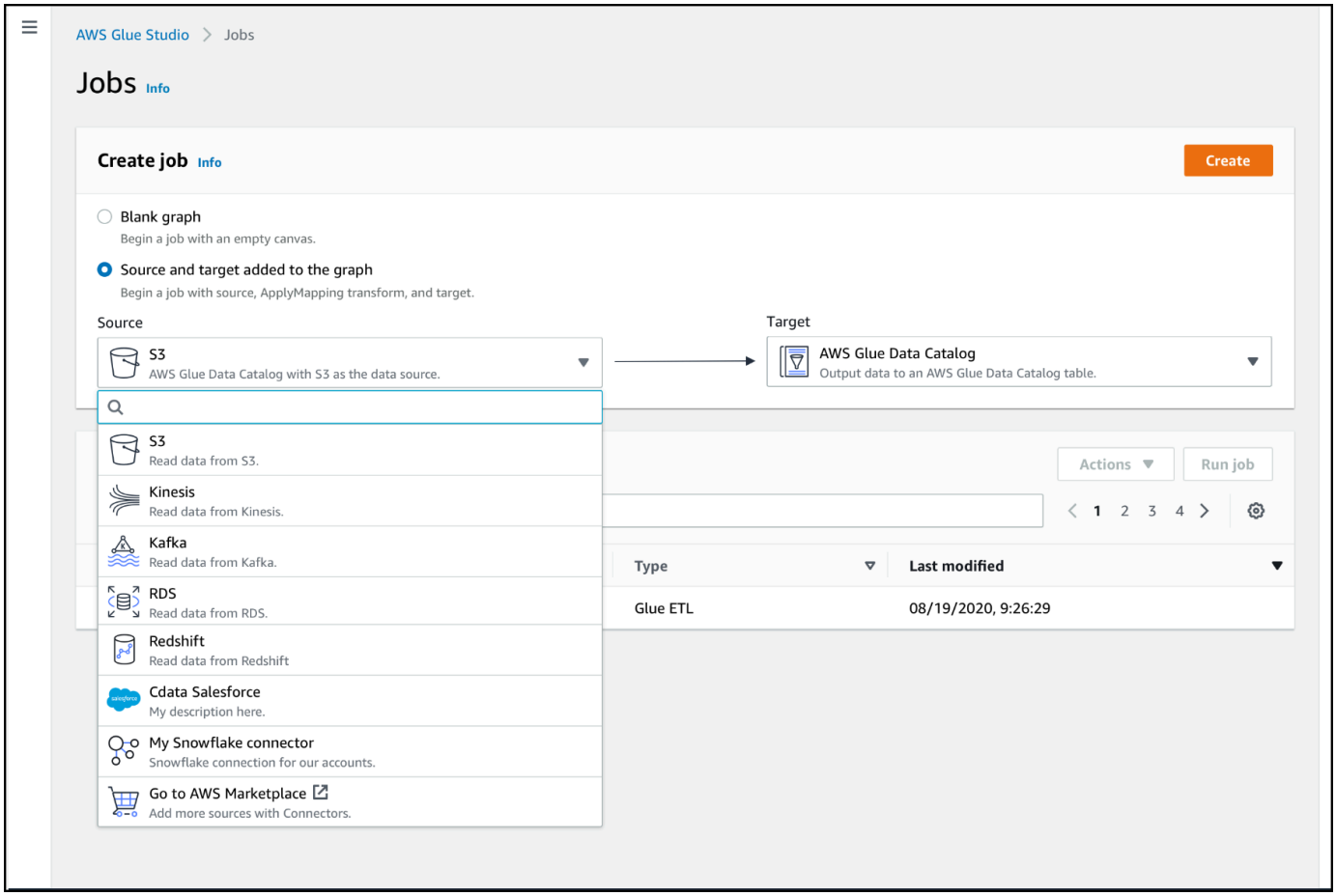
Quando si crea un nuovo processo, puoi scegliere un connettore per l'origine dati e le destinazioni dati.

Per creare un processo che utilizza connettori per l'origine dati o la destinazione dati

1. Accedi a Console di gestione AWS e apri la AWS Glue Studio console all'indirizzo https://console.aws.amazon.com/gluestudio/.

2. Nella pagina Connectors (Connettori), nell'elenco di risorse Your connections (Le tue connessioni), scegli la connessione da utilizzare nel processo, quindi scegli Create job (crea processo).

   In alternativa, nella pagina Jobs (Processi) di AWS Glue Studio, sotto Create job (Crea processo), scegli Source and target added to the graph (Origine e destinazione aggiunti al grafico). Nell'elenco a discesa Source (Origine), scegli il connettore personalizzato che desideri utilizzare nel processo. Puoi anche scegliere un connettore per Target (Destinazione).



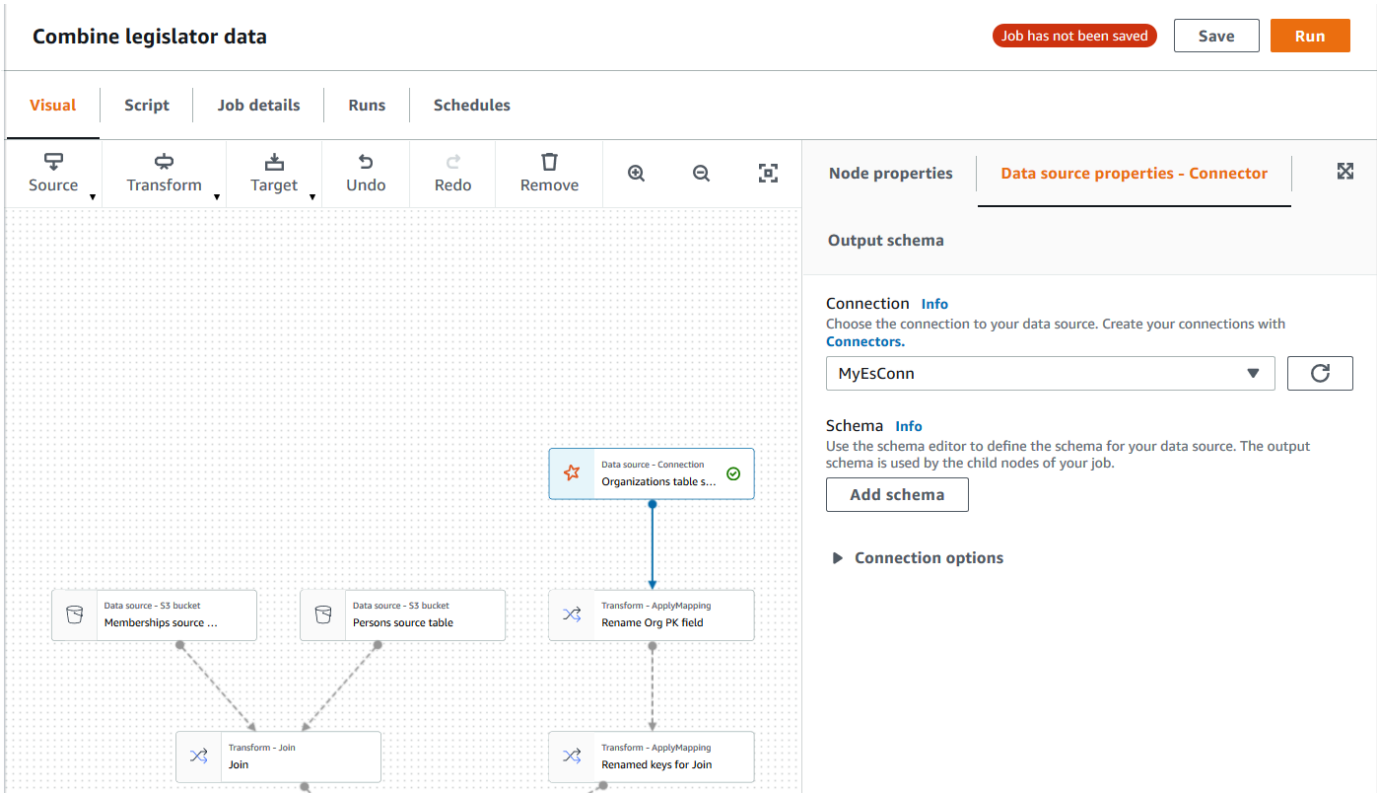

3. Scegli quindi Create (Crea) per aprire l'editor visivo dei processi.
4. Configura il nodo di origine dati, come descritto in Configurare le proprietà di origine per i nodi che utilizzano connettori.
5. Continua a creare il processo ETL aggiungendo trasformazioni, datastore aggiuntivi e destinazioni dati, come descritto in Avvio di processi ETL visivi in AWS Glue Studio.
6. Personalizza l'ambiente di esecuzione configurando le proprietà del processo, come descritto in Modificare le proprietà del processo.
7. Salva ed esegui il processo.

# Configurare le proprietà di origine per i nodi che utilizzano connettori

Dopo aver creato un processo che utilizza un connettore per l'origine dati, l'editor visivo dei processi mostra un grafico del processo con un nodo di origine dati configurato per il connettore. Devi configurare le proprietà dell'origine dati per tale nodo.

Per configurare le proprietà di un nodo di origine dati che utilizza un connettore

1. Scegli il nodo dell'origine dati del connettore nel grafico del processo oppure aggiungi un nuovo nodo e scegli il connettore per Node type (Tipo di nodo). Quindi, sulla destra, nel pannello dei dettagli dei nodi, scegli la scheda Data source properties (Proprietà dell'origine dati) se non è già selezionata.




2. Nella scheda Data source properties (Proprietà dell'origine dati), scegli la connessione da utilizzare per questo processo.

   Inserisci le informazioni aggiuntive necessarie per ciascun tipo di connessione:

JDBC

- Data source input type (Tipo di input dell'origine dati): scegli di specificare un nome di tabella o una query SQL come origine dati. A seconda del tipo, devi fornire le seguenti informazioni aggiuntive:
  - Table name (Nome tabella): il nome della tabella nell'origine dati. Se l'origine dati non utilizza la tabella dei termini, fornisci il nome di una struttura dati appropriata, come indicato dalle informazioni sull'utilizzo del connettore personalizzato (disponibili in Marketplace AWS).
  - Filter predicate (Predicato di filtro): una clausola di condizione da utilizzare durante la lettura dell'origine dati, simile a una clausola `WHERE`, che viene utilizzata per recuperare un sottoinsieme dei dati.
  - Query code (Codice query): inserisci una query SQL da utilizzare per recuperare un set di dati specifico dall'origine dati. Un esempio di query SQL di base è:

    ```
    SELECT column_list FROM
                            table_name WHERE where_clause
    ```

- Schema: poiché AWS Glue Studio utilizza le informazioni archiviate nella connessione per accedere all'origine dati anziché recuperare le informazioni sui metadati da una tabella del Data Catalog, devi fornire i metadati dello schema per l'origine dati. Scegli Add schema (Aggiungi schema) per aprire l'editor dello schema.

  Per istruzioni su come utilizzare l'editor dello schema, consulta Modifica dello schema in un nodo di trasformazione personalizzato.

- Partition column (Colonna di partizione): (facoltativo) puoi scegliere di partizionare le letture dei dati fornendo valori per Partition column (Colonna di partizione), Lower bound (Limite inferiore), Upper bound (Limite superiore) e Number of partitions (Numero di partizioni).

  I valori `lowerBound` e `upperBound` vengono utilizzati per decidere lo stride della partizione, non per filtrare le righe nella tabella. Tutte le righe della tabella vengono partizionate e restituite.

> **Note**
>
> Il partizionamento delle colonne aggiunge una condizione di partizionamento aggiuntiva alla query utilizzata per leggere i dati. Quando si utilizza una query anziché un nome di tabella, è necessario verificare che la query funzioni con la condizione di partizionamento specificata. Ad esempio:
>
> - Se il formato della query è `"SELECT col1 FROM table1"`, testa la query aggiungendo una clausola `WHERE` alla fine della query che utilizza la colonna della partizione.
> - Se il formato della query è `"SELECT col1 FROM table1 WHERE col2=val"`, testa la query estendendo la clausola `WHERE` con `AND` e un'espressione che utilizza la colonna della partizione.

- Data type casting (Casting del tipo di dati): se l'origine dati utilizza tipi di dati non disponibili in JDBC, utilizza questa sezione per specificare come convertire un tipo di dati dell'origine dati in tipi di dati JDBC. Puoi specificare fino a 50 conversioni di tipi di dati diverse. Tutte le colonne dell'origine dati che utilizzano lo stesso tipo di dati vengono convertite nello stesso modo.

  Ad esempio, se nell'origine dati sono presenti tre colonne che utilizzano il tipo di dati `Float` e si indica che il tipo di dati `Float` deve essere convertito nel tipo di dati `String` JDBC, tutte e tre le colonne che utilizzano il tipo di dati `Float` vengono convertite in `String`.

- Job bookmark keys (Chiavi di segnalibro di processo): i segnalibri del processo aiutano AWS Glue a mantenere le informazioni sullo stato e prevenire la rielaborazione dei dati precedenti. Specifica un'altra o più colonne come chiavi di segnalibro. AWS Glue Studio utilizza i tasti di segnalibro per tenere traccia dei dati già elaborati durante una precedente esecuzione del processo ETL. Tutte le colonne utilizzate per le chiavi di segnalibro personalizzati devono aumentare o diminuire in modo rigorosamente monotonico, ma sono ammessi spazi.

  Se inserisci più chiavi di segnalibro, queste vengono combinate per formare una singola chiave composta. Una chiave di segnalibro di processo composta non deve contenere colonne duplicate. Se non si specificano le chiavi di segnalibro, AWS Glue Studio per impostazione predefinita utilizza la chiave primaria come chiave di segnalibro, a condizione che sia in ordine crescente o decrescente sequenzialmente (senza spazi vuoti). Se la tabella non dispone di una chiave primaria, ma la proprietà segnalibro di processo è

abilitata, devi fornire chiavi personalizzate di segnalibro di processo. In caso contrario, la ricerca delle chiavi primarie da utilizzare come impostazione predefinita avrà esito negativo e l'esecuzione del processo avrà non riuscirà.

- Job bookmark keys sorting order (Ordinamento delle chiavi di segnalibro di processo): scegli se i valori chiave vengono aumentati o diminuiti in sequenza.

Spark

- Schema: poiché AWS Glue Studio utilizza le informazioni archiviate nella connessione per accedere all'origine dati anziché recuperare le informazioni sui metadati da una tabella del Data Catalog, devi fornire i metadati dello schema per l'origine dati. Scegli Add schema (Aggiungi schema) per aprire l'editor dello schema.

  Per istruzioni su come utilizzare l'editor dello schema, consulta Modifica dello schema in un nodo di trasformazione personalizzato.
- Connection options (Opzioni di connessione): inserisci ulteriori coppie chiave-valore in base alle esigenze per fornire ulteriori informazioni o opzioni di connessione. Ad esempio, puoi inserire un nome di database, un nome di tabella, un nome utente e una password.

  Ad esempio, per OpenSearch, si inseriscono le seguenti coppie chiave-valore, come descritto in: the section called "Tutorial: utilizzo del AWS Glue connettore per Elasticsearch"
  - `es.net.http.auth.user` : *`username`*
  - `es.net.http.auth.pass` : *`password`*
  - `es.nodes` : `https://`*`<Elasticsearch endpoint>`*
  - `es.port` : `443`
  - `path`: *`<Elasticsearch resource>`*
  - `es.nodes.wan.only` : `true`

Per un esempio delle opzioni di connessione minime da usare, vedete lo script di test di esempio MinimalSparkConnectorTest.scala on GitHub, che mostra le opzioni di connessione che normalmente fornireste in una connessione.

Athena

- Table name (Nome tabella): il nome della tabella nell'origine dati. Se utilizzi un connettore per leggere i log di CloudWatch Athena-log, devi inserire il nome della tabella. `all_log_streams`
- Athena schema name (Nome schema Athena): scegli lo schema nell'origine dati Athena corrispondente al database che contiene la tabella. Se si utilizza un connettore per la lettura da CloudWatch Athena-logs, è necessario immettere un nome di schema simile a. `/aws/glue/name`
- Schema: poiché AWS Glue Studio utilizza le informazioni archiviate nella connessione per accedere all'origine dati anziché recuperare le informazioni sui metadati da una tabella del Data Catalog, devi fornire i metadati dello schema per l'origine dati. Scegli Add schema (Aggiungi schema) per aprire l'editor dello schema.

  Per istruzioni su come utilizzare l'editor dello schema, consulta Modifica dello schema in un nodo di trasformazione personalizzato.
- Additional connection options (Opzioni di connessione aggiuntive): inserisci ulteriori coppie chiave-valore in base alle esigenze per fornire ulteriori informazioni o opzioni di connessione.

Per un esempio, vedi il `README.md` file in https://github.com/aws-samples/aws-glue-samples/. tree/master/GlueCustomConnectors/development/Athena Nei passaggi di questo documento, il codice di esempio mostra le opzioni di connessione minime richieste, che sono `tableName`, `schemaName` e `className`. L'esempio di codice specifica queste opzioni come parte della variabile `optionsMap`, ma puoi specificarle per la connessione e quindi utilizzarla.

3. (Facoltativo) Dopo aver configurato le proprietà del nodo e dell'origine dati, puoi visualizzare lo schema dei dati risultante per l'origine dati scegliendo la scheda Output schema (Schema di output) nel pannello dei dettagli del nodo. Lo schema visualizzato in questa scheda viene utilizzato da tutti i nodi figlio aggiunti al grafico del processo.
4. (Facoltativo) Dopo aver configurato le proprietà del nodo e dell'origine dati, puoi visualizzare il set di dati dall'origine dati scegliendo la scheda Data preview (Anteprima dei dati) nel pannello dei dettagli del nodo. La prima volta che si sceglie questa scheda per qualsiasi nodo del processo, viene richiesto di fornire un ruolo IAM per accedere ai dati. Esiste un costo per l'utilizzo di questa caratteristica e la fatturazione inizia non appena si fornisce un ruolo IAM.

## Configurare le proprietà di destinazione per i nodi che utilizzano connettori

Se usi un connettore per il tipo di destinazione dati, devi configurare le proprietà del nodo di destinazione dati.

Per configurare le proprietà di un nodo di destinazione dati che utilizza un connettore

1. Scegli il nodo di destinazione dati del connettore nel grafico del processo. Quindi, sulla destra, nel pannello dei dettagli dei nodi, scegli la scheda Data target properties (Proprietà della destinazione dati) se non è già selezionata.
2. Nella scheda Data target properties (Proprietà della destinazione dati), scegli la connessione da utilizzare per la scrittura nella destinazione.

   Inserisci le informazioni aggiuntive necessarie per ciascun tipo di connessione:

   JDBC

   - Connection (Connessione): scegli la connessione da utilizzare con il connettore. Per informazioni su come creare una connessione, vedi Creazione di connessioni per i connettori.
   - Table name (Nome tabella): il nome della tabella nella destinazione dati. Se la destinazione dei dati non utilizza la tabella dei termini, fornite il nome di una struttura dati appropriata, come indicato dalle informazioni sull'utilizzo del connettore personalizzato (disponibili in Marketplace AWS).
   - Batch size (Dimensione batch): (facoltativo): immetti il numero di righe o record da inserire nella tabella di destinazione in un'unica operazione. Il valore predefinito è 1000 righe.

   Spark

   - Connection (Connessione): scegli la connessione da utilizzare con il connettore. Se non hai creato una connessione in precedenza, scegli Create connection (Crea connessione) per crearne una. Per informazioni su come creare una connessione, vedi Creazione di connessioni per i connettori.
   - Connection options (Opzioni di connessione): inserisci ulteriori coppie chiave-valore in base alle esigenze per fornire ulteriori informazioni o opzioni di connessione. Puoi inserire un nome di database, un nome di tabella, un nome utente e una password.

Ad esempio, per OpenSearch, si inseriscono le seguenti coppie chiave-valore, come descritto in: the section called "Tutorial: utilizzo del AWS Glue connettore per Elasticsearch"

- `es.net.http.auth.user` : *`username`*
- `es.net.http.auth.pass` : *`password`*
- `es.nodes` : `https://`*`<Elasticsearch endpoint>`*
- `es.port` : `443`
- `path`: *`<Elasticsearch resource>`*
- `es.nodes.wan.only` : `true`

Per un esempio delle opzioni di connessione minime da usare, vedete lo script di test di esempio MinimalSparkConnectorTest.scala on GitHub, che mostra le opzioni di connessione che normalmente fornireste in una connessione.

3. Dopo aver configurato le proprietà del nodo e dell'origine dati, puoi visualizzare lo schema dei dati risultante per l'origine dati scegliendo la scheda Output schema (Schema di output) nel pannello dei dettagli del nodo.

## Gestione di connettori e connessioni

Per gestire i connettori e le connessioni, utilizza la pagina Connessioni di AWS Glue.

Argomenti

- Visualizzazione dei dettagli dei connettori e delle connessioni
- Modifica di connettori e connessioni
- Eliminazione di connettori e connessioni
- Annullare una sottoscrizione per un connettore

### Visualizzazione dei dettagli dei connettori e delle connessioni

Puoi visualizzare le informazioni di riepilogo sui connettori e sulle connessioni nelle tabelle delle risorse Your connectors (I tuoi connettori) e Your connections (Le tue connessioni) nella scheda Connectors (Connettori). Per visualizzare le informazioni dettagliate, procedi come segue.

Per visualizzare i dettagli del connettore o della connessione

1. Nella console AWS Glue Studio, scegli Connectors (Connettori) nel pannello di navigazione della console.
2. Scegli il connettore o la connessione di cui visualizzare le informazioni dettagliate.
3. Scegli Actions (Operazioni) e quindi View details (Visualizza i dettagli) per aprire la pagina relativa ai dettagli del connettore o della connessione.
4. Nella pagina prodotto, puoi scegliere di modificare o eliminare il connettore o la connessione.
   - Per i connettori, puoi scegliere Create connection (Crea connessione) per creare una nuova connessione che utilizza il connettore.
   - Per i connettori, puoi scegliere Create job (Crea processo) per creare un processo che utilizza il connettore.

## Modifica di connettori e connessioni

Per modificare le informazioni archiviate nei connettori e nelle connessioni, devi utilizzare la pagina Connectors (Connettori).

Per modificare un connettore o una connessione

1. Nella console AWS Glue Studio, scegli Connectors (Connettori) nel pannello di navigazione della console.
2. Scegli il connettore o la connessione da modificare.
3. Seleziona Actions (Operazioni), quindi scegli Edit (Modifica).

   Puoi anche scegliere View details (Visualizza i dettagli) e nella pagina dei dettagli del connettore o della connessione scegliere Edit (Modifica).
4. Nella pagina Edit connector (Modifica connettore) o Edit connection (Modifica connessione), aggiorna le informazioni, quindi scegli Save (Salva).

## Eliminazione di connettori e connessioni

Per eliminare connettori e connessioni, devi utilizzare la pagina Connectors (Connettori). Eliminando un connettore, è necessario eliminare anche tutte le connessioni create per tale connettore.

Come rimuovere i connettori da AWS Glue Studio

1. Nella console AWS Glue Studio, scegli Connectors (Connettori) nel pannello di navigazione della console.
2. Scegli il connettore o la connessione da eliminare.
3. Scegli Actions (Operazioni), quindi Delete (Elimina).

   Puoi anche scegliere View details (Visualizza i dettagli) e nella pagina dei dettagli del connettore o della connessione scegliere Delete (Elimina).
4. Conferma di voler rimuovere il connettore o la connessione inserendo **`Delete`** e quindi scegli Delete (Elimina).

   Eliminando un connettore, è necessario eliminare anche tutte le connessioni create per tale connettore.

I processi che utilizzano una connessione eliminata non funzioneranno più. Puoi modificare i processi per utilizzare un datastore diverso oppure rimuoverli. Per informazioni su come eliminare un processo, consulta Eliminazione dei processi.

Se elimini un connettore, la sottoscrizione per il connettore non viene annullata in Marketplace AWS. Per rimuovere una sottoscrizione per un connettore eliminato, segui le istruzioni riportate in Annullare una sottoscrizione per un connettore.

## Annullare una sottoscrizione per un connettore

Dopo aver eliminato le connessioni e il connettore daAWS Glue Studio, puoi annullare l'abbonamento Marketplace AWS se non ti serve più il connettore.

> **Note**
>
> Se annulli la sottoscrizione a un connettore, il connettore o la connessione non vengono rimossi dall'account. Qualsiasi processo che utilizza il connettore e le connessioni correlate non sarà più in grado di utilizzare il connettore e avrà esito negativo.
> Prima di annullare l'iscrizione o sottoscrivere nuovamente un connettore Marketplace AWS, è necessario eliminare le connessioni e i connettori esistenti associati a quel Marketplace AWS prodotto.

Per annullare l'iscrizione a un connettore in Marketplace AWS

1. Accedi alla Marketplace AWS console all'indirizzo https://console.aws.amazon.com/marketplace.
2. Scegli Manage subscriptions (Gestisci sottoscrizioni).
3. Nella pagina Manage subscriptions (Gestione degli abbonamenti), scegli Manage (Gestisci) accanto alla sottoscrizione del connettore che vuoi annullare.
4. Scegli Actions (Operazioni), quindi Cancel Subscriptions (Annulla sottoscrizione).
5. Seleziona la casella di controllo per acconsentire che le istanze in esecuzione siano addebitate all'account, quindi scegli Yes, cancel subscription (Sì, annulla sottoscrizione).

## Sviluppo di connettori personalizzati

Puoi scrivere il codice che legge o scrive i dati nel datastore e formatta i dati per l'utilizzo con i processi AWS Glue Studio. Puoi creare connettori per datastore Spark, Athena e JDBC. Il codice di esempio pubblicato su GitHub fornisce una panoramica delle interfacce di base da implementare.

Per creare il codice del connettore è necessario un ambiente di sviluppo locale. Puoi usare qualsiasi IDE o anche solo un editor della riga di comando per scrivere il connettore. Esempi di ambienti di sviluppo includono:

- Un ambiente Scala locale con una libreria Maven ETL AWS Glue locale, come descritto in Sviluppo in locale con Scala nella Guida per gli sviluppatori di AWS Glue .
- IntelliJ IDE, scaricando l'IDE da. https://www.jetbrains.com/idea/

Argomenti

- Sviluppo dei connettori Spark
- Sviluppo di connettori Athena
- Sviluppo di connettori JDBC
- Esempi di utilizzo di connettori personalizzati con AWS Glue Studio
- AWS GlueSviluppo di connettori per Marketplace AWS

### Sviluppo dei connettori Spark

Puoi creare un connettore Spark con Spark DataSource API V2 (Spark 2.4) per leggere i dati.

Per creare un connettore Spark personalizzato

Segui i passaggi nella libreria di AWS Glue GitHub esempio per lo sviluppo di connettori Spark, che si trova in/.md. https://github.com/aws-samples/ aws-glue-samples tree/master/GlueCustomConnectors/development/Spark/README

## Sviluppo di connettori Athena

Puoi creare un connettore Athena utilizzabile da AWS Glue e AWS Glue Studio per eseguire query su un'origine dati personalizzata.

Per creare un connettore Athena personalizzato

Segui i passaggi nella libreria di AWS Glue GitHub esempio per lo sviluppo dei connettori Athena, che si trova in https://github.com/aws-samples/aws-glue-samples/. tree/master/GlueCustomConnectors/development/Athena

## Sviluppo di connettori JDBC

Puoi creare un connettore che utilizza JDBC per accedere ai datastore.

Per creare un connettore JDBC personalizzato

1. Installa le librerie runtime Spark AWS Glue nel tuo ambiente di sviluppo locale. Fate riferimento alle istruzioni nella libreria di AWS Glue GitHub esempio all'indirizzo https://github.com/aws-samples/aws-glue-samples/tree/master/GlueCustomConnectors/development/GlueSparkRuntime/README.md.
2. Implementa il driver JDBC responsabile del recupero dei dati dall'origine dati. Fai riferimento alla documentazione Java per Java SE 8.

   Crea un punto di ingresso all'interno del tuo codice che AWS Glue Studio utilizza per individuare il connettore. Il campo Class name (Nome classe) dovrebbe essere il percorso completo del driver JDBC.
3. Usa l'API `GlueContext` per leggere i dati con il connettore. Se necessario, gli utenti possono aggiungere altre opzioni di input nella console AWS Glue Studio per configurare la connessione all'origine dati. Per un esempio di codice che mostra come leggere e scrivere su un database JDBC con un connettore JDBC personalizzato, vedete Valori Custom e ConnectionType. Marketplace AWS

### Esempi di utilizzo di connettori personalizzati con AWS Glue Studio

Fai riferimento ai seguenti blog per esempi di utilizzo di connettori personalizzati:

- Sviluppo, test e implementazione di connettori personalizzati per gli archivi dati con AWS Glue
- Apache Hudi: Scrittura in tabelle Apache Hudi usando il connettore personalizzato AWS Glue
- Google BigQuery: migrazione dei dati da Google BigQuery ad Amazon S3 AWS Glue tramite connettori personalizzati
- Snowflake (JDBC): Esecuzione di trasformazione dati con Snowflake e AWS Glue
- SingleStore: Creazione di un sistema ETL rapido utilizzando e SingleStore AWS Glue
- Salesforce: inserisci i dati di Salesforce in Amazon S3 utilizzando il connettore personalizzato JDBC con - CData AWS Glue
- MongoDB: Creazione di processi AWS Glue Spark ETL con Amazon DocumentDB (con compatibilità MongoDB) e MongoDB
- Amazon Relational Database Service (Amazon RDS): crea offerte di lavoro AWS Glue Spark ETL introducendo i tuoi driver JDBC per Amazon RDS
- MySQL (JDBC): /.scala https://github.com/aws-samples/ aws-glue-samples blob/master/ GlueCustomConnectors/development/Spark/SparkConnectorMySQL

### AWS GlueSviluppo di connettori per Marketplace AWS

In qualità di AWS partner, puoi creare connettori personalizzati e caricarli su Marketplace AWS per venderli ai AWS Glue clienti.

Il processo per lo sviluppo del codice del connettore è lo stesso dei connettori personalizzati, ma il processo di caricamento e verifica del codice del connettore è più dettagliato. Consulta le istruzioni contenute nella sezione Creazione di connettori Marketplace AWS disponibili sul GitHub sito Web.

## Restrizioni per l'utilizzo di connettori e connessioni in AWS Glue Studio

Quando utilizzi connettori personalizzati o connettori di Marketplace AWS, prendi nota delle seguenti restrizioni:

- L'API TestConnection non è supportata con le connessioni create per i connettori personalizzati.
- La crittografia delle password di connessione al catalogo dati non è supportata con connettori personalizzati.

- Non è possibile utilizzare i segnalibri di processo se specifichi un predicato di filtro per un nodo di origine dati che utilizza un connettore JDBC.
- La creazione di una connessione Marketplace non è supportata al di fuori AWS Glue Studio dell'interfaccia utente.

# Test di una connessione AWS Glue

Come best practice, prima di utilizzare una connessione AWS Glue in un processo ETL, utilizza la console AWS Glue per testare la connessione. AWS Glue utilizza i parametri nella connessione per confermare che può accedere al datastore e segnala eventuali errori. Per informazioni sulle connessioni AWS Glue, consulta Connessione ai dati.

Per testare una connessione AWS Glue

1. Accedi a Console di gestione AWS e apri la AWS Glue console all'indirizzo https://console.aws.amazon.com/glue/.
2. Nel riquadro di navigazione, in Catalogo dati, seleziona Connessioni. È inoltre possibile scegliere Connessioni dati sopra Catalogo dati nel riquadro di navigazione.
3. In Connessioni, seleziona la casella di controllo accanto alla connessione desiderata, quindi scegli Operazioni. Nel menu a discesa, seleziona Testa connessione.
4. Nella finestra di dialogo Test connection, seleziona un ruolo o scegli Crea ruolo IAM per accedere alla console AWS Identity and Access Management (IAM) e creare un nuovo ruolo. Il ruolo deve disporre delle autorizzazioni per il datastore.
5. Scegli Conferma.

   Il test inizia e può richiedere diversi minuti per completare. Se il test fallisce, scegli Risoluzione dei problemi per visualizzare i passaggi per risolvere il problema.
6. Scegli Registri per visualizzare i log in. CloudWatch Devi disporre delle autorizzazioni IAM necessarie per visualizzare i log. Per ulteriori informazioni, consulta AWS Managed (Predefined) Policies for CloudWatch Logs nella Amazon CloudWatch Logs User Guide.

# Configurazione delle AWS chiamate in modo che passino attraverso il tuo VPC

Il parametro job speciale ti `disable-proxy-v2` consente di indirizzare le chiamate a servizi come Amazon S3 e AWS Glue tramite il CloudWatch tuo VPC. Per impostazione predefinita, AWS Glue utilizza un proxy locale per inviare traffico attraverso il AWS Glue VPC per scaricare script e librerie da Amazon S3, inviare richieste CloudWatch per la pubblicazione di log e metriche e inviare richieste di accesso ai cataloghi di dati. AWS Glue Questo proxy consente al processo di funzionare normalmente anche se il tuo VPC non configura un percorso corretto verso altri AWS servizi, come Amazon S3 CloudWatch, e. AWS Glue AWS Glue ora offre un parametro per disattivare questo comportamento. Per ulteriori informazioni, vedere Parametri Job usati da AWS Glue. AWS Glue continuerà a utilizzare il proxy locale per pubblicare CloudWatch i registri dei AWS Glue lavori.

> **Note**
>
> - Questa funzionalità è supportata per i AWS Glue lavori con la AWS Glue versione 2.0 e successive. Quando utilizzi questa caratteristica, devi assicurarti che il tuo VPC abbia configurato un percorso verso Amazon S3, tramite un endpoint NAT o VPC di servizio.
> - Il parametro del processo obsoleto `disable-proxy` inoltra le tue chiamate ad Amazon S3 solo per il download di script e librerie tramite il tuo VPC. Consigliamo invece di utilizzare il nuovo parametro `disable-proxy-v2`.

Esempio di utilizzo

Crea un AWS Glue lavoro con`disable-proxy-v2`:

```
aws glue create-job \
    --name no-proxy-job \
    --role GlueDefaultRole \
    --command "Name=glueetl,ScriptLocation=s3://my-bucket/glue-script.py" \
    --connections Connections="traffic-monitored-connection" \
    --default-arguments '{"--disable-proxy-v2" : "true"}'
```

# La connessione a un archivio dati JDBC in un VPC

In genere, le risorse vengono create in Amazon Virtual Private Cloud (Amazon VPC) per impedirne l'accesso tramite rete internet pubblica. Per impostazione predefinita, non è AWS Glue possibile accedere alle risorse all'interno di un VPC. Per consentire l'accesso AWS Glue alle risorse all'interno del VPC, è necessario fornire ulteriori informazioni di configurazione specifiche del VPC che includono la sottorete VPC e il gruppo di sicurezza. IDs IDs AWS Glue utilizza queste informazioni per configurare interfacce di rete elastiche che consentono alla tua funzione di connettersi in modo sicuro ad altre risorse nel tuo VPC privato.

Quando utilizzi un endpoint VPC, aggiungilo alla tabella di routing. Per ulteriori informazioni, consulta le pagine Creating an interface VPC endpoint for AWS Glue e Prerequisiti.

Quando utilizzi la crittografia in Catalogo dati, crea l'endpoint dell'interfaccia KMS e aggiungilo alla tabella di routing. Per ulteriori informazioni, consulta la pagina Creating a VPC endpoint for AWS KMS.

## Accesso a dati VPC mediante interfacce di rete elastiche

Quando AWS Glue si connette a un data store JDBC in un VPC, AWS Glue crea un'interfaccia di rete elastica (con il prefisso`Glue_`) nell'account per accedere ai dati VPC. Non puoi eliminare questa interfaccia di rete finché è collegata a. AWS Glue Come parte della creazione dell'elastic network interface, AWS Glue associa uno o più gruppi di sicurezza ad essa. AWS Glue Per consentire la creazione dell'interfaccia di rete, i gruppi di sicurezza associati alla risorsa devono consentire l'accesso in entrata con una regola di origine. Questa regola contiene un gruppo di sicurezza associato alla risorsa. In questo modo, l'interfaccia di rete elastica accede all'archivio dati con lo stesso gruppo di sicurezza.

AWS Glue Per consentire la comunicazione con i suoi componenti, specifica un gruppo di sicurezza con una regola di ingresso autoreferenziale per tutte le porte TCP. Se crei una regola autoreferenziata, puoi limitare l'origine allo stesso gruppo di sicurezza nel VPC e non aprirla a tutte le reti. Il gruppo di sicurezza predefinito per il tuo VPC potrebbe già avere una regola autoreferenziata in entrata per `ALL Traffic`.

Puoi creare regole nella console Amazon VPC. Per aggiornare le impostazioni della regola tramite Console di gestione AWS, passa alla console VPC (https://console.aws.amazon.com/vpc/) e seleziona il gruppo di sicurezza appropriato. Specifica la regola in entrata per `ALL TCP` così da avere come origine lo stesso nome gruppo di sicurezza. Per ulteriori informazioni sulle regole dei gruppi di sicurezza, consulta Gruppi di sicurezza per il tuo VPC.

A ogni elastic network interface viene assegnato un indirizzo IP privato dall'intervallo di indirizzi IP nelle sottoreti specificate. All'interfaccia di rete non viene assegnato alcun indirizzo IP pubblico. AWS Glue richiede l'accesso a Internet (ad esempio, per accedere a AWS servizi che non dispongono di endpoint VPC). È·possibile configurare un'istanza NAT (Network Address Translation) all'interno del VPC oppure puoi usare il gateway NAT Amazon VPC. Per ulteriori informazioni, consulta Gateway NAT nella Guida per l'utente di Amazon VPC. Non puoi utilizzare direttamente un gateway internet collegato al VPC come percorso nella tabella di instradamento della sottorete perché ciò richiede che l'interfaccia di rete abbia indirizzi IP pubblici.

Gli attributi di rete VPC `enableDnsHostnames` e `enableDnsSupport` devono essere impostati su true. Per ulteriori informazioni, consulta Utilizzo del DNS con il tuo VPC.

> **Important**
>
> Non inserire l'archivio dati in una sottorete pubblica o in una sottorete privata che non abbia accesso a Internet. Collegalo invece solo alle sottoreti private che hanno accesso a Internet tramite un'istanza NAT o un gateway NAT Amazon VPC.

## Proprietà dell'interfaccia di rete elastica

Per creare l'interfaccia di rete elastica, sono necessarie le seguenti proprietà:

VPC

Il nome del VPC che contiene l'archivio dati.

Sottorete

Il nome della sottorete nel VPC che contiene l'archivio dati.

Gruppi di sicurezza

Gruppi di sicurezza associati all'archivio dati. AWS Glue associa questi gruppi di sicurezza all'interfaccia di rete elastica collegata alla sottorete VPC. Per consentire ai componenti di AWS Glue di comunicare e di impedire l'accesso da altre reti, almeno un gruppo di sicurezza a scelta deve specificare una regola in entrata autoreferenziale per tutte le porte TCP.

Per informazioni sulla gestione di un VPC con Amazon Redshift, consulta Gestione di cluster in Amazon Virtual Private Cloud (VPC).

Per ulteriori informazioni sulla gestione di un VPC con Amazon Relational Database Service (Amazon RDS), consulta Uso di un'istanza database Amazon RDS in un VPC.

## Utilizzo di una connessione MongoDB o MongoDB Atlas

Dopo aver creato una connessione per MongoDB o MongoDB Atlas, puoi utilizzarla nel processo ETL. Si crea una tabella in AWS Glue Data Catalog e si specifica la connessione MongoDB o MongoDB Atlas per l'attributo della tabella. `connection`

AWS Glue memorizza la connessione `url` e le credenziali nella connessione MongoDB. I formati di URI di connessione sono i seguenti:

- Per MongoDB: mongodb://host:port/database. L'host può essere un nome host, un indirizzo IP o un socket di dominio UNIX. Se la stringa di connessione non specifica una porta, utilizza la porta MongoDB predefinita, 27017.
- Per MongoDB Atlas: mongodb+srv://server.example.com/database. L'host può essere un nome host che corrisponde a un record DNS SRV. Il formato SRV non richiede una porta e utilizzerà la porta MongoDB predefinita, 27017.

Altre opzioni possono essere specificate nello script di processo. Per ulteriori informazioni, consulta the section called “Connessione MongoDB”.

## Crawling di un archivio di dati Amazon S3 utilizzando un endpoint VPC

Per motivi di sicurezza, audit o controllo, puoi consentire l'accesso all'archivio dati Amazon S3 o alle tabelle Catalogo dati supportate da Amazon S3 solo tramite un ambiente Amazon Virtual Private Cloud (Amazon VPC). In questo argomento viene descritto come creare e testare una connessione all'archivio dati Amazon S3 o alle tabelle Catalogo dati supportate da Amazon S3 in un endpoint VPC utilizzando il tipo di connessione `Network`.

Esegui le attività seguenti per eseguire un crawler nell'archivio dati:

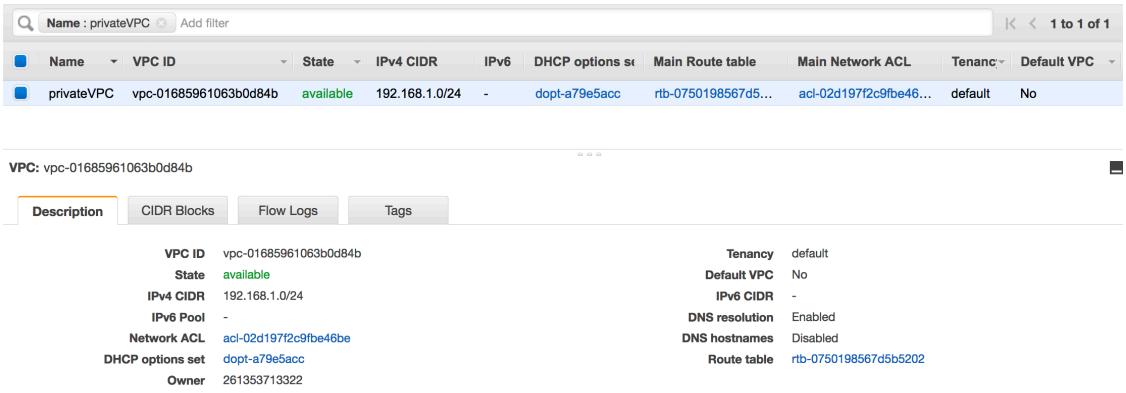
- the section called “Prerequisiti”
- the section called “Creazione della connessione ad Amazon S3”
- the section called “Test della connessione ad Amazon S3”

- the section called “Creazione di un crawler per un archivio di dati Amazon S3”
- the section called “Esecuzione di un crawler”

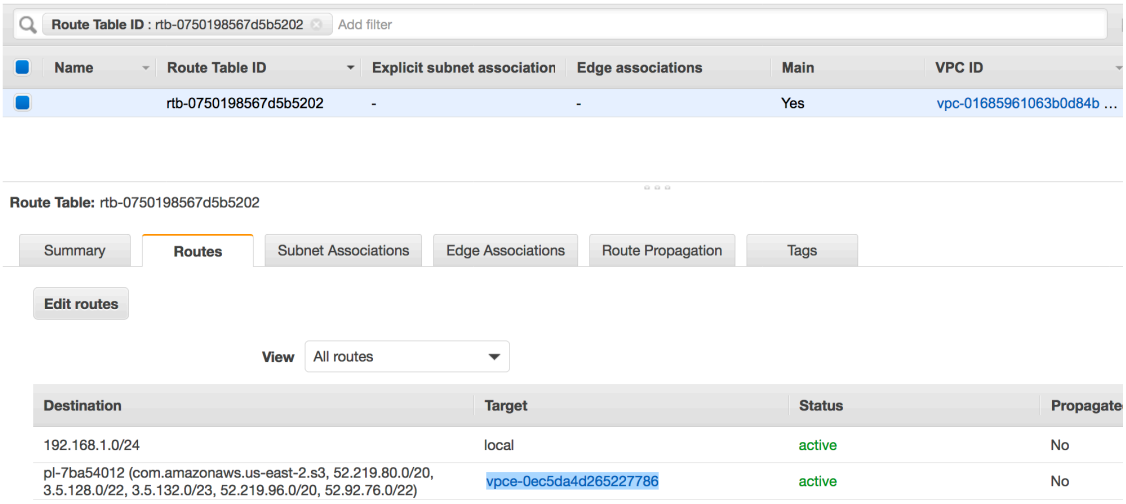
## Prerequisiti

Verifica di aver soddisfatto questi prerequisiti per configurare il datastore Amazon S3 affinché vi si possa accedere solo tramite un ambiente Amazon Virtual Private Cloud (Amazon VPC).

- Un VPC configurato. Ad esempio: vpc-01685961063b0d84b. Per ulteriori informazioni, consulta le Nozioni di base su Amazon VPC nella Guida per l'utente di Amazon VPC.
- Un endpoint Amazon S3 collegato al VPC. Ad esempio: vpc-01685961063b0d84b. Per ulteriori informazioni, consulta Endpoint per Amazon S3 nella Guida per l'utente di Amazon VPC.




- Una voce route che punta all'endpoint VPC. Ad esempio vpce-0ec5da4d265227786 nella tabella di routing utilizzata dall'endpoint VPC (vpce-0ec5da4d265227786).




- Una lista di controllo degli accessi di rete collegata al VPC consente il traffico.

- Un gruppo di sicurezza collegato al VPC consente il traffico.

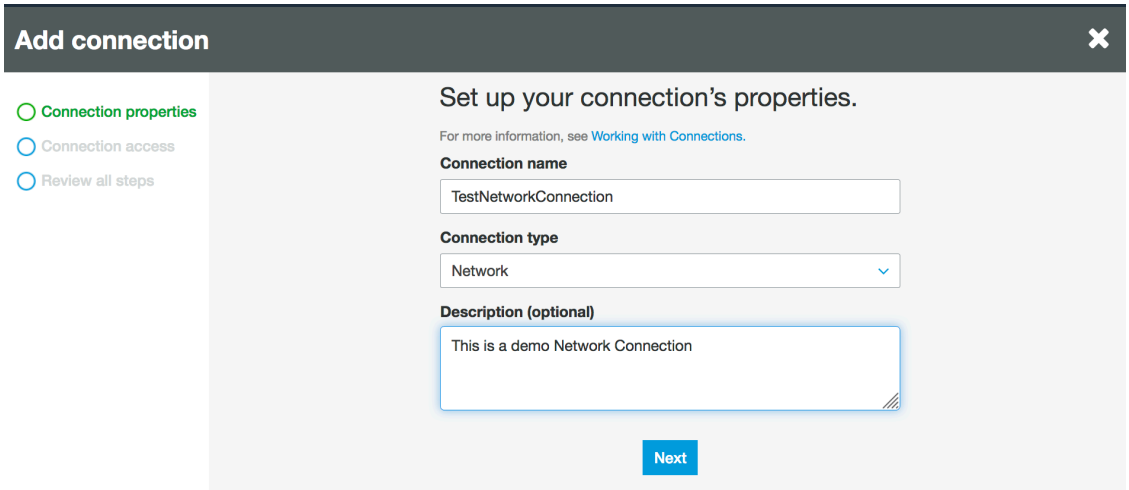
## Creazione della connessione ad Amazon S3

In genere, le risorse vengono create in Amazon Virtual Private Cloud (Amazon VPC) per impedirne l'accesso tramite rete internet pubblica. Per impostazione predefinita, non è AWS Glue possibile accedere alle risorse all'interno di un VPC. Per consentire l'accesso AWS Glue alle risorse all'interno del VPC, è necessario fornire ulteriori informazioni di configurazione specifiche del VPC che includono la sottorete VPC e il gruppo di sicurezza. IDs IDs Per creare una connessione `Network`, è necessario specificare le informazioni seguenti:

- Un ID VPC
- Una sottorete all'interno del VPC
- Un gruppo di sicurezza

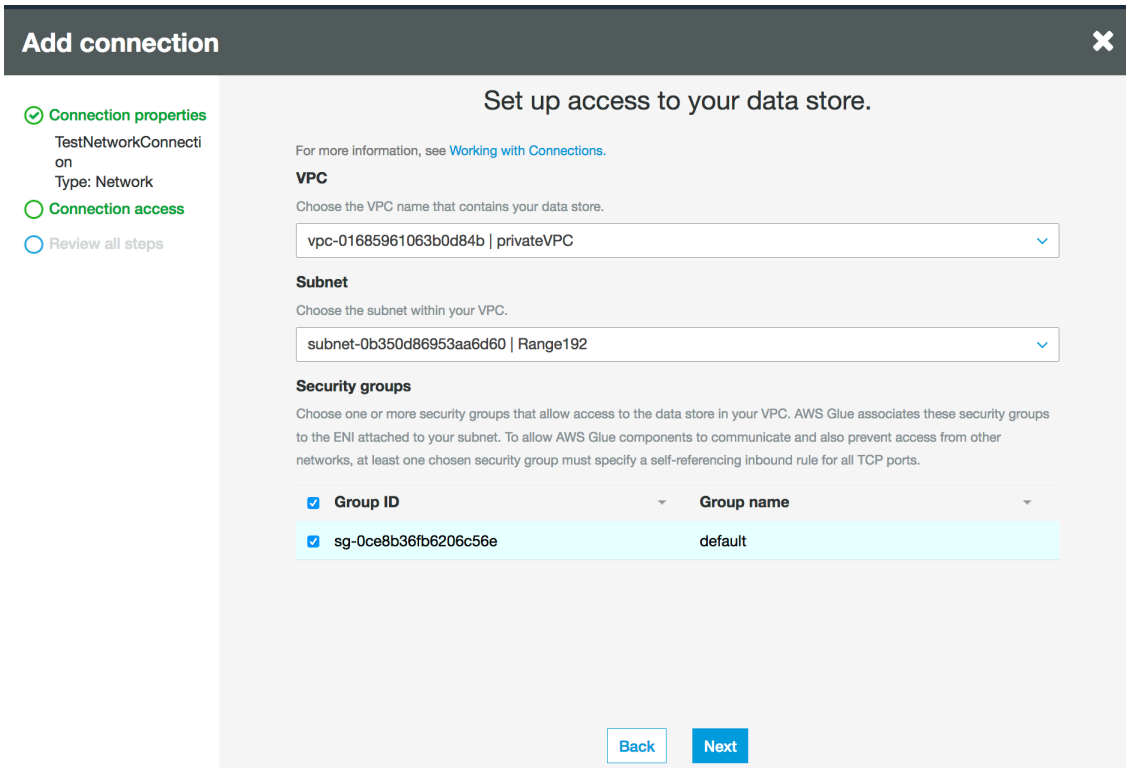
Per impostare una connessione `Network`:

1. Scegli Add connection (Aggiungi connessione) nel pannello di navigazione della console AWS Glue .
2. Inserisci il nome della connessione e scegli Network (Rete) come tipo di connessione. Scegli Next (Successivo).

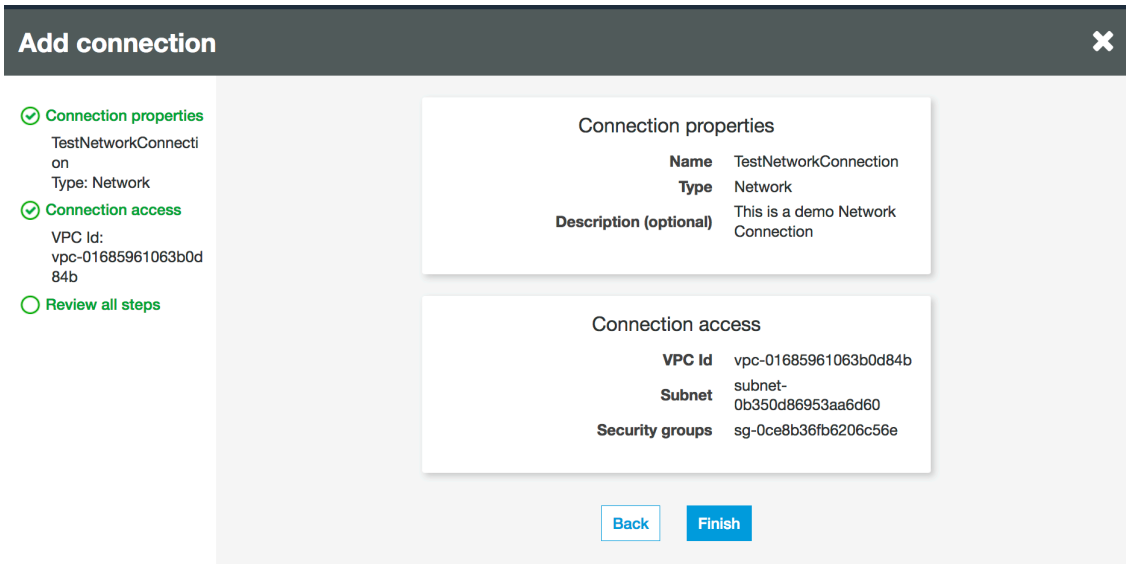



3. Configura le informazioni su VPC, sottorete e gruppi di sicurezza.
   - VPC: scegli il nome del VPC che contiene l'archivio dati.
   - Sottorete: scegli una sottorete nel VPC.

- Gruppi di sicurezza: scegli uno o più gruppi di protezione che consentono l'accesso all'archivio dati nel VPC.




4. Seleziona Next (Successivo).
5. Verifica le informazioni di connessione e scegli Finish (Termina).



# Test della connessione ad Amazon S3

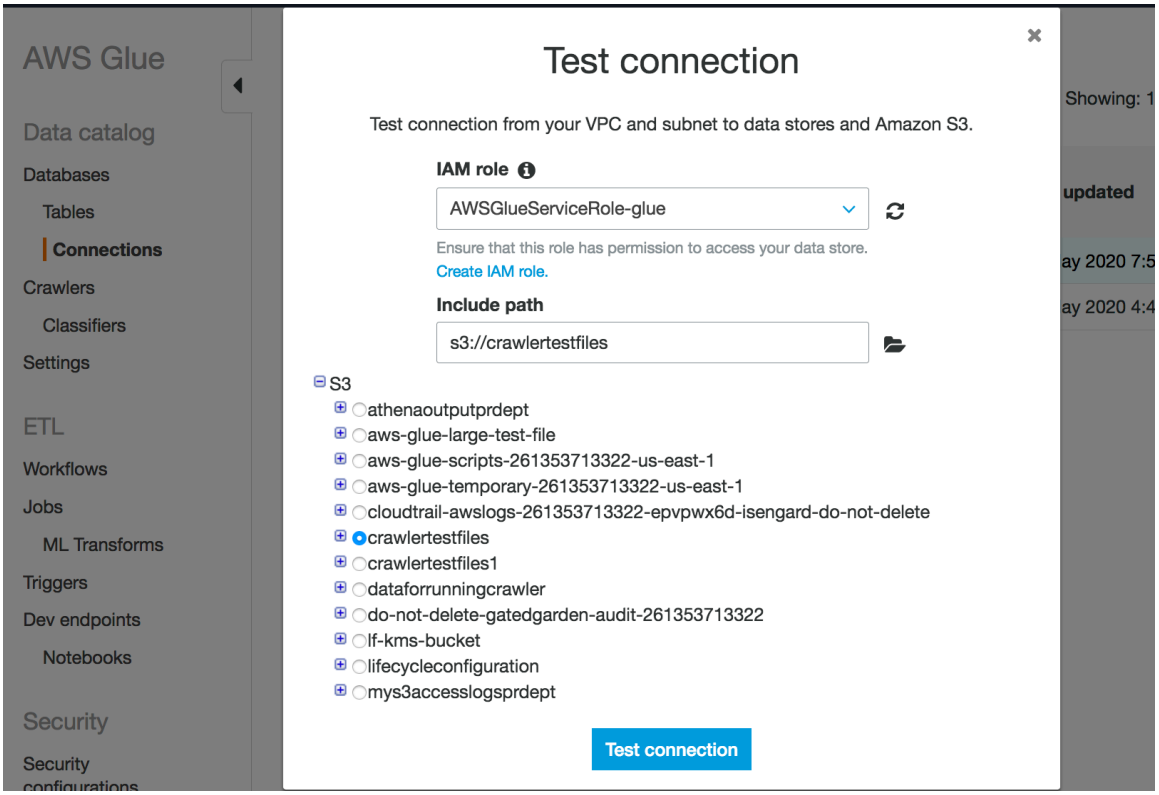
Una volta creata la connessione di `Network`, puoi testare la connettività al tuo archivio dati Amazon S3 in un endpoint VPC.

Durante il test di una connessione possono verificarsi i seguenti errori:

- ERRORE DI CONNESSIONE A INTERNET: indica un problema di connessione a Internet
- ERRORE DI BUCKET NON VALIDO: indica un problema con il bucket Amazon S3
- ERRORE DI CONNESSIONE S3: indica un errore di connessione ad Amazon S3
- ERRORE DI TIPO DI CONNESSIONE: indica che il tipo di connessione non ha il valore previsto, `NETWORK`
- TIPO DI TEST DI CONNESSIONE NON VALIDO: indica un problema con il tipo di test della connessione di rete
- DESTINAZIONE NON VALIDA: indica che il bucket Amazon S3 non è stato specificato correttamente

Per testare una connessione `Network`:

1. Seleziona la connessione Network (Rete) nella console AWS Glue .
2. Scegli Test Connection (Connessione di prova).
3. Scegli il ruolo IAM creato nel passaggio precedente e specifica un bucket Amazon S3.
4. Per verificare la connessione, scegli Test connection (Testa connessione). Potrebbero essere necessari alcuni istanti prima che il risultato venga visualizzato.




Se viene visualizzato un errore, controlla quanto segue:

- I privilegi corretti vengono forniti al ruolo selezionato.
- Viene fornito il bucket Amazon S3 corretto.
- I gruppi di sicurezza e la lista di controllo degli accessi di rete consentono il traffico in entrata e in uscita necessario.
- Il VPC specificato è connesso a un endpoint VPC Amazon S3.

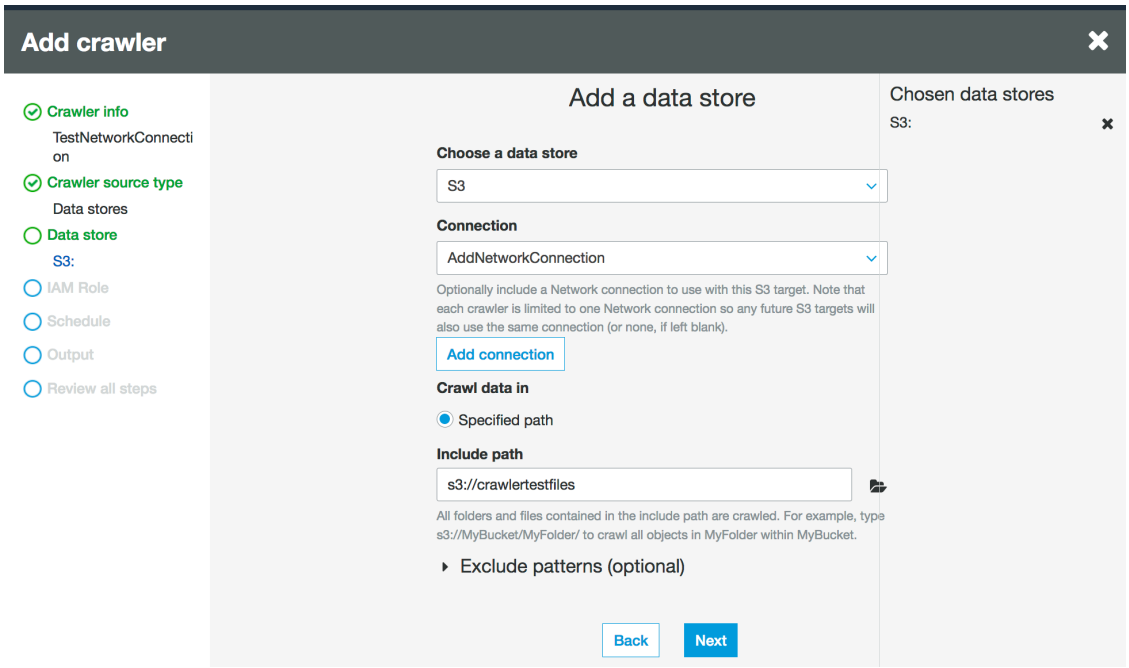
Dopo aver testato correttamente la connessione, è possibile creare un crawler.

## Creazione di un crawler per un archivio di dati Amazon S3

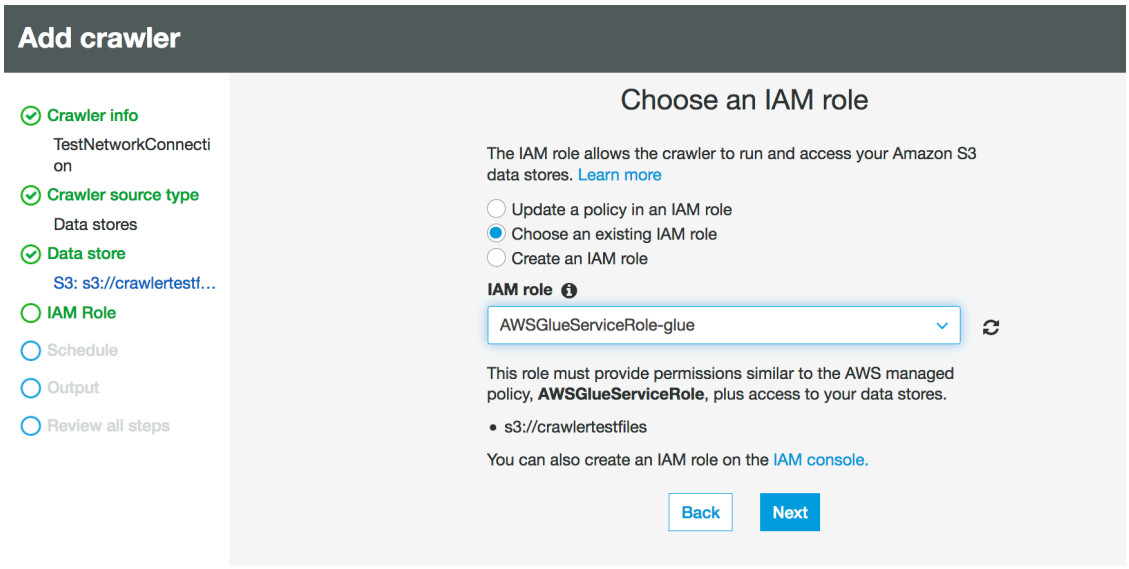
Ora è possibile creare un crawler che specifichi la connessione di `Network` creata. Per ulteriori dettagli sulla creazione di un crawler, consulta Configurazione di un crawler.

1. Inizia selezionando Crawler nel riquadro di navigazione della console. AWS Glue
2. Scegli Add crawler (Aggiungi crawler).
3. Specifica il nome del crawler, quindi scegli Next (Avanti).

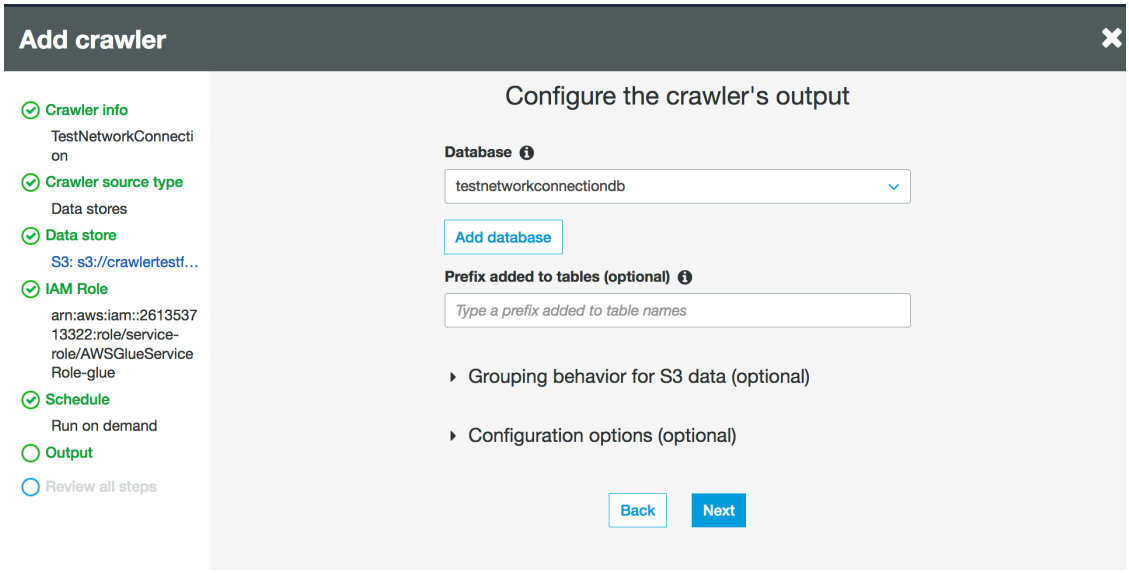
4. Quando viene richiesto di specificare l'origine dati, seleziona S3 e specifica il prefisso del bucket Amazon S3 e la connessione creata in precedenza.




5. Se necessario, aggiungi un altro archivio dati sulla stessa connessione di rete.
6. Scegli il ruolo IAM. Il ruolo IAM deve consentire l'accesso al servizio AWS Glue e al bucket Amazon S3. Per ulteriori informazioni, consulta the section called “Configurazione di un crawler”.




7. Definisci la pianificazione per il crawler.
8. Scegli un database esistente nel catalogo dati oppure crea una nuova voce del database.

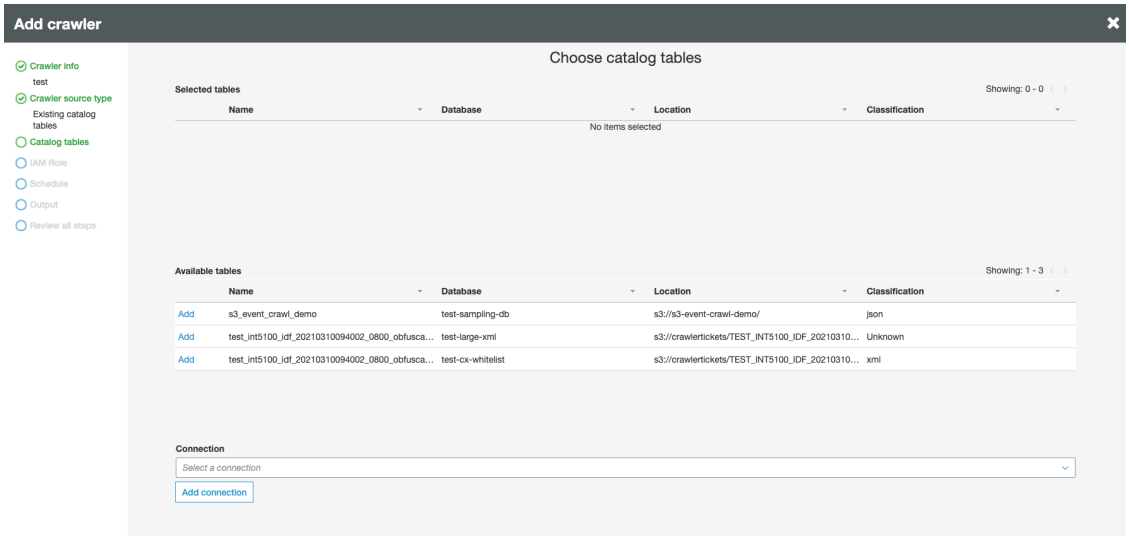



9. Completa la configurazione rimanente.

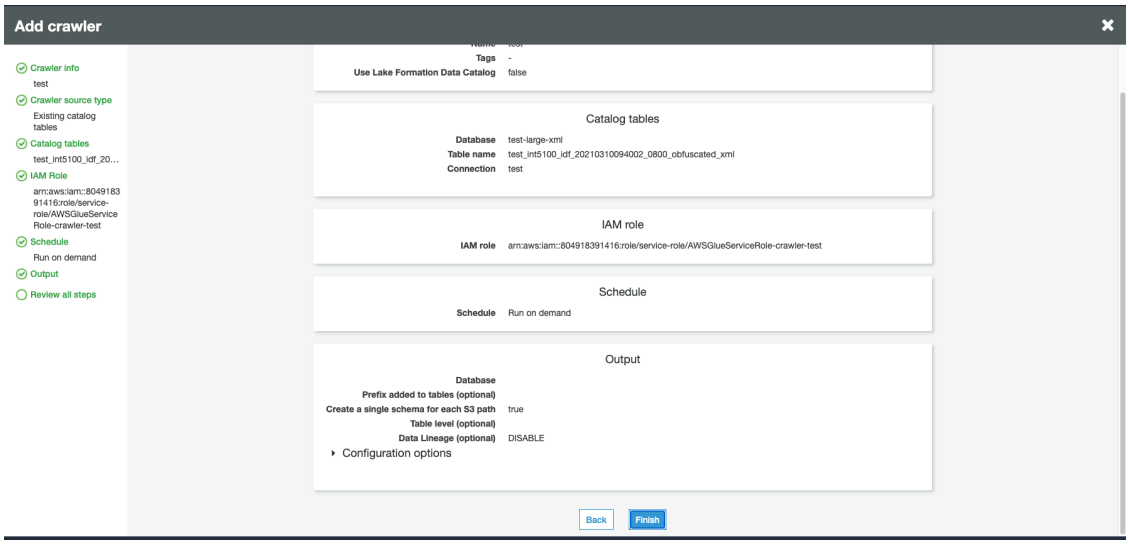
## Creazione di un crawler per le tabelle del Catalogo dati supportate da Amazon S3

Ora è possibile creare un crawler che specifichi la connessione di `Network` creata e il tipo di fonte del Catalogo. Per ulteriori dettagli sulla creazione di un crawler, consulta Configurazione di un crawler.

1. Inizia scegliendo Crawler nel riquadro di navigazione della console. AWS Glue
2. Scegli Add crawler (Aggiungi crawler).
3. Specifica il nome del crawler, quindi scegli Next (Avanti).
4. Quando viene richiesto il tipo di origine crawler, scegliere Existing catalog tables (Tabelle di catalogo esistenti) e specificare le tabelle di catalogo esistenti da eseguire per il crawling dall'elenco delle tabelle disponibili.

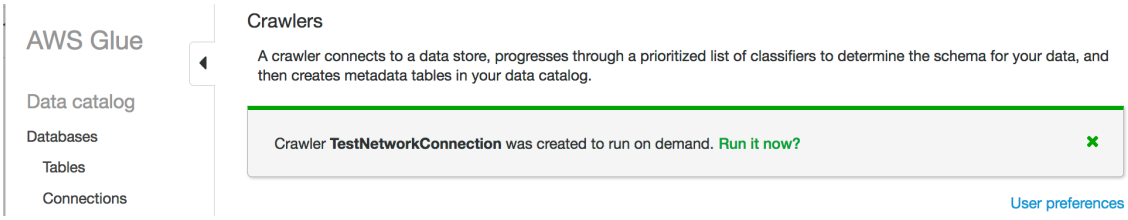



5. Scegli il ruolo IAM. Il ruolo IAM deve consentire l'accesso al servizio AWS Glue e al bucket Amazon S3. Per ulteriori informazioni, consulta the section called "Configurazione di un crawler".
6. Definisci la pianificazione per il crawler.
7. Scegli un database esistente nel catalogo dati oppure crea una nuova voce del database.
8. Completa la configurazione rimanente e rivedi i passaggi.




# Esecuzione di un crawler

Esegui il crawler.




## Risoluzione dei problemi

Per la risoluzione dei problemi relativi ai bucket Amazon S3 che utilizzano un gateway VPC, consulta l'argomento relativo alle difficoltà di connessione a un bucket S3 usando un endpoint VPC gateway.

# Lavorare con AWS Lake Formation dati protetti

## Accesso completo alla tabella

AWS Glue supporta l'accesso completo alla tabella (FTA) per le AWS Lake Formation tabelle protette. Ciò consente ai processi ETL di leggere e scrivere dati con autorizzazioni complete per le tabelle.

### Prerequisiti

- Autorizzazioni e ruoli IAM appropriati
- AWS Lake Formation configurato per il tuo catalogo di dati
- Tipi di tabelle compatibili (Hive o Iceberg)

### Considerazioni chiave

#### Autorizzazioni richieste

- Autorizzazione IAM `lakeformation:GetDataAccess`
- AWS Lake Formation autorizzazioni per le tabelle
- Autorizzazioni di accesso dei bucket di Amazon S3

#### Tipi di tabella supportati

- Tabelle Hive
- Tabelle Iceberg

Limitazioni

- Non compatibile con Spark Streaming
- Non si può utilizzare contemporaneamente al controllo granulare degli accessi
- Non supporta le tabelle Delta o Hudi

### Best practice

1. Assicurarsi che i ruoli IAM siano configurati correttamente prima dell'esecuzione del processo
2. Testare le autorizzazioni di accesso in un ambiente di sviluppo
3. Monitorare i registri di esecuzione dei processi per problemi relativi alle autorizzazioni
4. Conservare una documentazione chiara dei modelli di accesso

### Risoluzione dei problemi

I problemi comuni includono:

- Autorizzazioni IAM mancanti
- Configurazione errata AWS Lake Formation
- Problemi di compatibilità del tipo di tabella

Per istruzioni complete sulla configurazione e dettagli di configurazione, consulta Utilizzo AWS Lake Formation con accesso completo alla tabella.

## Risoluzione dei problemi di connessione in AWS Glue

Quando un AWS Glue crawler o un processo utilizza le proprietà di connessione per accedere a un data store, è possibile che si verifichino errori durante il tentativo di connessione. AWS Glueutilizza indirizzi IP privati nella sottorete quando crea interfacce di rete elastiche nel cloud privato virtuale (VPC) e nella sottorete specificati. I gruppi di sicurezza specificati nella connessione vengono applicati a ciascuna delle interfacce di rete elastiche. Verifica se i gruppi di sicurezza consentono l'accesso in uscita e la connettività al cluster database.

Inoltre, Apache Spark richiede la connettività bidirezionale tra nodi driver ed executor. Uno dei gruppi di sicurezza deve consentire le regole in ingresso su tutte le porte TCP. Puoi evitare che vengano

aperte a tutti limitando l'origine del gruppo di sicurezza a sé stesso con un gruppo di sicurezza autoreferenziata.

Di seguito sono elencate alcune operazioni tipiche che puoi eseguire per risolvere i problemi di connessione:

- Controlla l'indirizzo di porta della tua connessione.
- Controlla la stringa del nome utente e della password nella connessione o nel segreto.
- Per un archivio dati JDBC, verifica che consenta le connessioni in entrata.
- Verifica che il tuo archivio dati sia accessibile all'interno di VPC.
- Se memorizzi le credenziali di connessione utilizzando Gestione dei segreti AWS, assicurati che il tuo ruolo IAM for AWS Glue disponga dell'autorizzazione ad accedere al tuo segreto. Per ulteriori informazioni, consulta la pagina Esempio: Autorizzazione per recuperare valori segreti nella Guida per l'utente di Gestione dei segreti AWS . In base alla configurazione della rete, potrebbe essere necessario anche creare un endpoint VPC per stabilire una connessione privata tra il VPC e Secrets Manager. Per ulteriori informazioni, consulta Utilizzo di un Gestione dei segreti AWS endpoint VPC.

# Tutorial: utilizzo del AWS Glue connettore per Elasticsearch

Elasticsearch è un diffuso motore di ricerca e analisi dei dati open source per casi d'uso come analisi dei dati dei log, monitoraggio delle applicazioni in tempo reale e analisi dei dati di clickstream. Puoi utilizzarlo OpenSearch come archivio dati per i tuoi lavori di estrazione, trasformazione e caricamento (ETL) configurando il Connector for Elasticsearch in. AWS Glue AWS Glue Studio Questo connettore è disponibile gratuitamente da Marketplace AWS.

> **Note**
>
> Marketplace AWS Elasticsearch Spark Connector è diventato obsoleto. Utilizza invece un Connettore AWS Glue per Elasticsearch.

In questo tutorial, mostreremo come connetterti ai tuoi nodi Amazon OpenSearch Service con un numero minimo di passaggi.

Argomenti

- Prerequisiti

- Passaggio 1: (Facoltativo) Crea un AWS segreto per le informazioni sul OpenSearch cluster
- Fase 2: sottoscrizione al connettore
- Fase 3: Attivare il connettore AWS Glue Studio e creare una connessione
- Fase 4: configurazione di un ruolo IAM per il processo ETL
- Passaggio 5: Creare un lavoro che utilizzi la connessione OpenSearch
- Fase 6: esecuzione del processo


## Prerequisiti

Per utilizzare questo tutorial, è necessario disporre di quanto segue:

- Accesso a AWS Glue Studio
- Accesso a un OpenSearch cluster nel AWS cloud
- (Facoltativo) Accesso a Gestione dei segreti AWS.

## Passaggio 1: (Facoltativo) Crea un AWS segreto per le informazioni sul OpenSearch cluster

Per archiviare e utilizzare in modo sicuro le credenziali di connessione, salvale in Gestione dei segreti AWS. Il segreto creato verrà utilizzato più avanti nel tutorial dalla connessione. Le coppie chiave-valore delle credenziali verranno inserite nel AWS Glue Connector for Elasticsearch come normali opzioni di connessione.

Per ulteriori informazioni sulla creazione dei segreti, consulta Creazione e gestione di segreti con Gestione dei segreti AWS nella Guida per l'utente di Gestione dei segreti AWS .

Per creare un segreto AWS

1. Accedi alla console Gestione dei segreti AWS.
2. Nella pagina di introduzione del servizio o nella pagina dell'elenco Secrets (Segreti), scegli Store a new secret (Archivia un nuovo segreto).
3. Nella pagina Store a new secret (Archivia un nuovo segreto), scegli Other type of secret (Altro tipo di segreto). Questa opzione indica che devi fornire la struttura e i dettagli del tuo segreto.
4. Aggiungi una coppia chiave e valore per il nome utente del OpenSearch cluster. Esempio:

   ```
   es.net.http.auth.user: username
   ```

5. Scegli + Add row (+ Aggiungi riga) e inserisci un'altra coppia chiave-valore per la password. Esempio:

   `es.net.http.auth.pass`: *`password`*
6. Seleziona Next (Successivo).
7. Immetti il nome di un segreto. Ad esempio: my-es-secret. Facoltativamente, puoi inserire una descrizione.

   Registra il nome del segreto, che viene utilizzato più avanti in questo tutorial, quindi scegli Next (Successivo).
8. Scegli di nuovo Next (Successivo), quindi scegli Store (Archivia) per creare il segreto.

### Approfondimenti

Fase 2: sottoscrizione al connettore

## Fase 2: sottoscrizione al connettore

Il AWS Glue Connector for Elasticsearch è disponibile gratuitamente da. Marketplace AWS

Per abbonarsi al AWS Glue Connector for Elasticsearch su Marketplace AWS

1. Se non hai già configurato il tuo AWS account per l'utilizzo di License Manager, procedi come segue:

   a. Apri la AWS License Manager console in https://console.aws.amazon.com/license-manager.
   b. Scegli Create customer managed license (Crea una licenza gestita dal cliente).
   c. Nella finestra delle autorizzazioni IAM (configurazione unica), scegli Concedo le autorizzazioni richieste, quindi scegli Concedi AWS License Manager le autorizzazioni.

      Se non vedi questa finestra, hai già configurato le autorizzazioni necessarie.
2. Apri la console AWS Glue Studio all'indirizzo https://console.aws.amazon.com/gluestudio/.
3. Nella AWS Glue Studio console, espandi l'icona del menu (☰), quindi scegli Connettori nel riquadro di navigazione.
4. Nella pagina Connectors (Connectors), scegli Go to Marketplace AWS (Vai su Marketplace AWS).

5. Nella sezione Cerca AWS Glue Studio prodotti Marketplace AWS, inserisci AWS Glue Connector for Elasticsearch nel campo di ricerca, quindi premi Invio.
6. Seleziona il nome del connettore, Connettore AWS Glue per Elasticsearch.
7. Nella pagina prodotto del connettore, utilizza le schede per visualizzare le relative informazioni. Quando vuoi continuare, scegli Continue to Subscribe (Continua con la sottoscrizione).
8. Rivedi i termini di utilizzo del software. Fai clic su Accetta termini.
9. Al termine del processo di sottoscrizione, verrà visualizzata una notifica: "Grazie per esserti registrato a questo prodotto! Adesso puoi configurare il software". Sopra il banner ci sarà il pulsante Passa alla configurazione. Scegli Continue to Configuration (Passa alla configurazione).
10. Scegli l'opzione Fulfillment (Compimento) sulla pagina Configure this software (Configura questo software). Puoi scegliere tra AWS Glue 1.0/2.0 o 3.0. AWS Glue Quindi, scegli Continue to Launch (Continua con l'avvio).

### Approfondimenti

Fase 3: Attivare il connettore AWS Glue Studio e creare una connessione

## Fase 3: Attivare il connettore AWS Glue Studio e creare una connessione

Dopo aver scelto Continua all'avvio, viene visualizzata la pagina Avvia questo software in Marketplace AWS. Dopo aver utilizzato il collegamento per attivare il connettore AWS Glue Studio, si crea una connessione.

Per distribuire il connettore e creare una connessione in AWS Glue Studio

1. Nella pagina Avvia questo software nella Marketplace AWS console, scegli Istruzioni per l'uso, quindi scegli il link nella finestra che appare.

   Il browser viene reindirizzato alla pagina Crea connessione al marketplace della AWS Glue Studio console.
2. Inserisci un nome per la connessione. Ad esempio: my-es-connection.
3. Nella sezione Connection access (Accesso alla connessione), per Connection credential type (Tipo di credenziali di connessione), scegli User name and password (Nome utente e password).
4. Nel campo AWS secret (Segreto AWS ), inserisci il nome del tuo segreto. Ad esempio: my-es-secret.

5. Nella sezione Opzioni di rete, inserisci le informazioni sul VPC per connetterti al OpenSearch cluster.
6. Scegli Create connection and activate connector (Crea una connessione e attiva il connettore).

### Approfondimenti

Fase 4: configurazione di un ruolo IAM per il processo ETL

## Fase 4: configurazione di un ruolo IAM per il processo ETL

Quando si crea il job AWS Glue ETL, si specifica un ruolo AWS Identity and Access Management (IAM) per il job da utilizzare. Il ruolo deve concedere l'accesso a tutte le risorse utilizzate dal processo, incluso Amazon S3 (per qualsiasi origine, destinazione, script, file di driver e directory temporanee) e anche agli oggetti. AWS Glue Data Catalog

Il ruolo IAM assunto per il job AWS Glue ETL deve inoltre avere accesso al segreto creato nella sezione precedente. Per impostazione predefinita, il ruolo gestito da AWS `AWSGlueServiceRole` non ha accesso al segreto. Per impostare il controllo dell'accesso per i tuoi segreti, consulta Autenticazione e controllo degli accessi per Gestione dei segreti AWS e Limitazione dell'accesso a segreti specifici.

Per configurare un ruolo IAM per il processo ETL

1. Configura le autorizzazioni descritte in the section called "Esaminare le autorizzazioni IAM necessarie per i processi ETL".
2. Configura le autorizzazioni aggiuntive necessarie quando usi i connettori con AWS Glue Studio, come descritto in. the section called "Autorizzazioni richieste per l'utilizzo dei connettori"

### Approfondimenti

Passaggio 5: Creare un lavoro che utilizzi la connessione OpenSearch

## Passaggio 5: Creare un lavoro che utilizzi la connessione OpenSearch

Dopo aver creato un ruolo per il tuo job ETL, puoi crearne uno AWS Glue Studio che utilizzi la connessione e il connettore per Open Spark. ElasticSearch

Se il processo viene eseguito all'interno di un Amazon Virtual Private Cloud (Amazon VPC), verifica che questo sia configurato correttamente. Per ulteriori informazioni, consulta the section called "Configurazione di un VPC per il tuo processo ETL".

Per creare un processo che utilizza il connettore Spark Elasticsearch

1. In AWS Glue Studio, scegli Connettori.
2. Nell'elenco Your connections (Le tue connessioni), seleziona la connessione appena creata e scegli Create job (Crea processo).
3. Nell'editor visivo dei processi, scegli il nodo di origine dati. A destra, nella scheda Data source properties - Connector (Proprietà origine dati - Connettore), configura ulteriori informazioni per il connettore.

   a. Scegli Add Schema (Aggiungi schema) e inserisci lo schema del set di dati nell'origine dati. Le connessioni non utilizzano tabelle memorizzate nel Catalogo dati, il che AWS Glue Studio significa che non conosce lo schema dei dati. Devi fornire queste informazioni sullo schema manualmente. Per istruzioni su come utilizzare l'editor dello schema, consulta the section called "Modifica dello schema in un nodo di trasformazione personalizzato".

   b. Espandi Connection options (Opzioni di connessione).

   c. Scegli Aggiungi nuova opzione e inserisci le informazioni necessarie per il connettore che non sono state inserite nel AWS segreto:

      - es.nodes: https://< endpoint di OpenSearch dominio>
      - es.port: 443
      - path: test
      - es.nodes.wan.only: true

      Per una spiegazione di queste opzioni di connessione, fai riferimento a: https://www.elastic.co/guide/en/elasticsearch/hadoop/current/configuration.html.
4. Aggiungi un nodo di destinazione al grafico.

   La destinazione dati può essere Amazon S3 oppure le informazioni provenienti da un AWS Glue Data Catalog o un connettore possono essere usate per scrivere dati in una posizione diversa. Ad esempio, è possibile utilizzare una tabella del catalogo dati per scrivere in un database in Amazon RDS oppure utilizzare un connettore come destinazione dati per scrivere in archivi dati non supportati in modo nativo in AWS Glue.

Se si sceglie un connettore per la destinazione dati, è necessario scegliere una connessione creata per tale connettore. Inoltre, se richiesto dal provider del connettore, è necessario aggiungere opzioni per fornire ulteriori informazioni al connettore. Se si utilizza una connessione che contiene informazioni relative a un AWS segreto, non è necessario fornire il nome utente e la password di autenticazione nelle opzioni di connessione.

5. Facoltativamente, aggiungi ulteriori origini dati e uno o più nodi di trasformazione come descritto in the section called "Trasformare i dati con AWS Glue trasformazioni gestite".
6. Configura le proprietà del processo come descritto in the section called "Modificare le proprietà del processo", iniziando dalla fase 3, e salva il lavoro.

### Approfondimenti

Fase 6: esecuzione del processo

## Fase 6: esecuzione del processo

Dopo aver salvato il processo, puoi eseguire il processo per eseguire le operazioni ETL.

Per eseguire il lavoro che hai creato per il AWS Glue Connector for Elasticsearch

1. Utilizzando la AWS Glue Studio console, nella pagina dell'editor visivo, scegli Esegui.
2. Nel banner che indica l'esito positivo, scegli Run Details (Dettagli esecuzione), oppure puoi scegliere la scheda Runs (Esecuzioni) dell'editor visivo per visualizzare le informazioni sull'esecuzione del processo.

# Creazione AWS Glue di posti di lavoro con sessioni interattive

Grazie alle sessioni interattive in AWS Glue, i data engineer possono creare processi AWS Glue con una velocità e una semplicità mai viste prima.

Argomenti


- Panoramica delle sessioni interattive in AWS Glue
- Nozioni di base sulle sessioni interattive AWS Glue
- Configurazione delle sessioni interattive di AWS Glue per Jupyter e notebook AWS Glue Studio
- Convertire uno script o un taccuino in un lavoro AWS Glue
- Utilizzo delle operazioni di streaming in sessioni interattive di AWS Glue
- Prezzi delle sessioni interattive di AWS Glue
- Sviluppo e test degli script di lavoro AWS Glue a livello locale
- Endpoint di sviluppo


## Panoramica delle sessioni interattive in AWS Glue

Con le sessioni interattive di AWS Glue, puoi creare, testare ed eseguire rapidamente applicazioni di preparazione e analisi dei dati. Le sessioni interattive forniscono un'interfaccia programmatica e visiva per la creazione e il test di script di estrazione, trasformazione e caricamento (ETL) per la preparazione dei dati. Le sessioni interattive eseguono applicazioni di analisi dei dati Apache Spark e forniscono accesso on demand a un ambiente di runtime Spark remoto. AWS Glue gestisce in modo trasparente Spark serverless per queste sessioni interattive.

Le sessioni interattive sono flessibili, pertanto ti permettono di creare e testare le applicazioni dall'ambiente che preferisci. Puoi creare e lavorare con sessioni interattive tramite AWS Command Line Interface e l'API. Puoi utilizzare i notebook compatibili con Jupyter per creare e testare visivamente gli script. Le sessioni interattive forniscono un kernel Jupyter open source che si integra quasi ovunque funzioni Jupyter, inclusa l'integrazione con IntelliJ e VS Code. IDEs PyCharm Ciò consente di creare codice nell'ambiente locale ed eseguirlo senza problemi sul backend delle sessioni interattive.

Utilizzando l'API delle sessioni interattive, i clienti possono eseguire in modo programmatico applicazioni che utilizzano l'analisi dei dati di Apache Spark senza dover gestire l'infrastruttura Spark. È possibile eseguire una o più istruzioni Spark in una singola sessione interattiva.

Le sessioni interattive forniscono quindi un modo più rapido, economico e flessibile per creare ed eseguire applicazioni di preparazione e analisi dei dati. Per informazioni sull'utilizzo di sessioni interattive, consulta la documentazione in questa sezione. Magic supportati da AWS Glue

## Limitazioni

- I segnalibri del processo non sono supportati nelle sessioni interattive.
- La creazione AWS Command Line Interface di lavori su notebook utilizzando il non è supportata.
- AWS Glue Studio i notebook non supportano Scala.

# Nozioni di base sulle sessioni interattive AWS Glue

Queste sezioni descrivono come eseguire le sessioni interattive AWS Glue localmente.

## Prerequisiti per impostare le sessioni interattive a livello locale

Di seguito sono indicati i prerequisiti per l'installazione delle sessioni interattive:

- Sono supportate le versioni di Python dalla 3.6 alla 3.10 e successive.
- Vedi le sezioni seguenti per MacOS/Linux le istruzioni di Windows.
- Consulta la documentazione sui prezzi delle sessioni interattive per comprendere la struttura dei costi.

## Installazione di Jupyter e sessioni AWS Glue interattive, kernel Jupyter.

Utilizza quanto segue per installare il kernel localmente.

Il comando `install-glue-kernels` installa il kernelspec jupyter sia per i kernel pyspark sia per quelli spark e installa anche i loghi nella directory corretta.

```
pip3 install --upgrade jupyter boto3 aws-glue-sessions
```

```
install-glue-kernels
```

## Esecuzione di Jupyter

Completa i seguenti passaggi per eseguire Jupyter Notebook.

1. Per avviare il notebook Jupyter utilizzare il seguente comando.

```
jupyter notebook
```

2. Scegliere New (Nuovo), quindi scegliere uno dei kernel AWS Glue per iniziare a creare codice rispetto a AWS Glue.

## Configurazione delle credenziali di sessione e della regione

### Istruzioni per MacOS/Linux

Le sessioni interattive di AWS Glue richiedono le stesse autorizzazioni IAM di Processi e Dev Endpoint di AWS Glue. Specificare il ruolo utilizzato con le sessioni interattive in uno dei due modi seguenti:

1. Con `%iam_role` e `%region` magic
2. Con una linea aggiuntiva in `~/.aws/config`

Configurazione di un ruolo di sessione con magic

Nella prima cella digita `%iam_role <YourGlueServiceRole>` nella prima cella eseguita.

Configurazione di un ruolo di sessione con **`~/.aws/config`**

AWS GlueIl ruolo del servizio per le sessioni interattive può essere specificato nel notebook stesso o memorizzato insieme alla configurazione. AWS CLI Se hai un ruolo che normalmente usi con processi AWS Glue, questo sarà questo ruolo. Se non disponi ancora di un ruolo per i processi AWS Glue, segui la guida Configuring IAM permissions for AWS Glue per configurarne uno.

Per impostare questo ruolo come ruolo predefinito per le sessioni interattive:

1. Con un editor di testo, apri `~/.aws/config`.
2. Cerca il profilo che usi per AWS Glue. Se non usi un profilo, usa il profilo `[Default]`.
3. Aggiungi una riga nel profilo per il ruolo che intendi usare come `glue_role_arn=<AWSGlueServiceRole>`.

4. [Facoltativo]: se sul tuo profilo non è impostata una regione predefinita, è consigliabile aggiungerne una con `region=us-east-1`, sostituendo `us-east-1` con la regione desiderata.
5. Salvare la configurazione.

Per ulteriori informazioni, consulta Sessioni Interattive con IAM.

### Istruzioni per Windows

Le sessioni interattive di AWS Glue richiedono le stesse autorizzazioni IAM di Processi e Dev Endpoint di AWS Glue. Specificare il ruolo utilizzato con le sessioni interattive in uno dei due modi seguenti:

1. Con `%iam_role` e `%region` magic
2. Con una linea aggiuntiva in `~/.aws/config`

Configurazione di un ruolo di sessione con magic

Nella prima cella digita `%iam_role <YourGlueServiceRole>` nella prima cella eseguita.

Configurazione di un ruolo di sessione con **`~/.aws/config`**

AWS GlueIl ruolo di servizio per le sessioni interattive può essere specificato nel notebook stesso o memorizzato insieme alla AWS CLI configurazione. Se hai un ruolo che normalmente usi con processi AWS Glue, questo sarà questo ruolo. Se non disponi ancora di un ruolo per processi AWS Glue, segui questa guida,Impostazione delle autorizzazioni IAM per AWS Glue, per configurarne uno.

Per impostare questo ruolo come ruolo predefinito per le sessioni interattive:

1. Con un editor di testo, apri `~/.aws/config`.
2. Cerca il profilo che usi per AWS Glue. Se non usi un profilo, usa il profilo `[Default]`.
3. Aggiungi una riga nel profilo per il ruolo che intendi usare come `glue_role_arn=<AWSGlueServiceRole>`.
4. [Facoltativo]: se sul tuo profilo non è impostata una regione predefinita, è consigliabile aggiungerne una con `region=us-east-1`, sostituendo `us-east-1` con la regione desiderata.
5. Salvare la configurazione.

Per ulteriori informazioni, consulta Sessioni Interattive con IAM.

## Aggiornamento dall'anteprima delle sessioni interattive

Il kernel è stato aggiornato con nuovi nomi quando è stato rilasciato con la versione 0.27. Per pulire le versioni di anteprima dei kernel, esegui quanto segue da un terminale o. PowerShell

> **Note**
>
> Se fai parte di qualsiasi altra anteprima di AWS Glue che richiede un modello di servizio personalizzato, rimuovendo il kernel sarà rimosso anche questo modello.

```
# Remove Old Glue Kernels
jupyter kernelspec remove glue_python_kernel
jupyter kernelspec remove glue_scala_kernel

# Remove Custom Model
cd ~/.aws/models
rm -rf glue/
```

## Utilizzo di sessioni interattive con SageMaker AI Studio

AWS Glue Interactive Sessions è un ambiente di runtime Apache Spark on-demand e senza server che data scientist e ingegneri possono utilizzare per creare, testare ed eseguire rapidamente applicazioni di preparazione e analisi dei dati. È possibile avviare una sessione AWS Glue interattiva avviando un notebook Studio Classic. Amazon SageMaker AI

Per ulteriori informazioni, consulta Preparare i dati usando le sessioni interattive AWS Glue.

## Utilizzo di sessioni interattive con codice Microsoft Visual Studio

Prerequisiti

- Installa sessioni interattive AWS Glue e verifica che funzioni con notebook Jupyter.
- Scarica e installa Visual Studio Code con Jupyter. Per informazioni dettagliate, consulta Notebook Jupyter in VS Code

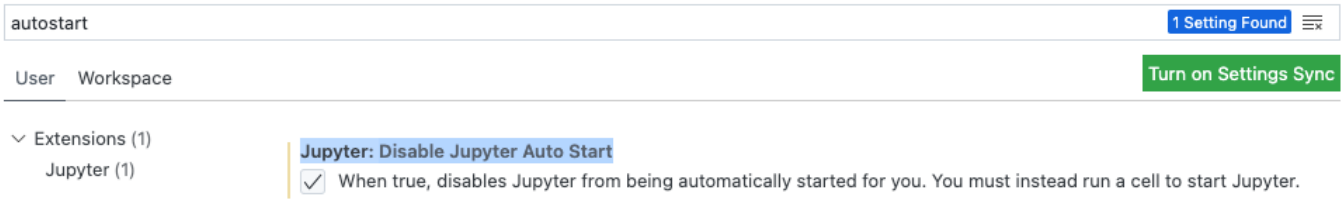
Per iniziare con sessioni interattive con VSCode

1. Disattiva Jupyter AutoStart in VS Code.

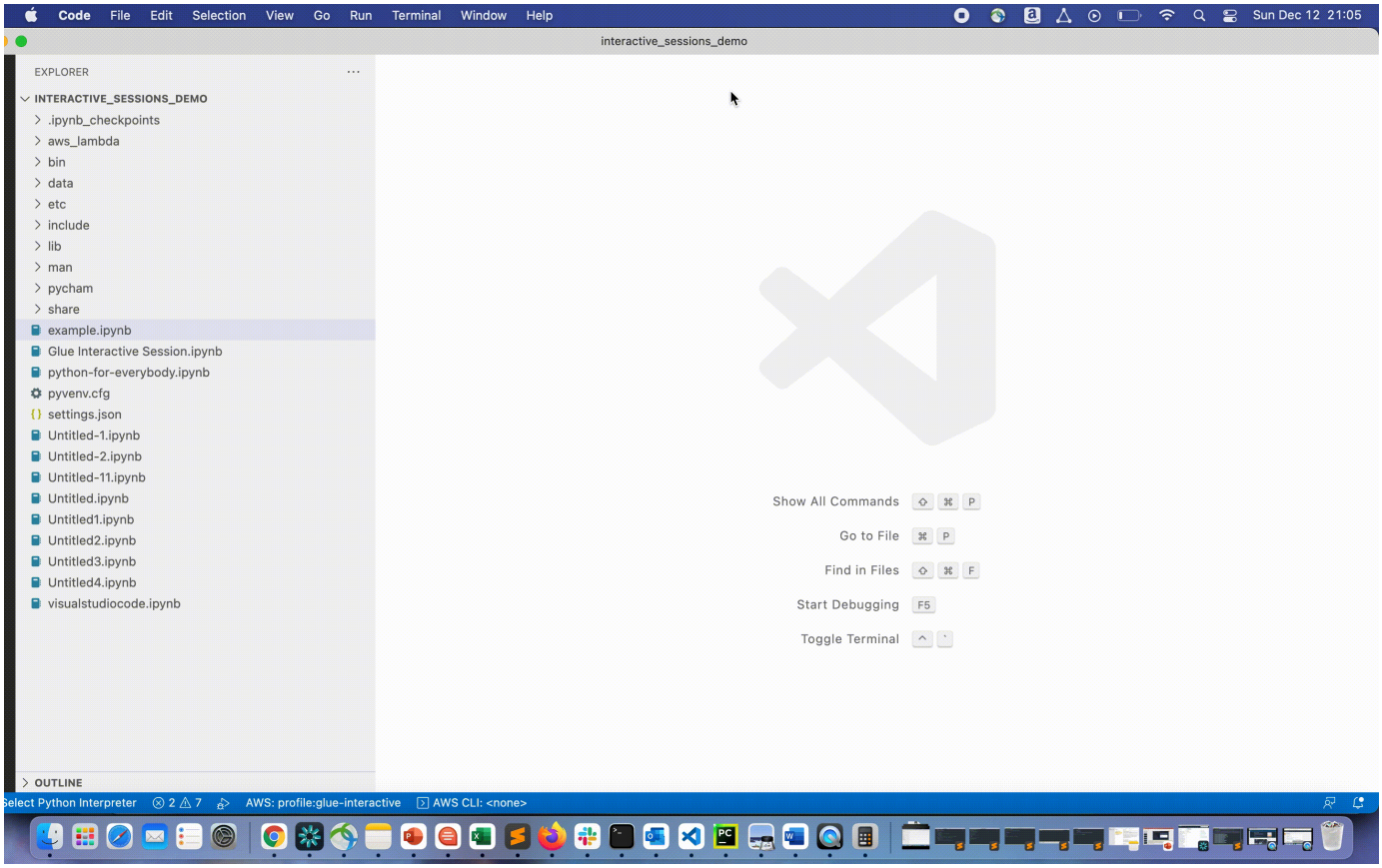
   In Visual Studio Code, i kernel di Jupyter vengono avviati automaticamente. Questo impedisce ai magic di entrare in vigore poiché la sessione è già stata avviata. Per disabilitare Auto Start su Windows, apri File > Preferences > Extensions > Jupyter, fai clic con il pulsante destro del mouse su Jupyter, quindi scegli Extension Settings.

   Su macOS, apri Code > Settings > Extensions > Jupyter, fai clic con il pulsante destro del mouse su Jupyter, quindi scegli Extension Settings.

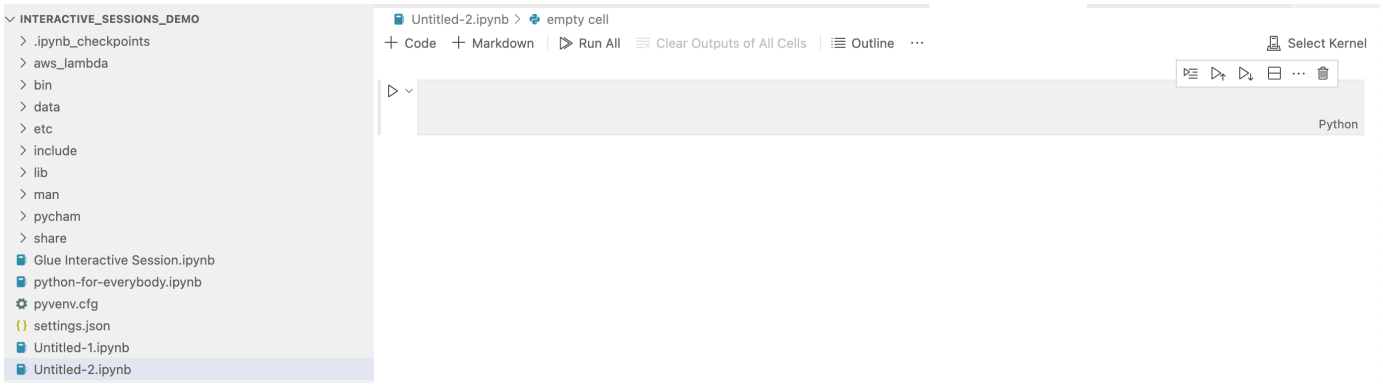
   Scorri verso il basso fino a visualizzare Jupyter: Disable Jupyter Auto Start. Seleziona la casella "When true, disables Jupyter from being automatically started for you. È necessario invece eseguire una cella per avviare Jupyter."




2. Vai su File (File) > New File (Nuovo file) > Save (Salva) per salvare questo file con il nome di tua scelta come estensione `.ipynb` o seleziona jupyter sotto Select a language (Seleziona una lingua) e salva il file.




3. Fare doppio clic sul file. Viene visualizzata la shell di Jupyter e verrà aperto un notebook.




4. Su Windows, quando crei un file per la prima volta, per impostazione predefinita, non è selezionato alcun kernel. Clicca su Select Kernel (Seleziona kernel) per visualizzare un elenco di kernel disponibili. Scegli Glue PySpark.

   Su macOS, se non vedi il PySpark kernel Glue, prova i seguenti passaggi:

   1. Esegui una sessione locale di Jupyter per ottenere l'URL.

      Ad esempio, per avviare il notebook Jupyter, utilizza il seguente comando.

```
jupyter notebook
```

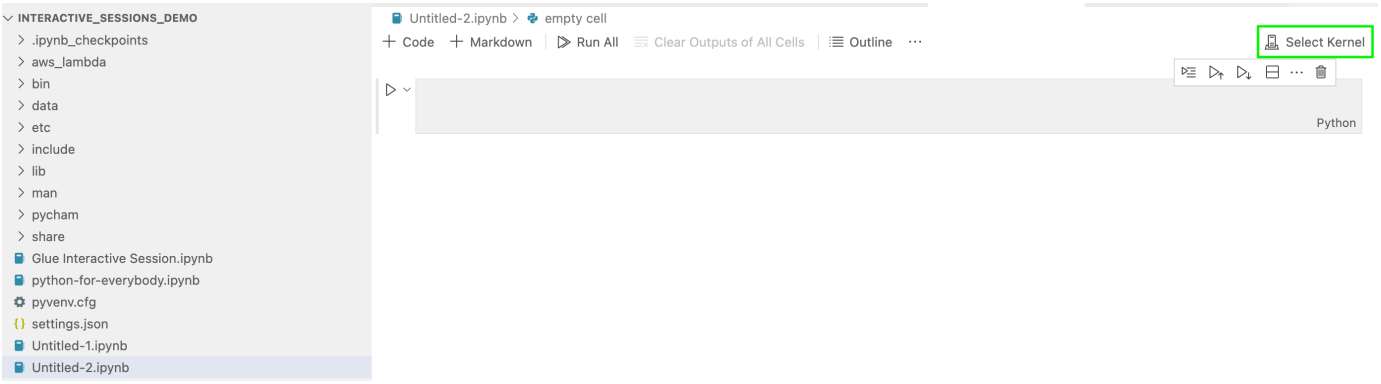
Quando il notebook viene eseguito per la prima volta, verrà visualizzato un URL simile a `http://localhost:8888/?token=3398XXXXXXXXXXXXXXX`.

Copia l'URL.

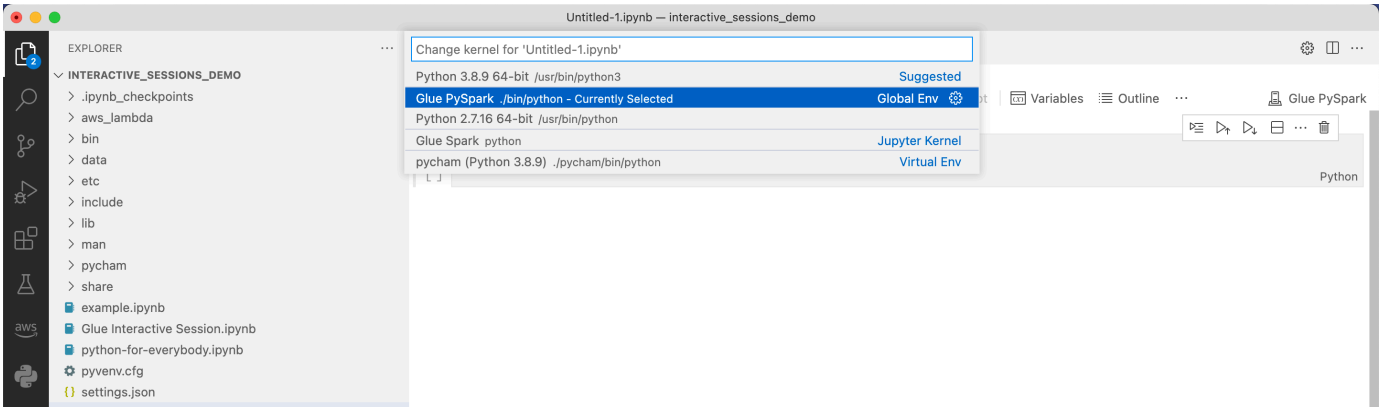
2. In VS Code, fai clic sul kernel corrente, quindi seleziona Select Another Kernel... e successivamente Existing Jupyter Server.... Incolla l'URL che hai copiato dal passaggio precedente.

   Se ricevi un messaggio di errore, consulta il wiki VS Code Jupyter.

3. In caso di successo, questo imposterà il kernel su  PySparkGlue.




Scegliete il kernel Glue PySpark o Glue Spark (rispettivamente per Python e Scala).




Se non vedi AWS Glue PySparki kernel AWS GlueSpark nell'elenco a discesa, assicurati di aver installato il AWS Glue kernel nel passaggio precedente o che

l'`python.defaultInterpreterPath`impostazione in Visual Studio Code sia corretta. Per ulteriori informazioni, vedi python. defaultInterpreterPath descrizione dell'impostazione.

5. Crea una sessione interattiva AWS Glue. Procedi alla creazione di una sessione nello stesso modo in cui è stato fatto nel notebook Jupyter. Specifica qualsiasi magic nella parte superiore della prima cella ed esegui un'istruzione di codice.

## Sessioni Interattive con IAM

Queste sezioni descrivono le considerazioni sulla sicurezza delle sessioni interattive di AWS Glue.

Argomenti

- Principali IAM utilizzati con le sessioni interattive
- Configurazione di un principale del client
- Configurazione di un ruolo runtime
- Rendi privata la tua sessione con TagOnCreate
- Considerazioni sulle policy IAM

### Principali IAM utilizzati con le sessioni interattive

Con le sessioni interattive AWS Glue vengono utilizzati due principali IAM.

- Principale client: il principale del client (un utente o un ruolo) autorizza le operazioni dell'API per sessioni interattive da un client AWS Glue configurato con le credenziali basate sull'identità del principale. Ad esempio, potrebbe trattarsi di un ruolo IAM che in genere si utilizza per accedere alla console di AWS Glue. Questo potrebbe anche essere un ruolo assegnato a un utente in IAM le cui credenziali vengono utilizzate per il AWS Command Line Interface, o un AWS Glue client utilizzato dalle sessioni interattive del kernel Jupyter.
- Ruolo Runtime: il Ruolo runtime è un ruolo IAM che il principale del client passa alle operazioni API delle sessioni interattive. AWS Glue utilizza questo ruolo per eseguire istruzioni nella sessione. Ad esempio, questo ruolo potrebbe essere quello utilizzato per l'esecuzione dei processi ETL di AWS Glue.

  Per ulteriori informazioni, consulta Configurazione di un ruolo runtime.

## Configurazione di un principale del client

È necessario allegare una policy di identità al principale del client per consentirgli di chiamare l'API delle sessioni interattive. Questo ruolo deve avere l'accesso `iam:PassRole` al ruolo di esecuzione che si passa per le operazioni API delle sessioni interattive come `CreateSession`. Ad esempio, puoi collegare la policy AWSGlueConsoleFullAccessgestita a un ruolo IAM che consente agli utenti del tuo account a cui è associata la policy di accedere a tutte le sessioni create nel tuo account (come l'istruzione di runtime o l'istruzione di cancellazione).

Se desideri proteggere la tua sessione e renderla privata solo per determinati ruoli IAM, come quelli associati all'utente che ha creato la sessione, puoi utilizzare il Tag Based Authorization Control di AWS Glue Interactive Session chiamato TagOnCreate. Per ulteriori informazioni, scopri Rendi privata la tua sessione con TagOnCreate come una policy gestita con ambito e basata su tag proprietari può rendere privata la tua sessione con. TagOnCreate Per ulteriori informazioni sulle policy basate sull'identità, consulta Policy basate sull'identità per AWS Glue.

## Configurazione di un ruolo runtime

È necessario passare un ruolo IAM all'operazione CreateSession API per consentire AWS Glue di assumere ed eseguire istruzioni in sessioni interattive. Il ruolo deve disporre delle stesse autorizzazioni IAM necessarie per l'esecuzione dei processi AWS Glue tipici. Ad esempio, puoi creare un ruolo di servizio utilizzando la AWSGlueServiceRolepolitica che consente di AWS Glue chiamare AWS i servizi per tuo conto. Se utilizzi la console AWS Glue, la console creerà automaticamente un ruolo di servizio per tuo conto o ne utilizzerà uno esistente. Puoi anche creare il tuo ruolo IAM personalizzato e allegare la policy IAM per concedere autorizzazioni simili.

Se desideri proteggere la tua sessione e renderla privata solo per l'utente che ha creato la sessione, puoi utilizzare il controllo di autorizzazione basato su tag di AWS Glue Interactive Session chiamato TagOnCreate. Per ulteriori informazioni, scopri Rendi privata la tua sessione con TagOnCreate in che modo una politica gestita basata su tag proprietari e basata su tag del proprietario può rendere privata la tua sessione con. TagOnCreate Per ulteriori informazioni sulle policy basate sull'identità, consulta la pagina Politiche basate sull'identità per Glue AWS. Se stai creando il ruolo di esecuzione da solo dalla console IAM e desideri rendere privato il servizio con TagOnCreate funzionalità, segui i passaggi seguenti.

1. Creare un ruolo IAM con il tipo di ruolo impostato su `Glue`.
2. Allega questa policy AWS Glue gestita: AwsGlueSessionUserRestrictedServiceRole

3. Prefissa il nome del ruolo con il nome della politica. AwsGlueSessionUserRestrictedServiceRole Ad esempio, puoi creare un ruolo con name AwsGlueSessionUserRestrictedServiceRole-myrole e allegare AWS Glue una policy gestita. AwsGlueSessionUserRestrictedServiceRole
4. Allegare una policy di attendibilità come di seguito per consentire a AWS Glue di assumere il ruolo:

   JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "glue.amazonaws.com"
        ]
      },
      "Action": [
        "sts:AssumeRole"
      ]
    }
  ]
}
```

Per le sessioni interattive del kernel Jupyter, puoi specificare la chiave nel tuo profilo. `iam_role` AWS Command Line Interface Per ulteriori informazioni, consulta Configurazione di sessioni con ~/.aws/config . Se stai interagendo con sessioni interattive usando un notebook AWS Glue, puoi passare il ruolo di esecuzione in `%iam_role` magic nella prima cella che esegui.

## Rendi privata la tua sessione con TagOnCreate

Le sessioni interattive di AWS Glue supportano il tagging e l'autorizzazione basata sui tag (TBAC) per le sessioni interattive come risorsa denominata. Oltre all'utilizzo di TagResource and da parte di TBAC UntagResource APIs, le sessioni AWS Glue interattive supportano la TagOnCreate funzionalità di «taggare» una sessione con un determinato tag solo durante la creazione della sessione con CreateSession operazione. Ciò significa anche che quei tag verranno rimossi il DeleteSession, alias. UntagOnDelete

TagOnCreate offre un potente meccanismo di sicurezza per rendere la sessione privata al creatore della sessione. Ad esempio, puoi allegare una policy IAM con «owner» RequestTag e valore $ {aws:userId} a un client principal (come un utente) per consentire la creazione di una sessione solo se su richiesta viene fornito un tag «owner» con il valore corrispondente dell'userID del chiamante. CreateSession Questa policy consente alle sessioni interattive di AWS Glue di creare una risorsa di sessione e taggare la sessione con il tag userId solo durante il tempo di creazione della sessione. Inoltre, puoi limitare l'accesso (ad esempio le istruzioni in esecuzione) alla tua sessione solo al creatore (alias tag owner con valore $ {aws:userId}) della sessione allegando una policy IAM con «owner» ResourceTag al ruolo di esecuzione che hai passato durante CreateSession.

Per semplificare l'utilizzo della TagOnCreate funzionalità che rende privata una sessione per il creatore della sessione, AWS Glue fornisce politiche gestite e ruoli di servizio specializzati.

Se desideri creare una sessione AWS Glue interattiva utilizzando un AssumeRole principale IAM (ovvero utilizzando credenziali fornite assumendo un ruolo IAM) e desideri rendere la sessione privata per il creatore, utilizza politiche simili rispettivamente a e. AWSGlueSessionUserRestrictedNotebookPolicyAWSGlueSessionUserRestrictedNotebookServiceRole Queste politiche consentono di AWS Glue utilizzare $ {aws:PrincipalTag} per estrarre il valore del tag owner. Ciò richiede il passaggio di un tag userID con valore $ {aws:userId} come nella credenziale di assunzione del ruolo. SessionTag Consulta Tag della sessione ID. Se utilizzi un'istanza Amazon EC2 con un profilo di istanza che vende la credenziale e desideri creare una sessione o interagire con la sessione dall'interno dell'istanza Amazon EC2, dovrai passare un tag userId con valore $ {aws:userId} come nella credenziale di assunzione del ruolo. SessionTag

Ad esempio, se stai creando una sessione utilizzando una credenziale AssumeRole principale IAM e desideri rendere privato il tuo servizio con TagOnCreate funzionalità, procedi nel seguente modo.

1. Crea tu stesso un ruolo runtime dalla console IAM. Allega questa policy AWS Glue gestita AwsGlueSessionUserRestrictedNotebookServiceRolee aggiungi il prefisso al nome del ruolo con il nome della policy. AwsGlueSessionUserRestrictedNotebookServiceRole Ad esempio, puoi creare un ruolo con nome AwsGlueSessionUserRestrictedNotebookServiceRole-myrole e allegare AWS Glue una policy gestita. AwsGlueSessionUserRestrictedNotebookServiceRole
2. Allega una policy di attendibilità come di seguito per consentire ad AWS Glue di assumere il ruolo:

   JSON

```
{
```

```
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "glue.amazonaws.com"
        ]
      },
      "Action": [
        "sts:AssumeRole"
      ]
    }
  ]
}
```

3. Crea un altro ruolo denominato con un prefisso AwsGlueSessionUserRestrictedNotebookPolicye allega la policy AWS Glue gestita AwsGlueSessionUserRestrictedNotebookPolicyper rendere privata la sessione. Oltre alla policy gestita, allega la seguente policy in linea per consentire iam: PassRole al ruolo che hai creato nel passaggio 1.

   JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": [
        "arn:aws:iam::*:role/
AwsGlueSessionUserRestrictedNotebookServiceRole*"
      ],
      "Condition": {
        "StringLike": {
          "iam:PassedToService": [
            "glue.amazonaws.com"
          ]
        }
      }
```

```
    }
  ]
}
```

4. Allega una policy di attendibilità come di seguito al AWS Glue IAM riportato qui sopra per consentirgli di assumere il ruolo.

   JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "glue.amazonaws.com"
        ]
      },
      "Action": [
        "sts:AssumeRole",
        "sts:TagSession"
      ]
    }
  ]
}
```

> **Note**
>
> Facoltativamente, puoi utilizzare un singolo ruolo (ad esempio, il ruolo notebook) e allegare entrambe le politiche AwsGlueSessionUserRestrictedNotebookServiceRolegestite sopra riportate e. AwsGlueSessionUserRestrictedNotebookPolicy Allega anche la policy inline aggiuntiva per consentire `iam:passrole` del tuo ruolo di AWS Glue. E infine allega la policy di fiducia di cui sopra per consentire `sts:AssumeRole` e `sts:TagSession`.

AWSGlueSessionUserRestrictedNotebookPolicy

AWSGlueSessionUserRestrictedNotebookPolicy Fornisce l'accesso per creare una sessione AWS Glue interattiva da un notebook solo se il tag chiave «proprietario» e il valore corrispondono AWS all'ID utente del principale (utente o ruolo). Per ulteriori informazioni, consulta Casi in cui è possibile utilizzare le variabili di policy. Questa policy è collegata al principale (utente o ruolo IAM) che crea notebook di sessioni interattive AWS Glue da AWS Glue Studio. Questa politica consente inoltre un accesso sufficiente al AWS Glue Studio notebook per interagire con le risorse della sessione AWS Glue Studio interattiva create con il valore del tag «owner» che corrisponde all'ID AWS utente del principale. Questa policy nega l'autorizzazione di modificare o rimuovere i tag "proprietario" da una risorsa di sessione AWS Glue dopo la creazione della sessione.

AWSGlueSessionUserRestrictedNotebookServiceRole

AWSGlueSessionUserRestrictedNotebookServiceRoleFornisce un accesso sufficiente al AWS Glue Studio notebook per interagire con le risorse della sessione AWS Glue interattiva create con il valore del tag «owner» che corrisponde all'ID AWS utente del principale (utente o ruolo) del creatore del notebook. Per ulteriori informazioni, consulta Casi in cui è possibile utilizzare le variabili di policy. Questa policy relativa ai ruoli di servizio è allegata al ruolo che viene passato per magia a un notebook o passato come ruolo di esecuzione all' CreateSession API. Questa politica consente inoltre di creare una sessione AWS Glue interattiva da un notebook solo se il tag chiave «proprietario» e il valore corrispondono AWS all'ID utente del principale. Questa policy nega l'autorizzazione di modificare o rimuovere i tag "proprietario" da una risorsa di sessione AWS Glue dopo la creazione della sessione. Questa policy include anche le autorizzazioni per la scrittura e la lettura da bucket Amazon S3, la CloudWatch scrittura di log, la creazione e l'eliminazione di tag per le risorse Amazon EC2 utilizzate da. AWS Glue

Rendere private le sessioni con gli utenti

Puoi collegare i AWSGlueSessionUserRestrictedPolicydue ruoli IAM associati a ciascuno degli utenti del tuo account per impedire loro di creare una sessione solo con un tag proprietario con un valore corrispondente al proprio $ {aws:userId}. Invece di utilizzare AWSGlueSessionUserRestrictedNotebookPolicye, è AWSGlueSessionUserRestrictedNotebookServiceRolenecessario utilizzare politiche simili a e rispettivamente. AWSGlueSessionUserRestrictedPolicyAWSGlueSessionUserRestrictedServiceRole Per ulteriori informazioni, consulta la pagina Using identity based policies. Questa policy limita l'accesso a una sessione solo al creatore, il valore ${aws:userID} dell'utente che ha creato la sessione con tag proprietario svelando il proprio ${aws:userID}. Se hai creato tu stesso il ruolo

di esecuzione utilizzando la console IAM seguendo i passaggi indicatiConfigurazione di un ruolo runtime, oltre ad allegare la policy AwsGlueSessionUserRestrictedPolicygestita, collega anche la seguente policy in linea a ciascuno degli utenti del tuo account `iam:PassRole` per consentire il ruolo di esecuzione che hai creato in precedenza.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": [
        "arn:aws:iam::*:role/AwsGlueSessionUserRestrictedServiceRole*"
      ],
      "Condition": {
        "StringLike": {
          "iam:PassedToService": [
            "glue.amazonaws.com"
          ]
        }
      }
    }
  ]
}
```

AWSGlueSessionUserRestrictedPolicy

AWSGlueSessionUserRestrictedPolicyFornisce l'accesso alla creazione di una sessione AWS Glue interattiva utilizzando l' CreateSession API solo se vengono forniti una chiave di tag «proprietario» e un valore corrispondenti al AWS relativo ID utente. Questa politica di identità è allegata all'utente che richiama l' CreateSession API. Questa politica consente inoltre di interagire con le risorse della sessione AWS Glue interattiva che sono state create con un tag «proprietario» e un valore corrispondenti AWS all'ID utente. Questa policy nega l'autorizzazione di modificare o rimuovere i tag "proprietario" da una risorsa di sessione AWS Glue dopo la creazione della sessione.

AWSGlueSessionUserRestrictedServiceRole

AWSGlueSessionUserRestrictedServiceRoleFornisce l'accesso completo a tutte AWS Glue le risorse ad eccezione delle sessioni e consente agli utenti di creare e utilizzare solo le sessioni interattive associate all'utente. Questa politica include anche altre autorizzazioni necessarie AWS Glue per gestire le risorse Glue in altri AWS servizi. La politica consente inoltre di aggiungere tag alle AWS Glue risorse di altri AWS servizi.

### Considerazioni sulle policy IAM

Le sessioni interattive sono risorse IAM in AWS Glue. Poiché si tratta di risorse IAM, l'accesso e l'interazione a una sessione sono regolati dalle policy IAM. In base alle policy IAM allegate a un principale del client o al ruolo di esecuzione configurato da un amministratore, un principale del client (utente o ruolo) sarà in grado di creare nuove sessioni e interagire con le proprie e con altre.

Se un amministratore ha allegato una policy IAM come AWSGlue ConsoleFullAccess o AWSGlue ServiceRole che consente l'accesso a tutte le AWS Glue risorse di quell'account, il responsabile del cliente sarà in grado di collaborare tra loro. Ad esempio, un utente sarà in grado di interagire con le sessioni create da altri utenti se le policy lo consentono.

Per configurare una policy su misura per le tue esigenze specifiche, consulta la documentazione IAM sulla configurazione delle risorse per una policy. Ad esempio, per isolare le sessioni che appartengono a un utente, puoi utilizzare la TagOnCreate funzionalità supportata dalle sessioni AWS Glue interattive. Per informazioni, consulta Rendi privata la tua sessione con TagOnCreate.

Le sessioni interattive supportano la limitazione della creazione di sessioni in base a determinate condizioni del VPC. Per informazioni, consulta Policy di controllo che controllano le impostazioni utilizzando le chiavi di condizione.

# Configurazione delle sessioni interattive di AWS Glue per Jupyter e notebook AWS Glue Studio

## Introduzione ai magic di Jupyter

I magic di Jupyter sono comandi che possono essere eseguiti all'inizio di una cella o come un intero corpo di una cella. I magic di linea iniziano per `%`, i magic di cella per `%%`. I magic di linea come `%region` e `%connections` possono essere eseguiti con più magic in una cella o con codice incluso nel corpo della cella come nell'esempio seguente.

```
%region us-east-2
%connections my_rds_connection
dy_f = glue_context.create_dynamic_frame.from_catalog(database='rds_tables',
 table_name='sales_table')
```

I magic di cella devono utilizzare l'intera cella e possono avere il comando esteso su più righe. Un esempio di `%%sql` è riportato di seguito.

```
%%sql
select * from rds_tables.sales_table
```

## Magic supportati dalle sessioni interattive di AWS Glue per Jupyter

Di seguito sono elencati i magic che puoi utilizzare con le sessioni interattive AWS Glue per Jupyter Notebook.

Sessioni Magic

| Name | Tipo | Description |
|---|---|---|
| `%help` | N/A | Restituisce un elenco di descrizioni e tipi di input per tutti i comandi magic. |
| `%profile` | Stringa | Specificate un profilo nella AWS configura zione da utilizzare come fornitore di credenziali. |
| `%region` | Stringa | Specificare il Regione AWS; in cui inizializ zare una sessione. Impostazione predefini ta da `~/.aws/configure`.<br><br>Ad esempio: `%region us-west-1` |
| `%idle_timeout` | Int | Il numero di minuti di inattività dopo i quali una sessione andrà in timeout in seguito all'esecuzione di una cella. Il valore predefinito del timeout di inattività per le |

| Name | Tipo | Description |
| --- | --- | --- |
| | | sessioni Spark ETL è il timeout predefini to, pari a 2.880 minuti (48 ore). Per altri tipi di sessione, consulta la documentazione relativa al tipo di sessione specifico.<br><br>Ad esempio: `%idle_timeout 3000` |
| `%session_id` | N/A | Restituisce l'ID della sessione in esecuzion e. |
| `%session_id_prefix` | Stringa | Definite una stringa che precederà tutte le sessioni IDs nel formato [session_id_prefix] - [session_id]. Se non viene fornito un ID di sessione, verrà generato un UUID casuale. Questo magic non è supportato quando esegui un notebook Jupyter in AWS Glue Studio.<br><br>Ad esempio: `%session_id_prefix 001` |
| `%status` | | Restituisce lo stato dell'attuale sessione AWS Glue inclusi la durata, la configura zione e l'utente/ruolo che la esegue. |
| `%stop_session` | | Arresta la sessione corrente. |
| `%list_sessions` | | Elenca tutte le sessioni attualmente in esecuzione per nome e ID. |
| `%session_type` | Stringa | Imposta il tipo di sessione su Flussi di dati, ETL o Ray.<br><br>Ad esempio: `%session_type Streaming` |

| Name | Tipo | Description |
| --- | --- | --- |
| `%glue_version` | Stringa | La versione di AWS Glue che questa sessione dovrà utilizzare.<br><br>Ad esempio: `%glue_version 3.0` |

Magic per la selezione dei tipi di processo

| Name | Tipo | Description |
| --- | --- | --- |
| `%streaming` | Stringa | Cambia il tipo di sessione in Streaming AWS Glue. |
| `%etl` | Stringa | Cambia il tipo di sessione in ETL AWS Glue. |
| %glue_ray | Stringa | Cambia AWS Glue il tipo di sessione in for Ray. Vedi Magics supportato dalle sessioni interattive di AWS Glue Ray. |

Magic per la configurazione di AWS Glue per Spark

Il magic `%%configure` è un dizionario formattato json composto da tutti i parametri di configurazione per una sessione. Ciascun parametro può essere specificato qui o tramite magic individuali.

| Name | Tipo | Description |
| --- | --- | --- |
| `%%configure` | Dizionario | Specifica un dizionario formattato JSON composto da tutti i parametri di configurazione per una sessione. Ciascun parametro può essere specificato qui o tramite magic individuali.<br><br>Per un elenco di parametri ed esempi su come utilizzare `%%configure` , |

| Name | Tipo | Description |
| --- | --- | --- |
| | | consultaargomenti del magic di cella %%configure. |
| `%iam_role` | Stringa | Specifica un ruolo IAM ARN con cui eseguire la sessione. Predefinito da ~/.aws/configure.<br><br>Ad esempio: `%iam_role AWSGlueServiceRole` |
| `%number_of_workers` | Int | Il numero di dipendenti di un specifico worker-type allocati quando viene eseguito un processo. Deve essere impostato anche `worker_type` . Il `number_of_workers` predefinito è 5.<br><br>Ad esempio: `%number_of_workers 2` |
| `%additional_python_modules` | List | Elenco separato da virgole di moduli Python aggiuntivi da includere nel cluster (possono provenire da PyPI o S3).<br><br>Esempio: `%additional_python_modules pandas, numpy` . |

| Name | Tipo | Description |
| --- | --- | --- |
| `%%tags` | Stringa | Aggiunge tag a una sessione. Specifica i tag tra parentesi graffe { }. Ogni coppia di nomi di tag è racchiusa tra parentesi (" ") e separata da una virgola (,).<br><br>`%%tags`<br>`{"billing":"Data-Platform",`<br>`"team":"analytics"}`<br><br>Utilizza il magic `%status` per visualizzare i tag associati alla sessione.<br><br>`%status`<br><br>`Session ID: <sessionId>`<br>`Status: READY`<br>`Role: <example-role>`<br>`CreatedOn: 2023-05-26 11:12:17.`<br>`056000-07:00`<br>`GlueVersion: 3.0`<br>`Job Type: glueetl`<br>`Tags: {'owner':'example-owner',`<br>`'team':'analytics', 'billing'`<br>`:'Data-Platform'}`<br>`Worker Type: G.4X`<br>`Number of Workers: 5`<br>`Region: us-west-2`<br>`Applying the following default`<br>`arguments:`<br>`--glue_kernel_version 0.38.0`<br>`--enable-glue-datacatalog true`<br>`Arguments Passed: ['--glue_`<br>`kernel_version: 0.38.0', '--`<br>`enable-glue-datacatalog: true']` |

| Name | Tipo | Description |
| --- | --- | --- |
| `%%assume_role` | Dizionario | Specifica un dizionario in formato json o una stringa ARN del ruolo IAM per creare una sessione per l'accesso multi-account.<br><br>Esempio con ARN:<br><br>`%%assume_role`<br>`{`<br>`  'arn:aws:iam::XXXXXXXXXXXX:`<br>`role/AWSGlueServiceRole'`<br>`}`<br><br>Esempio con credenziali:<br><br>` %%assume_role`<br>`{{`<br>`    "aws_access_key_id" =`<br>` "XXXXXXXXXXXX",`<br>`    "aws_secret_access_key" =`<br>` "XXXXXXXXXXXX",`<br>`    "aws_session_token" =`<br>` "XXXXXXXXXXXX"`<br>`}}` |

## argomenti del magic di cella %%configure

Il magic `%%configure` è un dizionario formattato json composto da tutti i parametri di configurazione per una sessione. Ciascun parametro può essere specificato qui o tramite magic individuali. Di seguito sono riportati alcuni esempi di argomenti supportati dal magic di cella `%%configure`. Utilizza il prefisso `--` per gli argomenti di esecuzione specificati per il lavoro. Esempio:

```
%%configure
{
   "--user-jars-first": "true",
   "--enable-glue-datacatalog": "false"
}
```

Per ulteriori informazioni sui parametri del lavoro, consulta Parametri Job.

Configurazione della sessione

| Parametro | Tipo | Description |
| --- | --- | --- |
| `max_retries` | Int | Il numero massimo di tentativi per riprovare il processo se ha esito negativo.<br>`%%configure`<br>`{`<br>`  "max_retries": "0"`<br>`}` |
| `max_concurrent_runs` | Int | Il numero massimo di esecuzioni simultane e consentite per un processo.<br>Esempio:<br>`%%configure`<br>`{`<br>`  "max_concurrent_runs": "3"`<br>`}` |

Parametri della sessione

| Parametro | Tipo | Descrizione |
| --- | --- | --- |
| `--enable-spark-ui` | Booleano | Abilita l'interfaccia utente di Spark per monitorare ed eseguire il debug dei processi ETL AWS Glue.<br>`%%configure`<br>`{`<br>`  "--enable-spark-ui": "true"` |

| Parametro | Tipo | Descrizione |
| --- | --- | --- |
| | | `}` |
| `--spark-event-logs-path` | Stringa | Specifica un percorso Amazon S3. Quando si utilizza la funzionalità di monitoraggio dell'interfaccia utente Spark.<br><br>Esempio:<br><br>`%%configure`<br>`{`<br>`  "--spark-event-logs-path":`<br>`"s3://path/to/event/logs/"`<br>`}` |
| `--script_location` | Stringa | Specifica il percorso S3 per uno script che esegue un processo.<br><br>Esempio:<br><br>`%%configure`<br>`{`<br>`  "script_location": "s3://new-folder-here"`<br>`}` |

| Parametro | Tipo | Descrizione |
| --- | --- | --- |
| `--SECURITY_CONFIGURATION` | Stringa | Il nome di una configurazione di AWS Glue sicurezza<br><br>Esempio:<br><br>`%%configure`<br>`{`<br>`"--security_configuration": { "encryption_type": "kms", "kms_key_id": "YOUR_KMS_KEY_ARN" }`<br>`}` |
| `--job-language` | Stringa | Il linguaggio di programmazione script. Accetta un valore di "scala" o "python". L'impostazione predefinita è "python".<br><br>Esempio:<br><br>`%%configure`<br>`{`<br>`"--job-language": "scala"`<br>`}` |

| Parametro | Tipo | Descrizione |
| --- | --- | --- |
| `--class` | Stringa | La classe Scala che funge da punto di accesso per lo script Scala. L'impostazione predefinita è null.<br><br>Esempio:<br><br>`%%configure`<br>`{`<br>`  "--class": "className"`<br>`}` |
| `--user-jars-first` | Booleano | Assegna la priorità ai file JAR aggiuntivi del cliente nel classpath. L'impostazione predefinita è null.<br><br>Esempio:<br><br>`%%configure`<br>`{`<br>`  "--user-jars-first": "true"`<br>`}` |
| `--use-postgres-driver` | Booleano | Assegna la priorità al driver JDBC Postgres nel percorso della classe per evitare conflitti con il driver JDBC. Amazon Redshift L'impostazione predefinita è null.<br><br>Esempio:<br><br>`%%configure`<br>`{`<br>`  "--use-postgres-driver": "true"`<br>`}` |

| Parametro | Tipo | Descrizione |
| --- | --- | --- |
| `--extra-files` | List(string) | I percorsi Amazon S3 dei file aggiuntivi, come file di configurazione, che AWS Glue copia nella directory di lavoro del tuo script prima di eseguirlo.<br><br>Esempio:<br><br>`%%configure`<br>`{`<br>`  "--extra-files": "s3://path/to/additional/files/"`<br>`}` |
| `--job-bookmark-option` | Stringa | Controlla il comportamento di un segnalibro del processo. Accetta il valore '', '' o ''. job-bookmark-enable job-bookmark-disable job-bookmark-pause L'impostazione predefinita è 'job-bookmark-disable'.<br><br>Esempio:<br><br>`%%configure`<br>`{`<br>`  "--job-bookmark-option": "job-bookmark-enable"`<br>`}` |

| Parametro | Tipo | Descrizione |
| --- | --- | --- |
| `--TempDir` | Stringa | Specifica un percorso Amazon S3 a un bucket utilizzabile come directory temporanea per il processo. L'impostazione predefinita è null.<br><br>Esempio:<br><br>`%%configure`<br>`{`<br>`  "--TempDir": "s3://path/to/temp/dir"`<br>`}` |
| `--enable-s3-parquet-optimized-committer` | Booleano | Abilita il committer ottimizzato EMRFS Amazon S3 per la scrittura dei dati Parquet in Amazon S3. Il valore predefinito è "true".<br><br>Esempio:<br><br>`%%configure`<br>`{`<br>`  "--enable-s3-parquet-optimized-committer": "false"`<br>`}` |

| Parametro | Tipo | Descrizione |
| --- | --- | --- |
| `--enable-rename-algorithm-v2` | Booleano | Imposta la versione dell'algoritmo di ridenominazione EMRFS alla versione 2. Il valore predefinito è "true".<br><br>Esempio:<br><br>`%%configure`<br>`{`<br>`  "--enable-rename-algorithm-v2": "true"`<br>`}` |
| `--enable-glue-datacatalog` | Booleano | Consente di utilizzare Catalogo dati AWS Glue come metastore Apache Spark Hive.<br><br>Esempio:<br><br>`%%configure`<br>`{`<br>`  "--enable-glue-datacatalog": "true"`<br>`}` |
| `--enable-metrics` | Booleano | Abilita la raccolta di parametri per la profilatura del processo per l'esecuzione. Il valore predefinito è "false".<br><br>Esempio:<br><br>`%%configure`<br>`{`<br>`  "--enable-metrics": "true"`<br>`}` |

| Parametro | Tipo | Descrizione |
| --- | --- | --- |
| `--enable-continuous-cloudwatch-log` | Booleano | Abilita la registrazione continua in tempo reale per i processi AWS Glue. Il valore predefinito è "false".<br><br>Esempio:<br><br>`%%configure`<br>`{`<br>`  "--enable-continuous-cloudwatch-log": "true"`<br>`}` |
| `--enable-continuous-log-filter` | Booleano | Specifica un filtro standard o nessun filtro durante la creazione o la modifica di un processo abilitato per la registrazione continua. Il valore predefinito è "true".<br><br>Esempio:<br><br>`%%configure`<br>`{`<br>`  "--enable-continuous-log-filter": "true"`<br>`}` |

| Parametro | Tipo | Descrizione |
| --- | --- | --- |
| `--continuous-log-stream-prefix` | Stringa | Specificate un prefisso di Amazon CloudWatch log stream personalizzato per un job abilitato alla registrazione continua. L'impostazione predefinita è null.<br><br>Esempio:<br><br>`%%configure`<br>`{`<br>`  "--continuous-log-stream-prefix": "prefix"`<br>`}` |
| `--continuous-log-conversionPattern` | Stringa | Specifica un modello di log di conversione personalizzato per un processo abilitato per la registrazione continua. L'impostazione predefinita è null.<br><br>Esempio:<br><br>`%%configure`<br>`{`<br>`  "--continuous-log-conversionPattern": "pattern"`<br>`}` |

| Parametro | Tipo | Descrizione |
| --- | --- | --- |
| `--conf` | Stringa | Controlla i parametri di configurazione di Spark. È per casi d'uso avanzati. Utilizza `--conf` prima di ogni parametro. Esempio:<br>`%%configure`<br>`{`<br>`"--conf": "spark.hadoop.hive`<br>`.metastore.glue.catalogid=1`<br>`23456789012 --conf hive.meta`<br>`store.client.factory.class=`<br>`com.amazonaws.glue.catalog.`<br>`metastore.AWSGlueDataCatalo`<br>`gHiveClientFactory --conf`<br>` hive.metastore.schema.verif`<br>`ication=false"`<br>`}` |
| timeout | Int | Determina il tempo massimo di attesa che la sessione Spark deve attendere per una dichiarazione prima di terminarla.<br>`%%configure`<br>`{`<br>`"timeout": "30"`<br>`}` |
| dimensionamento automatico | Booleano | Determina se utilizzare o meno il dimensionamento automatico.<br>`%%configure`<br>`{`<br>`"--enable-auto-scaling": "true"`<br>`}` |

## Magic per processi Spark (ETL e flussi di dati)

| Name | Tipo | Description |
| --- | --- | --- |
| `%worker_type` | Stringa | Standard, G.1X, G.2X, G.4X, G.8X, G.12X, G.16X, R.1X, R.2X, R.4X o R.8X. Deve essere impostato anche `number_of_workers` . Il worker_type predefinito è G.1X. |
| `%connections` | List | Specifica un elenco separato da virgola di connessioni da utilizzare nella sessione.<br><br>Esempio:<br><br>`%connections my_rds_connection`<br>`dy_f =`<br>`glue_context.create_dynamic_frame.from_catalog(database='rds_tables', table_name='sales_table')` |
| `%extra_py_files` | List | Specifica un elenco separato da virgola di file Python aggiuntivi da Amazon S3. |
| `%extra_jars` | List | Specifica un elenco separato da virgola di jar aggiuntivi da includere nel cluster. |
| `%spark_conf` | Stringa | Specifica le configurazioni Spark personalizzate per la sessione. Ad esempio, `%spark_conf spark.serializer=org.apache.spark.serializer.KryoSerializer` . |

## Magic per processi Ray

| Name | Tipo | Description |
|---|---|---|
| `%min_workers` | Int | Il numero minimo di worker allocati a un processo Ray. Default: 1.<br><br>Ad esempio: `%min_workers 2` |
| `%object_memory_head` | Int | La percentuale di memoria libera sul nodo principale dell'istanza dopo un avvio a caldo. Minimo: 0 Massimo: 100<br><br>Ad esempio: `%object_memory_head 100` |
| %object_memory_worker | Int | La percentuale di memoria libera sui nodi worker dell'istanza dopo un avvio a caldo. Minimo: 0 Massimo: 100<br><br>Ad esempio: `%object_memory_worker 100` |

## Magic operativi

| Name | Tipo | Description |
|---|---|---|
| `%%sql` | Stringa | Eseguire codice SQL. Tutte le righe dopo che il magic `%%sql` iniziale verrà passato come parte del codice SQL.<br><br>Ad esempio: `%%sql select * from rds_tables.sales_table` |
| `%matplot` | Figura matplotlib | Visualizza i dati utilizzando la libreria matplotlib.<br><br>Esempio: |

| Name | Tipo | Description |
| --- | --- | --- |
| | | `import matplotlib.pyplot as plt`<br><br>`# Set X-axis and Y-axis values`<br>`x = [5, 2, 8, 4, 9]`<br>`y = [10, 4, 8, 5, 2]`<br><br>`# Create a bar chart`<br>`plt.bar(x, y)`<br><br>`# Show the plot`<br>`%matplot plt` |
| `%plotly` | Figura plotly | Visualizza i dati utilizzando la libreria plotly.<br><br>Esempio:<br><br>`import plotly.express as px`<br><br>`#Create a graphical figure`<br>`fig = px.line(x=["a","b","c"],`<br>` y=[1,3,2], title="sample`<br>` figure")`<br><br>`#Show the figure`<br>`%plotly fig` |

## Sessioni di denominazione

AWS Gluele sessioni interattive sono AWS risorse e richiedono un nome. I nomi devono essere univoci per ogni sessione e possono essere limitati dagli amministratori IAM. Per ulteriori informazioni, consulta Sessioni Interattive con IAM. Il kernel Jupyter genera automaticamente nomi di sessione univoci per tuo conto. Tuttavia, le sessioni possono essere nominate manualmente in due modi:

1. Utilizzando il file di AWS Command Line Interface configurazione che si trova in`~.aws/config`. Vedere Configurazione di AWS Config con. AWS Command Line Interface

2. Utilizzo dei magic `%session_id_prefix`. Per informazioni, consulta Magic supportati dalle sessioni interattive di AWS Glue per Jupyter.

Il nome di una sessione viene generato come segue:

- Quando vengono forniti il prefisso e session_id: il nome della sessione sarà {prefix}-{UUID}.
- Quando non viene fornito nulla: il nome della sessione sarà {UUID}.

Il prefisso dei nomi delle sessioni ti consente di riconoscere la tua sessione quando la elenchi nella console AWS CLI o.

## Specifica di un ruolo IAM per le sessioni interattive

È necessario specificare un ruolo AWS Identity and Access Management (IAM) da utilizzare AWS Glue con il codice ETL eseguito con sessioni interattive.

Il ruolo deve disporre delle stesse autorizzazioni IAM di quelle necessarie per l'esecuzione dei processi AWS Glue. Consulta Creazione di un ruolo IAM per AWS Glue per ulteriori informazioni sulla creazione di un ruolo per processi AWS Glue e sessioni interattive.

I ruoli IAM possono essere specificati in due modi:

- Utilizzando il file di AWS Command Line Interface configurazione che si trova in `~.aws/config` (consigliato). Per ulteriori informazioni, consulta Configurazione di sessioni con ~/.aws/config .

  > **Note**
  >
  > Quando il magic `%profile` è in uso, la configurazione per `glue_iam_role` di quel profilo è mantenuta.

- Usando il magic %iam_role. Per ulteriori informazioni, consulta Magic supportati dalle sessioni interattive di AWS Glue per Jupyter.

## Configurazione di sessioni con profili denominati

AWS Gluele sessioni interattive utilizzano le stesse credenziali di AWS Command Line Interface o boto3 e le sessioni interattive rispettano e funzionano con profili denominati come quelli AWS CLI

trovati in (`~/.aws/config`Linux e macOS) o (Windows). `%USERPROFILE%\.aws\config` Per ulteriori informazioni, consulta la sezione Using named profiles.

Le sessioni interattive sfruttano i vantaggi dei profili denominati consentendo al prefisso ruolo di servizio e ID sessione AWS Glue di essere specificato in un profilo. Per configurare un ruolo del profilo, aggiungi una riga per la `iam_role` chiave and/or `session_id_prefix` al tuo profilo denominato come mostrato di seguito. Il valore `session_id_prefix` non richiede virgolette. Ad esempio, se desideri aggiungere un `session_id_prefix`, inserisci il valore di `session_id_prefix=myprefix`.

```
[default]
region=us-east-1
aws_access_key_id=AKIAIOSFODNN7EXAMPLE
aws_secret_access_key=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
glue_iam_role=arn:aws:iam::<AccountID>:role/<GlueServiceRole>
session_id_prefix=<prefix_for_session_names>

[user1]
region=eu-west-1
aws_access_key_id=AKIAI44QH8DHBEXAMPLE
aws_secret_access_key=je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY
glue_iam_role=arn:aws:iam::<AccountID>:role/<GlueServiceRoleUser1>
session_id_prefix=<prefix_for_session_names_for_user1>
```

Se si dispone di un metodo personalizzato per generare credenziali, è anche possibile configurare il profilo in modo che utilizzi il parametro `credential_process` nel file `~/.aws/config`. Ad esempio:

```
[profile developer]
region=us-east-1
credential_process = "/Users/Dave/generate_my_credentials.sh" --username helen
```

Puoi trovare ulteriori dettagli sulle credenziali di approvvigionamento tramite il parametro `credential_process` qui: Credenziali di approvvigionamento con un processo esterno.

Se nel profilo che state usando non sono impostate una regione o un `iam_role`, dovete specificarli usando le `%region` e i `%iam_role` magic nella prima cella che eseguite.

# Convertire uno script o un taccuino in un lavoro AWS Glue

È possibile convertire uno script o un notebook in un processo AWS Glue in due modi:

- Utilizza nbconvert per convertire il tuo file di documento notebook Jupyter `.ipynb` in un file `.py`. Per ulteriori informazioni, consulta nbconvert: Convert Notebooks to other formats (nbconvert: convertire notebook in altri formati).
- Carica il file nei notebook di AWS Glue Studio.
  - Nella console AWS Glue Studio, dal menu di navigazione a sinistra scegliere Processi.
  - Nella sezione Create job (Crea processo), scegli Jupyter Notebook (Notebook Jupyter).
  - Nella sezione Options (Opzioni), scegli Upload and edit an existing notebook (Carica e modifica un notebook esistente).
  - Seleziona Choose file (Scegli file) per effettuare il caricamento di un file `.ipynb`.

# Utilizzo delle operazioni di streaming in sessioni interattive di AWS Glue

## Commutazione del tipo di sessione streaming

Utilizzo della configurazione magic sessioni interattive AWS Glue, `%streaming`, per definire il processo che si sta eseguendo e inizializzare una sessione interattiva in streaming.

## Flusso di input di campionamento per lo sviluppo interattivo

Uno strumento che abbiamo sviluppato per migliorare l'esperienza interattiva nelle sessioni AWS Glue interattive è l'aggiunta di un nuovo metodo `GlueContext` per ottenere un'istantanea di uno stream in modalità statica DynamicFrame. `GlueContext`consente di ispezionare, interagire e implementare il flusso di lavoro.

Con l'istanza di classe `GlueContext`, sarai in grado di localizzare il metodo `getSampleStreamingDynamicFrame`. Gli argomenti richiesti per questo metodo sono:

- `dataFrame`: Lo Spark Streaming DataFrame
- `options`: vedi le opzioni disponibili di seguito

Le opzioni disponibili includono:

- windowSize: viene chiamato anche durata del microbatch. Questo parametro determinerà la durata di attesa di una query di streaming dopo l'attivazione del batch precedente. Il valore del parametro deve essere inferiore a `pollingTimeInMs`.
- pollingTimeInMs: La durata totale dell'esecuzione del metodo. Esso spedirà almeno un micro batch per ottenere registri campione dal flusso di input.
- recordPollingLimit: Questo parametro consente di limitare il numero totale di record che verranno esaminati dallo stream.
- (Facoltativo) È possibile utilizzare anche `writeStreamFunction` per applicare questa funzione personalizzata a ogni funzione di campionamento del registro. Vedi di seguito alcuni esempi in Scala e Python.

Scala

```
val sampleBatchFunction = (batchDF: DataFrame, batchId: Long) => {//Optional but
 you can replace your own forEachBatch function here}
val jsonString: String = s"""{"pollingTimeInMs": "10000", "windowSize": "5
 seconds"}"""
val dynFrame = glueContext.getSampleStreamingDynamicFrame(YOUR_STREAMING_DF,
 JsonOptions(jsonString), sampleBatchFunction)
dynFrame.show()
```

Python

```
def sample_batch_function(batch_df, batch_id):
       //Optional but you can replace your own forEachBatch function here
options = {
            "pollingTimeInMs": "10000",
            "windowSize": "5 seconds",
        }
glue_context.getSampleStreamingDynamicFrame(YOUR_STREAMING_DF, options,
 sample_batch_function)
```

> **Note**
>
> Se il `DynFrame` d'esempio è vuoto, le ragioni possono essere varie:

- La fonte di streaming è impostata su "Più recente" e non sono stati inseriti nuovi dati durante il periodo di campionamento.
- Il tempo del polling non è sufficiente per elaborare i registri importati. I dati non verranno visualizzati a meno che l'intero batch non sia stato elaborato.

## Esecuzione di applicazioni di streaming in sessioni interattive

Nelle sessioni interattive AWS Glue, è possibile eseguire un'applicazione di streaming AWS Glue nello stesso modo in cui si creerebbe un'applicazione di streaming nella Console AWS Glue. Poiché le sessioni interattive sono basate su sessione, l'individuazione di eccezioni nel runtime non provoca l'interruzione della sessione. Ora abbiamo l'ulteriore vantaggio di sviluppare iterativamente la funzione batch. Ad esempio:

```
def batch_function(data_frame, batch_id):
    log.info(data_frame.count())
    invalid_method_call()
glueContext.forEachBatch(frame=streaming_df, batch_function = batch_function, options =
 {**})
```

Nell'esempio precedente, abbiamo incluso un utilizzo non valido di un metodo e a differenza dei normali processi AWS Glue che usciranno dall'intera applicazione, le definizioni e il contesto di codifica dell'utente vengono conservati interamente e la sessione è ancora operativa. Non è necessario avviare un nuovo cluster e rieseguire tutte le trasformazioni precedenti. Ciò consente di concentrarsi sull'iterazione rapida delle implementazioni delle funzioni batch per ottenere risultati desiderati.

È importante notare che le sessioni interattive valutano ogni istruzione in modo bloccante per far sì che la sessione esegua una sola istruzione alla volta. Poiché le query di streaming sono continue e senza fine, le sessioni con query di streaming attive non saranno in grado di gestire alcuna istruzione di follow-up a meno che non vengano interrotte. Puoi emettere il comando di interruzione direttamente da Jupyter Notebook e il nostro kernel gestirà la cancellazione per te.

Prendi come esempio la seguente sequenza di istruzioni in attesa dell'esecuzione:

```
Statement 1:
      val number = df.count()
```

```
      #Spark Action with deterministic result
      Result: 5

Statement 2:
      streamingQuery.start().awaitTermination()
      #Spark Streaming Query that will be executing continously
      Result: Constantly updated with each microbatch

Statement 3:
      val number2 = df.count()
      #This will not be executed as previous statement will be running indefinitely
```

# Prezzi delle sessioni interattive di AWS Glue

AWS addebiti per le sessioni AWS Glue interattive in base alla durata della sessione e al numero di unità di elaborazione dati (DPU) utilizzate. Ti viene addebitata una tariffa oraria per il numero di carichi di lavoro DPUs utilizzati per eseguire i carichi di lavoro, fatturati in incrementi di un secondo. AWS Glue le sessioni interattive assegnano un valore predefinito di 5 DPUs e ne richiedono almeno 2. DPUs È inoltre prevista una durata di fatturazione minima di un minuto per ogni sessione interattiva. Per visualizzare le AWS Glue tariffe e gli esempi di prezzo o per stimare i costi utilizzando il Calcolatore AWS dei prezzi, consulta AWS Glue la pagina dei prezzi.

## Configura le sessioni interattive di AWS Glue

Puoi usare Jupyter magics nella tua sessione AWS interattiva per modificare i parametri di sessione e configurazione. Le magie sono brevi comandi con il prefisso % all'inizio delle celle di Jupyter che forniscono un modo semplice e veloce per aiutarti a controllare l'ambiente. Ad esempio, se desideri modificare il numero di worker assegnati al tuo lavoro dal valore predefinito di 5 a 10, puoi specificare `%number_of_workers 10`. Se desideri configurare la sessione in modo che si interrompa dopo 10 minuti di inattività anziché il 2880 predefinito, puoi specificare `%idle_timeout 10`.

Per l'elenco completo delle AWS magics disponibili, consulta Configurazione delle sessioni AWS interattive per Jupyter e notebook. AWS Glue Studio

# Sviluppo e test degli script di lavoro AWS Glue a livello locale

Quando sviluppi e testi gli script di lavoro di AWS Glue for Spark, sono disponibili diverse opzioni:

- AWS Console Glue Studio
  - Visual editor (Editor visivo)
  - Editor di script
  - AWS Taccuino Glue Studio
- Sessioni interattive
  - Jupyter Notebook
- immagine Docker
  - Sviluppo locale
  - Sviluppo remoto

Puoi scegliere una delle opzioni sopra elencate in base alle tue esigenze.

Se preferisci l'assenza di codice o una minore esperienza di programmazione, l'editor visivo di AWS Glue Studio è una buona scelta.

Se preferisci un'esperienza di notebook interattiva, il notebook AWS Glue Studio è una buona scelta. Per ulteriori informazioni, vedere Uso dei notebook con Glue Studio e AWS Glue. AWS Se desideri utilizzare il tuo ambiente locale, le sessioni interattive sono un'ottima scelta. Per ulteriori informazioni, consulta Utilizzo di sessioni interattive con AWS Glue.

Se preferisci un'esperienza di local/remote sviluppo, l'immagine Docker è una buona scelta. Questo ti aiuta a sviluppare e testare gli script di lavoro di AWS Glue for Spark ovunque preferisci senza incorrere nei costi di Glue AWS .

Se preferisci lo sviluppo locale senza Docker, installare localmente la directory della libreria AWS Glue ETL è una buona scelta.

## Sviluppo con AWS Glue Studio

L'editor visivo di AWS Glue Studio è un'interfaccia grafica che semplifica la creazione, l'esecuzione e il monitoraggio di lavori di estrazione, trasformazione e caricamento (ETL) in AWS Glue. Puoi comporre visivamente flussi di lavoro di trasformazione dei dati ed eseguirli senza problemi sul motore ETL serverless basato su Apache AWS Spark di Glue. È possibile esaminare lo schema e i risultati dei dati in ogni fase del processo. Per ulteriori informazioni, consultare la Guida per l'utente di AWS Glue Studio.

## Sviluppo tramite le sessioni interattive

Le sessioni interattive consentono di creare e testare le applicazioni dall'ambiente che preferisci. Per ulteriori informazioni, consulta Utilizzo di sessioni interattive con AWS Glue.

## Sviluppa e testa i AWS Glue lavori localmente utilizzando un'immagine Docker

Per una piattaforma di dati pronta per la produzione, il processo di sviluppo e la CI/CD pipeline dei AWS Glue lavori sono un argomento chiave. Puoi sviluppare e testare AWS Glue lavori in modo flessibile in un contenitore Docker. AWS Glue ospita immagini Docker su Docker Hub per configurare l'ambiente di sviluppo con utilità aggiuntive. Puoi usare il tuo IDE, notebook o REPL preferito utilizzando la libreria ETL. AWS Glue Questo argomento descrive come sviluppare e testare i job della AWS Glue versione 5.0 in un contenitore Docker utilizzando un'immagine Docker.

### Immagini Docker disponibili

Le seguenti immagini Docker sono disponibili per AWS Glue Amazon ECR:.

- Per la AWS Glue versione 5.0: `public.ecr.aws/glue/aws-glue-libs:5`
- Per la AWS Glue versione 4.0: `public.ecr.aws/glue/aws-glue-libs:glue_libs_4.0.0_image_01`
- Per la AWS Glue versione 3.0: `public.ecr.aws/glue/aws-glue-libs:glue_libs_3.0.0_image_01`
- Per la AWS Glue versione 2.0: `public.ecr.aws/glue/aws-glue-libs:glue_libs_2.0.0_image_01`

> **Note**
>
> AWS Glue Le immagini Docker sono compatibili sia con x86_64 che con arm64.

In questo esempio, utilizziamo public.ecr.aws/glue/aws-glue-libs:5 ed eseguiamo il container su un computer locale (Mac, Windows o Linux). Questa immagine del contenitore è stata testata per AWS Glue i job Spark della versione 5.0. L'immagine contiene quanto segue:

- Amazon Linux 2023

- AWS Glue Libreria ETL
- Apache Spark 3.5.4
- Librerie in formato tabella aperta; Apache Iceberg 1.7.1, Apache Hudi 0.15.0 e Delta Lake 3.3.0
- AWS Glue Client Data Catalog
- Amazon Redshift connettore per Apache Spark
- Amazon DynamoDB connettore per Apache Hadoop

Per configurare il container, estrai l'immagine da ECR Public Gallery e quindi esegui il container. Questo argomento illustra come eseguire il container con i seguenti metodi, a seconda delle esigenze:

- `spark-submit`
- REPL shell (`pyspark`)
- `pytest`
- Visual Studio Code

## Prerequisiti

Prima di iniziare, verifica che Docker sia installato e che il daemon Docker sia in esecuzione. Per istruzioni sull'installazione, consulta la documentazione Docker per Mac o Linux. La macchina su cui è in esecuzione il Docker ospita il contenitore. AWS Glue Inoltre, verifica di avere almeno 7 GB di spazio su disco per l'immagine sull'host su cui è in esecuzione Docker.

Per ulteriori informazioni sulle restrizioni relative allo sviluppo locale AWS Glue del codice, consulta Restrizioni allo sviluppo locale.

### Configurazione AWS

Per abilitare le chiamate AWS API dal contenitore, configura AWS le credenziali seguendo i passaggi seguenti. Nelle sezioni seguenti, utilizzeremo questo profilo AWS denominato.

1. Crea un profilo con AWS nome.
2. Apri `cmd` su Windows o su un terminale Mac/Linux ed esegui il seguente comando in un terminale:

```
PROFILE_NAME="<your_profile_name>"
```

Nelle sezioni seguenti, utilizziamo questo profilo AWS denominato.

Se stai eseguendo Docker su Windows, scegli l'icona Docker (fai clic con il pulsante destro del mouse) e scegli Passa ai container Linux prima di estrarre l'immagine.

Esegui il comando seguente per estrarre l'immagine da ECR Public:

```
docker pull public.ecr.aws/glue/aws-glue-libs:5
```

## Esegui il container

Ora puoi eseguire un container utilizzando questa immagine. Puoi scegliere una delle opzioni seguenti in base alle tue esigenze.

spark-submit

È possibile eseguire uno script di AWS Glue lavoro eseguendo il `spark-submit` comando sul contenitore.

1. Scrivi il tuo script e salvalo come `sample.py` nell'esempio riportato di seguito, quindi salvalo nella directory `/local_path_to_workspace/src/` utilizzando i seguenti comandi:

   ```
   $ WORKSPACE_LOCATION=/local_path_to_workspace
   $ SCRIPT_FILE_NAME=sample.py
   $ mkdir -p ${WORKSPACE_LOCATION}/src
   $ vim ${WORKSPACE_LOCATION}/src/${SCRIPT_FILE_NAME}
   ```

2. Queste variabili vengono usate nel comando docker run riportato di seguito. Il codice di esempio (sample.py) utilizzato nel comando spark-submit riportato di seguito è incluso nell'appendice alla fine di questo argomento.

   Esegui il comando seguente per eseguire il comando `spark-submit` sul container per inviare una nuova applicazione Spark:

   ```
   $ docker run -it --rm \
       -v ~/.aws:/home/hadoop/.aws \
       -v $WORKSPACE_LOCATION:/home/hadoop/workspace/ \
       -e AWS_PROFILE=$PROFILE_NAME \
       --name glue5_spark_submit \
       public.ecr.aws/glue/aws-glue-libs:5 \
   ```

```
    spark-submit /home/hadoop/workspace/src/$SCRIPT_FILE_NAME
```

3. (Facoltativo) Configura `spark-submit` in modo che corrisponda all'ambiente in uso. Ad esempio, puoi trasferire le dipendenze con la configurazione `--jars`. Per ulteriori informazioni, consulta Caricamento dinamico delle proprietà Spark nella documentazione di Spark.

Shell REPL (Pyspark)

Puoi eseguire la shell REPL (`read-eval-print loops`) per lo sviluppo interattivo. Esegui il comando seguente per eseguire il PySpark comando sul contenitore per avviare la shell REPL:

```
$ docker run -it --rm \
    -v ~/.aws:/home/hadoop/.aws \
    -e AWS_PROFILE=$PROFILE_NAME \
    --name glue5_pyspark \
    public.ecr.aws/glue/aws-glue-libs:5 \
    pyspark
```

Vedrai l'output seguente:

```
Python 3.11.6 (main, Jan  9 2025, 00:00:00) [GCC 11.4.1 20230605 (Red Hat 11.4.1-2)] on
 linux
Type "help", "copyright", "credits" or "license" for more information.
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use
 setLogLevel(newLevel).
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 3.5.4-amzn-0
      /_/

Using Python version 3.11.6 (main, Jan  9 2025 00:00:00)
Spark context Web UI available at None
Spark context available as 'sc' (master = local[*], app id = local-1740643079929).
SparkSession available as 'spark'.
>>>
```

Con questa shell REPL è possibile programmare ed eseguire test in modo interattivo.

Pytest

Per i test unitari, puoi utilizzare `pytest` gli script di lavoro di AWS Glue Spark. Esegui i comandi seguenti per la preparazione.

```
$ WORKSPACE_LOCATION=/local_path_to_workspace
$ SCRIPT_FILE_NAME=sample.py
$ UNIT_TEST_FILE_NAME=test_sample.py
$ mkdir -p ${WORKSPACE_LOCATION}/tests
$ vim ${WORKSPACE_LOCATION}/tests/${UNIT_TEST_FILE_NAME}
```

Esegui il seguente comando per eseguire `pytest` usando `docker run`:

```
$ docker run -i --rm \
    -v ~/.aws:/home/hadoop/.aws \
    -v $WORKSPACE_LOCATION:/home/hadoop/workspace/ \
    --workdir /home/hadoop/workspace \
    -e AWS_PROFILE=$PROFILE_NAME \
    --name glue5_pytest \
    public.ecr.aws/glue/aws-glue-libs:5 \
    -c "python3 -m pytest --disable-warnings"
```

Una volta che `pytest` avrà terminato l'esecuzione dei test di unità, il risultato sarà simile al seguente:

```
============================= test session starts ==============================
platform linux -- Python 3.11.6, pytest-8.3.4, pluggy-1.5.0
rootdir: /home/hadoop/workspace
plugins: integration-mark-0.2.0
collected 1 item

tests/test_sample.py .                                                   [100%]

======================== 1 passed, 1 warning in 34.28s =========================
```

Configurazione del container per l'utilizzo di Visual Studio Code

Per configurare il container con Visual Studio Code, completa i seguenti passaggi:

1. Installare Visual Studio Code.
2. Installare Python.

3. Installare Visual Studio Code Remote - Containers (Visual Studio Code Remote - Container)
4. Aprire la cartella del WorkSpace in Visual Studio Code.
5. Premi `Ctrl+Shift+P` (Windows/Linux) o `Cmd+Shift+P` (Mac).
6. Tipo `Preferences: Open Workspace Settings (JSON)`.
7. Premere Invio.
8. Incollare il seguente codice JSON e salvarlo.

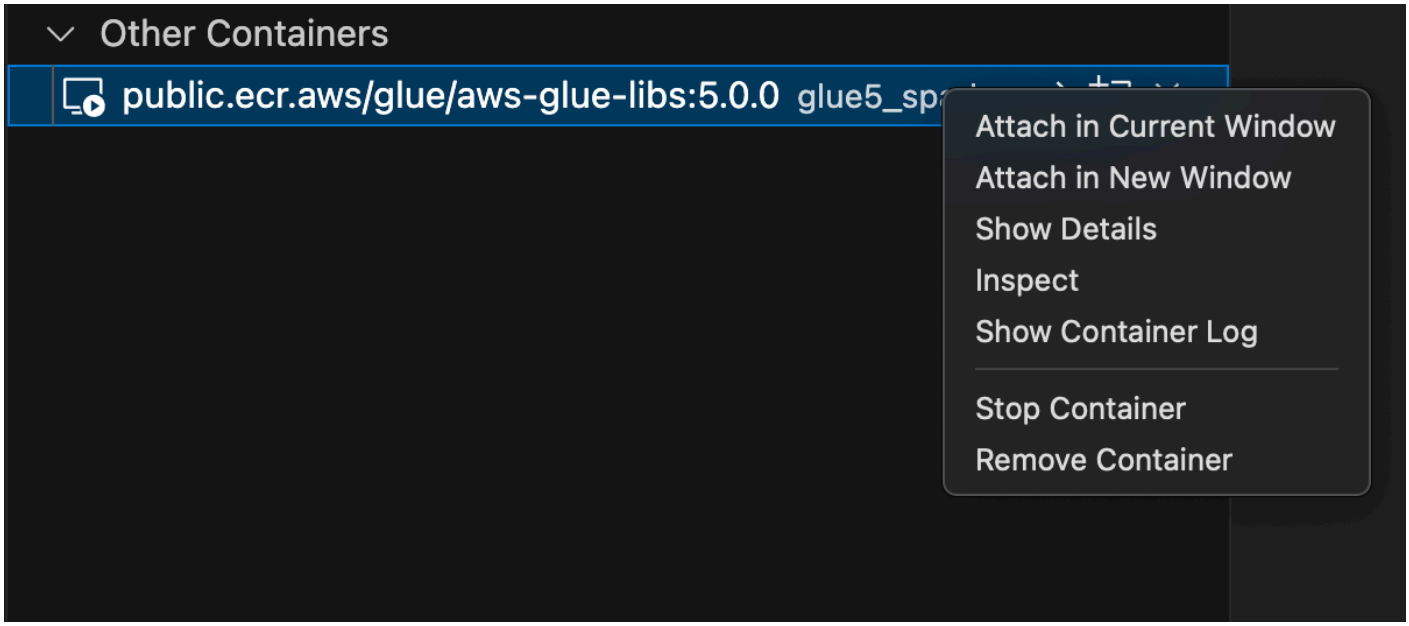
```
{
    "python.defaultInterpreterPath": "/usr/bin/python3.11",
    "python.analysis.extraPaths": [
        "/usr/lib/spark/python/lib/py4j-0.10.9.7-src.zip:/usr/lib/spark/python/:/usr/
lib/spark/python/lib/",
    ]
}
```

Per configurare il container:

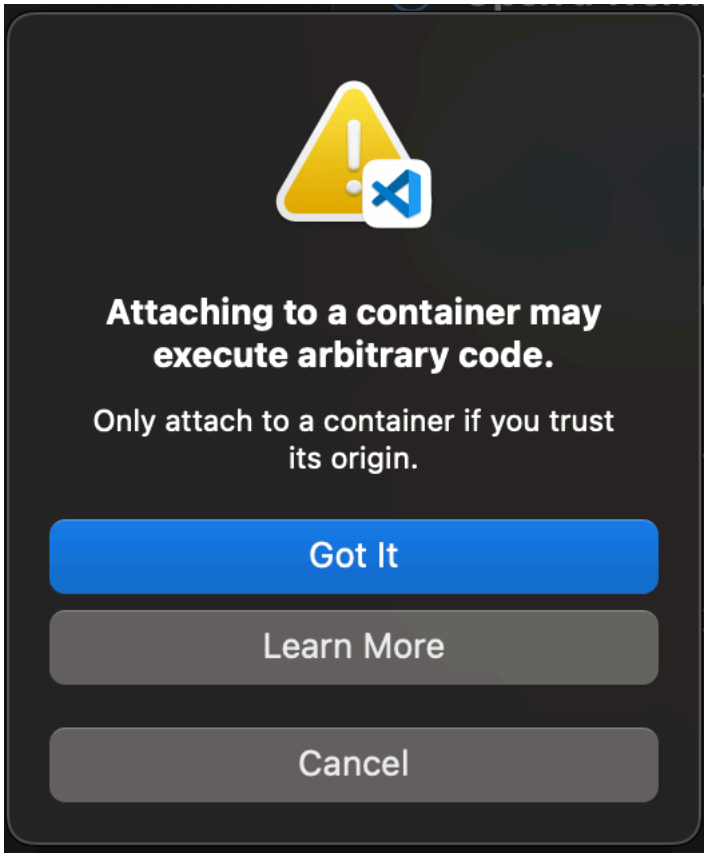
1. Eseguire il container Docker.

```
$ docker run -it --rm \
    -v ~/.aws:/home/hadoop/.aws \
    -v $WORKSPACE_LOCATION:/home/hadoop/workspace/ \
    -e AWS_PROFILE=$PROFILE_NAME \
    --name glue5_pyspark \
    public.ecr.aws/glue/aws-glue-libs:5 \
    pyspark
```

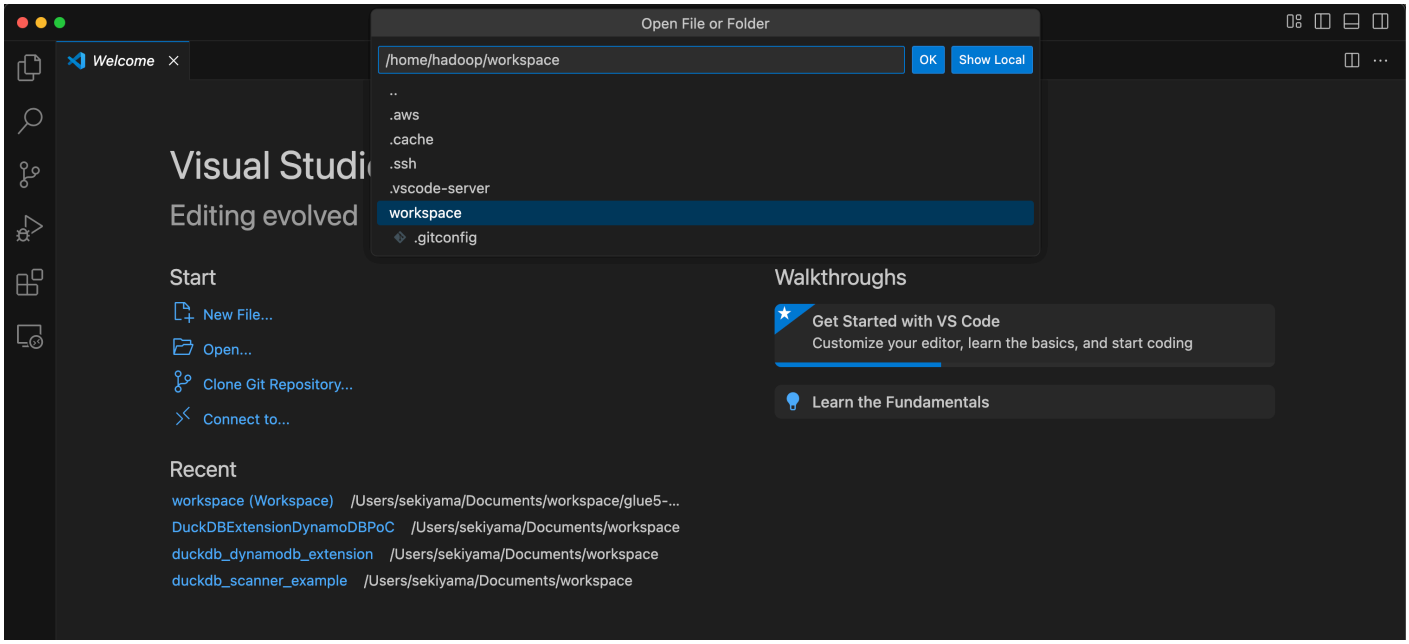
2. Avviare Visual Studio Code.
3. Scegliere Remote Explorer nel menu a sinistra, quindi `amazon/aws-glue-libs:glue_libs_4.0.0_image_01`.
4. Right-click e scegli Allega nella finestra corrente.




5. Se viene visualizzata la seguente finestra di dialogo, seleziona Capito.

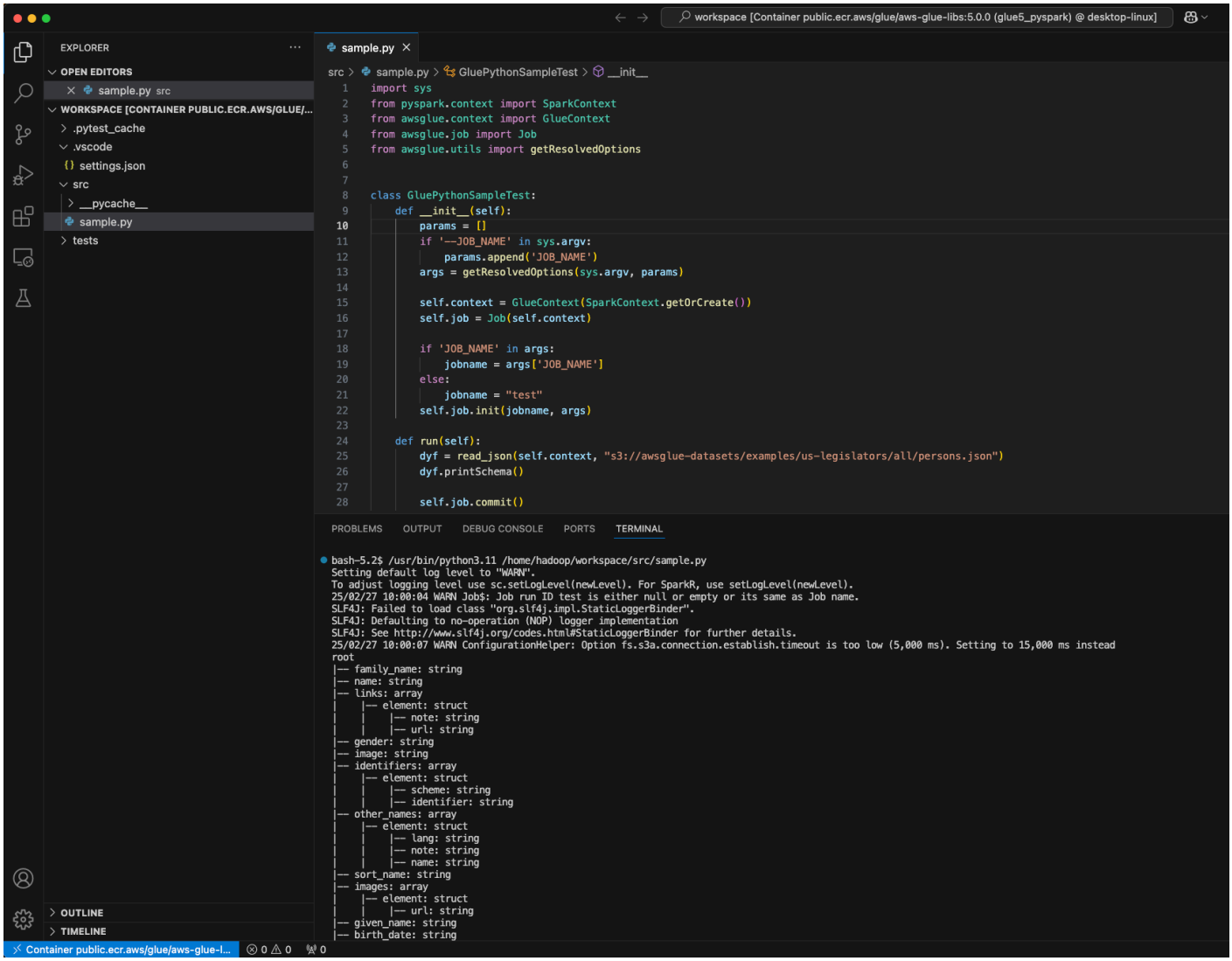



6. Aprire `/home/handoop/workspace/`.




7. Crea uno AWS Glue PySpark script e scegli Esegui.

   Visualizzerai l'esecuzione corretta dello script.




## Modifiche tra l'immagine Docker AWS Glue 4.0 e AWS Glue 5.0

Le principali modifiche tra l'immagine Docker AWS Glue 4.0 e AWS Glue 5.0:

- Nella AWS Glue versione 5.0, esiste un'unica immagine contenitore per i lavori in batch e in streaming. Ciò differisce da Glue 4.0, dove era presente un'immagine per i processi in batch e un'altra per quelli in streaming.
- Nella AWS Glue versione 5.0, il nome utente predefinito del contenitore èhadoop. Nella AWS Glue versione 4.0, il nome utente predefinito eraglue_user.
- Nella AWS Glue versione 5.0, diverse librerie aggiuntive, JupyterLab tra cui Livy, sono state rimosse dall'immagine. Puoi installarle manualmente.

- Nella AWS Glue versione 5.0, tutte le librerie Iceberg, Hudi e Delta sono precaricate di default e la variabile di ambiente `DATALAKE_FORMATS` non è più necessaria. Prima della AWS Glue 4.0, la variabile `DATALAKE_FORMATS` di ambiente veniva utilizzata per specificare quali formati di tabella specifici dovevano essere caricati.

L'elenco precedente è specifico dell'immagine Docker. Per ulteriori informazioni sugli aggiornamenti AWS Glue 5.0, consulta Introduzione alla AWS Glue versione 5.0 per Apache Spark e Migrazione dei job AWS Glue for Spark alla versione 5.0. AWS Glue

## Considerazioni

Tieni presente che le seguenti funzionalità non sono supportate quando usi l'immagine del AWS Glue contenitore per sviluppare script di lavoro localmente.

- Segnalibri di processo
- AWS Glue Parquet writer (utilizzo del formato Parquet in AWS Glue)
- FillMissingValues trasformare
- FindMatches trasformare
- Lettore CSV SIMD vettorializzato
- La proprietà personalizzata JdbcDriverS3Path per caricare il driver JDBC dal percorso Amazon S3
- AWS Glue Qualità dei dati
- Rilevamento di dati sensibili
- AWS Lake Formation vendita di credenziali basata su autorizzazioni

## Appendice: aggiunta di driver JDBC e librerie Java

Per aggiungere un driver JDBC attualmente non disponibile nel container, puoi creare una nuova directory nel workspace con i file JAR necessari e montare la directory in `/opt/spark/jars/` nel comando docker run. I file JAR presenti nella director `/opt/spark/jars/` all'interno del container vengono aggiunti automaticamente al percorso di classe Spark e saranno disponibili per l'uso durante l'esecuzione del processo.

Ad esempio, usa il seguente comando docker run per aggiungere i jar dei driver JDBC alla shell REPL. PySpark

```
docker run -it --rm \
```

```
    -v ~/.aws:/home/hadoop/.aws \
    -v $WORKSPACE_LOCATION:/home/hadoop/workspace/ \
    -v $WORKSPACE_LOCATION/jars/:/opt/spark/jars/ \
    --workdir /home/hadoop/workspace \
    -e AWS_PROFILE=$PROFILE_NAME \
    --name glue5_jdbc \
    public.ecr.aws/glue/aws-glue-libs:5 \
    pyspark
```

Come evidenziato in Considerazioni, l'opzione di `customJdbcDriverS3Path` connessione non può essere utilizzata per importare un driver JDBC personalizzato da Amazon S3 nelle immagini dei container. AWS Glue

## Endpoint di sviluppo

> **Note**
>
> L'esperienza su console per gli endpoint di sviluppo è stata rimossa a partire dal 31 marzo 2023. La creazione, l'aggiornamento e il monitoraggio degli endpoint di sviluppo sono ancora disponibili tramite l'API endpoint di sviluppo e la CLI AWS Glue.

Consigliamo vivamente di eseguire la migrazione dagli endpoint di sviluppo alle sessioni interattive per i motivi elencati di seguito. Per le azioni necessarie per la migrazione dagli endpoint di sviluppo alle sessioni interattive, consulta Migrazione dagli endpoint di sviluppo alle sessioni interattive.

| Description | Endpoint dev | Sessioni interattive |
| --- | --- | --- |
| Supporto per la versione Glue | Supporta AWS Glue versioni 0.9 e 1.0 | Supporta AWS Glue versione 2.0 e successive |
| Gli endpoint di sviluppo non sono disponibili nelle Regioni Asia Pacifico (Giacarta) (`ap-southeast-3` ), Medio Oriente (Emirati Arabi Uniti) (`me-central-1` ), Europa (Spagna) (`eu-south-2` ), | Le sessioni interattive non sono attualmente disponibili in Medio Oriente (Emirati Arabi Uniti) (`me-central-1` ), ma potrebbero essere rese disponibili in futuro | |

| Description | Endpoint dev | Sessioni interattive |
| --- | --- | --- |
| Europa (Zurigo) (`eu-central-2` ) o altre nuove Regioni in futuro | | |
| Metodo di accesso al cluster Spark | Supporta SSH, shell REPL, notebook Jupyter, IDE (ad es.) PyCharm | supporta AWS Glue Studio notebook, notebook Jupyter, vari notebook IDEs (ad esempio Visual Studio Code) e AI PyCharm SageMaker |
| Tempo per la prima query | Richiede 10-15 minuti per la configurazione di un cluster Spark | Può richiedere fino a 1 minuto per la configurazione di un cluster Spark temporaneo |
| Modello dei prezzi | AWS addebiti per gli endpoint di sviluppo in base all'ora in cui viene fornito l'endpoint e al numero di. DPUs Gli endpoint di sviluppo non scadono. È prevista una durata di fatturazione minima di 10 minuti per ogni endpoint di sviluppo fornito. Inoltre, vengono AWS addebitati i costi per i notebook Jupyter su EC2 istanze Amazon e i notebook SageMaker AI quando li configuri con endpoint di sviluppo. | AWS i costi per le sessioni interattive si basano sul tempo in cui la sessione è attiva e sul numero di sessioni interattive con timeout di DPUs inattività configurabili. AWS Glue Studioi notebook forniscono un'interfaccia integrata per le sessioni interattive e sono offerti senza costi aggiuntivi. È prevista una durata di fatturazione minima di 1 minuto per ogni sessione interattiva. I notebook AWS Glue Studio forniscono un'interfaccia integrata per sessioni interattive e sono offerti senza costi aggiuntivi |
| Esperienza della console | Disponibile solo tramite CLI e API | Disponibile tramite AWS Glue console, CLI e APIs |

# Migrazione dagli endpoint di sviluppo alle sessioni interattive

Utilizza la seguente lista di controllo per determinare il metodo appropriato per eseguire la migrazione dagli endpoint di sviluppo alle sessioni interattive.

Il tuo script dipende da funzionalità specifiche di AWS Glue 0.9 o 1.0 (ad esempio, HDFS, YARN, ecc.)?

Se la risposta è sì, consulta Migrazione dei processi AWS Glue a AWS Glue versione 3.0 per scoprire come migrare da Glue 0.9 o 1.0 a Glue 3.0 e versioni successive.

Quale metodo utilizzi per accedere all'endpoint di sviluppo?

| Se usi questo metodo | Allora completa le seguenti operazioni |
| --- | --- |
| SageMaker Notebook AI, notebook Jupyter o JupyterLab | Esegui la migrazione al notebook AWS Glue Studio scaricando i file `.ipynb` su Jupyter e crea un nuovo processo del notebook AWS Glue Studio caricando il file `.ipynb`. In alternativa, puoi anche usare SageMaker AI Studio e selezionare il kernel. AWS Glue |
| Notebook Zeppelin | Converti il notebook in un notebook Jupyter manualmente copiando e incollando il codice o automaticamente utilizzando un convertitore di terze parti come ze2nb. Quindi, usa il notebook nel AWS Glue Studio notebook o in SageMaker AI Studio. |
| IDE | Vedi AWS GlueLavori di creazione con PyCharm utilizzo di sessioni AWS Glue interattive o Utilizzo di sessioni interattive con Microsoft Visual Studio Code. |
| REPL | Installa `aws-glue-session package` in locale, quindi esegui il comando seguente:<br>• Per Python: `jupyter console --kernal glue_pyspark` |

| Se usi questo metodo | Allora completa le seguenti operazioni |
| --- | --- |
| | • Per Scala: `jupyter console --kernal glue_spark` |
| SSH | Non esiste un'opzione corrispondente per le sessioni interattive. In alternativa, puoi usare un'immagine Docker. Per ulteriori informazioni, consulta Sviluppare utilizzando un'immagine Docker |

Le seguenti sezioni forniscono informazioni sull'utilizzo degli endpoint di sviluppo per lo sviluppo di processi in AWS Glue versione 1.0.

Argomenti


- Utilizzo di endpoint per lo sviluppo di script
- Gestione di notebook


## Utilizzo di endpoint per lo sviluppo di script

> **Note**
>
> Gli endpoint di sviluppo sono supportati solo per le versioni AWS Glue precedenti alla 2.0. Per un ambiente interattivo in cui è possibile creare e testare script ETL, utilizza Notebooks on Studio. AWS Glue

AWS Glue può creare un ambiente, noto come endpoint di sviluppoche puoi utilizzare per sviluppare e testare iterativamente gli script ETL (Extract, Transform and Load). Puoi creare, modificare ed eliminare endpoint di sviluppo utilizzando la console AWS Glue o l'API.

### Gestione dell'ambiente di sviluppo

Quando crei un endpoint di sviluppo, devi fornire i valori di configurazione per effettuare il provisioning dell'ambiente di sviluppo. Questi valori indicano a AWS Glue come configurare la rete in modo da poter accedere all'endpoint di sviluppo in tutta sicurezza e che l'endpoint possa accedere agli archivi dati.

Quindi, crea un notebook che si connette all'endpoint di sviluppo e utilizza il notebook per creare e testare lo script ETL. Quando sei soddisfatto dei risultati del processo di sviluppo, è possibile creare un processo ETL che esegue il tuo script. Con questo processo è possibile aggiungere funzioni ed eseguire il debug dello script in modo interattivo.

Segui i tutorial in questa sezione per ulteriori informazioni su come utilizzare l'endpoint di sviluppo con i notebook.

Argomenti


- Flusso di lavoro degli endpoint di sviluppo
- Come funzionano gli endpoint di AWS Glue sviluppo con i notebook SageMaker
- Aggiunta di un endpoint di sviluppo
- Accesso all'endpoint di sviluppo
- Tutorial: configura un notebook Jupyter JupyterLab per testare ed eseguire il debug degli script ETL
- Tutorial: utilizza un notebook SageMaker AI con il tuo endpoint di sviluppo
- Tutorial: utilizzo di una shell (interprete di comandi) REPL con l'endpoint di sviluppo
- Tutorial: configura un PyCharm professionista con un endpoint di sviluppo
- Configurazione avanzata: condivisione degli endpoint di sviluppo tra più utenti


## Flusso di lavoro degli endpoint di sviluppo

Per utilizzare un endpoint di sviluppo AWS Glue, puoi seguire questo flusso di lavoro.

1. Crea un endpoint di sviluppo utilizzando l'API. Questo endpoint viene lanciato nel tuo cloud privato virtuale (VPC, Virtual Private Cloud) con i gruppi di sicurezza da te definiti.
2. L'API è in grado di eseguire il polling dell'endpoint di sviluppo fino a quando non viene allocato ed è pronto per l'utilizzo. Quando è pronto, connettiti all'endpoint di sviluppo utilizzando uno dei seguenti metodi per creare e testare gli script AWS Glue.
   - Crea un taccuino SageMaker AI nel tuo account. Per ulteriori informazioni su come creare un notebook, consulta the section called "Creazione di codice con notebook AWS Glue Studio".
   - È possibile aprire una finestra terminale per connettersi direttamente a un endpoint di sviluppo.

- Se hai l'edizione professionale dell'IDE JetBrains PyCharm Python, collegala a un endpoint di sviluppo e usala per sviluppare in modo interattivo. Se inserisci `pydevd` istruzioni nello script, PyCharm può supportare punti di interruzione remoti.

3. Una volta terminato il debug e il test dell'endpoint di sviluppo, è possibile eliminarlo.

## Come funzionano gli endpoint di AWS Glue sviluppo con i notebook SageMaker

Uno dei modi più comuni per accedere agli endpoint di sviluppo consiste nell'utilizzare Jupyter sui notebook. SageMaker Il notebook Jupyter è un'applicazione web open source ampiamente utilizzata nella visualizzazione, nell'analisi dei dati, nel machine learning, ecc. Un AWS Glue SageMaker notebook offre un'esperienza con i notebook Jupyter con endpoint di sviluppo. AWS Glue Nel AWS Glue SageMaker notebook, l'ambiente notebook Jupyter è preconfigurato con un plug-in Jupyter open source per SparkMagicinviare i lavori Spark a un cluster Spark remoto. Apache Livy è un servizio che consente l'interazione con un cluster Spark remoto tramite una REST API. Nel AWS Glue SageMaker notebook, SparkMagic è configurato per richiamare l'API REST su un server Livy in esecuzione su un endpoint di sviluppo. AWS Glue

Il seguente flusso di testo spiega come funziona ogni componente:

AWS Glue SageMaker notebook: (Jupyter → SparkMagic) → (rete) → endpoint di AWS Glue sviluppo: (Apache Livy → Apache Spark)

Una volta eseguito lo script Spark scritto in ogni paragrafo su un notebook Jupyter, il codice Spark viene inviato al server Livy tramite SparkMagic, quindi un job Spark chiamato «Livy-session-n» viene eseguito sul cluster Spark. Questo processo è denominato sessione Livy. Il processo Spark verrà eseguito mentre la sessione del notebook è attiva. Il processo Spark verrà terminato quando il kernel Jupyter viene arrestato dal notebook o quando la sessione è scaduta. Viene avviato un processo Spark per ogni file notebook (.ipynb).

È possibile utilizzare un singolo endpoint di sviluppo con più istanze di notebook. AWS Glue SageMaker È possibile creare più file di notebook in ogni istanza del SageMaker notebook. Quando apri un file di notebook ed esegui i paragrafi, viene avviata una sessione Livy per ogni file del taccuino sul cluster Spark tramite. SparkMagic Ogni sessione Livy corrisponde a un singolo processo Spark.

Comportamento predefinito per endpoint e notebook di AWS Glue sviluppo SageMaker

I processi Spark vengono eseguiti in base alla configurazione di Spark. Esistono diversi modi per impostare la configurazione di Spark (ad esempio, la configurazione del cluster Spark, la configurazione SparkMagic di Spark, ecc.).

Per impostazione predefinita, Spark alloca le risorse del cluster a una sessione Livy in base alla configurazione del cluster Spark. Nell'endpoint di sviluppo AWS Glue, la configurazione del cluster dipende dal tipo di worker. Ecco una tabella che spiega le configurazioni comuni per tipo di worker.

| | Standard | G.1X | G.2X |
|---|---|---|---|
| `spark.driver.memory` | 5G | 10G | 20G |
| `spark.executor.memory` | 5G | 10G | 20G |
| `spark.executor.cores` | 4 | 8 | 16 |
| `spark.dynamicAllocation.enabled` | TRUE | TRUE | TRUE |

Il numero massimo di executor Spark viene calcolato automaticamente in base alla combinazione di DPU (o `NumberOfWorkers`) e il tipo di dipendente.

| | Standard | G.1X | G.2X |
|---|---|---|---|
| Il numero massimo | `(DPU - 1) * 2 - 1` | `(NumberOfWorkers - 1)` | `(NumberOfWorkers - 1)` |

| | Standard | G.1X | G.2X |
|---|---|---|---|
| di executor Spark | | | |

Ad esempio, se l'endpoint di sviluppo ha 10 worker e il tipo di worker è `G.1X`, allora si avranno 9 executor Spark e l'intero cluster avrà 90G di memoria dell'executor, poiché ogni executor avrà 10G di memoria.

Indipendentemente dal tipo di worker specificato, verrà attivata l'allocazione dinamica delle risorse Spark. Se un set di dati è abbastanza grande, Spark potrebbe allocare tutti gli executor a una singola sessione Livy poiché `spark.dynamicAllocation.maxExecutors` non è configurata per impostazione predefinita. Ciò significa che altre sessioni Livy sullo stesso endpoint di sviluppo attenderanno per l'avvio di nuovi executor. Se il set di dati è piccolo, Spark sarà in grado di allocare gli esecutori a più sessioni Livy contemporaneamente.

> **Note**
>
> Per ulteriori informazioni sull'allocazione delle risorse in diversi casi d'uso e su come impostare una configurazione che modifichi il funzionamento, consulta Configurazione avanzata: condivisione degli endpoint di sviluppo tra più utenti.

## Aggiunta di un endpoint di sviluppo

Utilizza gli endpoint di sviluppo per lo sviluppo iterativo e per il test di script di estrazione, trasformazione e caricamento (ETL) in AWS Glue. È possibile utilizzare gli endpoint di sviluppo solo tramite la AWS Command Line Interface.

1. In una finestra a riga di comando, immetti un comando simile al seguente.

```
aws glue create-dev-endpoint --endpoint-name "endpoint1" --role-arn
 "arn:aws:iam::account-id:role/role-name" --number-of-nodes "3" --glue-version
 "1.0" --arguments '{"GLUE_PYTHON_VERSION": "3"}' --region "region-name"
```

   Questo comando specifica AWS Glue versione 1.0. Poiché questa versione supporta sia Python 2 sia Python 3, puoi utilizzare il parametro `arguments` per indicare la versione di Python

desiderata. Se il parametro `glue-version` viene omesso, viene utilizzato AWS Glue versione 0.9. Per ulteriori informazioni sulle versioni di AWS Glue, consulta Glue version job property.

Per informazioni sui parametri aggiuntivi della riga di comando, vedere create-dev-endpointnel AWS CLI Command Reference.

2. (Facoltativo) Immetti il comando seguente per controllare lo stato dell'endpoint di sviluppo. Quando lo stato cambia in `READY`, l'endpoint di sviluppo è pronto per l'uso.

```
aws glue get-dev-endpoint --endpoint-name "endpoint1"
```

## Accesso all'endpoint di sviluppo

Quando crei un endpoint di sviluppo in un cloud privato virtuale (VPC, Virtual Private Cloud), AWS Glue restituisce solo un indirizzo IP privato. Il campo dell'indirizzo IP pubblico non è compilato. Quando crei un endpoint non di sviluppo non VPC, AWS Glue restituisce solo un indirizzo IP pubblico.

Se l'endpoint di sviluppo dispone di un Public address (Indirizzo pubblico), conferma che sia raggiungibile con la chiave privata SSH per l'endpoint di sviluppo, come nell'esempio seguente.

```
ssh -i dev-endpoint-private-key.pem glue@public-address
```

Supponi che l'endpoint di sviluppo disponga di un Private address (Indirizzo privato), che la sottorete VPC sia instradabile dalla rete Internet pubblica e che i relativi gruppi di sicurezza consentano l'accesso in entrata dal client. In questo caso, segui questa procedura per collegare un indirizzo IP elastico a un endpoint di sviluppo per consentire l'accesso da Internet.

> **Note**
>
> Se desideri utilizzare indirizzi IP elastici, la sottorete utilizzata richiede un gateway Internet associato tramite la tabella di routing.

Per accedere a un endpoint di sviluppo collegando un indirizzo IP elastico

1. Apri la AWS Glue console all'indirizzo https://console.aws.amazon.com/glue/.

2. Nel pannello di navigazione, scegli Dev endpoints (Endpoint di sviluppo) e passa alla pagina dei dettagli dell'endpoint di sviluppo. Registra il Private address (Indirizzo privato) da utilizzare nella fase successiva.
3. Apri la EC2 console Amazon all'indirizzo https://console.aws.amazon.com/ec2/.
4. Nel pannello di navigazione, in Network & Security (Rete e sicurezza), scegli Network Interfaces (Interfacce di rete).
5. Cerca il DNS privato (IPv4) che corrisponde all'indirizzo privato nella pagina dei dettagli dell'endpoint di sviluppo della AWS Glue console.

   Potrebbe essere necessario modificare le colonne visualizzate sulla EC2 console Amazon. Prendi nota del Network interface ID (ID interfaccia di rete, ENI) per questo indirizzo (ad esempio, `eni-12345678`).
6. Sulla EC2 console Amazon, in Rete e sicurezza, scegli Elastic IPs.
7. Scegli Allocate new address (Alloca nuovo indirizzo), quindi seleziona Allocate (Alloca) per allocare un nuovo indirizzo IP elastico.
8. Nella IPs pagina Elastic, scegli l'IP elastico appena allocato. Quindi scegli Actions (Operazioni), Associate address (Associa indirizzo).
9. Nella pagina Associate address (Associa indirizzo), procedi come segue:

   - Per il Resource type (Tipo di risorsa), scegli Network interface (Interfaccia di rete).
   - Nella casella Network interface (Interfaccia di rete) , immetti il Network interface ID (ID interfaccia di rete, ENI) dell'indirizzo privato.
   - Selezionare Associate (Associa).
10. Conferma che l'indirizzo IP elastico appena associato sia raggiungibile con la chiave privata SSH associata all'endpoint di sviluppo, come nell'esempio seguente.

    ```
    ssh -i dev-endpoint-private-key.pem glue@elastic-ip
    ```

    Per informazioni sull'utilizzo di un bastion host per ottenere l'accesso SSH all'indirizzo privato dell'endpoint di sviluppo, consulta il post AWS Security Blog Securely Connect to Linux Instances Running in a Private Amazon VPC.

# Tutorial: configura un notebook Jupyter JupyterLab per testare ed eseguire il debug degli script ETL

In questo tutorial, colleghi un notebook Jupyter in JupyterLab esecuzione sul tuo computer locale a un endpoint di sviluppo. Questa operazione serve per poter eseguire, eseguire il debug e testare in modo interattivo l'estrazione, la trasformazione e il caricamento di script (ETL) AWS Glue prima di distribuirli. Questo tutorial usa l'inoltro della porta Secure Shell (SSH) per connettere il computer locale a un endpoint di sviluppo AWS Glue. Per ulteriori informazioni, consulta Inoltro della porta in Wikipedia.

## Fase 1: Installazione e Sparkmagic JupyterLab

È possibile eseguire l'installazione JupyterLab utilizzando `conda` o `pip`. `conda` è un sistema di gestione dei pacchetti open source e un sistema di gestione dell'ambiente che funziona su Windows, macOS e Linux. `pip` è l'installatore di pacchetti per Python.

Se esegui l'installazione su macOS, devi avere Xcode installato prima di poter installare Sparkmagic.

1. Install JupyterLab, Sparkmagic e le relative estensioni.

```
$ conda install -c conda-forge jupyterlab
$ pip install sparkmagic
$ jupyter nbextension enable --py --sys-prefix widgetsnbextension
$ jupyter labextension install @jupyter-widgets/jupyterlab-manager
```

2. Controlla la directory `sparkmagic` da `Location`.

```
$ pip show sparkmagic | grep Location
Location: /Users/username/.pyenv/versions/anaconda3-5.3.1/lib/python3.7/site-
packages
```

3. Cambia la tua directory con quella restituita per`Location`, e installa i kernel per Scala e. PySpark

```
$ cd /Users/username/.pyenv/versions/anaconda3-5.3.1/lib/python3.7/site-packages
$ jupyter-kernelspec install sparkmagic/kernels/sparkkernel
$ jupyter-kernelspec install sparkmagic/kernels/pysparkkernel
```

4. Scarica un esempio di file `config`.

```
$ curl -o ~/.sparkmagic/config.json https://raw.githubusercontent.com/jupyter-incubator/sparkmagic/master/sparkmagic/example_config.json
```

In questo file di configurazione, è possibile configurare parametri relativi a Spark come `driverMemory` e `executorCores`.

Fase 2: Inizia JupyterLab

All'avvio JupyterLab, il browser Web predefinito viene aperto automaticamente e `http://localhost:8888/lab/workspaces/{workspace_name}` viene visualizzato l'URL.

```
$ jupyter lab
```

Fase 3: avviare l'inoltro alla porta SSH per la connessione all'endpoint di sviluppo

Usa quindi l'inoltro porta locale SSH per inoltrare una porta locale (qui, 8998) alla destinazione remota definita da AWS Glue (`169.254.76.1:8998`).

1. Apri una finestra separata del terminale che ti consente di accedere a SSH. Su Microsoft Windows, puoi utilizzare la shell BASH fornita da Git for Windows o installare Cygwin.
2. Esegui il comando SSH seguente, modificato come segue:
   - Sostituisci *`private-key-file-path`* con un percorso al file `.pem` contenente la chiave privata corrispondente alla chiave pubblica utilizzata per creare l'endpoint di sviluppo.
   - Se stai inoltrando una porta diversa da 8998, sostituisci 8998 con il numero di porta che stai effettivamente usando in locale. L'indirizzo `169.254.76.1:8998` è la porta remota e non è modificata da te.
   - Sostituisci *`dev-endpoint-public-dns`* con l'indirizzo DNS pubblico del tuo endpoint di sviluppo. Per trovare questo indirizzo, passa all'endpoint di sviluppo nella console AWS Glue, scegli il nome e copia il valore di Public address (Indirizzo pubblico) elencato nella pagina Endpoint details (Dettagli endpoint).

```
ssh -i private-key-file-path -NTL 8998:169.254.76.1:8998 glue@dev-endpoint-public-dns
```

È probabile che vedrai un messaggio di avviso, come il seguente:

```
The authenticity of host 'ec2-xx-xxx-xxx-xx.us-west-2.compute.amazonaws.com
 (xx.xxx.xxx.xx)'
can't be established.  ECDSA key fingerprint is SHA256:4e97875Brt+1wKzRko
+JflSnp21X7aTP3BcFnHYLEts.
Are you sure you want to continue connecting (yes/no)?
```

Entra **yes** e lascia aperta la finestra del terminale durante l'uso JupyterLab.

3. Verifica che l'inoltro porta SSH funzioni correttamente con l'endpoint di sviluppo.

```
$ curl localhost:8998/sessions
{"from":0,"total":0,"sessions":[]}
```

Fase 4: eseguire un semplice frammento di script in un paragrafo del notebook

Ora il notebook in JupyterLab dovrebbe funzionare con il tuo endpoint di sviluppo. Digita il frammento di script seguente nel notebook ed eseguilo.

1. Verifica che Spark sia in esecuzione correttamente. Il comando seguente indica a Spark di calcolare 1 e quindi stampare il valore.

```
spark.sql("select 1").show()
```

2. Verifica se AWS Glue Data Catalog l'integrazione funziona. Il comando seguente elenca le tabelle nel catalogo dati.

```
spark.sql("show tables").show()
```

3. Verifica che un frammento di script semplice che utilizza le librerie AWS Glue funzioni.

   Lo script seguente utilizza i metadati della tabella `persons_json` in AWS Glue Data Catalog per creare un`DynamicFrame`dai dati di esempio. Quindi, verranno stampati il conteggio item e lo schema di questi dati.

```
import sys
from pyspark.context import SparkContext
from awsglue.context import GlueContext

# Create a Glue context
```

```
glueContext = GlueContext(SparkContext.getOrCreate())

# Create a DynamicFrame using the 'persons_json' table
persons_DyF = glueContext.create_dynamic_frame.from_catalog(database="legislators",
 table_name="persons_json")

# Print out information about *this* data
print("Count:  ", persons_DyF.count())
persons_DyF.printSchema()
```

L'output dello script è il seguente.

```
Count:  1961
root
|-- family_name: string
|-- name: string
|-- links: array
|    |-- element: struct
|    |    |-- note: string
|    |    |-- url: string
|-- gender: string
|-- image: string
|-- identifiers: array
|    |-- element: struct
|    |    |-- scheme: string
|    |    |-- identifier: string
|-- other_names: array
|    |-- element: struct
|    |    |-- note: string
|    |    |-- name: string
|    |    |-- lang: string
|-- sort_name: string
|-- images: array
|    |-- element: struct
|    |    |-- url: string
|-- given_name: string
|-- birth_date: string
|-- id: string
|-- contact_details: array
|    |-- element: struct
|    |    |-- type: string
|    |    |-- value: string
```

```
|-- death_date: string
```

Risoluzione dei problemi

- Durante l'installazione di JupyterLab, se il computer è protetto da un proxy o firewall aziendale, è possibile che si verifichino errori HTTP e SSL dovuti ai profili di sicurezza personalizzati gestiti dai reparti IT aziendali.

  Di seguito è riportato un esempio di errore tipico che si verifica quando `conda` non è in grado di connettersi ai propri repository:

  ```
  CondaHTTPError: HTTP 000 CONNECTION FAILED for url <https://repo.anaconda.com/pkgs/
  main/win-64/current_repodata.json>
  ```

  Ciò potrebbe accadere perché la tua azienda può bloccare le connessioni a repository ampiamente utilizzati in Python JavaScript e nelle community. Per ulteriori informazioni, consulta Problemi di installazione sul JupyterLab sito Web.
- Se si verifica un errore di connessione rifiutata quando provi a connetterti all'endpoint di sviluppo, potresti utilizzare un endpoint di sviluppo obsoleto. Prova a creare un nuovo endpoint di sviluppo e a riconnetterti.

## Tutorial: utilizza un notebook SageMaker AI con il tuo endpoint di sviluppo

NelAWS Glue, puoi creare un endpoint di sviluppo e quindi creare un notebook SageMaker AI per aiutarti a sviluppare i tuoi script ETL e di machine learning. Un notebook SageMaker AI è un'istanza di calcolo di machine learning completamente gestita che esegue l'applicazione Jupyter Notebook.

1. Nella console AWS Glue seleziona Dev endpoints (Endpoint di sviluppo) per passare all'elenco degli endpoint di sviluppo.
2. Seleziona la casella di controllo accanto al nome di un endpoint di sviluppo che desideri utilizzare e, nel menu Azione, scegli Crea notebook. SageMaker
3. Compilare la pagina Create and configure a notebook (Crea e configura un notebook) come segue:

   a. Immettere il nome di un notebook.

   b. In Attach to development endpoint (Collega a endpoint di sviluppo), verificare l'endpoint di sviluppo.

c. Crea o scegli un ruolo AWS Identity and Access Management (IAM).

   Si consiglia di creare un ruolo. Se si utilizza un ruolo esistente, assicurarsi di avere le autorizzazioni necessarie. Per ulteriori informazioni, consulta the section called “Fase 6: Creare una policy IAM per i notebook SageMaker AI”.

d. (Facoltativo) Scegliere un VPC, una sottorete e uno o più gruppi di sicurezza.

e. (Facoltativo) Scegli una chiave di AWS Key Management Service crittografia.

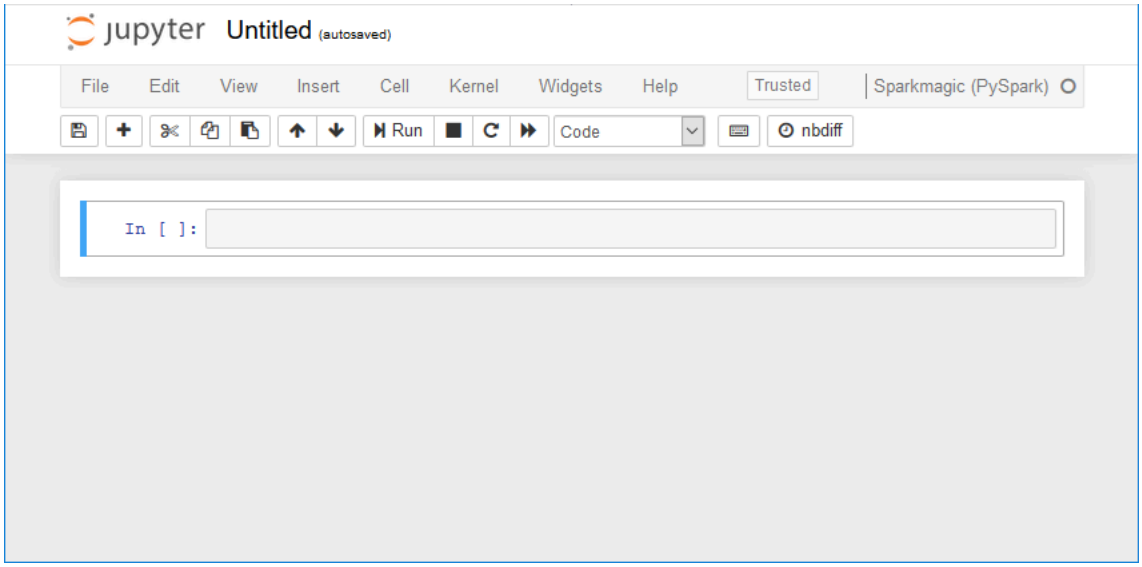
f. (Facoltativo) Aggiungere i tag per l'istanza del notebook.

4. Seleziona Crea notebook. Sulla pagina Notebooks (Notebook), scegli l'icona di aggiornamento in alto a destra e continua fino a quando la finestra Status (Stato) non mostra `Ready`.

5. Selezionare la casella di controllo accanto al nuovo nome del notebook, quindi scegliere Open notebook (Apri notebook).

6. Crea un nuovo taccuino: nella pagina di jupyter, scegli Nuovo, quindi scegli Sparkmagic (). PySpark

   La schermata dovrebbe essere simile alla seguente:

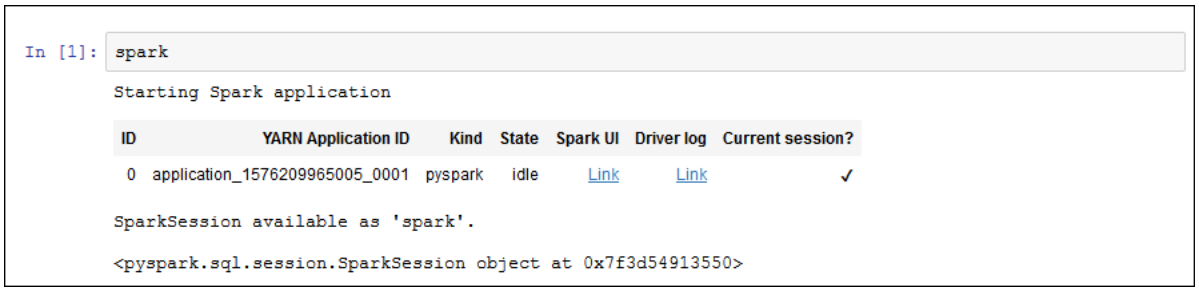



7. (Facoltativo) Nella parte superiore della pagina, scegliere Untitled (Senza titolo) e assegnare un nome al notebook.

8. Per avviare un'applicazione Spark, immettere il seguente comando nel notebook e quindi nella barra degli strumenti scegliere Run (Esegui).

```
spark
```

Dopo una breve attesa, viene visualizzata la seguente risposta:




9. Creare un frame dinamico ed eseguirvi una query: copiare, incollare ed eseguire il codice seguente, che restituisce il conteggio e lo schema della tabella `persons_json`.

```
import sys
from pyspark.context import SparkContext
from awsglue.context import GlueContext
from awsglue.transforms import *
glueContext = GlueContext(SparkContext.getOrCreate())
persons_DyF = glueContext.create_dynamic_frame.from_catalog(database="legislators",
 table_name="persons_json")
print ("Count:  ", persons_DyF.count())
persons_DyF.printSchema()
```

## Tutorial: utilizzo di una shell (interprete di comandi) REPL con l'endpoint di sviluppo

InAWS Glue, puoi creare un endpoint di sviluppo e quindi richiamare una shell REPL (Read—Evaluate—Print Loop) per eseguire il PySpark codice in modo incrementale in modo da poter eseguire il debug interattivo degli script ETL prima di distribuirli.

Per utilizzare una REPL su un endpoint di sviluppo, è necessario disporre dell'autorizzazione SSH all'endpoint.

1. Sul tuo computer locale, apri una finestra terminale che possa eseguire comandi SSH e incolla il comando SSH modificato. Esegui il comando .

   Se hai accettato AWS Glue versione 1.0 con Python 3 per l'endpoint di sviluppo, l'aspetto dell'output sarà simile al seguente:

```
Python 3.6.8 (default, Aug  2 2019, 17:42:44)
[GCC 4.8.5 20150623 (Red Hat 4.8.5-28)] on linux
Type "help", "copyright", "credits" or "license" for more information.
```

```
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/share/aws/glue/etl/jars/glue-assembly.jar!/
org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/lib/spark/jars/slf4j-log4j12-1.7.16.jar!/
org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use
 setLogLevel(newLevel).
2019-09-23 22:12:23,071 WARN  [Thread-5] yarn.Client (Logging.scala:logWarning(66))
 - Neither spark.yarn.jars nor spark.yarn.archive is set, falling back to uploading
 libraries under SPARK_HOME.
2019-09-23 22:12:26,562 WARN  [Thread-5] yarn.Client (Logging.scala:logWarning(66))
 - Same name resource file:/usr/lib/spark/python/lib/pyspark.zip added multiple
 times to distributed cache
2019-09-23 22:12:26,580 WARN  [Thread-5] yarn.Client (Logging.scala:logWarning(66))
 - Same path resource file:///usr/share/aws/glue/etl/python/PyGlue.zip added
 multiple times to distributed cache.
2019-09-23 22:12:26,581 WARN  [Thread-5] yarn.Client (Logging.scala:logWarning(66))
 - Same path resource file:///usr/lib/spark/python/lib/py4j-src.zip added multiple
 times to distributed cache.
2019-09-23 22:12:26,581 WARN  [Thread-5] yarn.Client (Logging.scala:logWarning(66))
 - Same path resource file:///usr/share/aws/glue/libs/pyspark.zip added multiple
 times to distributed cache.
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 2.4.3
      /_/

Using Python version 3.6.8 (default, Aug  2 2019 17:42:44)
SparkSession available as 'spark'.
>>>
```

2. Verifica che la REPL shell funzioni correttamente digitando l'istruzione `print(spark.version)`. Finché visualizza la versione Spark, la REPL è pronta per l'uso.
3. Ora puoi provare a eseguire il seguente script semplice, riga per riga, nella shell:

```
import sys
from pyspark.context import SparkContext
from awsglue.context import GlueContext
```

```
from awsglue.transforms import *
glueContext = GlueContext(SparkContext.getOrCreate())
persons_DyF = glueContext.create_dynamic_frame.from_catalog(database="legislators",
 table_name="persons_json")
print ("Count:  ", persons_DyF.count())
persons_DyF.printSchema()
```

## Tutorial: configura un PyCharm professionista con un endpoint di sviluppo

Questo tutorial mostra come connettere l'IDE PyCharm Professional Python in esecuzione sul computer locale a un endpoint di sviluppo in modo da poter eseguire, eseguire il debug e testare in modo interattivo gli script AWS Glue ETL (estrazione, trasferimento e caricamento) prima di distribuirli. Le istruzioni e le schermate del tutorial si basano sulla versione Professional 2019.3. PyCharm

Per connetterti a un endpoint di sviluppo in modo interattivo, devi avere installato Professional. PyCharm Non puoi farlo usando l'edizione gratuita.

> **Note**
>
> Il tutorial utilizza Amazon S3 come origine dati. Se invece desideri utilizzare un'origine dati JDBC, devi eseguire l'endpoint di sviluppo in un cloud privato virtuale (VPC). Per connetterti con SSH a un endpoint di sviluppo in un VPC, devi creare un tunnel SSH. Questo tutorial non include le istruzioni per la creazione di un tunnel SSH. Per informazioni sull'uso di SSH per connettersi a un endpoint di sviluppo in un VPC, consulta Securely Connect to Linux Instances Running in un Amazon VPC privato nel blog sulla sicurezza. AWS

Argomenti

- Connessione di un professionista a un endpoint di sviluppo PyCharm
- Distribuzione dello script nell'endpoint di sviluppo
- Configurazione di un interprete remoto
- Esecuzione dello script sull'endpoint di sviluppo

### Connessione di un professionista a un endpoint di sviluppo PyCharm

1. Crea un nuovo progetto Pure-Python in named. PyCharm `legislators`

2. Crea un file denominato `get_person_schema.py` nel progetto con il seguente contenuto:

```
from pyspark.context import SparkContext
from awsglue.context import GlueContext


def main():
    # Create a Glue context
    glueContext = GlueContext(SparkContext.getOrCreate())

    # Create a DynamicFrame using the 'persons_json' table
    persons_DyF =
 glueContext.create_dynamic_frame.from_catalog(database="legislators",
 table_name="persons_json")

    # Print out information about this data
    print("Count:  ", persons_DyF.count())
    persons_DyF.printSchema()


if __name__ == "__main__":
    main()
```

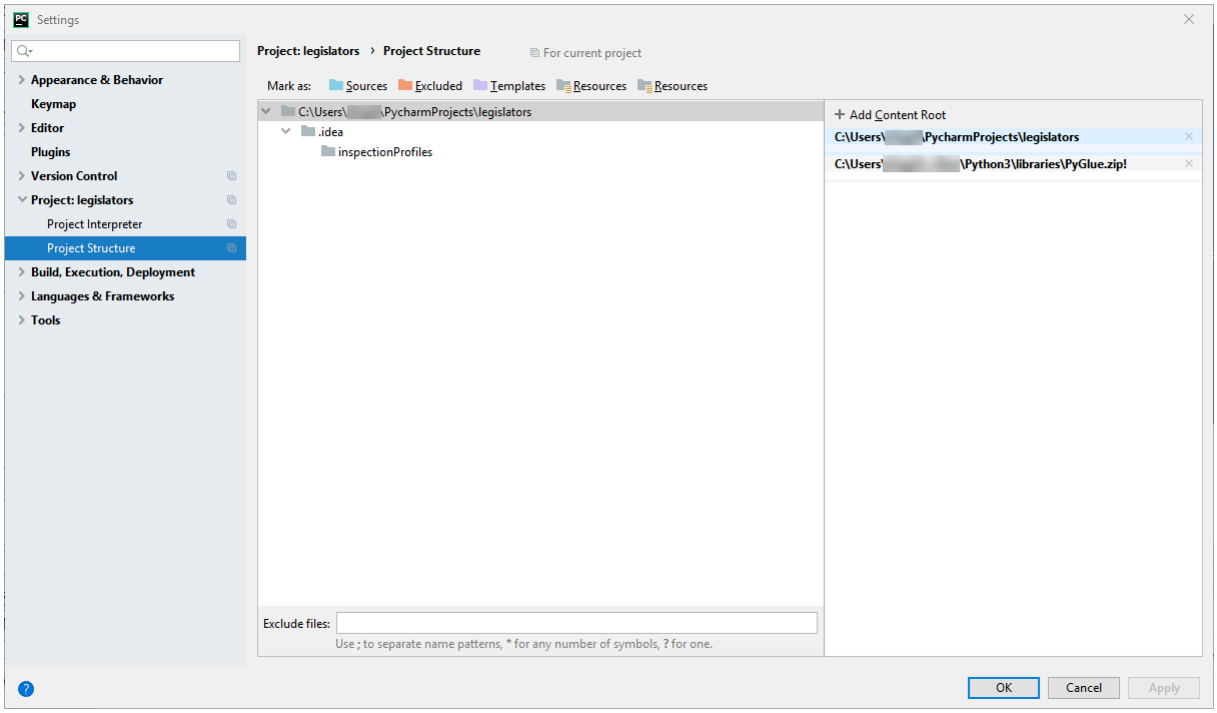
3. Esegui una delle seguenti operazioni:

   - Per AWS Glue versione 0.9, scarica il file della libreria Python AWS Glue, `PyGlue.zip`, da `https://s3.amazonaws.com/aws-glue-jes-prod-us-east-1-assets/etl/python/PyGlue.zip` a un percorso appropriato nel computer locale.
   - Per AWS Glue versione 1.0, scarica il file della libreria Python AWS Glue, `PyGlue.zip`, da `https://s3.amazonaws.com/aws-glue-jes-prod-us-east-1-assets/etl-1.0/python/PyGlue.zip` a un percorso appropriato nel computer locale.

4. Aggiungi `PyGlue.zip` come radice di contenuto per il tuo progetto in: PyCharm

   - In PyCharm, scegliete File, Impostazioni per aprire la finestra di dialogo Impostazioni. Puoi anche premere `Ctrl+Alt+S`.
   - Espandi il progetto `legislators` e scegli Project Structure (Struttura del progetto). Quindi nel riquadro di destra, scegli + Add Content Root (+ Aggiungi Content Root).
   - Seleziona il percorso in cui hai salvato `PyGlue.zip`, selezionalo, quindi scegli Apply (Applica).

La schermata Settings (Impostazioni) deve avere un aspetto simile al seguente:




Lascia aperta la finestra di dialogo Settings (Impostazioni) dopo aver scelto Apply (Applica).

5. Configura le opzioni di distribuzione per caricare lo script locale sull'endpoint di sviluppo utilizzando SFTP (questa funzionalità è disponibile solo in PyCharm Professional):

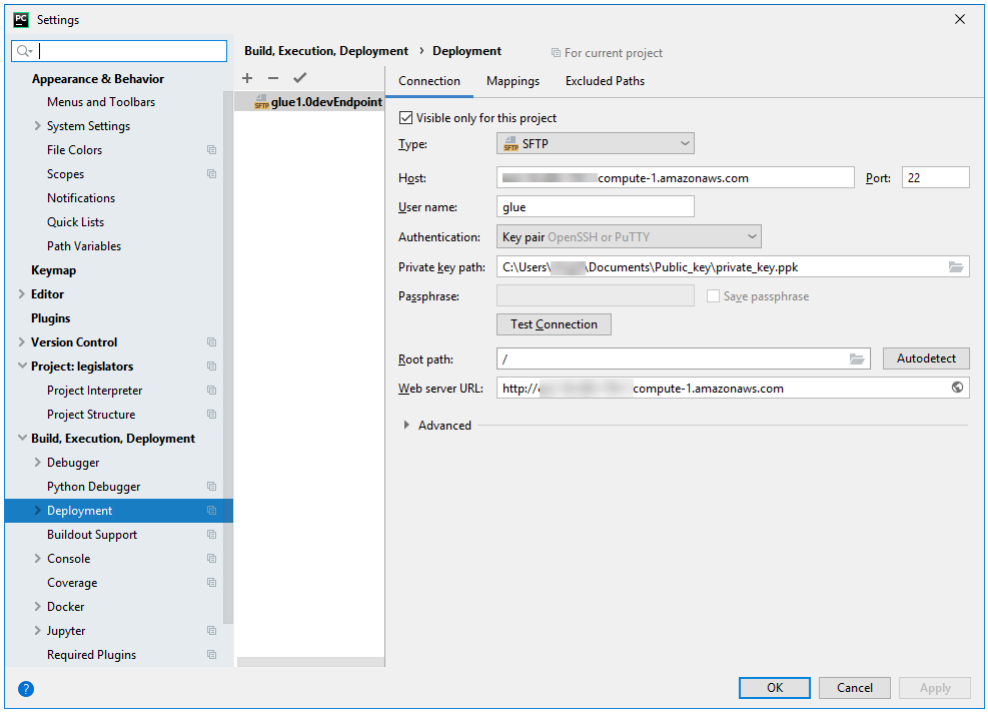
   - Nella finestra di dialogo Settings (Impostazioni), espandi la sezione Build, Execution, Deployment (Creazione, esecuzione e distribuzione). Scegli la sottosezione Deployment (Distribuzione).
   - Scegli l'icona + in alto nel riquadro centrale per aggiungere un nuovo server. Imposta il Type (Tipo) su `SFTP` e assegna un nome.
   - Imposta SFTP host (Host SFTP) sull'indirizzo pubblico dell'endpoint di sviluppo, come indicato nella pagina dei dettagli. Scegli il nome dell'endpoint di sviluppo nella console AWS Glue per visualizzare la pagina dei dettagli. Per un endpoint di sviluppo in esecuzione in un VPC, imposta l'host SFTP sull'indirizzo host e la porta locale del tunnel SSH sull'endpoint di sviluppo.
   - Imposta lo User name (Nome utente) su `glue`.
   - Imposta l'Auth type (Tipo di autenticazione) per la Key pair (OpenSSH or Putty) (Coppia di chiavi, OpenSSH o Putty). Imposta il Private key file (File della chiave privata) cercando il percorso in cui si trova il file della chiave privata dell'endpoint di sviluppo. Nota che supporta

PyCharm solo i tipi di chiave OpenSSH DSA, RSA ed ECDSA e non accetta chiavi nel formato privato di Putty. È possibile utilizzare una up-to-date versione di per generare un tipo di coppia di chiavi `ssh-keygen` che accetti, utilizzando una sintassi come la seguente: PyCharm

```
ssh-keygen -t rsa -f <key_file_name> -C "<your_email_address>"
```

- Scegli Test connection (Test connessione) e consenti il test della connessione. Se la connessione viene stabilita, scegli Apply (Applica).

La schermata Settings (Impostazioni) ora deve avere un aspetto simile al seguente:

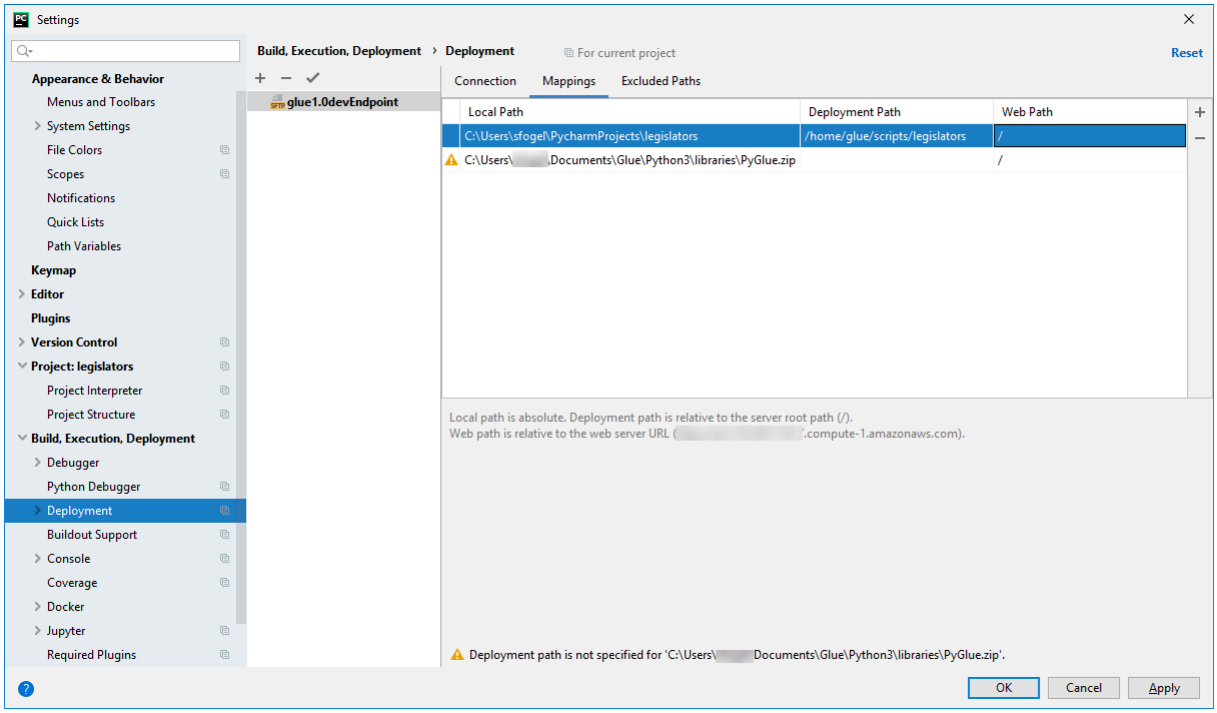



Lascia nuovamente aperta la finestra di dialogo Settings (Impostazioni) dopo aver scelto Apply (Applica).

6. Imposta la directory locale su una directory remota per la distribuzione:

   - Nel riquadro a destra della pagina Deployment (Distribuzione), scegli la scheda centrale nella parte superiore, contrassegnata dall'etichetta Mappings (Mappe).
   - Nella colonna Deployment Path (Percorso di distribuzione), immetti un percorso sotto `/home/glue/scripts/` per la distribuzione del percorso del progetto. Ad esempio: `/home/glue/scripts/legislators`.
   - Scegli Applica.

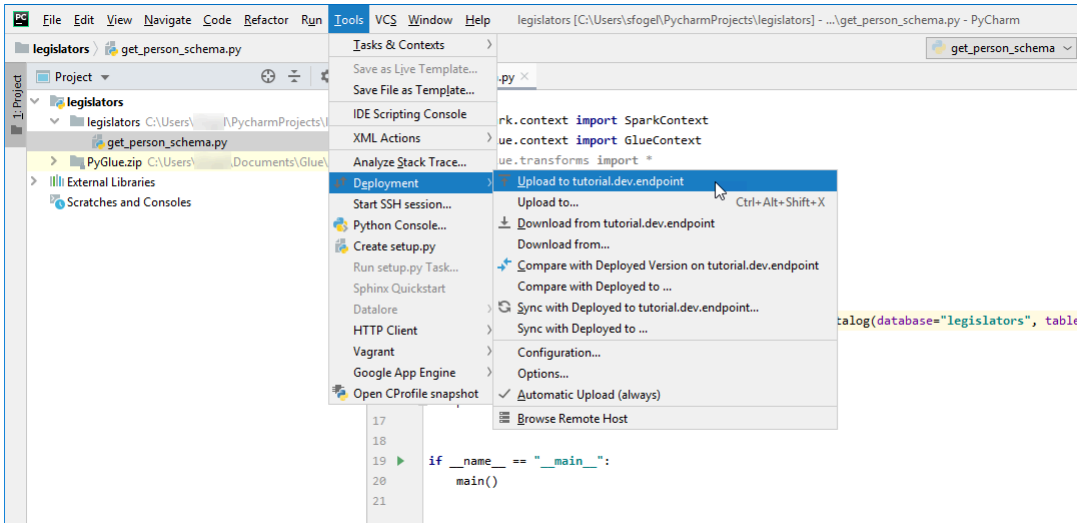
La schermata Settings (Impostazioni) ora deve avere un aspetto simile al seguente:



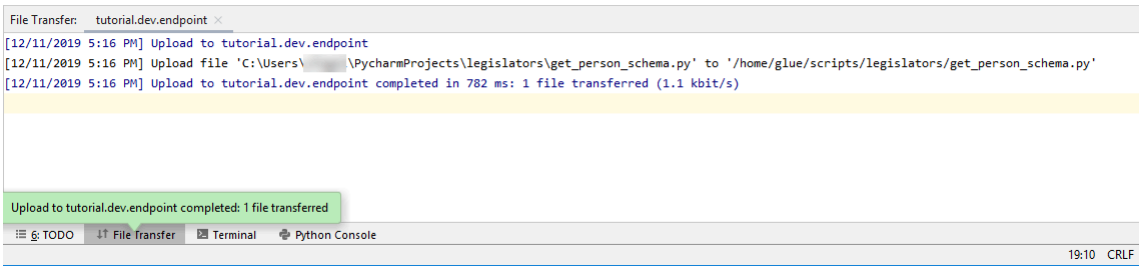

Scegli OK per chiudere la finestra di dialogo Settings (Impostazioni).

## Distribuzione dello script nell'endpoint di sviluppo

1. Scegli Tools (Strumenti), Deployment (Distribuzione), quindi scegli il nome con cui hai configurato l'endpoint di sviluppo, come illustrato nell'immagine seguente:



Dopo che lo script è stato distribuito, la parte inferiore della schermata deve avere un aspetto simile al seguente:




2. Nella barra dei menu scegli Tools (Strumenti), Deployment (Distribuzione), Automatic Upload (always) (Caricamento automatico (sempre)). Accertati che venga visualizzato un segno di spunta accanto a Automatic Upload (always) (Caricamento automatico (sempre)).

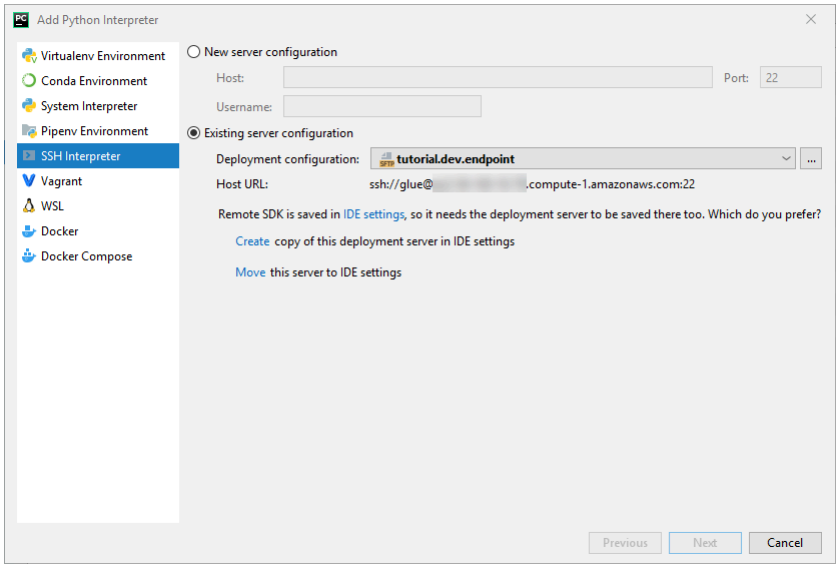
   Quando questa opzione è abilitata, carica PyCharm automaticamente i file modificati sull'endpoint di sviluppo.

### Configurazione di un interprete remoto

Configura PyCharm per utilizzare l'interprete Python sull'endpoint di sviluppo.

1. Nel menu File, scegli Settings (Impostazioni).
2. Espandi i legislators (legislatori) del progetto e scegli Project Interpreter (Interprete di progetto).
3. Scegli l'icona a forma di ingranaggio accanto all'elenco Project Interpreter (Interprete di progetto) quindi scegli Add (Aggiungi).
4. Nella finestra di dialogo Add Python Interpreter (Aggiungi interprete Python) nel riquadro di sinistra, scegli SSH Interpreter (Interprete SSH).
5. Scegli Existing server configuration (Configurazione server esistente) e nell'elenco Deployment configuration (Configurazione distribuzione) scegli la configurazione.

   La schermata ora deve avere un aspetto simile all'immagine seguente.




6. Scegli "Move this server to IDE settings (Sposta il server nelle impostazioni IDE", quindi seleziona Next (Successivo).
7. Nel campo Interpreter (Interprete) cambia il percorso in `/usr/bin/gluepython` se stai usando Python 2 o in `/usr/bin/gluepython3` se stai usando Python 3. Quindi scegli Finish (Fine).

Esecuzione dello script sull'endpoint di sviluppo

Per eseguire lo script:

- Nel riquadro sinistro, fai clic con il pulsante destro del mouse sul nome del file e scegli Esegui ". *<filename>*

  Dopo una serie di messaggi, l'output finale mostra il conteggio e lo schema.

```
Count:  1961
root
|-- family_name: string
|-- name: string
|-- links: array
|    |-- element: struct
|    |    |-- note: string
|    |    |-- url: string
|-- gender: string
|-- image: string
|-- identifiers: array
|    |-- element: struct
```

```
|    |    |-- scheme: string
|    |    |-- identifier: string
|-- other_names: array
|    |-- element: struct
|    |    |-- lang: string
|    |    |-- note: string
|    |    |-- name: string
|-- sort_name: string
|-- images: array
|    |-- element: struct
|    |    |-- url: string
|-- given_name: string
|-- birth_date: string
|-- id: string
|-- contact_details: array
|    |-- element: struct
|    |    |-- type: string
|    |    |-- value: string
|-- death_date: string


Process finished with exit code 0
```

Ora hai effettuato la configurazione per eseguire il debug dello script in remoto sul tuo endpoint di sviluppo.

## Configurazione avanzata: condivisione degli endpoint di sviluppo tra più utenti

Questa sezione spiega come sfruttare gli endpoint di sviluppo con SageMaker notebook nei casi d'uso tipici per condividere gli endpoint di sviluppo tra più utenti.

### Configurazione tenancy singola

Nei casi d'uso a tenant singolo, per semplificare l'esperienza degli sviluppatori ed evitare conflitti per le risorse, è consigliabile che ogni sviluppatore utilizzi il proprio endpoint di sviluppo per il progetto su cui sta lavorando. Ciò semplifica anche le decisioni relative al tipo di worker e al conteggio DPU, lasciandole a discrezione dello sviluppatore e del progetto su cui sta lavorando.

Non dovrai occuparti dell'allocazione delle risorse a meno che non vengano eseguiti simultaneamente più file notebook. Se esegui il codice in più file notebook contemporaneamente, verranno avviate contemporaneamente più sessioni Livy. Per separare le configurazioni del cluster

Spark al fine di eseguire più sessioni Livy contemporaneamente, puoi seguire i passaggi introdotti nei casi d'uso multi-tenant.

Ad esempio, se l'endpoint di sviluppo ha 10 worker e il tipo di worker è `G.1X`, allora si avranno 9 executor Spark e l'intero cluster avrà 90G di memoria dell'executor, poiché ogni executor avrà 10G di memoria.

Indipendentemente dal tipo di worker specificato, verrà attivata l'allocazione dinamica delle risorse Spark. Se un set di dati è abbastanza grande, Spark potrebbe allocare tutti gli executor a una singola sessione Livy poiché `spark.dynamicAllocation.maxExecutors` non è configurata per impostazione predefinita. Ciò significa che altre sessioni Livy sullo stesso endpoint di sviluppo attenderanno per l'avvio di nuovi executor. Se il set di dati è piccolo, Spark sarà in grado di allocare gli esecutori a più sessioni Livy contemporaneamente.

> **Note**
>
> Per ulteriori informazioni sull'allocazione delle risorse in diversi casi d'uso e su come impostare una configurazione che modifichi il funzionamento, consulta Configurazione avanzata: condivisione degli endpoint di sviluppo tra più utenti.

Configurazione tenancy multipla

> **Note**
>
> Tieni presente che gli endpoint di sviluppo hanno lo scopo di emulare l'ambiente ETL AWS Glue come ambiente a tenant singolo. Sebbene l'utilizzo di multi-tenant sia possibile, si tratta di un caso d'uso avanzato e alla maggior parte degli utenti si consiglia di mantenere un modello di tenancy singolo per ogni endpoint di sviluppo.

Nei casi d'uso multi-tenant, potresti doverti occupare dell'allocazione delle risorse. Il fattore chiave è il numero di utenti che utilizzano contemporaneamente un notebook Jupyter. Se il tuo team lavora in un flusso di lavoro follow-the-sun "" e c'è un solo utente Jupyter per fuso orario, il numero di utenti simultanei è solo uno, quindi non dovrai preoccuparti dell'allocazione delle risorse. Tuttavia, se il notebook è condiviso tra più utenti e ogni utente invia il codice ad hoc, sarà necessario considerare i seguenti punti.

Per partizionare le risorse del cluster Spark tra più utenti, puoi utilizzare le configurazioni. SparkMagic Esistono due modi diversi per configurare. SparkMagic

(A) Utilizzo della direttiva %%configure -f

Se vuoi modificare la configurazione per sessione Livy dal notebook, puoi eseguire la direttiva `%%configure -f` sul paragrafo di notebook.

Ad esempio, se vuoi eseguire l'applicazione Spark su 5 executor, puoi eseguire il comando seguente nel paragrafo di notebook.

```
%%configure -f
{"numExecutors":5}
```

Vedrai quindi solo 5 executor in esecuzione per il processo sull'interfaccia utente di Spark.

Si consiglia di limitare il numero massimo di executor per l'allocazione dinamica delle risorse.

```
%%configure -f
{"conf":{"spark.dynamicAllocation.maxExecutors":"5"}}
```

(B) Modifica il SparkMagic file di configurazione

SparkMagic funziona in base all'API Livy. SparkMagic crea sessioni Livy con configurazioni come`driverMemory`,,, `driverCores`,`executorMemory`, `executorCores` `numExecutors`, ecc. `conf` Questi sono i fattori chiave che determinano la quantità di risorse consumate dall'intero cluster Spark. SparkMagic consente di fornire un file di configurazione per specificare i parametri che vengono inviati a Livy. Puoi vedere un file di configurazione di esempio in questo repository Github.

Se vuoi modificare la configurazione di tutte le sessioni Livy da un notebook, puoi modificare `/home/ec2-user/.sparkmagic/config.json` per aggiungere `session_config`.

Per modificare il file di configurazione su un'istanza di SageMaker notebook, puoi seguire questi passaggi.

1. Apri un SageMaker taccuino.
2. Apri il kernel del terminale.
3. Esegui i comandi seguenti:

```
sh-4.2$ cd .sparkmagic
```

```
sh-4.2$ ls
config.json logs
sh-4.2$ sudo vim config.json
```

Ad esempio, puoi aggiungere queste righe a `/home/ec2-user/.sparkmagic/config.json` e riavviare il kernel Jupyter dal notebook.

```
  "session_configs": {
    "conf": {
      "spark.dynamicAllocation.maxExecutors":"5"
    }
  },
```

Linee guida e best practice

Per evitare questo tipo di conflitto di risorse, puoi utilizzare alcuni approcci di base come:

- Avere un cluster Spark più grande aumentando il `NumberOfWorkers` (dimensionamento orizzontale) e aggiornando il `workerType` (dimensionamento verticale)
- Allocare di meno risorse per utente (meno risorse per sessione Livy)

Il tuo approccio dipenderà dal caso d'uso. Se disponi di un endpoint di sviluppo più grande e non vi è una quantità enorme di dati, la possibilità di un conflitto di risorse diminuirà significativamente perché Spark può allocare risorse in base a una strategia di allocazione dinamica.

Come descritto sopra, il numero massimo di executor Spark viene calcolato automaticamente in base alla combinazione di DPU (o `NumberOfWorkers`) e il tipo di worker. Ogni applicazione Spark avvia un driver e più executor. Per il calcolo è necessario il `NumberOfWorkers` = `NumberOfExecutors` + 1. La matrice seguente illustra la capacità necessaria nell'endpoint di sviluppo in base al numero di utenti simultanei.

| Numero di utenti simultanei del notebook | Numero di executor Spark che vuoi allocare per utente | Totale NumberOfWorkers per il tuo endpoint di sviluppo |
|---|---|---|
| 3 | 5 | 18 |
| 10 | 5 | 60 |

| Numero di utenti simultanei del notebook | Numero di executor Spark che vuoi allocare per utente | Totale NumberOfWorkers per il tuo endpoint di sviluppo |
|---|---|---|
| 50 | 5 | 300 |

Se vuoi allocare meno risorse per utente, `spark.dynamicAllocation.maxExecutors` (o `numExecutors`) è il parametro più semplice da configurare come parametro di sessione Livy. Se imposti la configurazione seguente`/home/ec2-user/.sparkmagic/config.json`, SparkMagic assegnerà un massimo di 5 esecutori per sessione Livy. Questo aiuterà a separare le risorse per sessione Livy.

```
"session_configs": {
    "conf": {
      "spark.dynamicAllocation.maxExecutors":"5"
    }
  },
```

Supponiamo che ci sia un endpoint di sviluppo con 18 worker (G.1X) e 3 utenti del notebook contemporaneamente. Se la configurazione della sessione ha `spark.dynamicAllocation.maxExecutors=5`, ogni utente può utilizzare 1 driver e 5 executor. Anche eseguendo più paragrafi di notebook simultaneamente, non si verificheranno conflitti di risorse.

Pro e contro

Con questa configurazione di sessione `"spark.dynamicAllocation.maxExecutors":"5"`, potrai evitare errori di conflitto di risorse e non sarà necessario attendere l'allocazione delle risorse in caso di accessi utente simultanei. Tuttavia, anche quando ci sono molte risorse gratuite (ad esempio, non ci sono altri utenti simultanei), Spark non può assegnare più di 5 executor per la sessione Livy.

Altre note

È buona prassi interrompere il kernel Jupyter quando si smette di usare un notebook. Ciò consentirà di liberare risorse che altri utenti del notebook potranno utilizzare immediatamente, senza dover attendere la scadenza del kernel (spegnimento automatico).

Problemi comuni

Anche quando si seguono le linee guida, è possibile che si verifichino alcuni problemi.

Sessione non trovata

Se provi a eseguire un paragrafo del notebook anche se la sessione Livy è già stata terminata, verrà visualizzato il messaggio seguente. Per attivare la sessione Livy, devi riavviare il kernel Jupyter scegliendo Kernel >Restart (Riavvia) nel menu Jupyter, quindi eseguire nuovamente il paragrafo del notebook.

```
An error was encountered:
Invalid status code '404' from http://localhost:8998/sessions/13 with error payload:
 "Session '13' not found."
```

Risorse YARN insufficienti

Se provi a eseguire un paragrafo del notebook anche se il cluster Spark non dispone di risorse sufficienti per avviare una nuova sessione Livy, verrà visualizzato il messaggio seguente. Seguendo le linee guida è spesso possibile evitare questo problema, tuttavia potrebbe verificarsi. Per risolvere il problema, puoi verificare se sono presenti sessioni Livy attive non necessarie. In caso affermativo, sarà necessario terminarle per liberare le risorse del cluster. Per ulteriori informazioni, consulta la prossima sezione.

```
Warning: The Spark session does not have enough YARN resources to start.
The code failed because of a fatal error:
    Session 16 did not start up in 60 seconds..

Some things to try:
a) Make sure Spark has enough available resources for Jupyter to create a Spark
 context.
b) Contact your Jupyter administrator to make sure the Spark magics library is
 configured correctly.
c) Restart the kernel.
```

Monitoraggio e debug

In questa sezione vengono descritte le tecniche per il monitoraggio delle risorse e delle sessioni.

Monitoraggio e debug dell'allocazione delle risorse nel cluster

È possibile visualizzare l'interfaccia utente di Spark per monitorare il numero di risorse allocate per ogni sessione Livy e quali sono le configurazioni Spark effettive sul processo. Per attivare l'interfaccia utente di Spark, consulta Abilitazione dell'interfaccia utente Web di Apache Spark per endpoint di sviluppo.

(Facoltativo) Se è necessaria una visualizzazione in tempo reale dell'interfaccia utente di Spark, puoi configurare un tunnel SSH sul server di cronologia Spark in esecuzione sul cluster Spark.

```
ssh -i <private-key.pem> -N -L 8157:<development endpoint public address>:18080
 glue@<development endpoint public address>
```

Apri http://localhost:8157 nel browser per visualizzare l'interfaccia utente di Spark in locale.

Sessioni libere Livy non necessarie

Consulta queste procedure per arrestare le sessioni Livy non necessarie da un notebook o da un cluster Spark.

(a). Terminare le sessioni Livy da un notebook

Puoi chiudere il kernel su un notebook Jupyter per terminare sessioni Livy non necessarie.

(b). Terminare le sessioni Livy da un cluster Spark

Se sono ancora in esecuzione sessioni Livy non necessarie, puoi arrestare le sessioni Livy nel cluster Spark.

Come prerequisito per eseguire questa procedura, è necessario configurare la chiave pubblica SSH per l'endpoint di sviluppo.

Per accedere al cluster Spark, esegui il comando seguente:

```
$ ssh -i <private-key.pem> glue@<development endpoint public address>
```

Per visualizzare le sessioni attive Livy, esegui il comando seguente:

```
$ yarn application -list
20/09/25 06:22:21 INFO client.RMProxy: Connecting to ResourceManager at
 ip-255-1-106-206.ec2.internal/172.38.106.206:8032
Total number of applications (application-types: [] and states: [SUBMITTED, ACCEPTED,
 RUNNING]):2
Application-Id Application-Name Application-Type User Queue State Final-State Progress
 Tracking-URL
application_1601003432160_0005 livy-session-4 SPARK livy default RUNNING UNDEFINED 10%
 http://ip-255-1-4-130.ec2.internal:41867
application_1601003432160_0004 livy-session-3 SPARK livy default RUNNING UNDEFINED 10%
 http://ip-255-1-179-185.ec2.internal:33727
```

Puoi quindi chiudere la sessione Livy con il seguente comando:

```
$ yarn application -kill application_1601003432160_0005
20/09/25 06:23:38 INFO client.RMProxy: Connecting to ResourceManager at
 ip-255-1-106-206.ec2.internal/255.1.106.206:8032
Killing application application_1601003432160_0005
20/09/25 06:23:39 INFO impl.YarnClientImpl: Killed application
 application_1601003432160_0005
```

## Gestione di notebook

> **Note**
>
> Gli endpoint di sviluppo sono supportati solo per le versioni AWS Glue precedenti alla 2.0. Per un ambiente interattivo in cui è possibile creare e testare script ETL, utilizza Notebooks on Studio. AWS Glue

Un notebook consente lo sviluppo e il test interattivi degli script ETL (estrazione, trasformazione e caricamento) su un endpoint di sviluppo. AWS Gluefornisce un'interfaccia per i notebook SageMaker AI Jupyter. ConAWS Glue, crei e gestisci notebook AI. SageMaker Puoi anche aprire notebook SageMaker AI dalla console. AWS Glue

Inoltre, puoi utilizzare Apache Spark con l' SageMaker intelligenza artificiale su endpoint di AWS Glue sviluppo che supportano l'intelligenza SageMaker artificiale (ma non i job ETL). AWS Glue SageMaker Spark è una libreria Apache Spark open source per l'intelligenza artificiale. SageMaker Per ulteriori informazioni, consulta Usare Apache Spark con Amazon. SageMaker

> **Important**
>
> La gestione dei notebook SageMaker AI con endpoint di AWS Glue sviluppo è disponibile nelle seguenti regioni: AWS

| Region | Codice |
| --- | --- |
| Stati Uniti orientali (Ohio) | `us-east-2` |
| Stati Uniti orientali (Virginia settentrionale) | `us-east-1` |

| Region | Codice |
| --- | --- |
| Stati Uniti occidentali (California settentrionale) | `us-west-1` |
| Stati Uniti occidentali (Oregon) | `us-west-2` |
| Asia Pacifico (Tokyo) | `ap-northeast-1` |
| Asia Pacifico (Seul) | `ap-northeast-2` |
| Asia Pacifico (Mumbai) | `ap-south-1` |
| Asia Pacifico (Singapore) | `ap-southeast-1` |
| Asia Pacifico (Sydney) | `ap-southeast-2` |
| Canada (Centrale) | `ca-central-1` |
| Europa (Francoforte) | `eu-central-1` |
| Europa (Irlanda) | `eu-west-1` |
| Europa (Londra) | `eu-west-2` |

# Creazione di processi ETL visivi

## Crea lavori ETL visivi con AWS Glue Studio

AWS Glue Studio fornisce un'interfaccia visiva per la creazione, l'esecuzione e il monitoraggio di lavori Extract/Transform/Load (ETL) in. AWS Glue Un job in AWS Glue è costituito dalla logica aziendale che esegue il lavoro di estrazione, trasformazione e caricamento (ETL). Con AWS Glue Studio, puoi comporre visivamente flussi di lavoro di trasformazione dei dati ed eseguirli senza problemi sul motore ETL serverless basato su AWS Glue Apache Spark. Puoi creare processi che spostano e trasformano i dati tra vari archivi di dati e flussi utilizzando un' drag-and-dropinterfaccia senza dover imparare Spark o scrivere codice.

Un processo AWS Glue incapsula uno script che si connette ai dati di origine, lo elabora e quindi lo scrive nella destinazione dati. Di solito un processo esegue script di estrazione, trasformazione e caricamento (ETL). I processi possono eseguire script progettati per ambienti di runtime Apache Spark e Ray. I processi possono anche eseguire script Python generici (processi shell di Python). I trigger AWS Glue possono avviare processi in base a una pianificazione, un evento oppure on demand. È possibile monitorare le esecuzioni dei processi per comprendere i parametri di runtime come esito positivo, durata e ora di inizio.

È possibile utilizzare gli script generati da AWS Glue oppure è possibile fornire i propri. Con uno schema di origine e una posizione o uno schema di destinazione, il generatore di AWS Glue Studio codice può creare automaticamente uno script Apache Spark API (). PySpark Puoi usare questo script come punto di partenza e modificarlo per soddisfare gli obiettivi.

AWS Glue può scrivere file di output in diversi formati di dati. Ogni tipo di processo può supportare diversi formati di output. Per alcuni formati di dati, possono essere scritti formati comuni di compressione.

## Gestione dei AWS Glue lavori nella console AWS

Per visualizzare i lavori esistenti, accedi Console di gestione AWS e apri la AWS Glue console all'indirizzo https://console.aws.amazon.com/glue/. Quindi scegli scheda Jobs (Processi) in AWS Glue. L'elenco Jobs (Processi) mostra l'ubicazione dello script associato a ciascun processo quando il processo è stato modificato e l'opzione di segnalibro del processo attuale.

Puoi creare processi nella sezione ETL della console AWS Glue. Durante la creazione di un nuovo processo o dopo averlo salvato, è possibile utilizzare AWS Glue Studio per modificare i processi

ETL. Poi farlo modificando i nodi nell'editor visivo o modificando lo script del processo in modalità sviluppatore. È inoltre possibile aggiungere e rimuovere nodi nell'editor visivo per creare processi ETL più complicati.

### Passaggi successivi per la creazione di un processo in AWS Glue Studio

Puoi utilizzare l'editor visivo dei processi per configurare i nodi per il processo. Ogni nodo rappresenta un'azione, ad esempio la lettura di dati dalla posizione di origine o l'applicazione di una trasformazione ai dati. Ogni nodo aggiunto al processo dispone di proprietà che forniscono informazioni sulla posizione dei dati o sulla trasformazione.

I passaggi successivi per la creazione e la gestione dei lavori sono:

- Avvio di processi ETL visivi in AWS Glue Studio
- Visualizzare lo script del processo
- Modificare le proprietà del processo
- Salvare il lavoro
- Avviare un'esecuzione del processo
- Visualizzare le informazioni sulle esecuzioni dei processi recenti
- Accesso al pannello di controllo di monitoraggio dei processi

## Crea flussi ETL visivi con Amazon SageMaker

Con un flusso di lavoro di Amazon SageMaker Unified Studio, puoi configurare ed eseguire una serie di attività in Amazon SageMaker Unified Studio. I flussi di lavoro di Amazon SageMaker Unified Studio utilizzano Apache Airflow per modellare le procedure di elaborazione dei dati e orchestrare gli artefatti del codice di Amazon Unified Studio. SageMaker Per ulteriori informazioni, consulta Utilizzo dei flussi di lavoro in Amazon SageMaker Unified Studio.

## Avvio di processi ETL visivi in AWS Glue Studio

Puoi utilizzare la semplice interfaccia visiva di AWS Glue Studio per creare i processi ETL. Puoi la pagina Jobs (Processi) per creare nuovi processi. È inoltre possibile usare un editor di script o notebook per lavorare direttamente con il codice nello script del processo ETL di AWS Glue Studio.

Nella pagina Jobs (Processi) puoi visualizzare tutti i processi creati con AWS Glue Studio o AWS Glue. In questa pagina puoi visualizzare, gestire ed eseguire i processi.

Vedi anche il tutorial del blog per un altro esempio di come creare processi ETL con AWS Glue Studio.

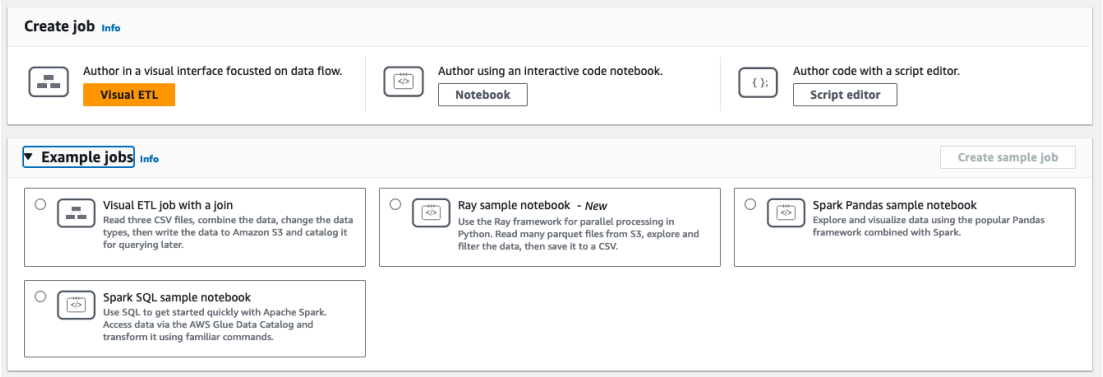
# Avvio di processi in AWS Glue Studio

AWS Glue consente di creare un processo tramite un'interfaccia visiva, un notebook di codice interattivo o un editor di script. È possibile avviare un processo facendo clic su una delle opzioni o creare un nuovo processo basato su un processo di esempio.

I processi di esempio creano processi con lo strumento che preferisci. Ad esempio, i processi di esempio consentono di creare un processo ETL visivo che esegue il join di file CSV in una tabella di catalogo, creare un processo in un notebook di codice interattivo con AWS Glue per Ray o AWS Glue per Spark quando si lavora con panda, oppure creare un processo in un notebook di codice interattivo con SparkSQL.

## Creazione di un processo AWS Glue Studio partendo da zero

1. Accedi a Console di gestione AWS e apri la AWS Glue Studio console all'indirizzo https://console.aws.amazon.com/gluestudio/.
2. Nel riquadro di navigazione, seleziona Processi ETL.
3. Nella sezione Crea processo, scegli un'opzione di configurazione per il processo.




Opzioni per creare un processo da zero:

- ETL visivo: crea il processo in un'interfaccia visiva incentrata sul flusso di dati
- Crea processi utilizzando un notebook a codice interattivo: crea processi in modo interattivo in un'interfaccia notebook basata su notebook Jupyter

Prima di selezionare questa opzione e creare una sessione di creazione di processi tramite notebook, è necessario fornire informazioni aggiuntive. Per ulteriori informazioni su come specificare queste informazioni, consulta Nozioni di base sui notebook in AWS Glue Studio.

- Crea codice con un editor di script: se hai familiarità con la programmazione e la scrittura di script ETL, scegli questa opzione per creare un nuovo processo ETL di Spark. Scegli il motore: shell Python, Ray, Spark (Python) o Spark (Scala). Quindi, scegli Inizia da zero o Carica script per caricare uno script esistente da un file locale. Se scegli di utilizzare l'editor di script, per progettare o modificare il tuo processo, non potrai utilizzare l'editor visivo dei processi.

  Un processo Spark viene eseguito in un ambiente Apache Spark gestito da AWS Glue. Per impostazione predefinita, i nuovi script sono codificati in Python. Per scrivere un nuovo script Scala, consulta Creazione e modifica di script Scala in AWS Glue Studio.

### Creazione di un processo AWS Glue Studio a partire da un processo di esempio

Puoi scegliere di creare un processo da un processo di esempio. Nella sezione Processi di esempio, scegli un processo di esempio, quindi scegli Crea processo di esempio. La creazione di un processo di esempio da una delle opzioni fornisce un modello rapido per iniziare a lavorare.

1. Accedi a Console di gestione AWS e apri la AWS Glue Studio console all'indirizzo https://console.aws.amazon.com/gluestudio/.
2. Nel riquadro di navigazione, seleziona Processi ETL.
3. Seleziona un'opzione per creare un processo da un processo di esempio:

   - Processo ETL visivo per eseguire il join di più origini: leggi tre file CSV, combina i dati, modifica i tipi di dati, quindi scrivi i dati su Amazon S3 e catalogali per le query successive.
   - Notebook Spark con Pandas: esplora e visualizza i dati utilizzando il popolare framework Pandas combinato con Spark.
   - Notebook Spark con SQL: inizia rapidamente a utilizzare Apache Spark tramite SQL. Accedi ai dati tramite Catalogo dati AWS Glue e trasformali utilizzando comandi familiari.
4. Scegli Crea un processo di esempio.

## Caratteristiche dell'editor dei processi

L'editor di processi offre le seguenti caratteristiche per la creazione e la modifica di processi.

- Un diagramma visivo del processo, con un nodo per ogni attività: nodi di origine dati per la lettura dei dati; nodi di trasformazione per la modifica dei dati; nodi di destinazione dati per la scrittura dei dati.

  È possibile visualizzare e configurare le proprietà di ciascun nodo nel diagramma del processo. È inoltre possibile visualizzare lo schema e i dati di esempio per ogni nodo nel diagramma del processo. Queste caratteristiche consentono di verificare che il processo stia modificando e trasformando i dati nel modo corretto, senza doverlo eseguire
- Una scheda di visualizzazione e modifica degli script, in cui è possibile modificare il codice generato per il processo.
- Una scheda per i dettagli del processo, in cui è possibile configurare diverse impostazioni per personalizzare l'ambiente in cui viene eseguito il processo ETL AWS Glue.
- Una scheda per le esecuzioni, in cui è possibile visualizzare le esecuzioni correnti e precedenti del processo, lo stato dell'esecuzione del processo e accedere ai registri per l'esecuzione del processo.
- Una scheda per la qualità dei dati, in cui è possibile applicare le regole sulla qualità dei dati al processo.
- Una scheda per le pianificazioni, in cui è possibile configurare l'ora di inizio del processo o impostare le esecuzioni del processo ricorrenti.
- Una scheda per il controllo della versione, in cui è possibile configurare un servizio Git da utilizzare con il processo.

## Utilizzo delle anteprime dello schema nell'editor visivo dei processi

Durante la creazione o la modifica del processo, è possibile utilizzare la scheda Output schema (Schema di output) per visualizzare lo schema dei dati.

Prima di poter visualizzare lo schema, l'editor dei processi necessita delle autorizzazioni per accedere all'origine dati. È possibile specificare un ruolo IAM nella scheda dei dettagli del processo dell'editor o nella scheda Output schema (Schema di output) per un nodo. Se il ruolo IAM dispone di tutte le autorizzazioni necessarie per accedere all'origine dati, è possibile visualizzare lo schema nella scheda Output schema (Schema di output) per un nodo.

## Utilizzo delle anteprime dei dati nell'editor visivo dei processi

Le anteprime dei dati consentono di creare e testare il processo, usando un esempio dei dati, senza doverlo eseguire ripetutamente. Utilizzando l'anteprima dei dati, puoi:

- Verifica un ruolo IAM per assicurarti di avere accesso alle origini dati o alle destinazioni dati.
- Controlla che la trasformazione stia modificando i dati nel modo previsto. Ad esempio, se utilizzi un filtro di trasformazione, puoi accertarti che il filtro stia selezionando il sottoinsieme di dati corretto.
- Controlla i dati. Se il set di dati contiene colonne con valori di più tipi, nell'anteprima dei dati viene visualizzato un elenco di tuple per tali colonne. Ogni tupla contiene il tipo di dato e il suo valore.

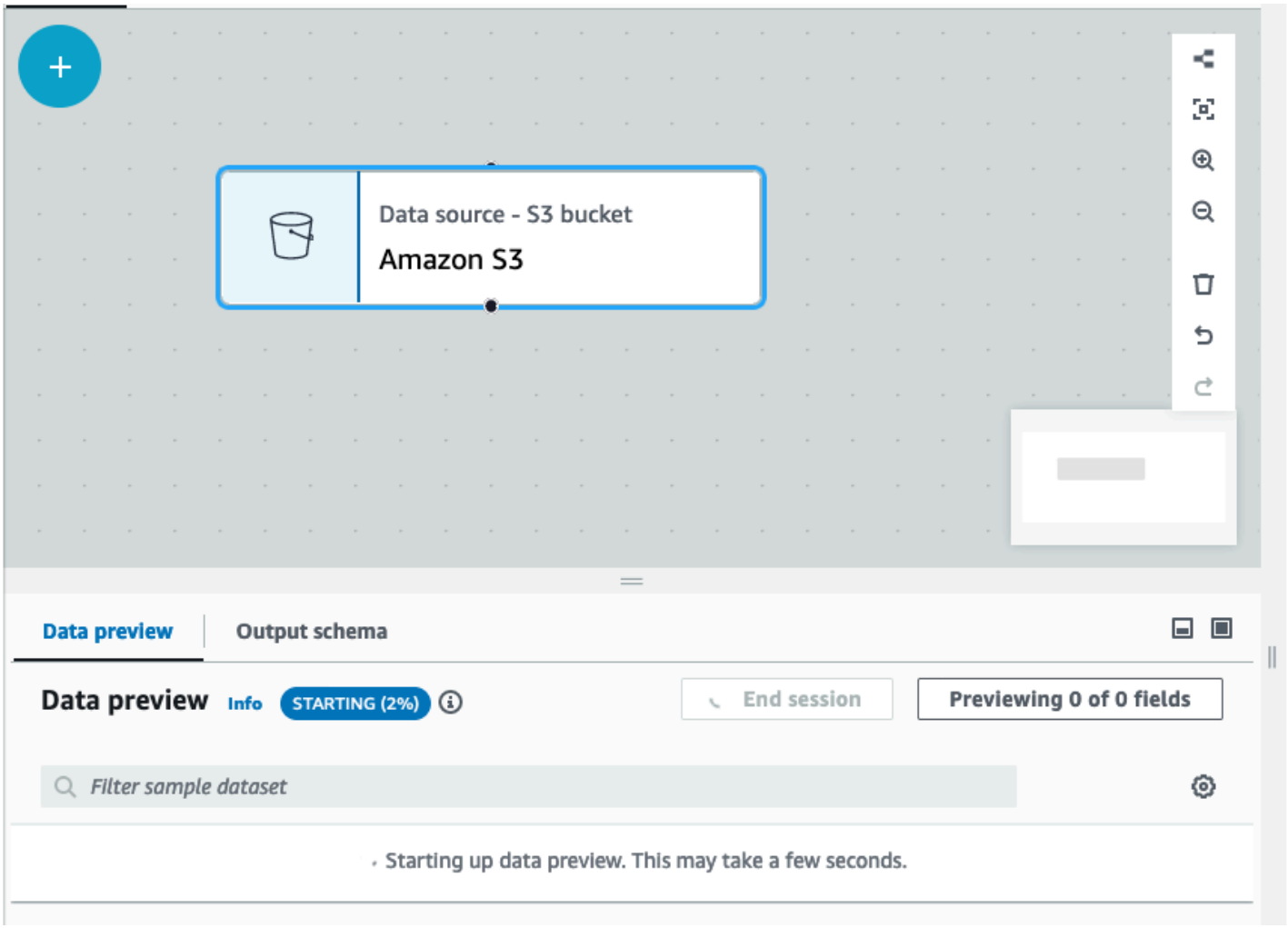
> **Note**
>
> Se utilizzi una sessione di anteprima dei dati e un nodo SQL o di codice personalizzato personalizzato, la sessione di anteprima dei dati eseguirà l'SQL o il blocco di codice così com'è per l'intero set di dati.

Durante la creazione o la modifica del processo, è possibile utilizzare la scheda Anteprima dei dati sotto il canvas del processo per visualizzare un campione dei dati. Una nuova sessione di anteprima dei dati verrà avviata automaticamente quando il ruolo è già configurato sul processo o è stato impostato un ruolo IAM predefinito nell'account. Se un ruolo non è stato configurato in precedenza, puoi avviare una sessione selezionando il ruolo.

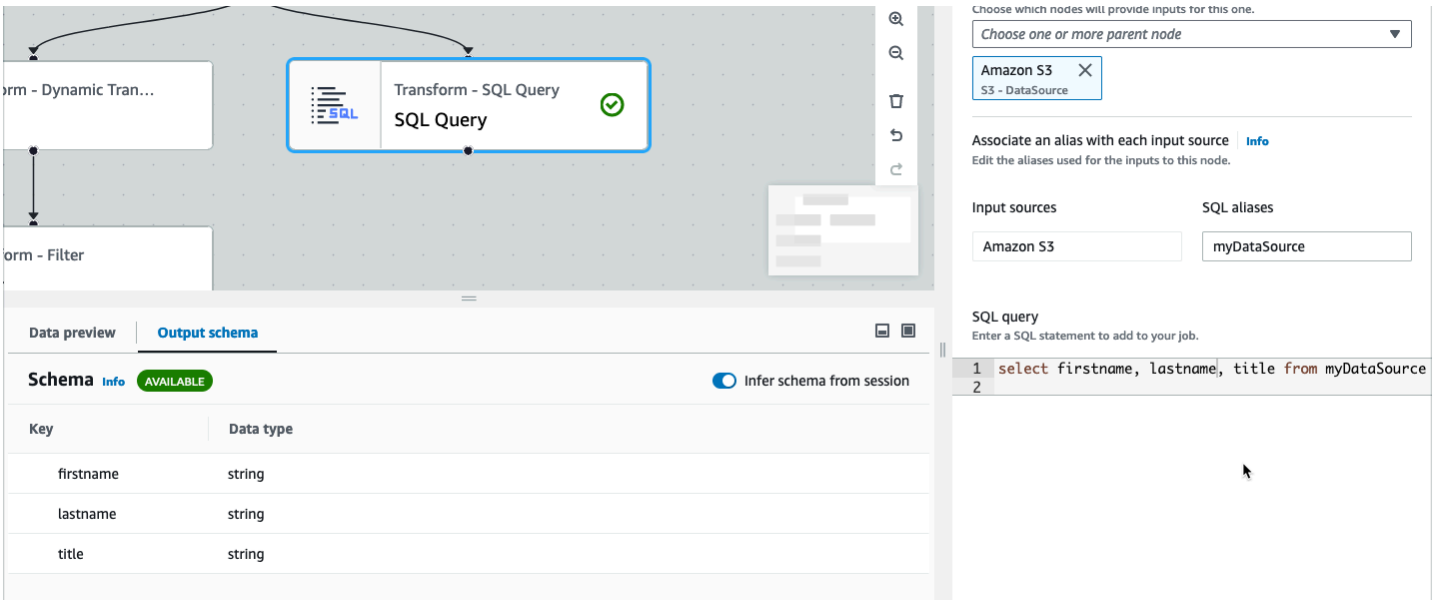
> **Note**
>
> Il ruolo scelto per la sessione di anteprima dei dati verrà utilizzato anche per il processo.

Puoi vedere lo stato e l'avanzamento della sessione, nonché i dettagli della sessione, facendo clic sull'icona delle informazioni.

Quando la sessione è pronta, AWS Glue Studio caricherà i dati per il nodo selezionato. È possibile visualizzare la percentuale di completamento man mano che procede.




Durante la creazione del processo visivo, AWS Glue Studio aggiornerà automaticamente lo schema per il nodo selezionato quando si attiva Deduci schema dalla sessione nella scheda Schema di output.




Per configurare le preferenze di anteprima dei dati:

Scegliere l'icona delle impostazioni (simbolo dell'ingranaggio) per configurare le preferenze per le anteprime dei dati. Queste impostazioni si applicano a tutti i nodi del diagramma del processo. Puoi:

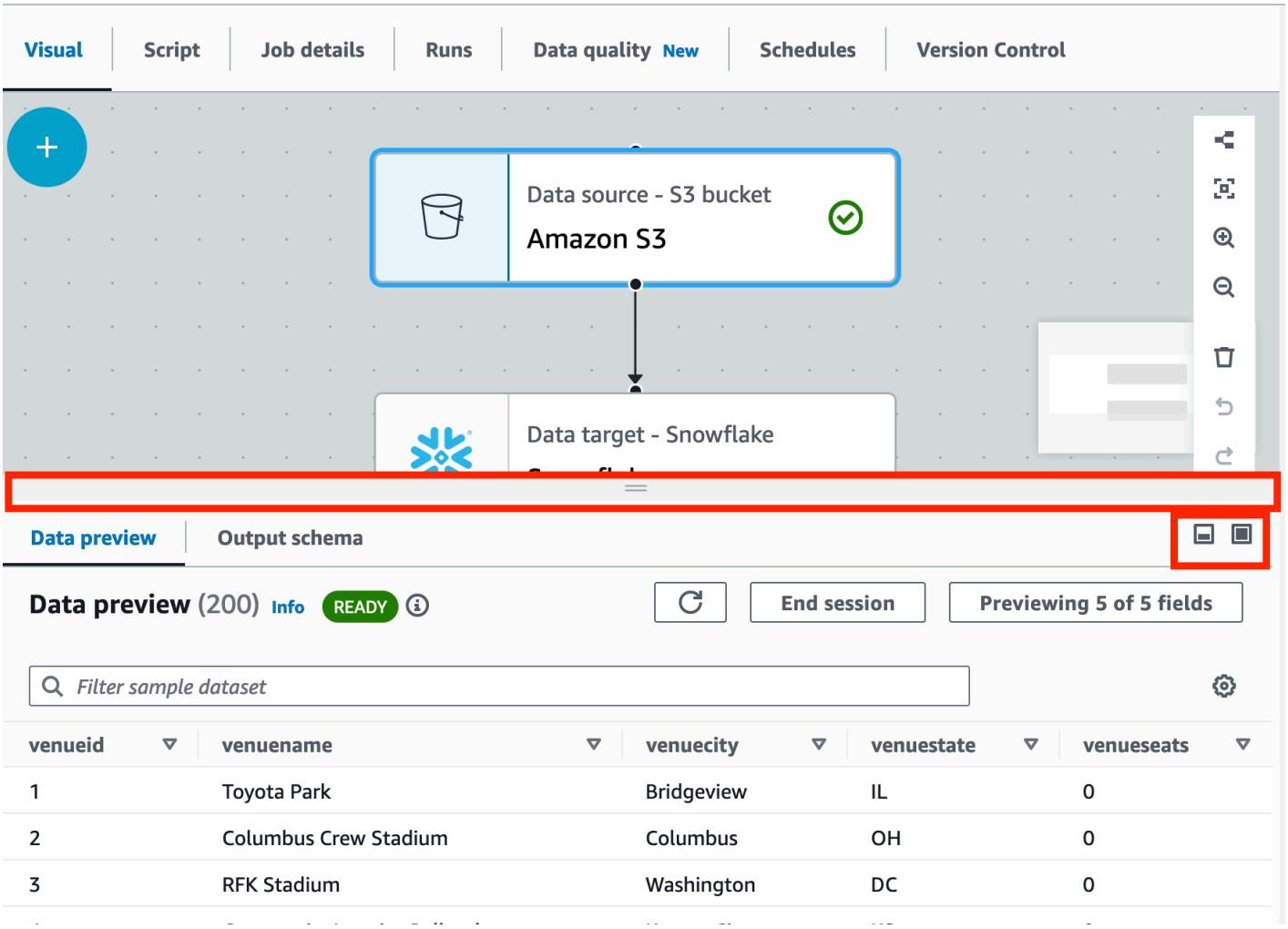
- Scegliere di avvolgere il testo da una riga all'altra. Per impostazione predefinita, questa opzione è abilitata.
- Modifica il numero di righe (il valore predefinito è 200)
- Scegli un ruolo IAM o creare un ruolo IAM, se necessario
- Scegli di avviare automaticamente una nuova sessione quando si crea un processo. Questo fornisce una nuova sessione interattiva durante la creazione dei processi. Questa impostazione si applica a livello di account. Una volta configurata, verrà applicata a tutti gli utenti dell'account durante la modifica di qualsiasi processo.
- Scegliere di dedurre automaticamente lo schema. Gli schemi di output verranno dedotti automaticamente per il nodo selezionato
- Scegli di importare automaticamente le librerie AWS Glue. Questo è utile perché impedirà che l'anteprima dei dati riavvii nuove sessioni quando si aggiungono nuove trasformazioni che richiedono il riavvio della sessione

Le funzionalità aggiuntive includono la possibilità di:

- Seleziona Previewing x of y fields (Anteprima dei campi x di y) per selezionare le colonne (campi) da visualizzare in anteprima. Quando si visualizzano in anteprima i dati utilizzando le impostazioni

di default, l'editor dei processi mostra le prime 5 colonne del set di dati. È possibile modificare questa impostazione per mostrare tutte o nessuna (non consigliato).

- Scorri la finestra di anteprima dei dati sia orizzontalmente che verticalmente.
- Per visualizzare meglio i dati e le strutture dei dati, utilizzare il pulsante di ingrandimento per espandere la scheda Anteprima dati e sovrapporre il grafico del processo. Allo stesso modo, utilizzare il pulsante di riduzione al minimo per ridurre al minimo la scheda Anteprima dei dati. È possibile anche selezionare la maniglia del riquadro e trascinarla verso l'alto per espandere la scheda Anteprima dei dati.




- Usa Termina sessione per interrompere l'anteprima dei dati. Quando interrompi la sessione, puoi scegliere un nuovo ruolo IAM e selezionare impostazioni aggiuntive (come attivare o disattivare le impostazioni) per avviare automaticamente una nuova sessione, dedurre lo schema o importare librerie AWS Glue, e riavviare la sessione.

## Restrizioni nell'utilizzo delle anteprime dei dati

Quando utilizzi le anteprime dati, potresti riscontrare le seguenti restrizioni o limitazioni.

- Selezionando la scheda Data preview (Anteprima dei dati) per la prima volta, ti verrà richiesto di scegliere un ruolo IAM. Questo ruolo deve disporre delle autorizzazioni necessarie per accedere ai dati e alle altre risorse necessarie per creare le anteprime dei dati.
- Dopo aver fornito un ruolo IAM, è necessario un po' di tempo prima che i dati siano disponibili per la visualizzazione. Per i set di dati con meno di 1 GB di dati, può essere necessario fino a un minuto. Se disponi di un set di dati di grandi dimensioni, utilizza le partizioni per ridurre il tempo di caricamento. Il caricamento dei dati direttamente da Amazon S3 offre le prestazioni migliori.
- Se disponi di un set di dati molto grande e sono necessari più di 15 minuti per eseguire query sui dati per l'anteprima, la richiesta scadrà. Le anteprime dei dati hanno un timeout di inattività di 30 minuti. Per ovviare a questo problema, riduci le dimensioni del set di dati per utilizzare le anteprime dei dati.
- Per impostazione predefinita, vengono visualizzate le prime 50 colonne nella scheda Anteprima dei dati. Se le colonne non contengono valori di dati, verrà visualizzato un messaggio che indica che non sono presenti dati da visualizzare. Puoi aumentare il numero di righe campionate o di colonne selezionate per visualizzare i valori dei dati.
- Le anteprime dei dati non sono attualmente supportate per le origini dati in streaming o per le origini dati che utilizzano connettori personalizzati.
- Gli errori su un nodo influiscono sull'intero processo. Se un nodo presenta un errore con le anteprime dei dati, l'errore verrà visualizzato su tutti i nodi finché non lo si corregge.
- Se si modifica un'origine dati per il processo, potrebbe essere necessario aggiornare i nodi figlio dell'origine dati in modo che corrispondano al nuovo schema. Ad esempio, se hai un ApplyMapping nodo che modifica una colonna e la colonna non esiste nell'origine dati sostitutiva, dovrai aggiornare il nodo di trasformazione. ApplyMapping
- Se visualizzi la scheda Data preview (Anteprima dei dati) per un nodo di trasformazione della query SQL e la query SQL utilizza un nome di campo non corretto, nella scheda viene visualizzato un errore.

## Generazione di codice dello script

Quando si utilizza l'editor visivo per creare un processo, il codice ETL viene generato automaticamente per te. AWS Glue Studio crea uno script di processo funzionale e completo e lo salva in una posizione Amazon S3.

Esistono due forme di codice generate da AWS Glue Studio: la versione originale o classica e una versione più recente e semplificata. Per impostazione predefinita, il nuovo generatore di codice viene utilizzato per creare lo script del processo. È possibile generare uno script di processo utilizzando il generatore di codice classico sulla scheda Script scegliendo il pulsante di attivazione Generate classic script (Genera script classico).

Alcune delle differenze nella nuova versione del codice generato includono:

- I blocchi di commenti di grandi dimensioni non vengono più aggiunti allo script
- Le strutture di output nel codice utilizzano il nome del nodo specificato nell'editor visivo. Nello script di classe, le strutture di output sono semplicemente denominate `DataSource0`, `DataSource1`, `Transform0`, `Transform1`, `DataSink0`, `DataSink1` e così via.
- I comandi lunghi sono divisi su più righe per eliminare la necessità di scorrere la pagina per visualizzare l'intero comando.

Le nuove caratteristiche di AWS Glue Studio richiedono la nuova versione di generazione del codice e non funziona con il classico script di codice. Quando si tenta di eseguire questi processi, viene richiesto di aggiornarli.

## Trasformare i dati con AWS Glue trasformazioni gestite

AWS Glue Studio offre due tipi di trasformazioni:

- Trasformazioni native di AWS Glue: gestite da AWS Glue e disponibili per tutti gli utenti.
- Trasformazioni visive personalizzate: consente di caricare le proprie trasformazioni da utilizzare in AWS Glue Studio

### Nodi di trasformazione dei dati gestiti da AWS Glue

AWS Glue Studio offre un set di trasformazioni predefinite che puoi usare per elaborare i dati. I dati passano da un nodo nel diagramma di processo a un altro in una struttura di dati denominata `DynamicFrame`, che è un'estensione di un SQL Apache Spark `DataFrame`.

Nel diagramma precompilato per un processo, tra i nodi di origine dati e di destinazione dati si trova il nodo di trasformazione Modifica schema. È possibile configurare questo nodo di trasformazione per modificare i dati oppure utilizzare ulteriori trasformazioni.

Le seguenti trasformazioni predefinite sono disponibili con AWS Glue Studio:

- ChangeSchema: mappa le chiavi di proprietà dei dati nell'origine dati alle chiavi di proprietà dei dati nella destinazione dati. È possibile rinominare le chiavi, modificare i tipi di dati per le chiavi e scegliere le chiavi da eliminare dal set di dati.
- SelectFields: Scegli le chiavi di proprietà dei dati che desideri conservare.
- DropFields: Scegli le chiavi di proprietà dei dati che desideri eliminare.
- RenameField: Rinomina una singola chiave di proprietà dei dati.
- Spigot: scrivi esempi dei dati in un bucket Amazon S3.
- Join: esegui il join di due set di dati in un set di dati utilizzando una frase di confronto sulle chiavi di proprietà dei dati specificate. È possibile utilizzare inner, outer, left, right, left semi e left anti join.
- Union: combina righe provenienti da più di un'origine dati che hanno lo stesso schema.
- SplitFields: divide le chiavi delle proprietà dei dati in due`DynamicFrames`. Output è una raccolta di `DynamicFrames`: uno con le chiavi di proprietà dei dati selezionate e uno con le chiavi di proprietà dei dati rimanenti.
- SelectFromCollection: Scegline una `DynamicFrame` da una raccolta di`DynamicFrames`. L'output è il `DynamicFrame` selezionato.
- FillMissingValues: individua i record nel set di dati che contengono valori mancanti e aggiungi un nuovo campo con un valore suggerito determinato mediante imputazione
- Filter: divide un set di dati in due, in base a una condizione di filtro.
- DropNullFields: rimuove le colonne dal set di dati se tutti i valori nella colonna sono "null".
- Elimina i duplicati: rimuove le righe dall'origine dati consentendo di scegliere se abbinare righe intere o specificare le chiavi.
- SQL: inserisce il codice SparkSQL in un campo di inserimento testo per utilizzare una query SQL e trasformare i dati. L'output è un singolo `DynamicFrame`.
- Aggregate: esegue un calcolo (ad esempio media, somma, min, max) su campi e righe selezionati e crea un nuovo campo con i valori appena calcolati.
- Flatten: estrae i campi all'interno delle strutture in campi di primo livello.
- UUID: aggiunge una colonna con un identificatore univoco universale per ogni riga.
- Identifier: aggiunge una colonna con un identificatore numerico per ogni riga.
- To timestamp: converte una colonna in un tipo di timestamp.
- Format timestamp: converte una colonna di timestamp in una stringa formattata.

- Conditional Router transform: applica più condizioni ai dati in ingresso. Ogni riga dei dati in ingresso viene valutata in base a una condizione di filtro di gruppo ed elaborata nel gruppo corrispondente.
- Trasformazione Concatena colonne: crea una nuova colonna di stringhe utilizzando i valori di altre colonne con un distanziatore opzionale.
- Trasformazione Dividi stringa: suddividi una stringa in un array di token utilizzando un'espressione regolare per definire come viene eseguita la suddivisione.
- Trasformazione Array a colonne: estrai alcuni o tutti gli elementi di una colonna di tipo array in nuove colonne.
- Trasformazione Aggiungi timestamp corrente: contrassegna le righe con l'ora in cui i dati sono stati elaborati. Ciò è utile per scopi di controllo o per tenere traccia della latenza nella pipeline di dati.
- Trasformazione Pivot: righe a colonne: aggrega una colonna numerica ruotando valori univoci su colonne selezionate che diventano nuove colonne. Se sono selezionate più colonne, i valori vengono concatenati per denominare le nuove colonne.
- Trasformazione Elimina pivot: righe a colonne: converti le colonne in valori di nuove colonne generando una riga per ogni valore univoco.
- Trasformazione Bilancia automaticamente elaborazione: ridistribuisci i dati tra i worker per migliorare le prestazioni. Ciò è utile nei casi in cui i dati non sono bilanciati o, poiché provengono dall'origine, non consentono un'elaborazione parallela sufficiente.
- Trasformazione Colonna derivata: definisci una nuova colonna basata su una formula matematica o un'espressione SQL in cui è possibile utilizzare altre colonne nei dati, oltre a costanti e valori letterali.
- Trasformazione Ricerca: aggiungi colonne da una tabella di catalogo definita quando le chiavi corrispondono alle colonne di ricerca definite nei dati.
- Trasformazione Espandi array o mappa: estrae i valori da una struttura annidata in singole righe più facili da manipolare.
- Trasformazione Corrispondenza dei record: richiama una trasformazione di classificazione dei dati di machine learning Corrispondenza dei record esistente.
- Trasformazione Rimuovi righe nulle: rimuove dal set di dati le righe che hanno tutte le colonne come nulle o vuote.
- Trasformazione Analizza colonna JSON: analizza una colonna di stringhe contenente dati JSON e convertila in una struttura o in una colonna di array, a seconda che il JSON sia rispettivamente un oggetto o un array.
- Trasformazione Estrai percorso JSON: estrai nuove colonne da una colonna di stringhe JSON.

- Trasformazione Estrai frammenti di stringa con un'espressione regolare: estrai frammenti di stringa utilizzando un'espressione regolare e crea a partire da essa una nuova colonna o anche più colonne, se si utilizzano gruppi di espressioni regolari.
- Custom transform: inserisce il codice in un campo di inserimento testo per utilizzare le trasformazioni personalizzate. L'output è una raccolta di `DynamicFrames`.

## Utilizzo di una ricetta di preparazione dei dati in AWS Glue Studio

La trasformazione della Ricetta di preparazione dei dati consente di creare una ricetta di preparazione dei dati da zero utilizzando un'interfaccia di creazione interattiva in stile griglia. Consente inoltre di importare una ricetta AWS Glue DataBrew esistente e quindi modificarla in AWS Glue Studio.

Il nodo Ricetta di preparazione dei dati è disponibile nel pannello Risorse. È possibile connettere il nodo Ricetta di preparazione dei dati a un altro nodo del flusso di processo visivo, che si tratti di un nodo Origine dati o di un altro nodo di trasformazione. Dopo aver scelto una AWS Glue DataBrew ricetta e una versione, i passaggi applicati nella ricetta sono visibili nella scheda delle proprietà del nodo.

### Prerequisiti

- Se si importa una AWS Glue DataBrew ricetta, si dispone delle autorizzazioni IAM richieste, come descritto in. Importa una AWS Glue DataBrew ricetta in AWS Glue Studio
- È necessario creare una sessione di anteprima dei dati.

### Limitazioni

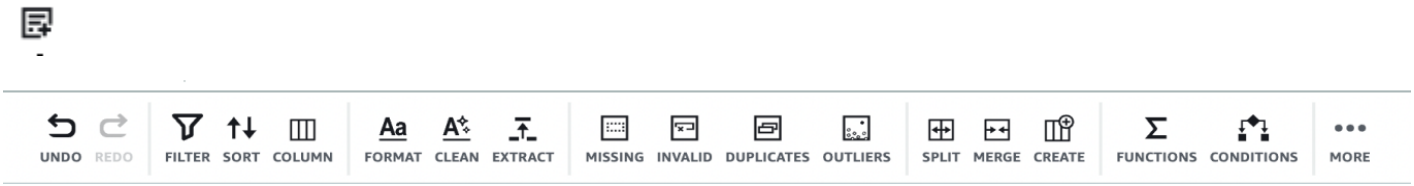
- AWS Glue DataBrew le ricette sono supportate solo nelle regioni commerciali DataBrew .
- Non tutte le AWS Glue DataBrew ricette sono supportate daAWS Glue. Alcune ricette non potranno essere eseguite in AWS Glue Studio.
  - Le ricette con trasformazioni `UNION` e `JOIN` non sono supportate, tuttavia AWS Glue Studio dispone già di nodi di trasformazione "Join" e "Union" che possono essere utilizzati prima o dopo un nodo Ricetta di preparazione dei dati.
- I nodi Ricetta di preparazione dei dati sono supportati per i processi a partire da AWS Glue versione 4.0. Questa versione verrà selezionata automaticamente dopo l'aggiunta di un nodo Ricetta di preparazione dei dati al processo.

- I nodi Ricetta di preparazione dei dati richiedono Python. Viene impostato automaticamente quando il nodo Ricetta di preparazione dei dati viene aggiunto al processo.
- L'aggiunta di un nuovo nodo Ricetta di preparazione dei dati al grafico visivo riavvierà automaticamente la sessione di Anteprima dei dati con le librerie corrette per utilizzare il nodo Ricetta di preparazione dei dati.
- Le seguenti trasformazioni non sono supportate per l'importazione o la modifica in un nodo Ricetta di preparazione dei dati: `GROUP_BY`, `PIVOT`, `UNPIVOT`, `eTRANSPOSE`.

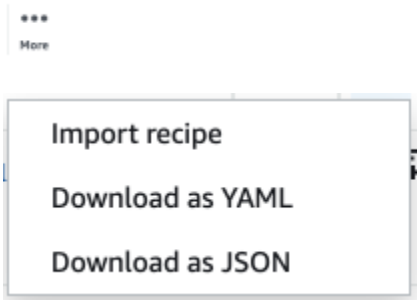
### Funzionalità aggiuntive

Dopo aver selezionato la trasformazione Ricetta di preparazione dei dati, hai la possibilità di intraprendere azioni aggiuntive dopo aver scelto Author recipe.

- Aggiungi passaggio: puoi aggiungere ulteriori passaggi a una ricetta, se necessario, scegliendo l'icona Aggiungi passaggio oppure utilizzare la barra degli strumenti nel riquadro di anteprima scegliendo un'azione.




- Importa ricetta: scegli Altro, quindi Importa ricetta da utilizzare nel tuo AWS Glue Studio processo.

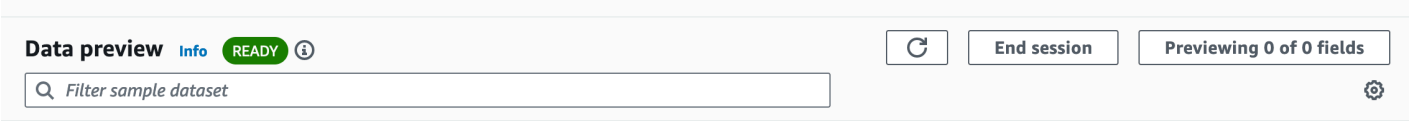



- Scarica come YAML: scegli Altro, quindi Scarica come YAML per scaricare la ricetta da salvare all'esterno. AWS Glue Studio
- Scarica come JSON: scegli Altro, quindi Scarica come JSON per scaricare la ricetta da salvare all'esterno. AWS Glue Studio
- Annulla e ripeti i passaggi della ricetta: puoi annullare e ripetere i passaggi della ricetta nel riquadro di anteprima quando lavori con i dati nella griglia.

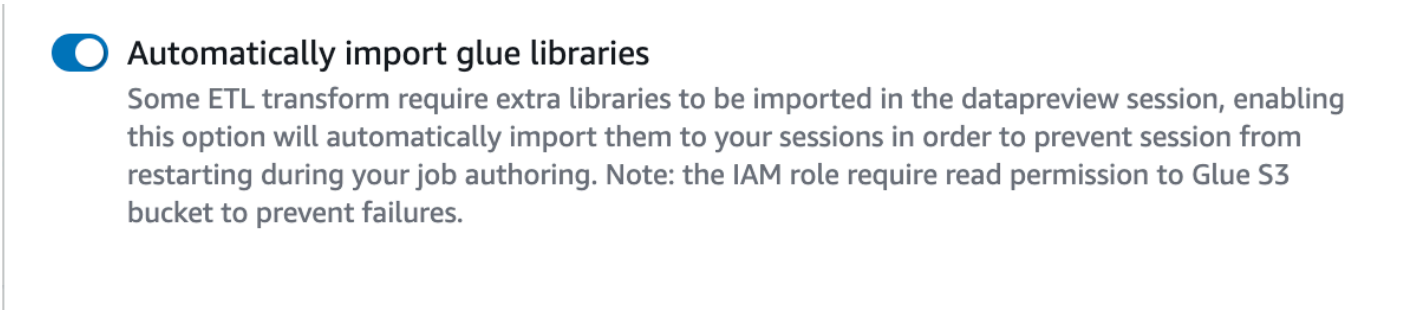
Crea ed esegui ricette di preparazione dei dati in un job ETL AWS Glue visivo

In questo scenario, puoi creare ricette di preparazione dei dati senza doverle prima creare. DataBrew Prima di iniziare a creare ricette, devi:

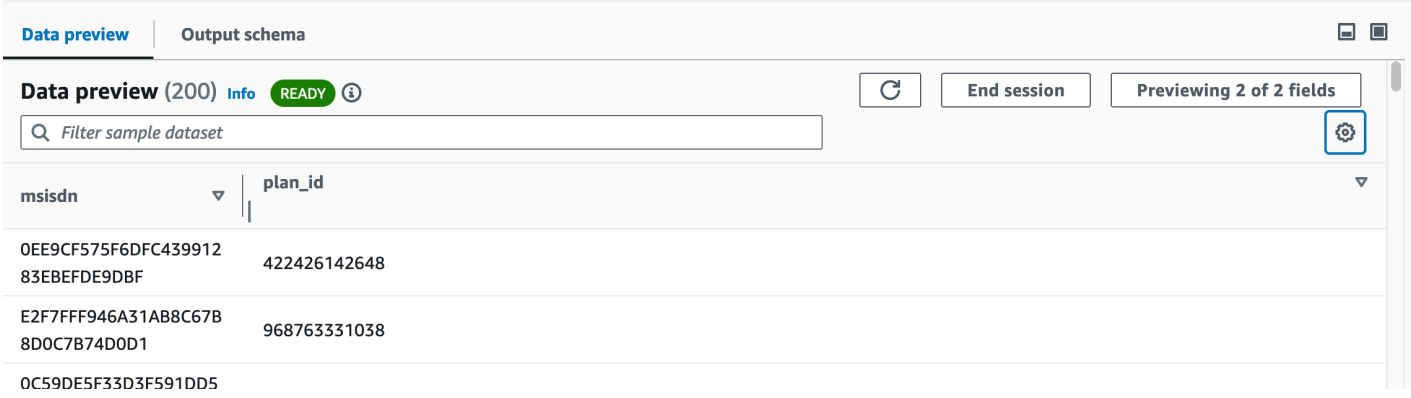
- Avere in esecuzione una sessione di Anteprima dati attiva. Quando la sessione di anteprima dei dati è PRONTA, Author Recipe diventerà attiva e potrai iniziare a creare o modificare la tua ricetta.




- Assicurati che l'interruttore per l'importazione automatica delle librerie glue sia abilitato.

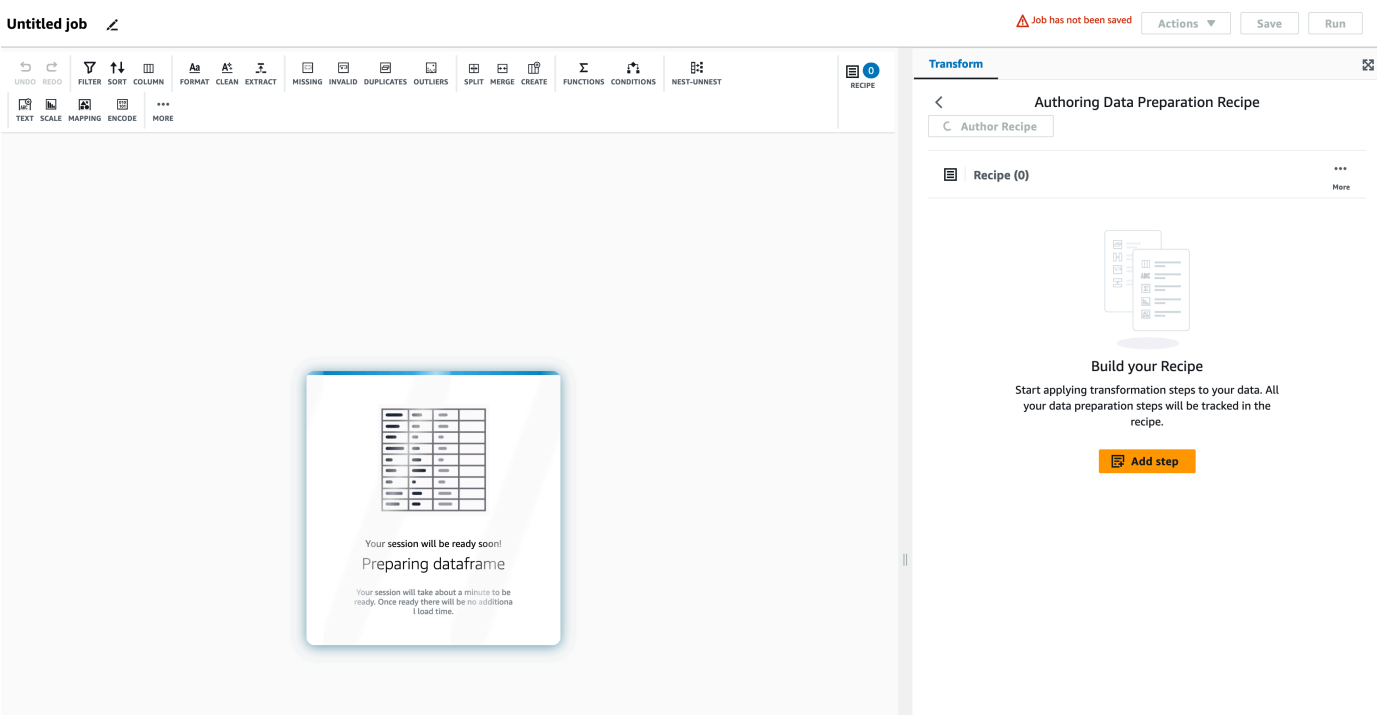



Puoi farlo scegliendo l'icona a forma di ingranaggio nel riquadro Anteprima dati.

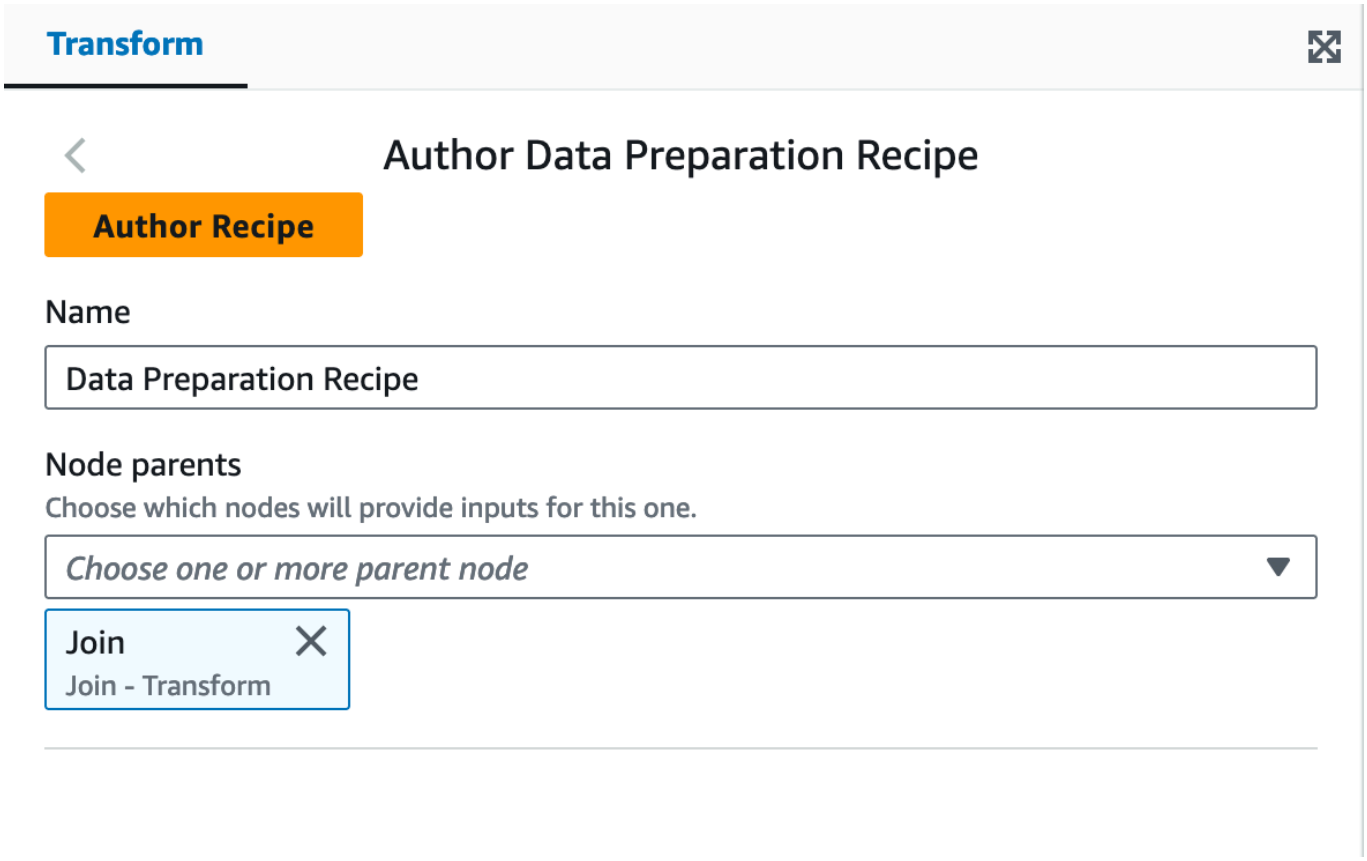


Per creare una ricetta di preparazione dei dati in AWS Glue Studio:

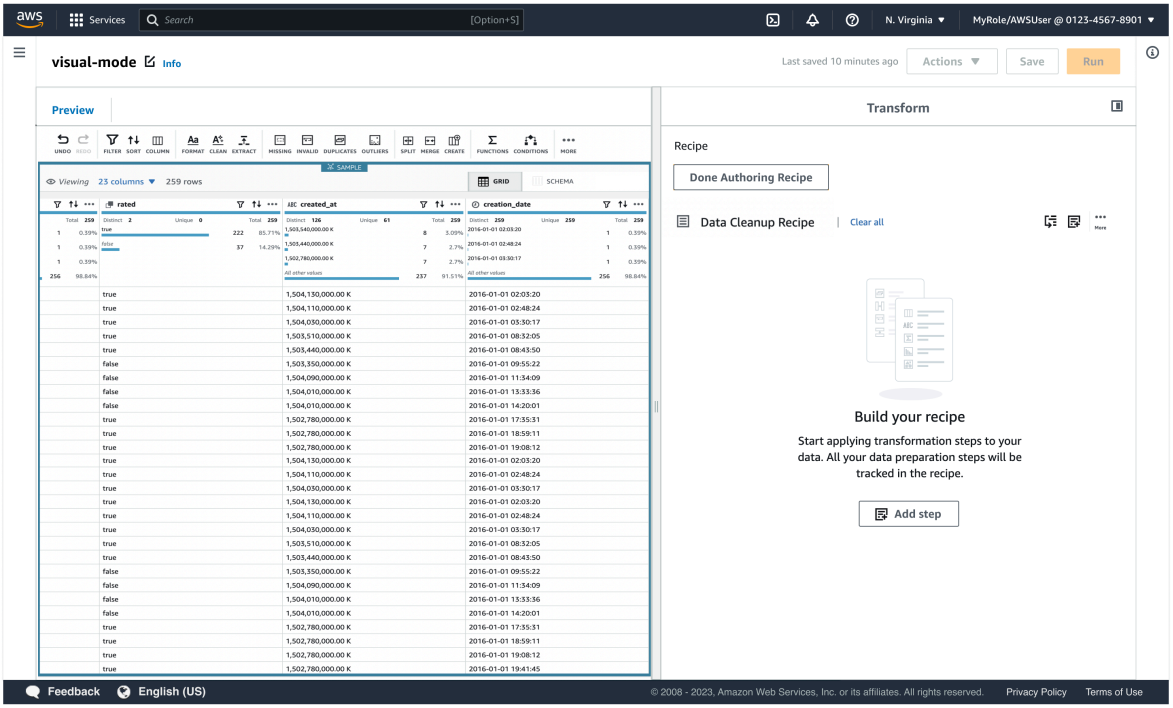
1. Aggiungi il trasformatore Data Preparation Recipe al canvas del lavoro. Il trasformatore deve essere connesso a un nodo di origine dati padre. Quando aggiungi il nodo Data Preparation Recipe, questo si riavvierà con le librerie appropriate e vedrai il frame dei dati in preparazione.



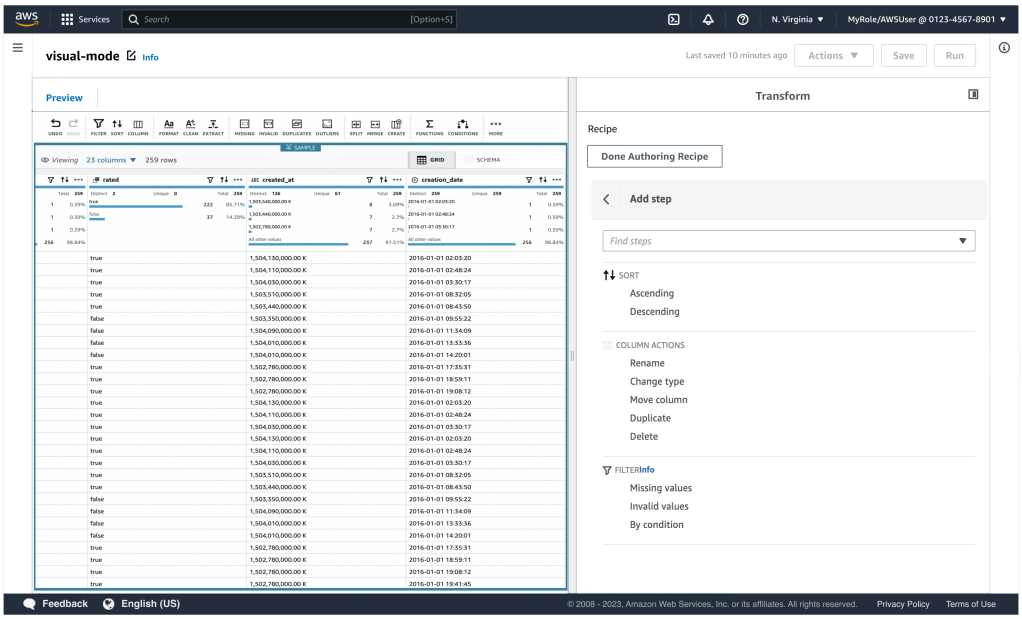

2. Una volta che la sessione di Anteprima dati è pronta, i dati con tutti i passaggi applicati in precedenza verranno visualizzati nella parte inferiore dello schermo.
3. Scegli Author Recipe. Ciò ti consentirà di iniziare una nuova ricetta in AWS Glue Studio.




4. Nel pannello Trasforma a destra del canvas di lavoro, inserisci un nome per la ricetta di preparazione dei dati.
5. Sul lato sinistro, il canvas verrà sostituito da una visualizzazione a griglia dei dati. A destra, il pannello Trasforma cambierà per mostrare i passaggi della ricetta. Scegli Aggiungi passaggio per aggiungere il primo passaggio della ricetta.

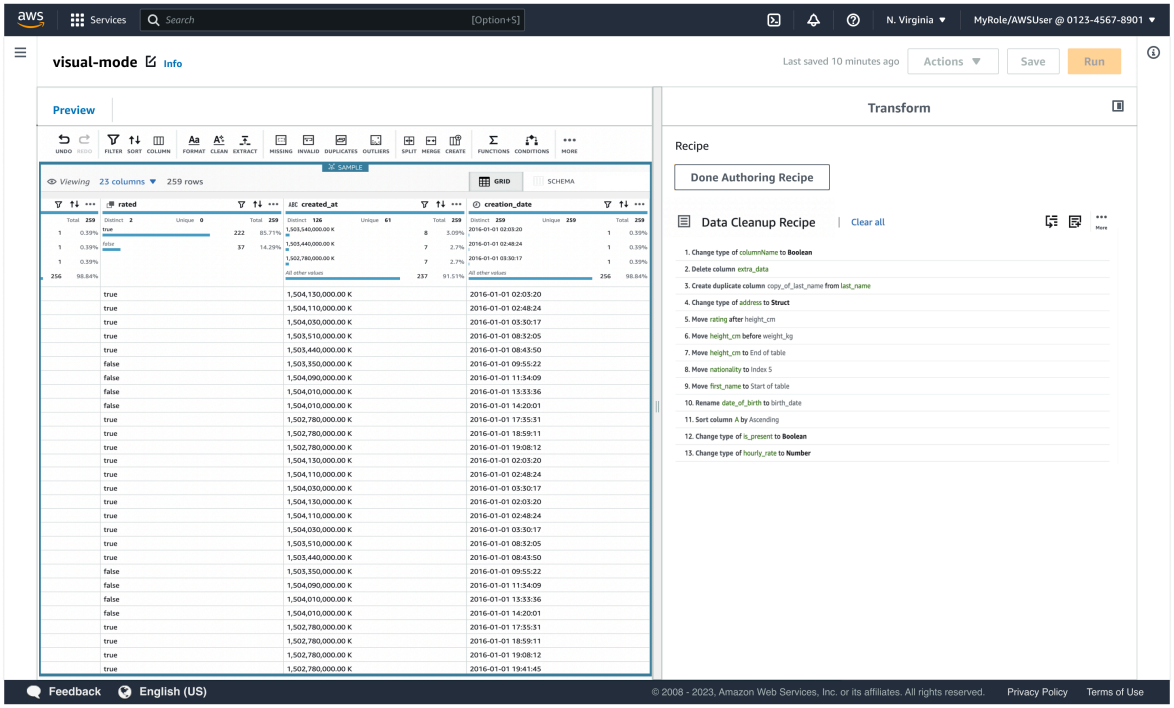



6. Nel pannello Trasforma puoi scegliere di ordinare, eseguire un'azione sulla colonna e filtrare i valori. Ad esempio, scegli Rinomina colonna.




7. Nel pannello Trasforma sul lato destro, le opzioni per rinominare una colonna consentono di scegliere la colonna di origine da rinominare e di inserire il nuovo nome della colonna. Dopo averlo fatto, scegli Applica.

   Puoi visualizzare in anteprima ogni passaggio, annullarne uno, riordinarli e utilizzare qualsiasi icona di azione (Filtra, Ordina, Dividi, Unisci, ecc.) Quando esegui azioni nella griglia di dati, i passaggi vengono aggiunti alla ricetta nel pannello Trasforma.

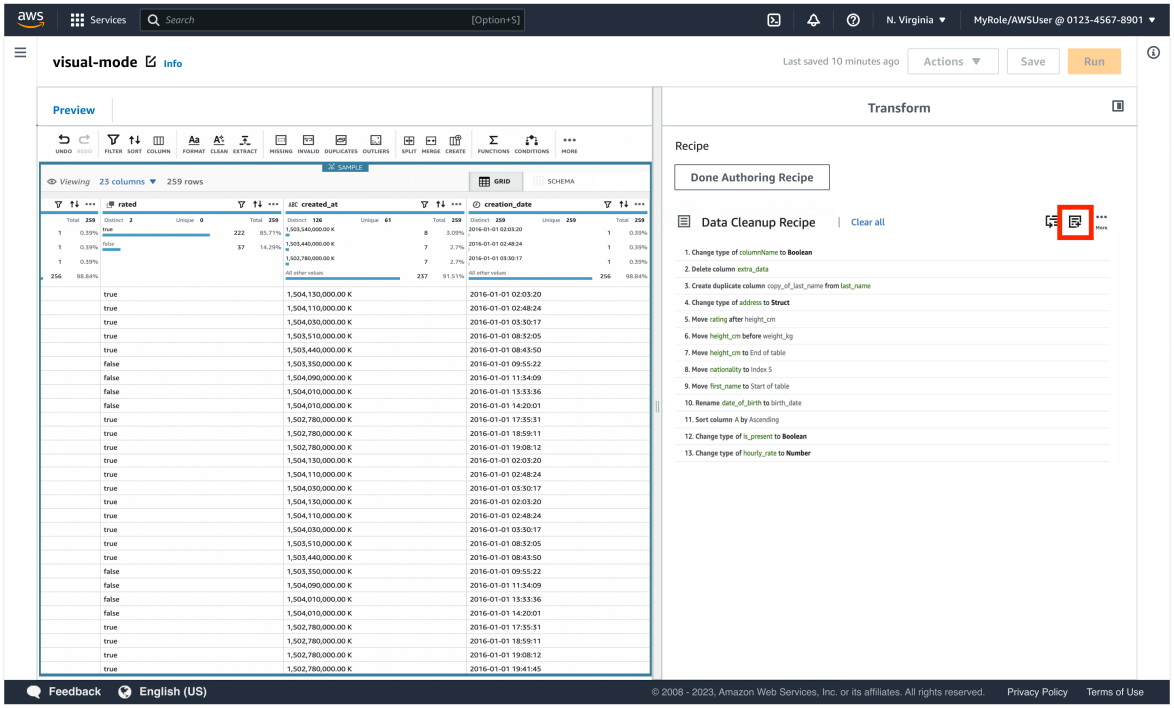



Se devi apportare una modifica, puoi farlo nel riquadro Anteprima visualizzando in anteprima il risultato di ogni passaggio, annullandolo e riordinandoli. Esempio:

- Undo/redo passaggio: annulla un passaggio scegliendo l'icona Annulla. Puoi ripetere un passaggio scegliendo l'icona Ripeti.

- Fase di riordino: quando riordini una fase, AWS Glue Studio convaliderà ogni passaggio e ti comunicherà se non è valido.

8. Dopo aver applicato un passaggio, il pannello Trasforma ti mostrerà tutti i passaggi della ricetta. Puoi cancellare tutti i passaggi per ricominciare da capo, aggiungerne altri scegliendo l'icona Aggiungi o scegliere Done Authoring Recipe.




9. Scegli Salva nella parte in alto a destra dello schermo. I passaggi della ricetta non verranno salvati finché non salverai il lavoro.

## Importa una AWS Glue DataBrew ricetta in AWS Glue Studio

In AWS Glue DataBrew, una ricetta è un insieme di passaggi di trasformazione dei dati. AWS Glue DataBrew recipes descrive come trasformare i dati che sono già stati letti e non descrive dove e come leggere i dati, né come e dove scrivere i dati. Questo è configurato nei nodi di origine e destinazione in AWS Glue Studio. Per ulteriori informazioni sulle ricette, consulta Creazione e utilizzo delle AWS Glue DataBrew ricette.

Per utilizzare AWS Glue DataBrew le ricette inAWS Glue Studio, iniziate con la creazione di ricette in AWS Glue DataBrew. Se disponi di ricette che desideri utilizzare, puoi ignorare questo passaggio.

## Autorizzazioni IAM per AWS Glue DataBrew

Questo argomento fornisce informazioni per aiutarti a comprendere le azioni e le risorse che un amministratore IAM può utilizzare in una policy AWS Identity and Access Management (IAM) per la trasformazione Data Preparation Recipe.

Per ulteriori informazioni sulla sicurezza in AWS Glue, consulta Gestione degli accessi.

> **Note**
>
> La tabella seguente elenca le autorizzazioni necessarie perché un utente possa importare una ricetta AWS Glue DataBrew esistente.

Azioni di trasformazione della ricetta di preparazione dei dati

| Azione | Description |
| --- | --- |
| `databrew:ListRecipes` | Concede l'autorizzazione a recuperare le ricette AWS Glue DataBrew . |
| `databrew:ListRecipeVersions` | Concede l'autorizzazione a recuperare le versioni delle ricette AWS Glue DataBrew . |
| `databrew:DescribeRecipe` | Concede l'autorizzazione a recuperare la descrizione della ricetta AWS Glue DataBrew . |

Il ruolo che stai utilizzando per accedere a questa funzionalità deve disporre di una policy che consenta diverse azioni AWS Glue DataBrew . A tale scopo, puoi utilizzare una policy `AWSGlueConsoleFullAccess` che includa le azioni necessarie o aggiungere la seguente policy incorporata al tuo ruolo:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
```

```
      "Action": [
        "databrew:ListRecipes",
        "databrew:ListRecipeVersions",
        "databrew:DescribeRecipe"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

Per utilizzare la trasformazione Ricetta di preparazione dei dati, devi aggiungere l'operazione `IAM:PassRole` alla policy delle autorizzazioni.

Autorizzazioni aggiuntive richieste

| Azione | Description |
| --- | --- |
| `iam:PassRole` | Concede a IAM l'autorizzazione per consentire all'utente di trasmettere i ruoli approvati. |

Senza queste autorizzazioni si verifica il seguente errore:

```
"errorCode": "AccessDenied"
"errorMessage": "User: arn:aws:sts::account_id:assumed-role/AWSGlueServiceRole is not
authorized to perform: iam:PassRole on resource: arn:aws:iam::account_id:role/service-
role/AWSGlueServiceRole
because no identity-based policy allows the iam:PassRole action"
```

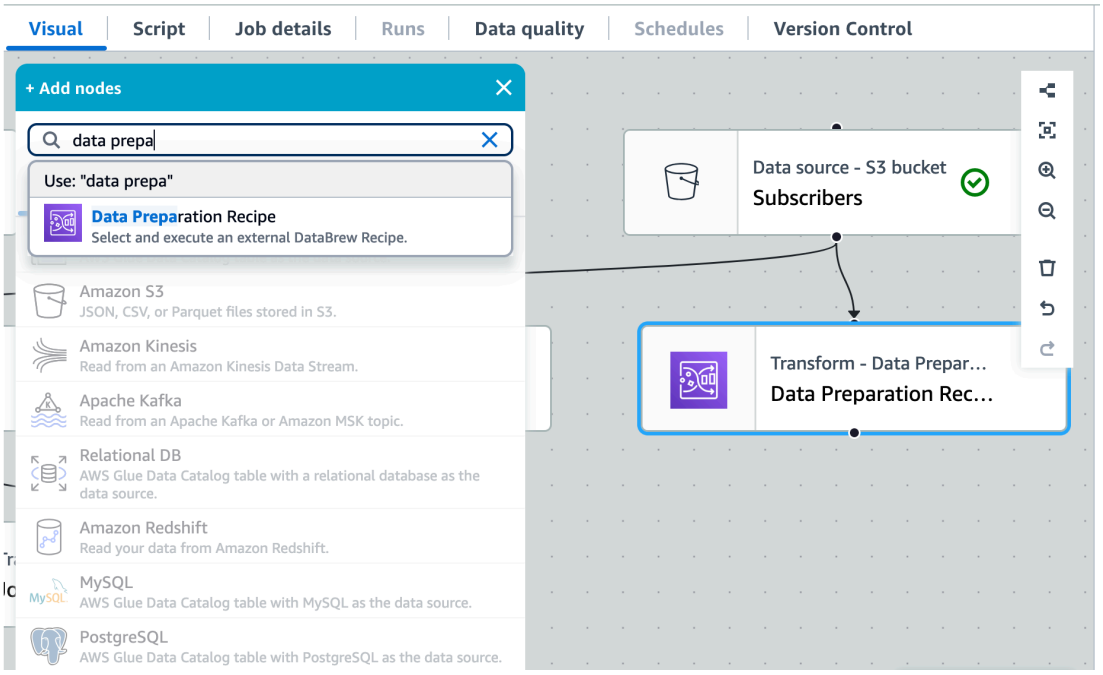
Importazione di una ricetta AWS Glue DataBrew

Per importare una AWS Glue DataBrew ricetta e utilizzarla inAWS Glue Studio:

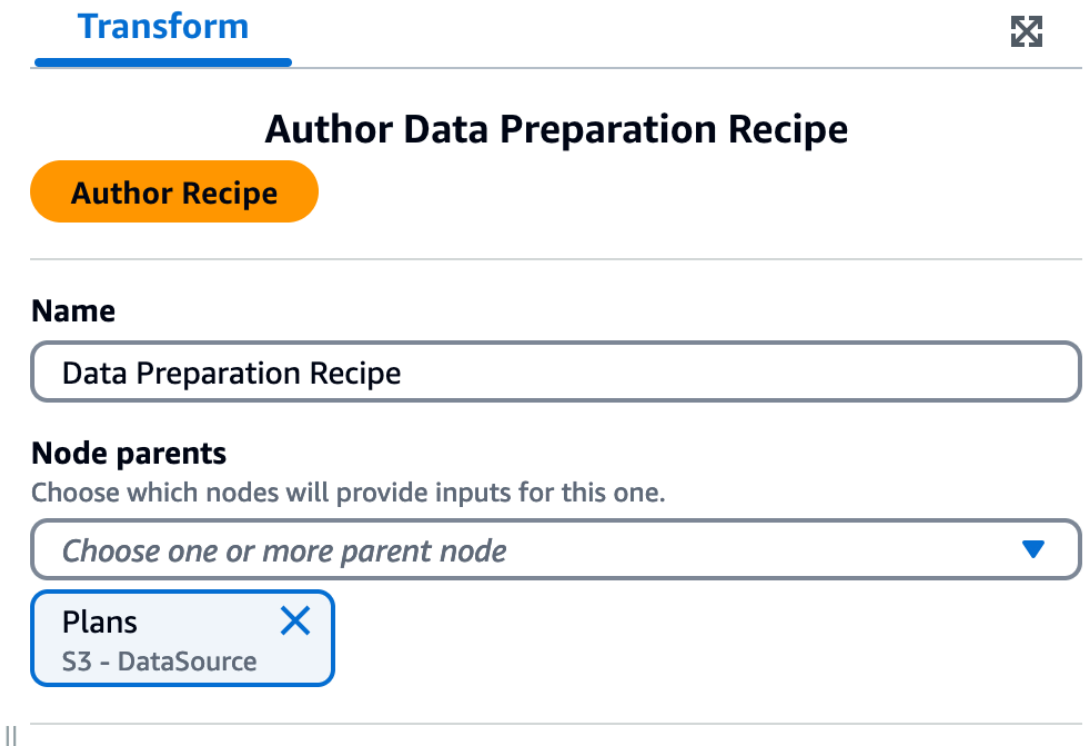
Se disponi di un nodo Data Preparation Recipe e desideri modificare i passaggi della ricetta direttamente in AWS Glue Studio, dovrai importare i passaggi della ricetta nel tuo AWS Glue Studio lavoro.

1. Avvia un processo AWS Glue con un'origine dati AWS Glue Studio.

2. Aggiungi il nodo Data Preparation Recipe al canvas del lavoro.

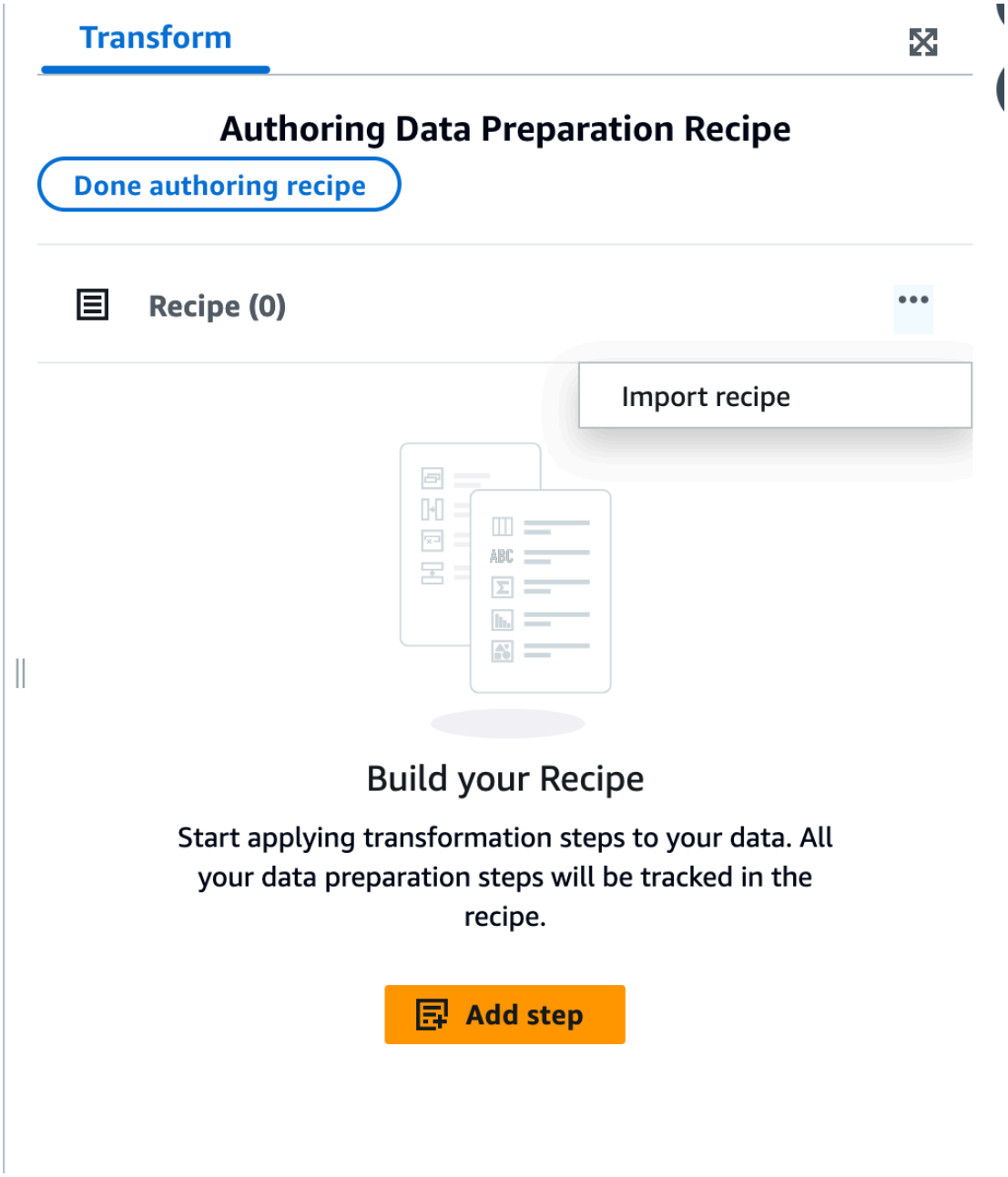



3. Nel pannello Trasforma, inserisci un nome per la ricetta.
4. Scegli uno o più nodi padre selezionando i nodi disponibili sul canvas dall'elenco a discesa.
5. Scegli Author Recipe. Se Author Recipe è grigio, non è disponibile fino a quando non sono stati selezionati i nodi padre e non è terminata una sessione di anteprima dei dati.

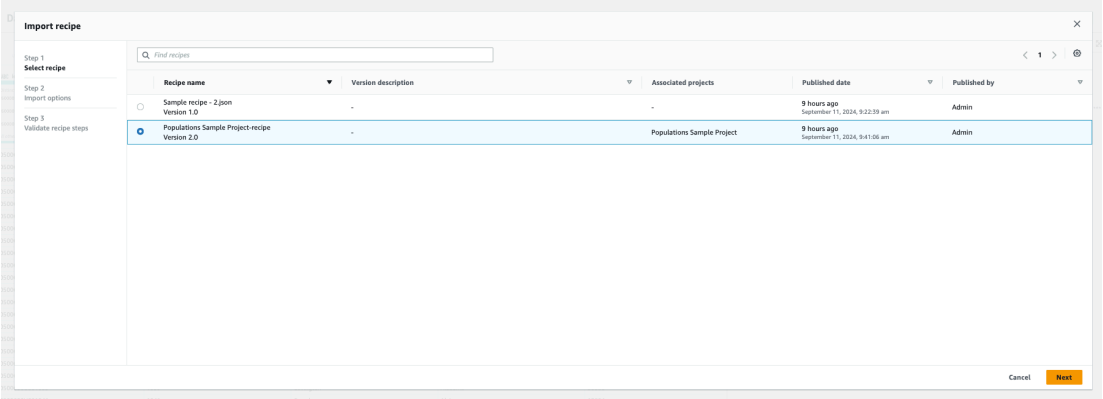


6. Il frame di dati viene caricato e mostra informazioni dettagliate sui dati di origine.

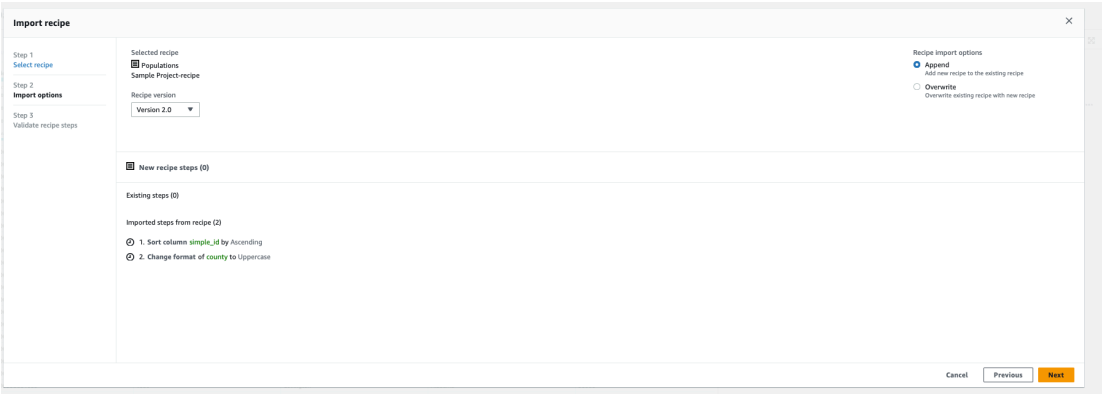
   Seleziona l'icona Altre azioni e scegli Importa ricetta.



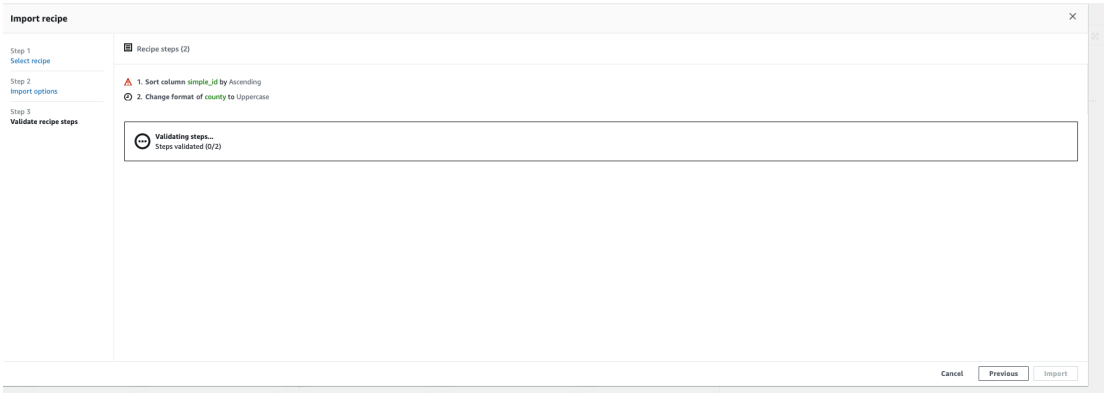

7. Usa la procedura guidata di importazione della ricetta per completare i passaggi. Nel passaggio 1, cerca la tua ricetta, selezionala e scegli Avanti.

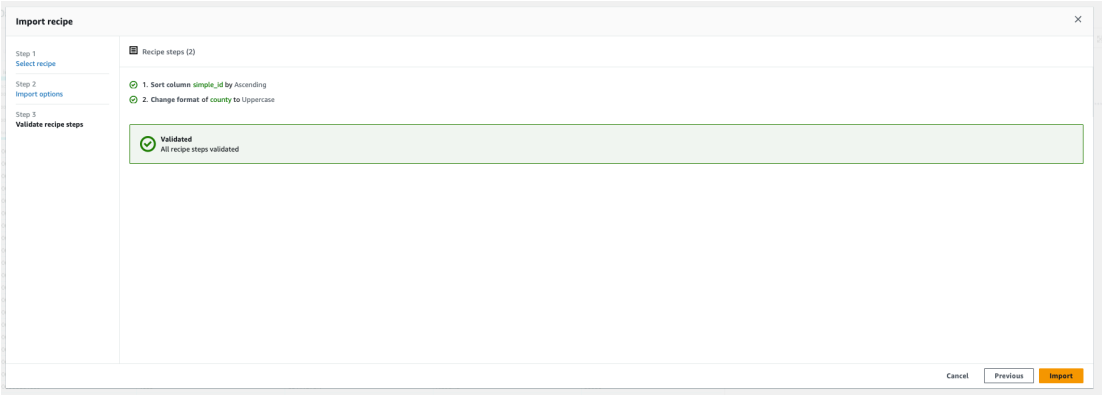



8. Nel passaggio 2, scegli le opzioni di importazione. Puoi scegliere di aggiungere una nuova ricetta a una ricetta esistente o di sovrascrivere una ricetta esistente. Scegli Next (Successivo).




9. Nel passaggio 3, convalida i passaggi della ricetta. Una volta importata la AWS Glue DataBrew ricetta, puoi modificarla direttamente in. AWS Glue Studio






10. Dopodiché, i passaggi verranno importati come parte del tuo lavoro. AWS Glue Apporta le modifiche di configurazione necessarie nella scheda Dettagli del lavoro, ad esempio assegnando un nome al lavoro e regolando la capacità allocata in base alle esigenze. Scegli Salva per salvare il lavoro e la ricetta.

> **Note**
>
> JOIN, UNION, GROUP_BY, PIVOT, UNPIVOT, TRANSPOSE non sono supportati per l'importazione delle ricette, né saranno disponibili nella modalità di creazione delle ricette.

11. Facoltativamente, puoi completare la creazione del lavoro aggiungendo altri nodi di trasformazione secondo necessità e aggiungendo i nodi di destinazione dei dati.

    Se riordini i passaggi dopo aver importato una ricetta, AWS Glue esegue la convalida di tali passaggi. Ad esempio, se hai rinominato e poi eliminato una colonna e hai spostato il passaggio di eliminazione in alto, il passaggio di ridenominazione non è valido. Puoi quindi modificare i passaggi per correggere l'errore di convalida.

## Migrazione da AWS Glue DataBrew AWS Glue Studio

Se hai delle ricette AWS Glue DataBrew, usa la seguente lista di controllo per migrare le tue ricette su AWS Glue Studio.

| Se vuoi | Allora completa le seguenti operazioni |
| --- | --- |
| Consenti agli utenti di recuperare AWS Glue DataBrew ricette, versioni e descrizioni delle ricette. | Aggiungi le autorizzazioni IAM a una policy che consenta al tuo ruolo di accedere alle |

| Se vuoi | Allora completa le seguenti operazioni |
| --- | --- |
| | azioni necessarie. Per informazioni, consulta Autorizzazioni IAM per AWS Glue DataBrew. |
| Importa una AWS Glue DataBrew ricetta esistente in AWS Glue Studio. | Segui la procedura riportata in Importazione di una ricetta AWS Glue DataBrew. |
| Importa una ricetta con JOIN e UNION. | Le ricette con le trasformazioni UNION e JOIN non sono supportate. Usa le trasformazioni Join e Union in AWS Glue Studio prima o dopo un nodo Data Preparation Recipe. |

## Utilizzo di Modifica schema per mappare nuovamente le chiavi delle proprietà dei dati

Una trasformazione Modifica schema mappa nuovamente le chiavi di proprietà dei dati di origine nella configurazione desiderata per i dati di destinazione. In un nodo di trasformazione Modifica schema, puoi:

- Modificare il nome di più chiavi di proprietà dati.
- Modificare il tipo di dati delle chiavi di proprietà dei dati, se il nuovo tipo di dati è supportato e esiste un percorso di trasformazione tra i due tipi di dati.
- Scegliere un sottoinsieme di chiavi di proprietà dei dati indicando quali chiavi di proprietà dei dati si desidera eliminare.

È anche possibile aggiungere ulteriori nodi Modifica schema al diagramma del processo in base alle esigenze, ad esempio per modificare origini dati aggiuntive o in seguito a una trasformazione Join.

Utilizzo di Modifica schema con tipo di dati decimale

Quando si utilizza la trasformazione Modifica schema con tipo di dati decimale, la trasformazione Modifica schema cambia la precisione portandola al valore predefinito di (10,2). Per modificare questa impostazione e impostare la precisione per il proprio caso d'uso, è possibile utilizzare la trasformazione SQL Query ed eseguire il cast delle colonne con una precisione specifica.

Ad esempio, se disponi di una colonna di input denominata DecimalCol "" di tipo Decimal e desideri rimapparla su una colonna di output denominata "OutputDecimalCol" con una precisione specifica di (18,6), dovresti:

1. Aggiungere una successiva trasformazione di SQL Query dopo la trasformazione Modifica schema.
2. Nella trasformazione di SQL Query, usare una query SQL per eseguire il cast della colonna rimappata con la precisione desiderata. E la query SQL risulterà simile a quanto segue:

   ```
   SELECT col1, col2, CAST(DecimalCol AS DECIMAL(18,6)) AS OutputDecimalCol
   FROM __THIS__
   ```

   Nella query SQL precedente.

   - "col1" e "col2" sono altre colonne dei dati da esaminare senza modifiche.
   - `DecimalCol` è il nome della colonna originale dei dati di input.
   - `CAST (DecimalCol AS DECIMAL (18,6)) `inserisce` `in un tipo decimale con una precisione di 18 cifre e 6 cifre decimali. DecimalCol
   - `AS `rinomina la colonna castata in` `. OutputDecimalCol OutputDecimalCol

Utilizzando la trasformazione SQL Query, è possibile sovrascrivere la precisione predefinita impostata dalla trasformazione Modifica schema e convertire in modo esplicito le colonne Decimal alla precisione desiderata. Questo approccio consente di sfruttare la trasformazione Modifica schema per rinominare e ristrutturare i dati gestendo al contempo i requisiti di precisione per le colonne Decimal attraverso la successiva trasformazione di SQL Query.

Aggiungere una trasformazione Modifica schema al processo

> **Note**
>
> La trasformazione Modifica schema non fa distinzione tra maiuscole e minuscole.

Aggiunta di un nodo di trasformazione Modifica schema al diagramma di processo

1. (Facoltativo) Apri il pannello Risorse, quindi scegli Modifica schema per aggiungere una nuova trasformazione al diagramma di processo, se necessario.
2. Nel pannello Proprietà del nodo, inserisci un nome per il nodo nel diagramma del processo. Se non è già selezionato un nodo padre, scegli un nodo dall'elenco Node parents (Nodi padre) da utilizzare come origine di input per la trasformazione.
3. Seleziona la scheda Trasforma nel pannello Proprietà del nodo.

4. Modifica lo schema di input:

   - Per rinominare una chiave di proprietà dati, inserisci il nuovo nome della chiave nel campo Target key (Chiave di destinazione).
   - Per modificare il tipo di dati per una chiave di proprietà dei dati, scegli il nuovo tipo di dati per la chiave dall'elenco Data type (Tipo di dati).
   - Per rimuovere una chiave di proprietà dati dallo schema di destinazione, scegli la casella di controllo Drop (Elimina) per quella chiave.

5. (Facoltativo) Dopo aver configurato le proprietà del nodo di trasformazione, puoi visualizzare lo schema modificato per i dati scegliendo la scheda Output schema (Schema di output) nel pannello dei dettagli del nodo. La prima volta che si sceglie questa scheda per qualsiasi nodo del processo, viene richiesto di fornire un ruolo IAM per accedere ai dati. Se non è stato specificato un ruolo IAM nella scheda Job details (Dettagli del processo), viene richiesto di immettere un ruolo IAM a questo punto.
6. (Facoltativo) Dopo aver configurato le proprietà del nodo e le proprietà di trasformazione, puoi visualizzare il set di dati modificato scegliendo la scheda Data preview (Anteprima dei dati) nel pannello dei dettagli del nodo. La prima volta che si sceglie questa scheda per qualsiasi nodo del processo, viene richiesto di fornire un ruolo IAM per accedere ai dati. Esiste un costo per l'utilizzo di questa caratteristica e la fatturazione inizia non appena si fornisce un ruolo IAM.

## Utilizzo di Elimina duplicati

La trasformazione Elimina duplicati offre due opzioni per rimuovere le righe dall'origine dati. È possibile scegliere di rimuovere le righe duplicate interamente uguali oppure selezionare alcuni campi e rimuovere le righe corrispondenti solo in base ai campi scelti.

Ad esempio, in questo set di dati sono presenti righe duplicate in cui tutti i valori in alcune righe sono esattamente gli stessi di un'altra riga mentre altri sono uguali o diversi.

| Riga | Name | Email | Età | Stato | Nota |
|---|---|---|---|---|---|
| 1 | Joy | joy@gmail | 33 | NY | |
| 2 | Tim | tim@gmail | 45 | OH | |
| 3 | Rose | rose@gmail | 23 | NJ | |

| Riga | Name | Email | Età | Stato | Nota |
|---|---|---|---|---|---|
| 4 | Tim | tim@gmail | 42 | OH | |
| 5 | Rose | rose@gmail | 23 | NJ | |
| 6 | Tim | tim@gmail | 42 | OH | Questa è una riga duplicata e corrisponde completamente in tutti i valori alla riga n. 4 |
| 7 | Rose | rose@gmail | 23 | NJ | Questa è una riga duplicata e corrisponde completamente in tutti i valori alla riga n. 5 |

Se scegli di abbinare righe intere, le righe 6 e 7 verranno rimosse dal set di dati. Il set di dati ora è:

| Riga | Name | Email | Età | Stato |
|---|---|---|---|---|
| 1 | Joy | joy@gmail | 33 | NY |
| 2 | Tim | tim@gmail | 45 | OH |
| 3 | Rose | rose@gmail | 23 | NJ |
| 4 | Tim | tim@gmail | 42 | OH |
| 5 | Rose | rose@gmail | 23 | NJ |

Se hai scelto di specificare le chiavi, puoi scegliere di rimuovere le righe che corrispondono a "nome" ed "e-mail". In questo modo puoi esercitare un maggiore controllo su che cosa si intende per "riga duplicata" per il tuo set di dati. Specificando "nome" ed "e-mail", il set di dati ora è:

| Riga | Name | Email | Età | Stato |
|---|---|---|---|---|
| 1 | Joy | joy@gmail | 33 | NY |
| 2 | Tim | tim@gmail | 45 | OH |
| 3 | Rose | rose@gmail | 23 | NJ |

Alcune cose da tenere a mente:

- Affinché le righe vengano riconosciute come duplicate, i valori fanno distinzione tra maiuscole e minuscole. Tutti i valori nelle righe devono avere la stessa successione di maiuscole e minuscole. Questo vale per entrambe le opzioni scelte (Abbina righe intere o Specifica le chiavi).
- Tutti i valori vengono letti come stringhe.
- La trasformazione Elimina duplicati utilizza il comando dropDuplicates di Spark.
- Quando si utilizza la trasformazione Elimina duplicati, la prima riga viene mantenuta e le altre righe vengono eliminate.
- La trasformazione Elimina duplicati non modifica lo schema del dataframe. Se scegli di specificare le chiavi, tutti i campi vengono conservati nel dataframe risultante.

## Utilizzo SelectFields per rimuovere la maggior parte delle chiavi di proprietà dei dati

È possibile creare un sottoinsieme di chiavi di proprietà dei dati dal set di dati utilizzando la SelectFieldstrasformazione. Puoi indicare quali chiavi di proprietà dei dati conservare e le altre vengono rimosse dal set di dati.

> **Note**
>
> La SelectFieldstrasformazione distingue tra maiuscole e minuscole. Usalo ApplyMappingse hai bisogno di un modo senza distinzione tra maiuscole e minuscole per selezionare i campi.

Per aggiungere un nodo di SelectFields trasformazione al diagramma del lavoro

1. (Facoltativo) Aprite il pannello Risorse, quindi scegliete SelectFieldsdi aggiungere una nuova trasformazione al diagramma del lavoro, se necessario.
2. Nella scheda Node properties (Proprietà del nodo), inserisci un nome per il nodo nel diagramma del processo. Se non è già selezionato un nodo padre, scegli un nodo dall'elenco Node parents (Nodi padre) da utilizzare come origine di input per la trasformazione.
3. Seleziona la scheda Transform (Trasformazione) nel pannello dei dettagli del nodo.
4. Sotto l'intestazione SelectFields, scegliete le chiavi delle proprietà dei dati nel set di dati che desiderate conservare. Tutte le chiavi di proprietà dei dati non selezionate vengono eliminate dal set di dati.

   Puoi anche selezionare la casella di controllo accanto all'intestazione di colonna Field (Campo) per scegliere automaticamente tutte le chiavi di proprietà dei dati nel set di dati. Quindi puoi deselezionare singolarmente le chiavi di proprietà dei dati per rimuoverle dal set di dati.
5. (Facoltativo) Dopo aver configurato le proprietà del nodo di trasformazione, puoi visualizzare lo schema modificato per i dati scegliendo la scheda Output schema (Schema di output) nel pannello dei dettagli del nodo. La prima volta che si sceglie questa scheda per qualsiasi nodo del processo, viene richiesto di fornire un ruolo IAM per accedere ai dati. Se non è stato specificato un ruolo IAM nella scheda Job details (Dettagli del processo), viene richiesto di immettere un ruolo IAM a questo punto.
6. (Facoltativo) Dopo aver configurato le proprietà del nodo e le proprietà di trasformazione, puoi visualizzare il set di dati modificato scegliendo la scheda Data preview (Anteprima dei dati) nel pannello dei dettagli del nodo. La prima volta che si sceglie questa scheda per qualsiasi nodo del processo, viene richiesto di fornire un ruolo IAM per accedere ai dati. Esiste un costo per l'utilizzo di questa caratteristica e la fatturazione inizia non appena si fornisce un ruolo IAM.

## Utilizzo DropFields per conservare la maggior parte delle chiavi di proprietà dei dati

È possibile creare un sottoinsieme di chiavi di proprietà dei dati dal set di dati utilizzando la DropFieldstrasformazione. Puoi indicare quali chiavi di proprietà dei dati rimuovere dal dataset e le altre vengono conservate.

> **Note**
>
> La DropFieldstrasformazione distingue tra maiuscole e minuscole. Utilizza Modifica schema se ti serve un modo per selezionare i campi che non faccia distinzione tra maiuscole e minuscole.

Per aggiungere un nodo di DropFields trasformazione al diagramma del lavoro

1. (Facoltativo) Aprite il pannello Risorse, quindi scegliete di DropFieldsaggiungere una nuova trasformazione al diagramma del lavoro, se necessario.
2. Nella scheda Node properties (Proprietà del nodo), inserisci un nome per il nodo nel diagramma del processo. Se non è già selezionato un nodo padre, scegli un nodo dall'elenco Node parents (Nodi padre) da utilizzare come origine di input per la trasformazione.
3. Seleziona la scheda Transform (Trasformazione) nel pannello dei dettagli del nodo.
4. Sotto l'intestazione DropFields, scegliete le chiavi di proprietà dei dati da eliminare dalla fonte dei dati.

   Puoi anche selezionare la casella di controllo accanto all'intestazione di colonna Field (Campo) per scegliere automaticamente tutte le chiavi di proprietà dei dati nel set di dati. Quindi puoi deselezionare singolarmente le chiavi di proprietà dei dati per mantenerle nel set di dati.
5. (Facoltativo) Dopo aver configurato le proprietà del nodo di trasformazione, puoi visualizzare lo schema modificato per i dati scegliendo la scheda Output schema (Schema di output) nel pannello dei dettagli del nodo. La prima volta che si sceglie questa scheda per qualsiasi nodo del processo, viene richiesto di fornire un ruolo IAM per accedere ai dati. Se non è stato specificato un ruolo IAM nella scheda Job details (Dettagli del processo), viene richiesto di immettere un ruolo IAM a questo punto.
6. (Facoltativo) Dopo aver configurato le proprietà del nodo e le proprietà di trasformazione, puoi visualizzare il set di dati modificato scegliendo la scheda Data preview (Anteprima dei dati) nel pannello dei dettagli del nodo. La prima volta che si sceglie questa scheda per qualsiasi nodo del processo, viene richiesto di fornire un ruolo IAM per accedere ai dati. Esiste un costo per l'utilizzo di questa caratteristica e la fatturazione inizia non appena si fornisce un ruolo IAM.

## Rinominare un campo nel set di dati

È possibile utilizzare la RenameFieldtrasformazione per modificare il nome di una singola chiave di proprietà nel set di dati.

> **Note**
>
> La RenameFieldtrasformazione distingue tra maiuscole e minuscole. Usalo ApplyMappingse hai bisogno di una trasformazione senza distinzione tra maiuscole e minuscole.

> **Tip**
>
> Utilizzando la trasformazione Modifica schema è possibile rinominare più chiavi di proprietà dei dati nel set di dati con una singola trasformazione.

Per aggiungere un nodo di RenameField trasformazione al diagramma del lavoro

1. (Facoltativo) Aprite il pannello Risorse, quindi scegliete di RenameFieldaggiungere una nuova trasformazione al diagramma del lavoro, se necessario.
2. Nella scheda Node properties (Proprietà del nodo), inserisci un nome per il nodo nel diagramma del processo. Se non è già selezionato un nodo padre, scegli un nodo dall'elenco Node parents (Nodi padre) da utilizzare come origine di input per la trasformazione.
3. Seleziona la scheda Transform (Trasformazione).
4. Sotto l'intestazione Data field (Campo dati), scegli una chiave di proprietà dai dati di origine, quindi inserisci un nuovo nome nel campo New field name (Nuovo nome campo).
5. (Facoltativo) Dopo aver configurato le proprietà del nodo di trasformazione, puoi visualizzare lo schema modificato per i dati scegliendo la scheda Output schema (Schema di output) nel pannello dei dettagli del nodo. La prima volta che si sceglie questa scheda per qualsiasi nodo del processo, viene richiesto di fornire un ruolo IAM per accedere ai dati. Se non è stato specificato un ruolo IAM nella scheda Job details (Dettagli del processo), viene richiesto di immettere un ruolo IAM a questo punto.
6. (Facoltativo) Dopo aver configurato le proprietà del nodo e le proprietà di trasformazione, puoi visualizzare il set di dati modificato scegliendo la scheda Data preview (Anteprima dei dati) nel pannello dei dettagli del nodo. La prima volta che si sceglie questa scheda per qualsiasi nodo del

processo, viene richiesto di fornire un ruolo IAM per accedere ai dati. Esiste un costo per l'utilizzo di questa caratteristica e la fatturazione inizia non appena si fornisce un ruolo IAM.

## Utilizzo di Spigot per campionare il set di dati

Per testare le trasformazioni eseguite dal processo, è possibile ottenere un campione dei dati, allo scopo di verificare che la trasformazione funzioni come previsto. La trasformazione Spigot scrive un sottoinsieme di regitri dal set di dati in un file JSON in un bucket Amazon S3. Il metodo di campionamento dei dati può essere un numero specifico di registri dall'inizio del file o un fattore di probabilità utilizzato per selezionare i registri.

Per aggiungere un nodo di trasformazione Spigot al diagramma di processo

1. (Facoltativo) Apri il pannello Risorse, quindi scegli Spigot per aggiungere una nuova trasformazione al diagramma di processo, se necessario.
2. Nella scheda Node properties (Proprietà del nodo), inserisci un nome per il nodo nel diagramma del processo. Se non è già selezionato un nodo padre, scegli un nodo dall'elenco Node parents (Nodi padre) da utilizzare come origine di input per la trasformazione.
3. Seleziona la scheda Transform (Trasformazione) nel pannello dei dettagli del nodo.
4. Inserisci un percorso Amazon S3 o scegli Browse S3 (Sfoglia S3) per scegliere una posizione in Amazon S3. Questa è la posizione in cui il processo scrive il file JSON che contiene l'esempio di dati.
5. Inserisci le informazioni per il metodo di campionamento. Puoi specificare un valore per Number of records (Numero di registri) da scrivere a partire dall'inizio del set di dati e una Probability threshold (Soglia di probabilità) (inserita sotto forma di valore decimale con un valore massimo di 1) di scelta di un dato registro.

   Ad esempio, per scrivere i primi 50 registri dal set di dati, è necessario impostare Number of records (Numero di registri) su 50 e Probability threshold (Soglia di probabilità) su 1 (100%).

## Unione di set di dati

La trasformazione Join consente di combinare due set di dati in uno. È possibile specificare i nomi delle chiavi nello schema di ogni set di dati da confrontare. L'output `DynamicFrame` contiene righe in cui le chiavi soddisfano la condizione di join. Le righe in ogni set di dati che soddisfano la condizione di join vengono combinate in una singola riga nell'output `DynamicFrame`, che contiene tutte le colonne trovate in entrambi i set di dati.

Per aggiungere un nodo di trasformazione Join al diagramma di processo

1. Se è disponibile una sola origine dati, è necessario aggiungere un nuovo nodo di origine dati al diagramma di processo.
2. Scegli uno dei nodi di origine per il join. Apri il pannello Risorse, quindi scegli Join per aggiungere una nuova trasformazione al diagramma del processo.
3. Nella scheda Node properties (Proprietà del nodo), inserisci un nome per il nodo nel diagramma del processo.
4. Nella scheda Node properties (Proprietà del nodo), sotto l'intestazione Node parents (Nodi padre), aggiungi un nodo padre in modo che ci siano due set di dati che forniscono input per il join. Il padre può essere un nodo di origine dati o un nodo di trasformazione.

   > **Note**
   >
   > Un join può avere solo due nodi padre.

5. Seleziona la scheda Transform (Trasformazione).

   Se viene visualizzato un messaggio che indica che esistono nomi di chiavi in conflitto, è possibile:

   - Scegli Risolvi per aggiungere automaticamente un nodo di ApplyMappingtrasformazione al diagramma del lavoro. Il ApplyMapping nodo aggiunge un prefisso a tutte le chiavi del set di dati che hanno lo stesso nome di una chiave nell'altro set di dati. Ad esempio, se utilizzi il valore predefinito di **`right`**, tutte le chiavi nel set di dati destro che hanno lo stesso nome di una chiave nel set di dati sinistro verranno rinominate in `(right)`*`key name`*.
   - Aggiungere manualmente un nodo di trasformazione in precedenza nel diagramma del processo per rimuovere o rinominare le chiavi in conflitto.
6. Scegli il tipo di join nell'elenco Join type (Tipo di join).

   - Inner join: restituisce una riga con colonne di entrambi i set di dati per ogni corrispondenza in base alla condizione di join. Le righe che non soddisfano la condizione di join non vengono restituite.
   - Left join: tutte le righe del set di dati sinistro e solo le righe del set di dati destro che soddisfano la condizione di join.
   - Right join: tutte le righe del set di dati destro e solo le righe del set di dati sinistro che soddisfano la condizione di join.

- Outer join: tutte le righe di entrambi i set di dati.
- Left semi join: tutte le righe del set di dati sinistro che hanno una corrispondenza nel set di dati destro in base alla condizione di join.
- Right semi join: tutte le righe del set di dati sinistro che non hanno una corrispondenza nel set di dati destro in base alla condizione di join.

7. Nella scheda Transform (Trasformazione), sotto l'intestazione Join conditions (Condizioni di join), scegli Add condition (Aggiungi condizione). Scegli una chiave di proprietà da ciascun set di dati da confrontare. Le chiavi di proprietà sul lato sinistro dell'operatore di confronto vengono definite come set di dati sinistro e le chiavi di proprietà a destra vengono definite come set di dati destro.

   Per condizioni di join più complesse, è possibile aggiungere ulteriori chiavi di corrispondenza scegliendo Add condition (Aggiungi condizione) più di una volta. Se si aggiunge accidentalmente una condizione, è possibile selezionare l'icona di eliminazione (🗑) per rimuoverla.

8. (Facoltativo) Dopo aver configurato le proprietà del nodo di trasformazione, puoi visualizzare lo schema modificato per i dati scegliendo la scheda Output schema (Schema di output) nel pannello dei dettagli del nodo. La prima volta che si sceglie questa scheda per qualsiasi nodo del processo, viene richiesto di fornire un ruolo IAM per accedere ai dati. Se non è stato specificato un ruolo IAM nella scheda Job details (Dettagli del processo), viene richiesto di immettere un ruolo IAM a questo punto.
9. (Facoltativo) Dopo aver configurato le proprietà del nodo e le proprietà di trasformazione, puoi visualizzare il set di dati modificato scegliendo la scheda Data preview (Anteprima dei dati) nel pannello dei dettagli del nodo. La prima volta che si sceglie questa scheda per qualsiasi nodo del processo, viene richiesto di fornire un ruolo IAM per accedere ai dati. Esiste un costo per l'utilizzo di questa funzionalità e la fatturazione inizia non appena si fornisce un ruolo IAM.

Per un esempio di schema di output del join, considera un join tra due set di dati con le seguenti chiavi di proprietà:

```
Left: {id, dept, hire_date, salary, employment_status}
Right: {id, first_name, last_name, hire_date, title}
```

Il join è configurato in modo che corrisponda alle chiavi `id` e `hire_date` utilizzando l'operatore di confronto =.

Perché entrambi i set di dati contengono le chiavi `id` e `hire_date`, scegli Resolve it (Risolvi) per aggiungere automaticamente il prefisso **`right`** alle chiavi nel set di dati giusto.

Le chiavi nello schema di output sarebbero:

```
{id, dept, hire_date, salary, employment_status,
(right)id, first_name, last_name, (right)hire_date, title}
```

## Utilizzo di Union per combinare le righe

Il nodo di trasformazione Union si utilizza quando si desidera combinare righe provenienti da più di un'origine dati aventi il medesimo schema.

Esistono due tipi di trasformazioni Union:

1. ALL: quando si applica ALL, l'unione risultante non rimuove le righe duplicate.
2. DISTINCT: quando si applica DISTINCT, l'unione risultante rimuove le righe duplicate.

Union e Join: differenze

Si utilizza Union per combinare le righe. Si utilizza Join per combinare le colonne.

Utilizzo della trasformazione Union nel canvas di ETL visivo

1. Aggiungi più di un'origine dati per eseguire una trasformazione Union. Per aggiungere un'origine dati, apri il pannello Risorse, quindi scegli l'origine dati dalla scheda Origini. Prima di utilizzare la trasformazione Union, devi assicurarti che tutte le origini dati coinvolte nell'unione abbiano lo stesso schema e la stessa struttura.
2. Quando hai almeno due origini dati che desideri combinare utilizzando la trasformazione Union, crea la trasformazione Union aggiungendola al canvas. Apri il pannello Risorse sul canvas e cerca "Union". In alternativa, scegli la scheda Trasformazioni nel pannello Risorse, scorri verso il basso fino a trovare la trasformazione Union, quindi scegli Union.
3. Seleziona il nodo Union nel canvas del processo. Nella finestra Proprietà del nodo, scegli i nodi padri da connettere alla trasformazione Union.
4. AWS Glue verifica la compatibilità per assicurarsi che la trasformazione Union possa essere applicata a tutte le origini dati. Se lo schema delle origini dati è lo stesso, l'operazione sarà consentita. Se le origini dati non hanno lo stesso schema, viene visualizzato un messaggio

di errore: "The input schemas of this union are not the same. Prendi in considerazione l'idea ApplyMapping di utilizzarlo per abbinare gli schemi.» Per risolvere questo problema, scegli Usa ApplyMapping.

5. Scegli il tipo di Union.

    1. All: per impostazione predefinita, è selezionato il tipo All Union; ciò comporterà la duplicazione delle righe, se presenti nella combinazione di dati.
    2. Distinct: scegli Distinct se desideri che le righe duplicate vengano rimosse dalla combinazione di dati risultante.

## Utilizzo SplitFields per dividere un set di dati in due

La SplitFieldstrasformazione consente di scegliere alcune delle chiavi di proprietà dei dati nel set di dati di input e inserirle in un set di dati e le chiavi non selezionate in un set di dati separato. L'output di questa trasformazione è una raccolta di `DynamicFrames`.

> **Note**
>
> È necessario utilizzare una SelectFromCollectiontrasformazione per convertire la raccolta di `DynamicFrames` in una singola `DynamicFrame` prima di poter inviare l'output a una posizione di destinazione.

La SplitFieldstrasformazione distingue tra maiuscole e minuscole. Aggiungi una ApplyMappingtrasformazione come nodo principale se hai bisogno di nomi di chiavi di proprietà senza distinzione tra maiuscole e minuscole.

Per aggiungere un nodo di SplitFields trasformazione al diagramma del lavoro

1. (Facoltativo) Aprite il pannello Risorse, quindi scegliete di SplitFieldsaggiungere una nuova trasformazione al diagramma del lavoro, se necessario.
2. Nella scheda Node properties (Proprietà del nodo), inserisci un nome per il nodo nel diagramma del processo. Se non è già selezionato un nodo padre, scegli un nodo dall'elenco Node parents (Nodi padre) da utilizzare come origine di input per la trasformazione.
3. Seleziona la scheda Transform (Trasformazione).
4. Scegli le chiavi di proprietà da inserire nel primo set di dati. Le chiavi non scelte vengono inserite nel secondo set di dati.

5. (Facoltativo) Dopo aver configurato le proprietà del nodo di trasformazione, puoi visualizzare lo schema modificato per i dati scegliendo la scheda Output schema (Schema di output) nel pannello dei dettagli del nodo. La prima volta che si sceglie questa scheda per qualsiasi nodo del processo, viene richiesto di fornire un ruolo IAM per accedere ai dati. Se non è stato specificato un ruolo IAM nella scheda Job details (Dettagli del processo), viene richiesto di immettere un ruolo IAM a questo punto.
6. (Facoltativo) Dopo aver configurato le proprietà del nodo e le proprietà di trasformazione, puoi visualizzare il set di dati modificato scegliendo la scheda Data preview (Anteprima dei dati) nel pannello dei dettagli del nodo. La prima volta che si sceglie questa scheda per qualsiasi nodo del processo, viene richiesto di fornire un ruolo IAM per accedere ai dati. Esiste un costo per l'utilizzo di questa caratteristica e la fatturazione inizia non appena si fornisce un ruolo IAM.
7. Configura un nodo di SelectFromCollectiontrasformazione per elaborare i set di dati risultanti.

## Panoramica della trasformazione SelectFromCollection

Alcune trasformazioni hanno come output più set di dati anziché un singolo set di dati, ad esempio. SplitFields La SelectFromCollectiontrasformazione seleziona un set di dati (`DynamicFrame`) da una raccolta di set di dati (una matrice di). `DynamicFrames` L'output per la trasformazione è il `DynamicFrame` selezionato.

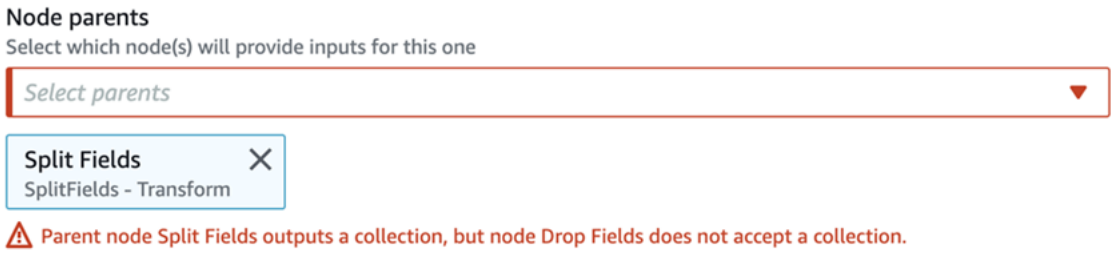
È necessario utilizzare questa trasformazione dopo aver utilizzato una trasformazione che crea una raccolta di `DynamicFrames`, come ad esempio:

- Trasformazioni di codice personalizzate
- SplitFields

Se non aggiungete un nodo di SelectFromCollectiontrasformazione al diagramma del lavoro dopo una di queste trasformazioni, riceverete un errore relativo al vostro lavoro.

Il nodo padre per questa trasformazione deve essere un nodo che restituisce una raccolta di `DynamicFrames`. Se per questo nodo di trasformazione si sceglie un padre che restituisce un singolo `DynamicFrame`, ad esempio Join, il processo restituisce un errore.

Analogamente, se si utilizza un SelectFromCollectionnodo nel diagramma del processo come elemento principale per una trasformazione che prevede un valore singolo `DynamicFrame` come input, il processo restituisce un errore.




## Utilizzo SelectFromCollection per scegliere quale set di dati conservare

Usa la SelectFromCollectiontrasformazione per convertire una raccolta di `DynamicFrames` in una singola`DynamicFrame`.

Per aggiungere un nodo di SelectFromCollection trasformazione al diagramma del lavoro

1. (Facoltativo) Aprite il pannello Risorse, quindi scegliete di SelectFromCollectionaggiungere una nuova trasformazione al diagramma del lavoro, se necessario.
2. Nella scheda Node properties (Proprietà del nodo), inserisci un nome per il nodo nel diagramma del processo. Se non è già selezionato un nodo padre, scegli un nodo dall'elenco Node parents (Nodi padre) da utilizzare come origine di input per la trasformazione.
3. Seleziona la scheda Transform (Trasformazione).
4. Sotto l'intestazione Frame index (Indice del frame), scegli il numero di indice della matrice che corrisponde al `DynamicFrame` da selezionare dalla raccolta di `DynamicFrames`.

   Ad esempio, se il nodo principale di questa trasformazione è una SplitFieldstrasformazione, nella scheda Schema di output di quel nodo è possibile visualizzare lo schema per ciascuno `DynamicFrame` di essi. Per mantenere il `DynamicFrame` associato allo schema per Output 2, devi selezionare **1** per il valore di Frame index (Indice di frame), che è il secondo valore nell'elenco.

   Solo il `DynamicFrame` scelto è incluso nell'output.
5. (Facoltativo) Dopo aver configurato le proprietà del nodo di trasformazione, puoi visualizzare lo schema modificato per i dati scegliendo la scheda Output schema (Schema di output) nel pannello dei dettagli del nodo. La prima volta che si sceglie questa scheda per qualsiasi nodo del processo, viene richiesto di fornire un ruolo IAM per accedere ai dati. Se non è stato specificato un ruolo IAM nella scheda Job details (Dettagli del processo), viene richiesto di immettere un ruolo IAM a questo punto.

6. (Facoltativo) Dopo aver configurato le proprietà del nodo e le proprietà di trasformazione, puoi visualizzare il set di dati modificato scegliendo la scheda Data preview (Anteprima dei dati) nel pannello dei dettagli del nodo. La prima volta che si sceglie questa scheda per qualsiasi nodo del processo, viene richiesto di fornire un ruolo IAM per accedere ai dati. Esiste un costo per l'utilizzo di questa caratteristica e la fatturazione inizia non appena si fornisce un ruolo IAM.

## Trovare e riempi i valori mancanti in un set di dati

È possibile utilizzare la FillMissingValuestrasformazione per individuare i record nel set di dati con valori mancanti e aggiungere un nuovo campo con un valore determinato dall'imputazione. Il set di dati di input viene utilizzato per addestrare il modello di Machine Learning (ML) che determina quale dovrebbe essere il valore mancante. Se si utilizzano set di dati incrementali, ogni set incrementale viene utilizzato come dati di addestramento per il modello ML, pertanto i risultati potrebbero non essere molto accurati.

Per utilizzare un nodo di FillMissingValues trasformazione nel diagramma del lavoro

1. (Facoltativo) Aprite il pannello Risorse, quindi scegliete di FillMissingValuesaggiungere una nuova trasformazione al diagramma del lavoro, se necessario.
2. Nella scheda Node properties (Proprietà del nodo), inserisci un nome per il nodo nel diagramma del processo. Se non è già selezionato un nodo padre, scegli un nodo dall'elenco Node parents (Nodi padre) da utilizzare come origine di input per la trasformazione.
3. Seleziona la scheda Transform (Trasformazione).
4. Per Data field (Campo dati), scegli il nome della colonna o del campo dai dati di origine da analizzare per i valori mancanti.
5. (Facoltativo) Nel campo New field name (Nuovo nome campo), inserisci un nome per il campo aggiunto a ciascun registro che conterrà il valore di sostituzione stimato per il campo analizzato. Se nel campo analizzato non ci sono valori mancanti, il valore nel campo analizzato viene copiato nel nuovo campo.

   Se non specifichi un nome per il nuovo campo, il nome predefinito è il nome della colonna analizzata con aggiunta di `_filled`. Ad esempio, se inserisci **`Age`** per Data field (Campo dati) senza specificare un valore per New field name (Nuovo nome campo), a ogni registro viene aggiunto un nuovo campo denominato **`Age_filled`**.
6. (Facoltativo) Dopo aver configurato le proprietà del nodo di trasformazione, puoi visualizzare lo schema modificato per i dati scegliendo la scheda Output schema (Schema di output) nel

pannello dei dettagli del nodo. La prima volta che si sceglie questa scheda per qualsiasi nodo del processo, viene richiesto di fornire un ruolo IAM per accedere ai dati. Se non è stato specificato un ruolo IAM nella scheda Job details (Dettagli del processo), viene richiesto di immettere un ruolo IAM a questo punto.

7. (Facoltativo) Dopo aver configurato le proprietà del nodo e le proprietà di trasformazione, puoi visualizzare il set di dati modificato scegliendo la scheda Data preview (Anteprima dei dati) nel pannello dei dettagli del nodo. La prima volta che si sceglie questa scheda per qualsiasi nodo del processo, viene richiesto di fornire un ruolo IAM per accedere ai dati. Esiste un costo per l'utilizzo di questa caratteristica e la fatturazione inizia non appena si fornisce un ruolo IAM.

## Filtro delle chiavi all'interno di un set di dati

Utilizzo della trasformazione Filter per creare un nuovo set di dati filtrando i registri dal set di dati di input in base a un'espressione regolare. Le righe che non soddisfano la condizione di filtro vengono rimosse dall'output.

- Per i tipi di dati stringa, è possibile filtrare le righe in cui il valore della chiave corrisponde a una stringa specificata.
- Per i tipi di dati numerici, è possibile filtrare le righe confrontando il valore della chiave con un valore specificato utilizzando gli operatori di confronto <, >, =, !=, <= e >=.

Se si specificano più condizioni di filtro, i risultati vengono combinati utilizzando `AND` per impostazione predefinita, ma è possibile anche scegliere `OR`.

La trasformazione Filter fa distinzione tra maiuscole e minuscole. Aggiungi una ApplyMappingtrasformazione come nodo principale se hai bisogno di nomi di chiavi di proprietà senza distinzione tra maiuscole e minuscole.

Per aggiungere un nodo di trasformazione Filter al diagramma di processo

1. (Facoltativo) Apri il pannello Risorse, quindi scegli Filtra per aggiungere una nuova trasformazione al diagramma di processo, se necessario.
2. Nella scheda Node properties (Proprietà del nodo), inserisci un nome per il nodo nel diagramma del processo. Se non è già selezionato un nodo padre, scegli un nodo dall'elenco Node parents (Nodi padre) da utilizzare come origine di input per la trasformazione.
3. Seleziona la scheda Transform (Trasformazione).

4. Scegli Globale AND o Global OR. Questo determina il modo in cui vengono combinate più condizioni di filtro. Tutte le condizioni sono combinate usando le operazioni `AND` o `OR`. Se hai una condizione di filtro singolo, puoi sceglierne una delle due.
5. Seleziona il pulsante Add condition (Aggiungi condizione) nella sezione Filter condition (Condizione di filtro) per aggiungere una condizione di filtro.

   Nel campo Key (Chiave), scegli il nome di una chiave di proprietà dal set di dati. Nel campo Operation (Operazione), seleziona l'operatore di confronto. Nel campo Value (Valore), inserisci il valore di confronto. Di seguito sono riportate alcuni esempi di condizioni di filtro.

   - `year >= 2018`
   - `State matches 'CA*'`

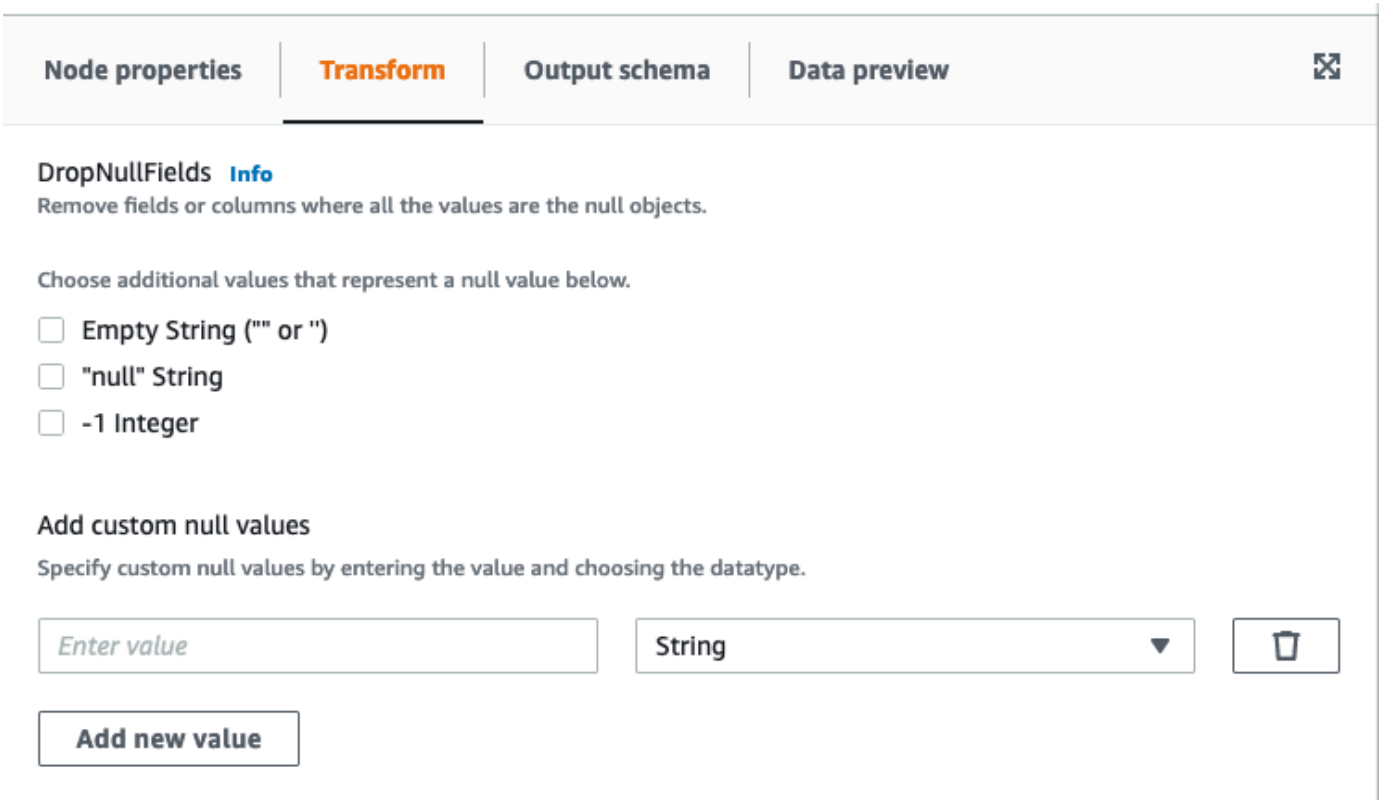
   Quando si filtrano i valori di stringa, è necessario assicurarsi che il valore di confronto utilizzi un formato di espressione regolare che corrisponda al linguaggio di script selezionato nelle proprietà del processo (Python o Scala).
6. Aggiungi ulteriori condizioni di filtro, se necessario.
7. (Facoltativo) Dopo aver configurato le proprietà del nodo di trasformazione, puoi visualizzare lo schema modificato per i dati scegliendo la scheda Output schema (Schema di output) nel pannello dei dettagli del nodo. La prima volta che si sceglie questa scheda per qualsiasi nodo del processo, viene richiesto di fornire un ruolo IAM per accedere ai dati. Se non è stato specificato un ruolo IAM nella scheda Job details (Dettagli del processo), viene richiesto di immettere un ruolo IAM a questo punto.
8. (Facoltativo) Dopo aver configurato le proprietà del nodo e le proprietà di trasformazione, puoi visualizzare il set di dati modificato scegliendo la scheda Data preview (Anteprima dei dati) nel pannello dei dettagli del nodo. La prima volta che si sceglie questa scheda per qualsiasi nodo del processo, viene richiesto di fornire un ruolo IAM per accedere ai dati. Esiste un costo per l'utilizzo di questa caratteristica e la fatturazione inizia non appena si fornisce un ruolo IAM.

## Utilizzo DropNullFields per rimuovere campi con valori nulli

Usa la DropNullFieldstrasformazione per rimuovere i campi dal set di dati se tutti i valori nel campo sono «nulli». Per impostazione predefinita, AWS Glue Studio riconoscerà gli oggetti nulli, ma alcuni valori come stringhe vuote, stringhe "null", -1 interi o altri segnaposto come zeri, non verranno riconosciuti automaticamente come null.

Per usare il DropNullFields

1. Aggiungere un DropNullFields nodo al diagramma del lavoro.
2. Nella scheda Node properties (Proprietà del nodo), scegli valori aggiuntivi che rappresentano un valore nullo. È possibile scegliere di selezionare nessuno o tutti i valori:




   - Stringa vuota ("" or ''): i campi che contengono stringhe vuote verranno eliminati
   - "stringa null": i campi che contengono la stringa con la parola "null" verranno eliminati
   - -1 numero intero: i campi che contengono un numero intero -1 (negativo) verranno eliminati
3. Se necessario, è anche possibile specificare valori nulli personalizzati. Si tratta di valori nulli che potrebbero essere univoci per il set di dati. Per aggiungere un valore nullo personalizzato, scegli Add new value (Aggiungi nuovo valore).
4. Inserisci il valore nullo personalizzato. Ad esempio, questo può essere zero o qualsiasi valore utilizzato per rappresentare un valore nullo nel set di dati.
5. Scegli il tipo di dati nel campo a discesa. I tipi di dati possono essere String o Integer.

> **Note**
>
> I valori nulli personalizzati e i relativi tipi di dati devono corrispondere esattamente, affinché i campi vengano riconosciuti come valori nulli e vengano quindi eliminati. Corrispondenze parziali in cui solo il valore nullo personalizzato corrisponde, ma il tipo di dati non comporta l'eliminazione dei campi.

## Utilizzo di una query SQL per trasformare i dati

Puoi utilizzare una trasformazione SQL per scrivere la tua trasformazione sotto forma di query SQL.

Un nodo di trasformazione SQL può avere più set di dati come input, ma produce solo un singolo set di dati come output. Contiene un campo di testo, in cui puoi inserire la query Apache SparkSQL. Puoi assegnare alias a ciascun set di dati utilizzato come input, in modo da semplificare la query SQL. Per ulteriori informazioni sulla sintassi SQL, consulta la documentazione di Spark SQL.

> **Note**
>
> Se utilizzi una trasformazione SQL Spark con un'origine dati situata in un VPC, aggiungi un endpoint VPC AWS Glue al VPC che contiene l'origine dati. Per ulteriori informazioni sulla configurazione degli endpoint di sviluppo, consulta Aggiunta di un endpoint di sviluppo, Impostazione dell'ambiente per endpoint di sviluppo e Accesso all'endpoint di sviluppo nella Guida per gli sviluppatori di AWS Glue .

Per aggiungere un nodo di trasformazione SQL al diagramma di processo

1. (Facoltativo) Aggiungi un nodo di trasformazione al diagramma di processo, se necessario. Scegliere SQL Query per il tipo di nodo.

> **Note**
>
> Se utilizzi una sessione di anteprima dei dati e un nodo SQL o di codice personalizzato personalizzato, la sessione di anteprima dei dati eseguirà l'SQL o il blocco di codice così com'è per l'intero set di dati.

2. Nella scheda Node properties (Proprietà del nodo), inserisci un nome per il nodo nel diagramma del processo. Se non è già selezionato un nodo padre, o se desideri più input per la trasformazione SQL, scegli un nodo dall'elenco Node parents (Nodi padre) da utilizzare come origine di input per la trasformazione. Aggiungi nodi padre aggiuntivi in base alle esigenze.
3. Seleziona la scheda Transform (Trasformazione) nel pannello dei dettagli del nodo.
4. I set di dati di origine per la query SQL sono identificati dai nomi specificati nel campo Name (Nome) per ogni nodo. Se non vuoi utilizzare questi nomi o se i nomi non sono adatti per una query SQL, puoi associare un nome a ciascun set di dati. La console fornisce alias predefiniti, ad esempio `MyDataSource`.

   Ad esempio, se un nodo padre per il nodo di trasformazione SQL è denominato `Rename Org PK field`, è possibile associare il nome `org_table` a questo set di dati. Questo alias può quindi essere utilizzato nella query SQL al posto del nome del nodo.
5. Nel campo di immissione testo sotto l'intestazione Code block (Blocco di codice), incolla o immetti la query SQL. Il campo di testo mostra la sintassi SQL evidenziata e i suggerimenti per le parole chiave.
6. Con il nodo di trasformazione SQL selezionato, scegli l'opzione Output schema (Schema di output), quindi scegli Edit (Modifica). Specifica le colonne e i tipi di dati che descrivono i campi di output della query SQL.

   Specifica lo schema utilizzando le azioni seguenti nella sezione Output schema (Schema di output) della pagina:

   - Per rinominare una colonna, posiziona il cursore nella casella di testo Key (Chiave) per la colonna (nota anche come field (campo) o property key (chiave di proprietà) e inserisci il nuovo nome.
   - Per modificare il tipo di dati per una colonna, seleziona il nuovo tipo di dati per la colonna dall'elenco a discesa.
   - Per aggiungere una nuova colonna di livello superiore allo schema, scegli l'opzione Overflow ( ⋯ ), quindi scegli Add root key (Aggiungi chiave root). Vengono aggiunte nuove colonne nella parte superiore dello schema.
   - Per rimuovere una colonna dallo schema, scegli l'icona di eliminazione ( 🗑 ) all'estrema destra del nome della chiave.

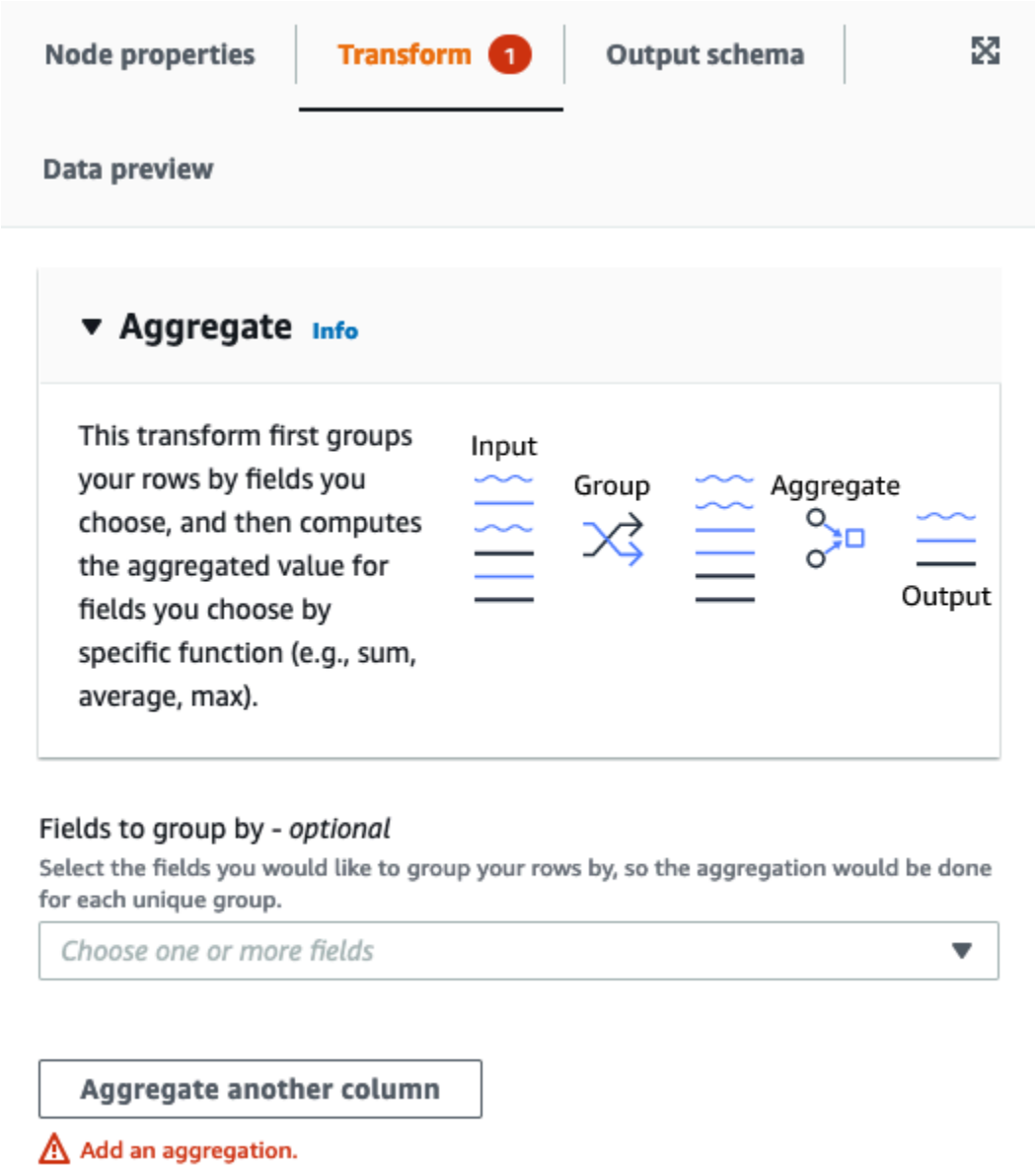
7. Una volta terminato di specificare lo schema di output, scegli Apply (Applica) per salvare le modifiche e uscire dall'editor dello schema. Se non vuoi salvare le modifiche, scegli Cancel (Annulla) per modificare l'editor dello schema.
8. (Facoltativo) Dopo aver configurato le proprietà del nodo e le proprietà di trasformazione, puoi visualizzare il set di dati modificato scegliendo la scheda Data preview (Anteprima dei dati) nel pannello dei dettagli del nodo. La prima volta che si sceglie questa scheda per qualsiasi nodo del processo, viene richiesto di fornire un ruolo IAM per accedere ai dati. Esiste un costo per l'utilizzo di questa caratteristica e la fatturazione inizia non appena si fornisce un ruolo IAM.

## Utilizzo di Aggregate per eseguire calcoli di riepilogo sui campi selezionati

Utilizzo della trasformazione Aggregate

1. Aggiungi il nodo Aggregate al diagramma del processo.
2. Nella scheda Node properties (Proprietà del nodo), scegli i campi da raggruppare selezionando il campo a discesa (facoltativo). È possibile selezionare più di un campo alla volta o cercare il nome del campo digitando nella barra di ricerca.

   Quando i campi sono selezionati, vengono visualizzati il nome e il tipo di dati. Per eliminare un campo, selezionare "X" sul campo.




3. Scegli Aggregate another column (Aggrega un'altra colonna). È necessario selezionare almeno un campo.

4. Scegli un campo nel menu a discesa Field to aggregate (Campo da aggregare).

5. Scegli la funzione di aggregazione da applicare al campo scelto:

   - avg. calcola la media
   - countDistinct: calcola il numero di valori univoci non nulli
   - count: calcola il numero di valori non null
   - first: restituisce il primo valore che soddisfa i criteri "raggruppa per"
   - last: restituisce l'ultimo valore che soddisfa i criteri "raggruppa per"
   - kurtosis: calcola la nitidezza del picco di una curva di distribuzione della frequenza
   - max: restituisce il valore più alto che soddisfa i criteri "raggruppa per"
   - min: restituisce il valore più basso che soddisfa i criteri "raggruppa per"
   - skewness: misura l'asimmetria della distribuzione di probabilità di una distribuzione normale
   - stddev_pop: calcola la deviazione standard della popolazione e restituisce la radice quadrata della varianza di popolazione
   - sum: la somma di tutti i valori del gruppo
   - SumDistinct: la somma di valori distinti nel gruppo
   - var_samp: la varianza campione del gruppo (ignora i valori nulli)
   - var_pop: la varianza di popolazione del gruppo (ignora i valori nulli)

## Appiattimento di strutture annidate

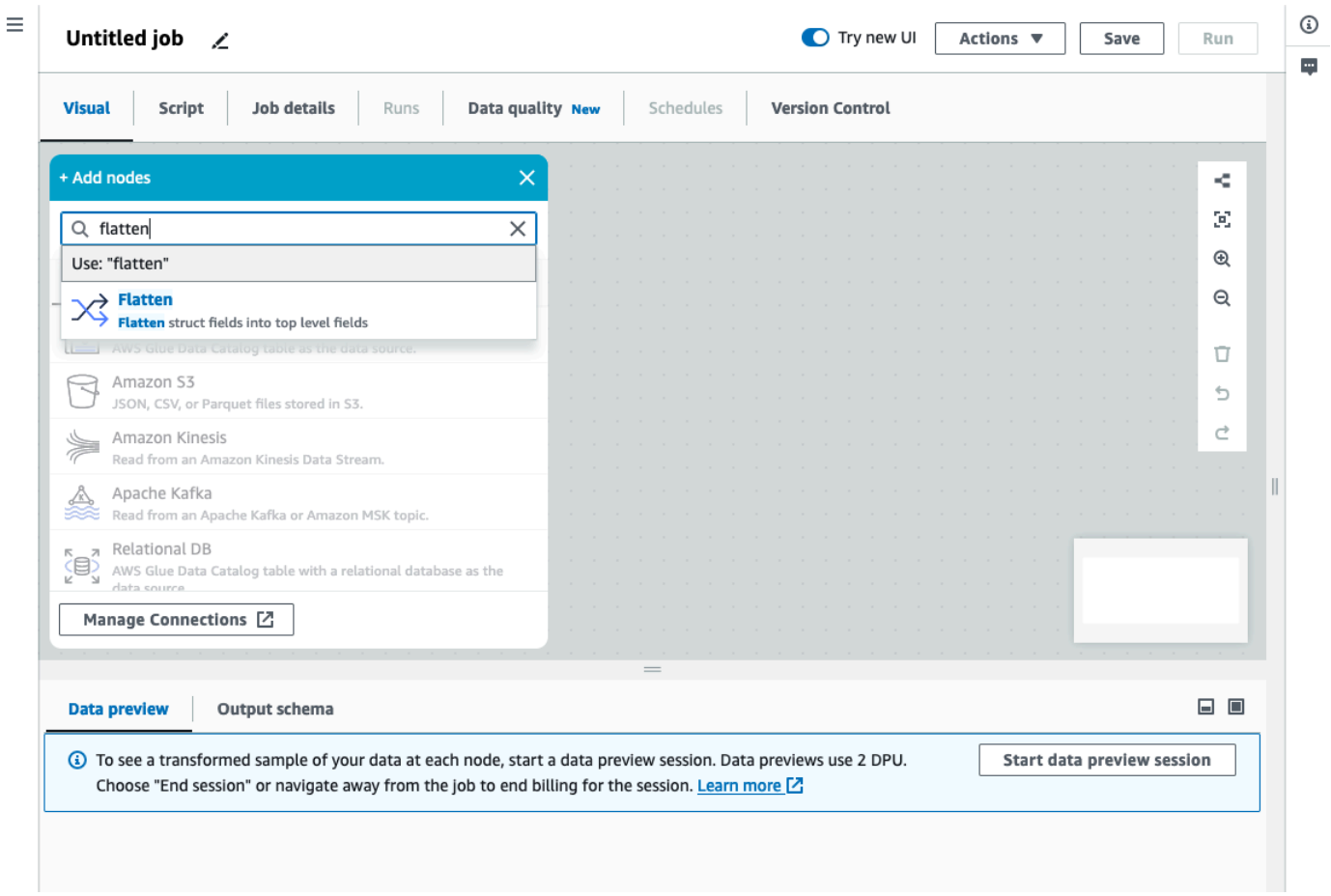
Appiattisci i campi delle strutture annidate nei dati, in modo che diventino campi di primo livello. I nuovi campi vengono denominati utilizzando il nome del campo preceduto dai nomi dei campi della struttura per raggiungerlo, separati da punti.

Ad esempio, prendiamo un caso in cui i dati hanno un campo di tipo Struct denominato "phone_numbers", che tra gli altri campi ne ha uno di tipo Struct denominato "home_phone" con due campi: "country_code" e "number". Una volta appiattiti, questi due campi diventeranno campi di primo livello denominati rispettivamente "phone_numbers.home_phone.country_code" e "phone_numbers.home_phone.number".

Aggiunta di un nodo di trasformazione Appiattisci nel diagramma di processo

1. Apri il pannello Risorse, scegli la scheda Trasformazioni, quindi Appiattisci per aggiungere una nuova trasformazione al diagramma di processo. È inoltre possibile digitare "Appiattisci" nella

barra di ricerca e poi fare clic sul nodo Appiattisci. Il nodo selezionato al momento dell'aggiunta del nodo ne sarà il nodo padre.




2. (Facoltativo) Nella scheda Proprietà del nodo, è possibile inserire un nome per il nodo nel diagramma del processo. Se non è già selezionato un nodo padre, scegli un nodo dall'elenco Node parents (Nodi padre) da utilizzare come origine di input per la trasformazione.
3. (Facoltativo) Nella scheda Trasforma, puoi limitare l'appiattimento del livello massimo di annidamento. Ad esempio, se si imposta tale valore su 1 significa che solo le strutture di primo livello verranno appiattite. Impostando il valore massimo su 2 verrà appiattito il primo livello e le strutture direttamente sottostanti.

## Aggiunta di una colonna UUID

Quando aggiungi una colonna UUID (Universally Unique Identified), a ogni riga verrà assegnata una stringa univoca di 36 caratteri.

Aggiunta di un nodo di trasformazione UUID al diagramma di processo

1. Apri il pannello Risorse, quindi scegli UUID per aggiungere una nuova trasformazione al diagramma di processo. Il nodo selezionato al momento dell'aggiunta del nodo ne sarà il nodo padre.
2. (Facoltativo) Nella scheda Proprietà del nodo, è possibile inserire un nome per il nodo nel diagramma del processo. Se non è già selezionato un nodo padre, scegli un nodo dall'elenco Node parents (Nodi padre) da utilizzare come origine di input per la trasformazione.
3. (Facoltativo) Nella scheda Trasforma, puoi personalizzare il nome della nuova colonna. Per impostazione predefinita, si chiamerà "uuid".

## Aggiunta di una colonna identificativa

Assegna un Identificatore numerico per ogni riga del set di dati.

Aggiunta di un nodo di trasformazione Identificatore nel diagramma di processo

1. Apri il pannello Risorse, quindi scegli Identificatore per aggiungere una nuova trasformazione al diagramma di processo. Il nodo selezionato al momento dell'aggiunta del nodo ne sarà il nodo padre.
2. (Facoltativo) Nella scheda Proprietà del nodo, è possibile inserire un nome per il nodo nel diagramma del processo. Se non è già selezionato un nodo padre, scegli un nodo dall'elenco Node parents (Nodi padre) da utilizzare come origine di input per la trasformazione.
3. (Facoltativo) Nella scheda Trasforma, puoi personalizzare il nome della nuova colonna. Per impostazione predefinita, verrà denominata "id".
4. (Facoltativo) Se il processo elabora e archivia i dati in modo incrementale, è necessario evitare che gli stessi ID vengano riutilizzati tra le esecuzioni del processo.

   Nella scheda Trasforma, seleziona l'opzione della casella di controllo Univoco. Includerà il timestamp del processo nell'identificatore, rendendolo univoco tra più esecuzioni. Per consentire l'inserimento di un numero maggiore, la colonna anziché di tipo lungo sarà decimale.

## Conversione di una colonna in tipo timestamp

È possibile utilizzare la trasformazione A timestamp per modificare il tipo di dati di una colonna numerica o di stringa in un timestamp, in modo da consentirne l'archiviazione con quel tipo di dati o l'applicazione ad altre trasformazioni che richiedono un timestamp.

Aggiunta di un nodo di trasformazione A timestamp nel diagramma di processo

1. Apri il pannello Risorse, quindi scegli A timestamp per aggiungere una nuova trasformazione al diagramma del processo. Il nodo selezionato al momento dell'aggiunta del nodo ne sarà il nodo padre.
2. (Facoltativo) Nella scheda Proprietà del nodo, è possibile inserire un nome per il nodo nel diagramma del processo. Se non è già selezionato un nodo padre, scegli un nodo dall'elenco Node parents (Nodi padre) da utilizzare come origine di input per la trasformazione.
3. Nella scheda Trasforma, inserisci il nome della colonna da convertire.
4. Nella scheda Trasforma, definisci come analizzare la colonna selezionata scegliendo il tipo.

   Se il valore è un numero, può essere espresso in secondi (timestamp Unix/Python), millisecondi o microsecondi, scegli l'opzione corrispondente.

   Se il valore è una stringa formattata, scegli il tipo "iso"; la stringa deve essere conforme a una delle varianti del formato ISO, ad esempio: "2022-11-02T14:40:59.915Z".

   Se a questo punto non conosci il tipo oppure righe diverse utilizzano tipi diversi, puoi scegliere "rilevamento automatico" e il sistema genererà la sua ipotesi migliore, a un costo di prestazioni ridotto.
5. (Facoltativo) Nella scheda Trasforma, invece di convertire la colonna selezionata, puoi crearne una nuova e mantenere l'originale inserendo un nome per la nuova colonna.

## Conversione di una colonna di timestamp in una stringa formattata

Formatta una colonna di timestamp in una stringa in base a uno schema. Puoi utilizzare Formatta timestamp per ottenere data e ora come stringa con il formato desiderato. Puoi definire il formato utilizzando la sintassi della data Spark e la maggior parte dei codici di data Python.

Ad esempio, se desideri che la stringa della data sia formattata come «2023-01-01 00:00», puoi definire tale formato utilizzando la sintassi Spark come «HH:mm» o i codici data Python equivalenti come «%y-%m-%D %H: %Myyyy-MM-dd »

Aggiunta di un nodo di trasformazione Formatta timestamp nel diagramma di processo

1. Apri il pannello Risorse, quindi scegli Formatta timestamp per aggiungere una nuova trasformazione al diagramma del processo. Il nodo selezionato al momento dell'aggiunta del nodo ne sarà il nodo padre.

2. (Facoltativo) Nella scheda Proprietà del nodo, è possibile inserire un nome per il nodo nel diagramma del processo. Se non è già selezionato un nodo padre, scegli un nodo dall'elenco Node parents (Nodi padre) da utilizzare come origine di input per la trasformazione.
3. Nella scheda Trasforma, inserisci il nome della colonna da convertire.
4. Nella scheda Trasforma, inserisci il modello di formato timestamp da utilizzare, espresso utilizzando la sintassi della data Spark o i codici di data Python.
5. (Facoltativo) Nella scheda Trasforma, invece di convertire la colonna selezionata, puoi crearne una nuova e mantenere l'originale inserendo un nome per la nuova colonna.

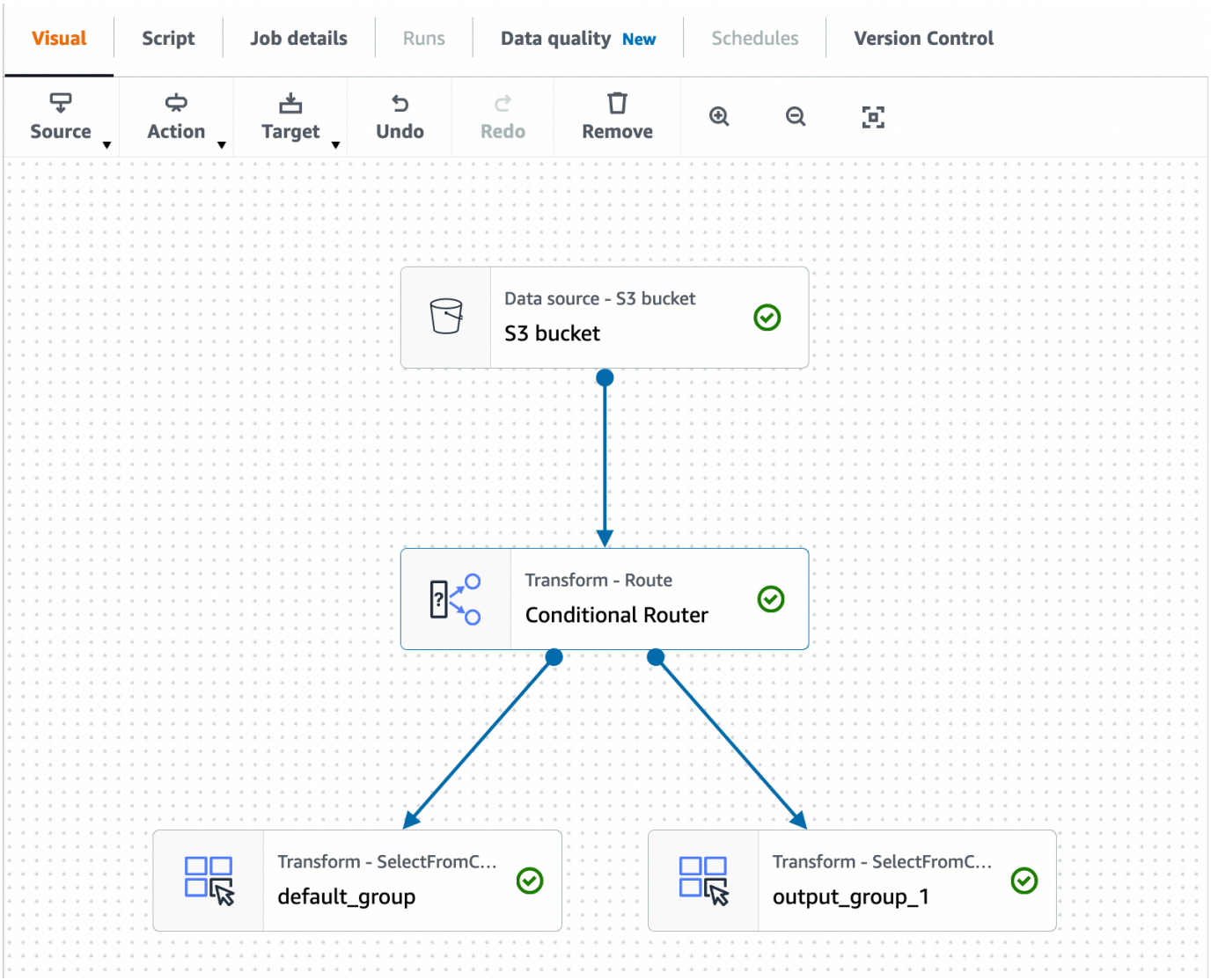
## Creazione di una trasformazione router condizionale

La trasformazione router condizionale consente di applicare più condizioni ai dati in ingresso. Ogni riga dei dati in ingresso viene valutata in base a una condizione di filtro di gruppo ed elaborata nel gruppo corrispondente. Se una riga soddisfa più di una condizione di filtro di gruppo, la trasformazione passa la riga a più gruppi. Se una riga non soddisfa alcuna condizione, può essere eliminata o indirizzata a un gruppo di output predefinito.

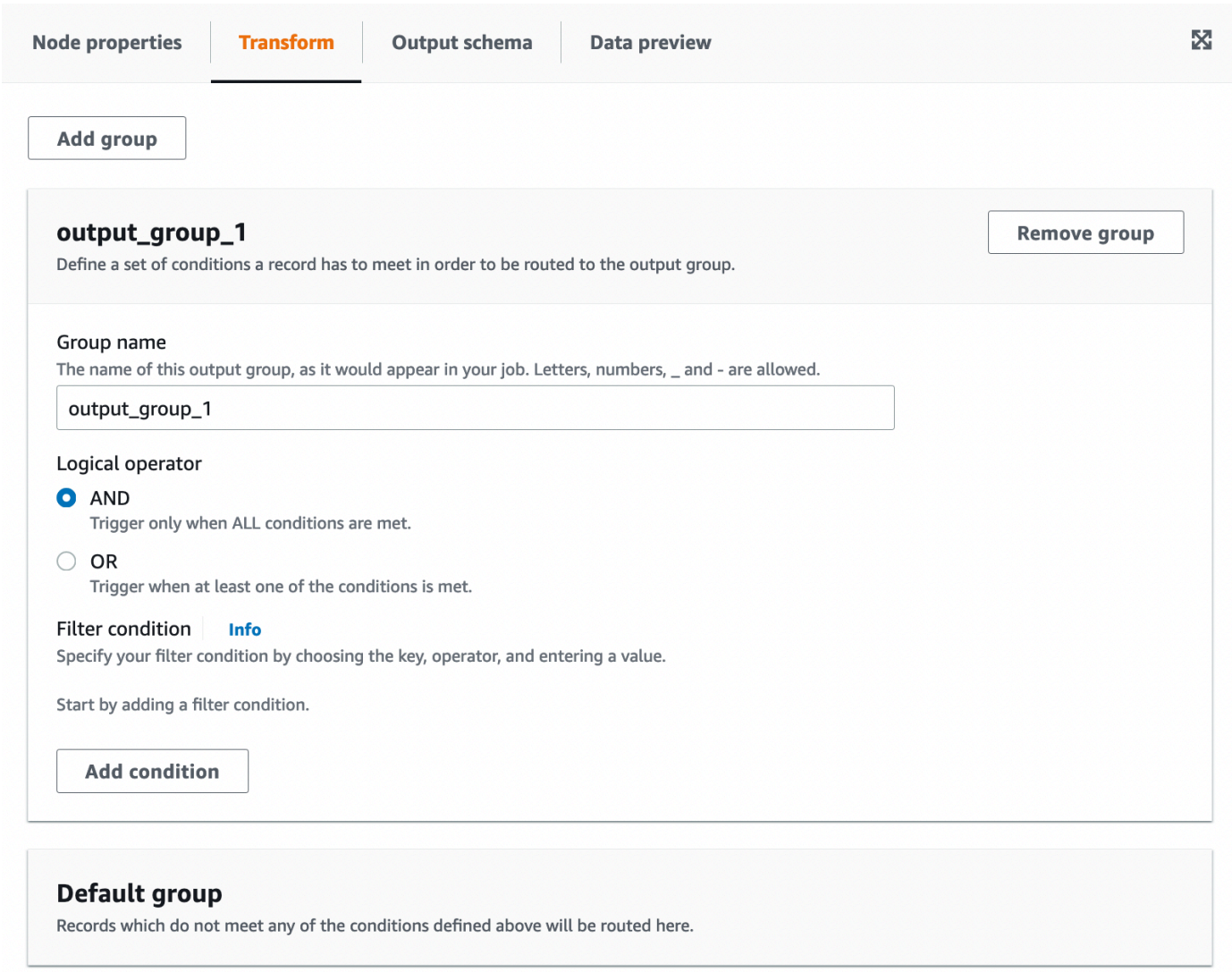
Questa trasformazione è simile alla trasformazione di filtro, ma utile per gli utenti che desiderano testare gli stessi dati di input su più condizioni.

Per aggiungere una trasformazione router condizionale:

1. Scegli un nodo in cui eseguire la trasformazione router condizionale. Può essere un nodo di origine o un'altra trasformazione.
2. Scegli Azione, quindi usa la barra di ricerca per trovare e scegliere "Router condizionale". Viene aggiunta una trasformazione Router condizionale insieme a due nodi di output. Un nodo di output, "Gruppo predefinito", contiene record che non soddisfano nessuna delle condizioni definite negli altri nodi di output. Il gruppo predefinito non può essere modificato.

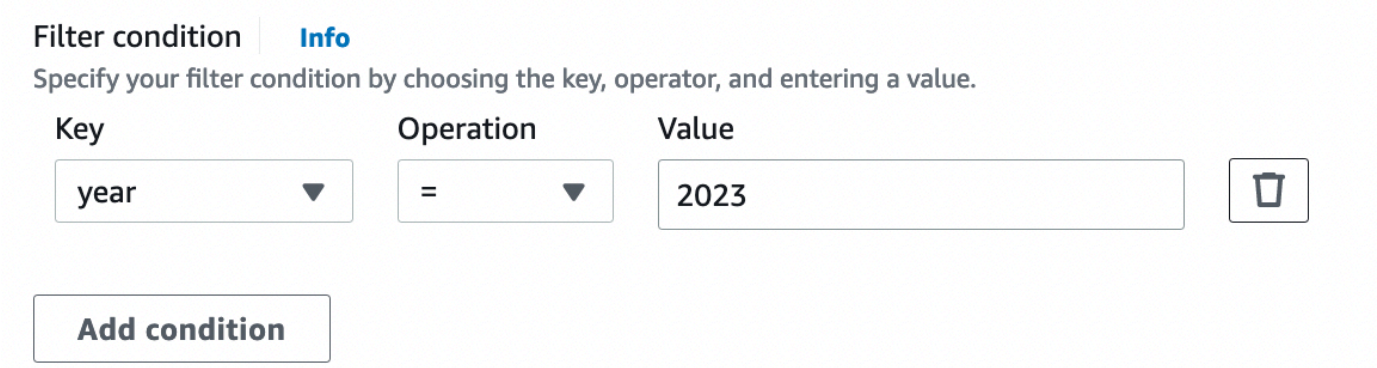



Puoi aggiungere altri gruppi di output scegliendo Aggiungi gruppo. Per ogni gruppo di output, è possibile assegnare un nome al gruppo e aggiungere condizioni di filtro e un operatore logico.




3. Rinomina il nome del gruppo di output inserendo un nuovo nome per il gruppo. AWS Glue Studio assegnerà automaticamente un nome ai tuoi gruppi (ad esempio, "output_group_1").
4. Scegli un operatore logico (AND, OR) e aggiungi una condizione di filtro specificando la chiave, l'operazione e il valore. Gli operatori logici consentono di implementare più di una condizione di filtro ed eseguire l'operatore logico su ogni condizione di filtro specificata.

   Quando si specifica la chiave, è possibile scegliere tra le chiavi disponibili nello schema. È quindi possibile scegliere l'operazione disponibile in base al tipo di chiave selezionata. Ad esempio, se il tipo di chiave è "stringa", l'operazione disponibile tra cui scegliere è "corrispondenze".




5. Inserisci il valore nel campo Valore. Per aggiungere condizioni di filtro aggiuntive, scegli Aggiungi condizione . Per rimuovere condizioni di filtro, scegli l'icona del cestino.

## Utilizzo della trasformazione Concatena colonne per aggiungere colonne

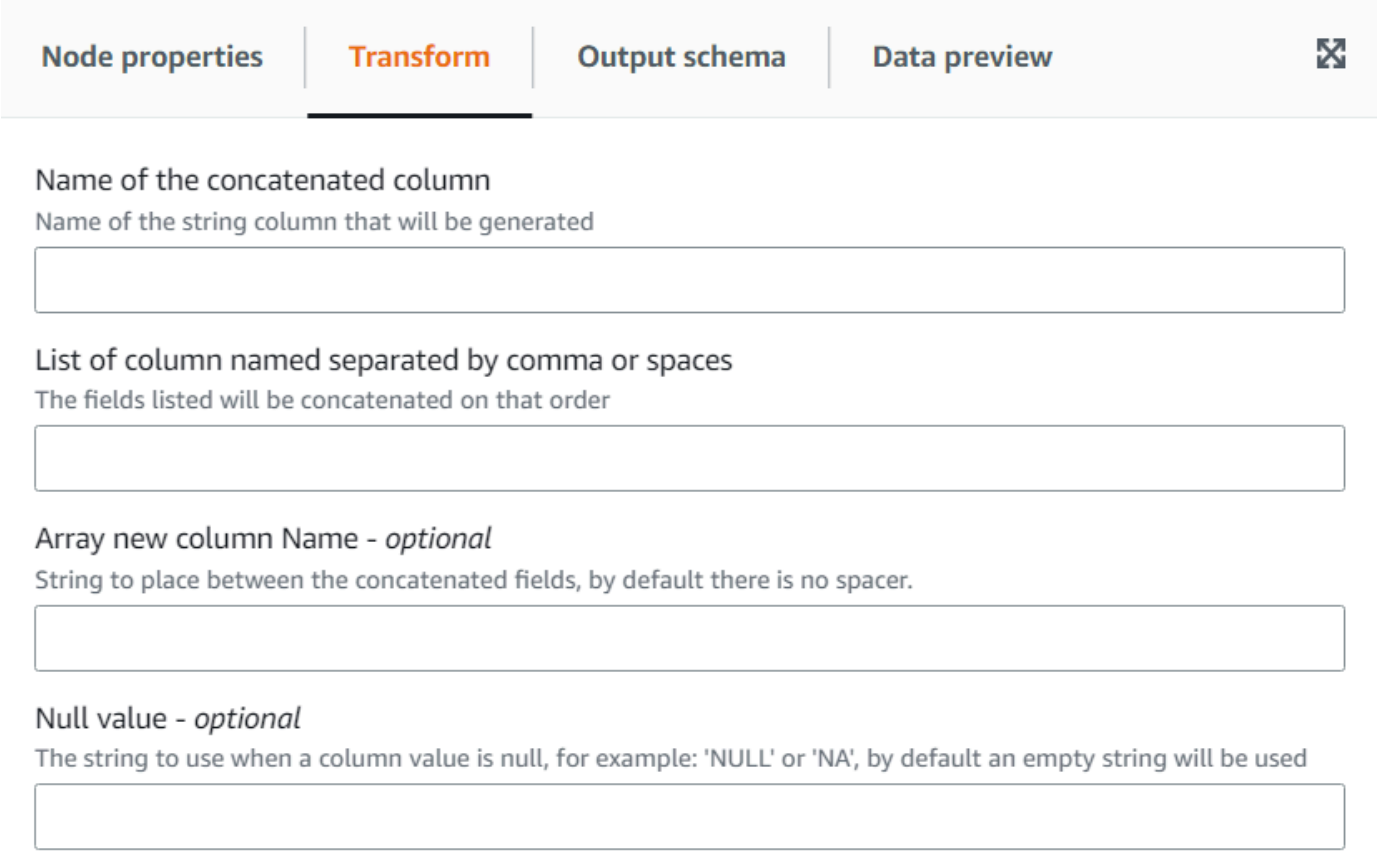
La trasformazione Concatena consente di creare una nuova colonna di stringhe utilizzando i valori di altre colonne con un distanziatore opzionale. Ad esempio, se definiamo una colonna concatenata "data" come concatenazione di "anno", "mese" e "giorno" (in quest'ordine) con "-" come spaziatore, otterremmo:

| giorno | mese | anno | data |
|---|---|---|---|
| 01 | 01 | 2020 | 2020-01-01 |
| 02 | 01 | 2020 | 2020-01-02 |
| 03 | 01 | 2020 | 2020-01-03 |
| 04 | 01 | 2020 | 2020-01-04 |

Per aggiungere una trasformazione Concatena:

1. Aprire il pannello Risorse. Quindi, scegliere Concatena colonne per aggiungere una nuova trasformazione al diagramma di processo. Il nodo selezionato al momento dell'aggiunta del nodo ne sarà il nodo padre.
2. (Facoltativo) Nella scheda Proprietà del nodo, è possibile inserire un nome per il nodo nel diagramma del processo. Se non è già selezionato un nodo padre, scegliere un nodo dall'elenco Nodi padre da utilizzare come origine di input per la trasformazione.

3. Nella scheda Trasforma, inserire il nome della colonna che conterrà la stringa concatenata e le colonne da concatenare. L'ordine in cui si selezionano le colonne nel menu a discesa sarà l'ordine utilizzato.




4. Spaziatore - facoltativo: inserire una stringa da inserire tra i campi concatenati. Per impostazione predefinita, non sono previsti spaziatori.
5. Valore nullo - facoltativo: inserire una stringa da utilizzare quando il valore di una colonna è nullo. Per impostazione predefinita, nei casi in cui le colonne hanno il valore “NULL” o “NA”, viene utilizzata una stringa vuota.

## Utilizzo della trasformazione Dividi stringa per suddividere una colonna di stringhe

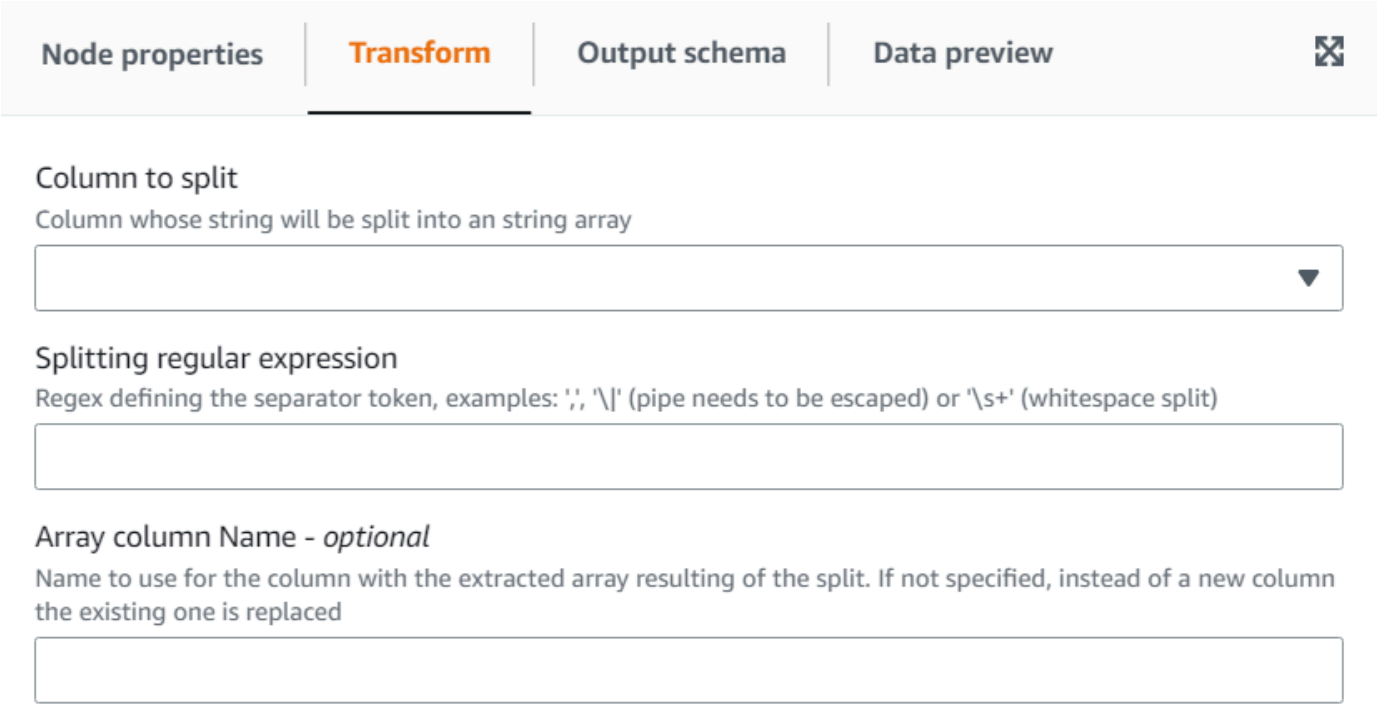
La trasformazione Dividi stringa consente di suddividere una stringa in un array di token utilizzando un'espressione regolare per definire come viene eseguita la suddivisione. È quindi possibile mantenere la colonna come tipo array o applicare una trasformazione Array a colonne successivamente a questa per estrarre i valori dell'array in campi di primo livello, supponendo che ogni token abbia un significato che conosciamo in precedenza. Inoltre, se l'ordine dei token è irrilevante (ad esempio, un insieme di categorie), è possibile utilizzare la trasformazione Espandi per generare una riga separata per ogni valore.

Ad esempio, è possibile dividere una colonna "categories" utilizzando una virgola come modello per aggiungere una colonna "categories_arr".

| product_id | categorie | categories_arr |
|---|---|---|
| 1 | sport,inverno | [sport, inverno] |
| 2 | giardino, attrezzi | [giardino, attrezzi] |
| 3 | videogiochi | [videogiochi] |
| 4 | gioco,gioco da tavolo,gioco di società | [gioco, gioco da tavolo, gioco di società] |

Per aggiungere una trasformazione Dividi stringa:

1. Apri il pannello Risorse, quindi scegli Dividi stringa per aggiungere una nuova trasformazione al diagramma di processo. Il nodo selezionato al momento dell'aggiunta del nodo ne sarà il nodo padre.
2. (Facoltativo) Nella scheda Proprietà del nodo, è possibile inserire un nome per il nodo nel diagramma del processo. Se non è già selezionato un nodo padre, scegliere un nodo dall'elenco Nodi padre da utilizzare come origine di input per la trasformazione.
3. Nella scheda Trasforma, scegli la colonna da dividere e inserisci il modello da utilizzare per dividere la stringa. Nella maggior parte dei casi puoi semplicemente inserire i caratteri, a meno che non abbiano un significato speciale come espressione regolare e debbano contenere caratteri di escape. I caratteri che richiedono escape sono `\.[]{}()<>*+-=!?^$|` e occorre aggiungere una barra rovesciata davanti al carattere. Ad esempio, se vuoi utilizzare un punto (".") come separatore, devi inserire `\.`. Tuttavia, la virgola non ha un significato speciale e può essere specificata così com'è: `,`.




4. (Facoltativo) Se desideri mantenere la colonna di stringhe originale, puoi inserire un nome per una nuova colonna di array: potrai così mantenere sia la colonna di stringhe originale sia la nuova colonna di array tokenizzata.

## Utilizzo della trasformazione Array a colonne per estrarre gli elementi di un array in colonne di primo livello

La trasformazione Array a colonne consente di estrarre alcuni o tutti gli elementi di una colonna di tipo array in nuove colonne. La trasformazione riempirà le nuove colonne il più possibile se l'array ha un numero sufficiente di valori da estrarre, prendendo facoltativamente gli elementi nelle posizioni specificate.

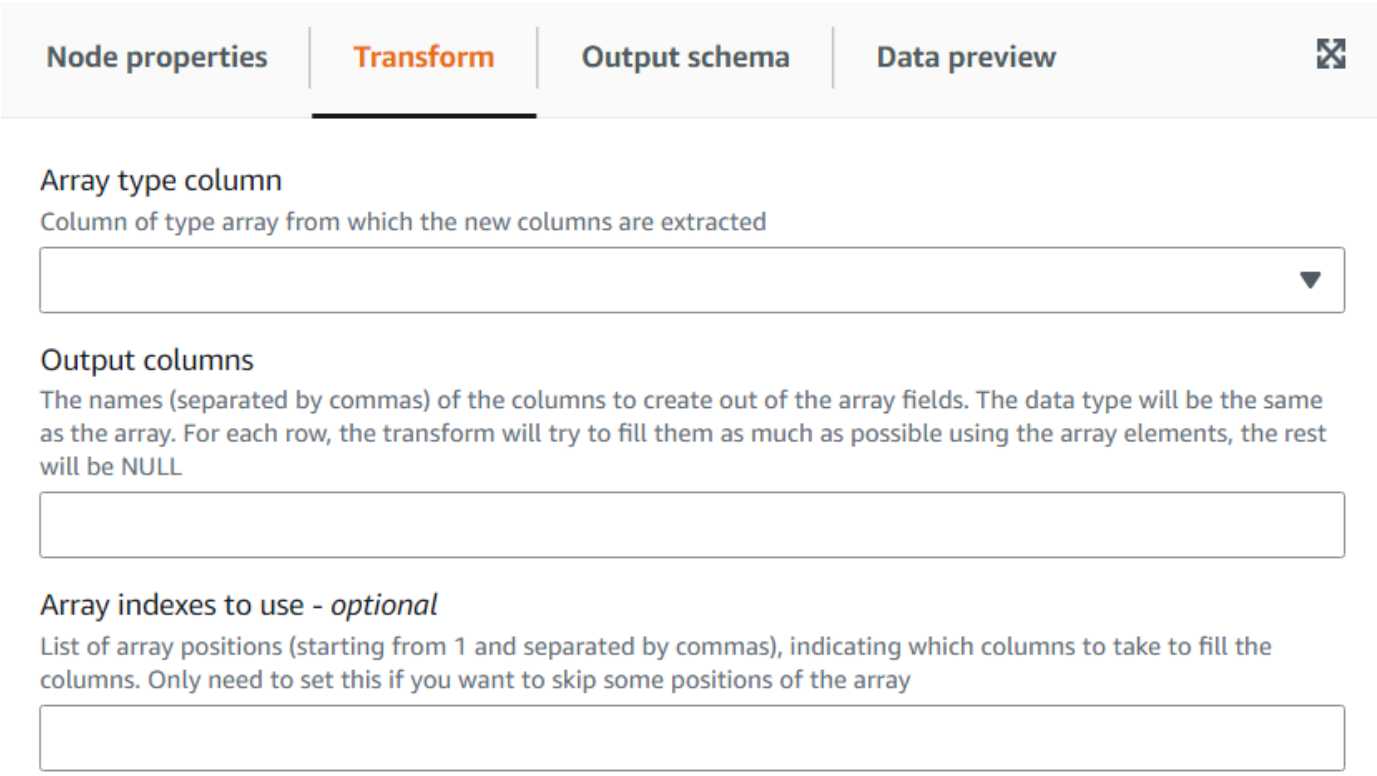
Ad esempio, se hai una colonna di array "subnet", che è il risultato dell'applicazione della trasformazione "Dividi stringa" su una sottorete ip v4, puoi estrarre la prima e la quarta posizione nelle nuove colonne "first_octect" e "fourth_octect". L'output della trasformazione in questo esempio sarebbe il seguente; nota che le ultime due righe hanno array più corti del previsto:

| sottorete | first_octect | fourth_octect |
|---|---|---|
| [54, 240, 197, 238] | 54 | 238 |

| sottorete | first_octect | fourth_octect |
|---|---|---|
| [192, 168, 0, 1] | 192 | 1 |
| [192, 168] | 192 | |
| [] | | |

Per aggiungere una trasformazione Array a colonne:

1. Apri il pannello Risorse, quindi scegli Array a colonne per aggiungere una nuova trasformazione al diagramma del processo. Il nodo selezionato al momento dell'aggiunta del nodo ne sarà il nodo padre.
2. (Facoltativo) Nella scheda Proprietà del nodo, è possibile inserire un nome per il nodo nel diagramma del processo. Se non è già selezionato un nodo padre, scegliere un nodo dall'elenco Nodi padre da utilizzare come origine di input per la trasformazione.
3. Nella scheda Trasforma, scegli la colonna dell'array da estrarre e inserisci l'elenco delle nuove colonne per i token estratti.



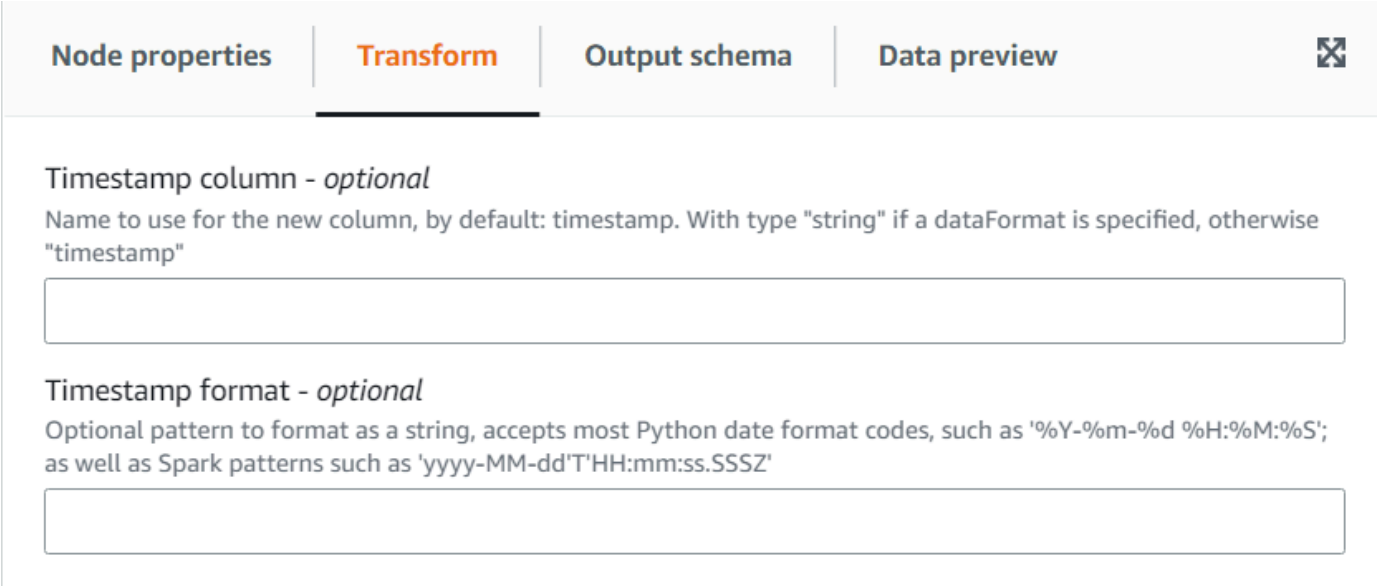
4. (Facoltativo) Se non vuoi prendere i token dell'array per assegnarli alle colonne, puoi specificare gli indici da prendere che verranno assegnati all'elenco di colonne nello stesso ordine specificato. Ad esempio, se le colonne di output sono "column1, column2, column3" e gli indici "4, 1, 3", il quarto elemento dell'array andrà alla column1, il primo alla column2 e il terzo alla column3 (se l'array è più corto del numero di indice, verrà impostato un valore NULL).

## Utilizzo della trasformazione Aggiungi timestamp corrente

La trasformazione Aggiungi timestamp corrente consente di contrassegnare le righe con l'ora in cui i dati sono stati elaborati. Ciò è utile per scopi di controllo o per tenere traccia della latenza nella pipeline di dati. È possibile aggiungere questa nuova colonna come tipo di dati timestamp o come stringa formattata.

Per aggiungere una trasformazione Aggiungi timestamp corrente:

1. Apri il pannello Risorse, quindi scegli Aggiungi timestamp corrente per aggiungere una nuova trasformazione al diagramma del processo. Il nodo selezionato al momento dell'aggiunta del nodo ne sarà il nodo padre.
2. (Facoltativo) Nella scheda Proprietà del nodo, è possibile inserire un nome per il nodo nel diagramma del processo. Se non è già selezionato un nodo padre, scegliere un nodo dall'elenco Nodi padre da utilizzare come origine di input per la trasformazione.




3. (Facoltativo) Nella scheda Trasforma, inserisci un nome personalizzato per la nuova colonna e un formato se preferisci che la colonna sia una stringa di data formattata.

## Utilizzo della trasformazione Pivot: righe a colonne

La trasformazione Pivot: righe a colonne consente di aggregare una colonna numerica ruotando valori univoci su colonne selezionate che diventano nuove colonne. Se sono selezionate più colonne, i valori vengono concatenati per denominare le nuove colonne. In questo modo, le righe vengono consolidate pur avendo più colonne con aggregazioni parziali per ogni valore univoco. Ad esempio, se disponi di questo set di dati sulle vendite per mese e paese (ordinato per facilitare la comprensione):

| anno | mese | paese | quantità |
|---|---|---|---|
| 2020 | Jan | uk | 32 |
| 2020 | Jan | de | 42 |
| 2020 | Jan | us | 64 |
| 2020 | Feb | uk | 67 |
| 2020 | Feb | de | 4 |
| 2020 | Feb | de | 7 |
| 2020 | Feb | us | 6 |
| 2020 | Feb | us | 12 |
| 2020 | Jan | us | 90 |

Se utilizzi quantità e paese come colonne di aggregazione, vengono create nuove colonne a partire dalla colonna paese originale. Nella tabella seguente sono presenti nuove colonne per de, uk e us anziché la colonna paese.

| anno | mese | de | uk | us |
|---|---|---|---|---|
| 2020 | Jan | 42 | 32 | 64 |
| 2020 | Jan | 11 | 67 | 18 |

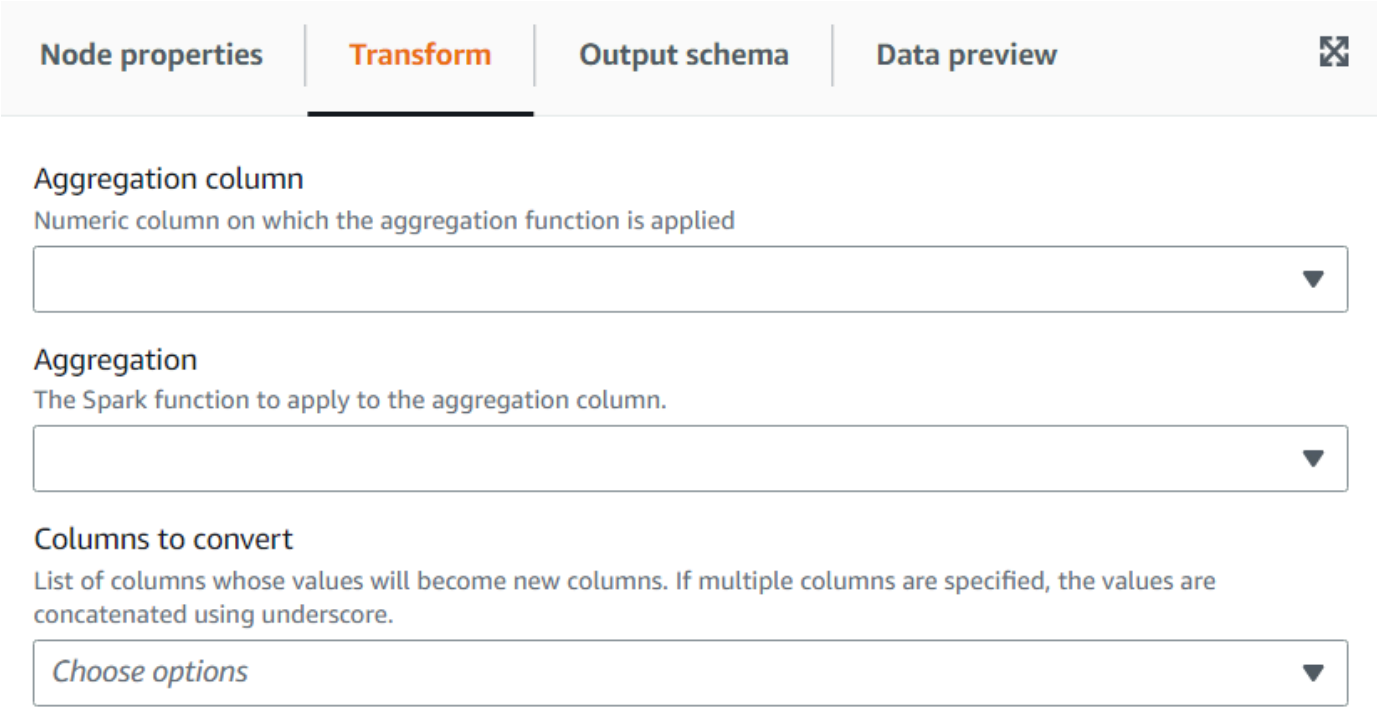
| anno | mese | de | uk | us |
|---|---|---|---|---|
| 2021 | Jan | | | 90 |

Se invece vuoi eseguire il pivot sia sul mese sia sul paese, otterrai una colonna per ogni combinazione dei valori di tali colonne:

| anno | Jan_de | Jan_uk | Jan_us | Feb_de | Feb_uk | Feb_us |
|---|---|---|---|---|---|---|
| 2020 | 42 | 32 | 64 | 11 | 67 | 18 |
| 2021 | | | 90 | | | |

Per aggiungere una trasformazione Pivot: righe a colonne:

1. Apri il pannello Risorse, quindi scegli Pivot: righe a colonne per aggiungere una nuova trasformazione al diagramma del processo. Il nodo selezionato al momento dell'aggiunta del nodo ne sarà il nodo padre.
2. (Facoltativo) Nella scheda Proprietà del nodo, è possibile inserire un nome per il nodo nel diagramma del processo. Se non è già selezionato un nodo padre, scegliere un nodo dall'elenco Nodi padre da utilizzare come origine di input per la trasformazione.
3. Nella scheda Trasforma, sceglie la colonna numerica che verrà aggregata per produrre i valori per le nuove colonne, la funzione di aggregazione da applicare e le colonne per convertire i valori univoci in nuove colonne.




## Utilizzo della trasformazione Elimina pivot: colonne a righe

La trasformazione Elimina pivot consente di convertire le colonne in valori di nuove colonne generando una riga per ogni valore univoco. È l'opposto del pivot, ma tieni presente che non è equivalente, in quanto non può separare le righe con valori identici che sono state aggregate o suddividere le combinazioni nelle colonne originali. Per fare queste operazioni, puoi utilizzare in seguito una trasformazione Dividi. Ad esempio, in presenza della tabella seguente:

| anno | mese | de | uk | us |
|---|---|---|---|---|
| 2020 | Jan | 42 | 32 | 64 |
| 2020 | Feb | 11 | 67 | 18 |
| 2021 | Jan | | | 90 |

È possibile eliminare il pivot dalle colonne "de", "uk" e "us" in una colonna "country" con il valore "amount" e ottenere quanto segue (ordinato qui a scopo illustrativo):

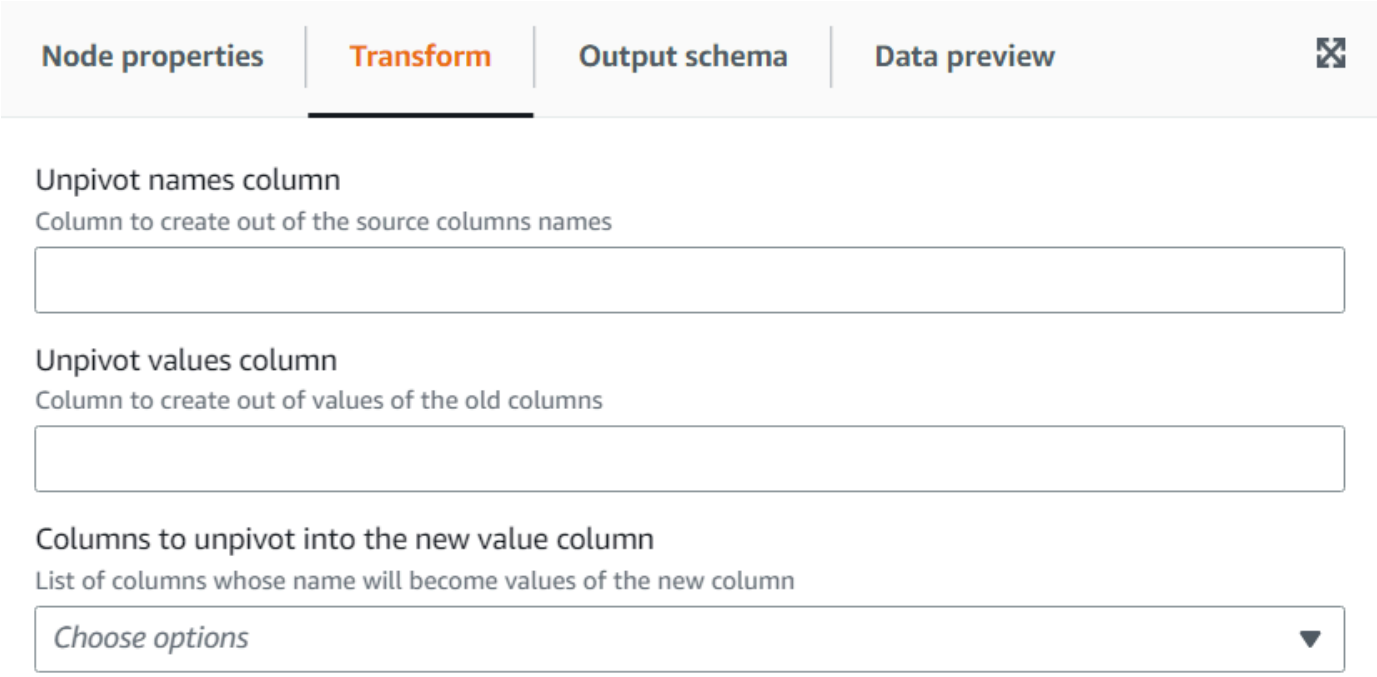
| anno | mese | paese | quantità |
|---|---|---|---|
| 2020 | Jan | uk | 32 |
| 2020 | Jan | de | 42 |
| 2020 | Jan | us | 64 |
| 2020 | Feb | uk | 67 |
| 2020 | Feb | de | 11 |
| 2020 | Feb | us | 18 |
| 2021 | Jan | us | 90 |

Nota che le colonne con un valore NULL ("de" e "uk" di gennaio 2021) non vengono generate per impostazione predefinita. È possibile abilitare questa opzione per ottenere:

| anno | mese | paese | quantità |
|---|---|---|---|
| 2020 | Jan | uk | 32 |
| 2020 | Jan | de | 42 |
| 2020 | Jan | us | 64 |
| 2020 | Feb | uk | 67 |
| 2020 | Feb | de | 11 |
| 2020 | Feb | us | 18 |
| 2021 | Jan | us | 90 |
| 2021 | Jan | de | |
| 2021 | Jan | uk | |

Per aggiungere una trasformazione Elimina pivot: colonne a righe:

1. Apri il pannello Risorse, quindi scegli Elimina pivot: colonne a righe per aggiungere una nuova trasformazione al diagramma del processo. Il nodo selezionato al momento dell'aggiunta del nodo ne sarà il nodo padre.
2. (Facoltativo) Nella scheda Proprietà del nodo, è possibile inserire un nome per il nodo nel diagramma del processo. Se non è già selezionato un nodo padre, scegliere un nodo dall'elenco Nodi padre da utilizzare come origine di input per la trasformazione.
3. Nella scheda Trasforma, inserisci le nuove colonne da creare per contenere i nomi e i valori delle colonne dalle quali intendi eliminare il pivot.

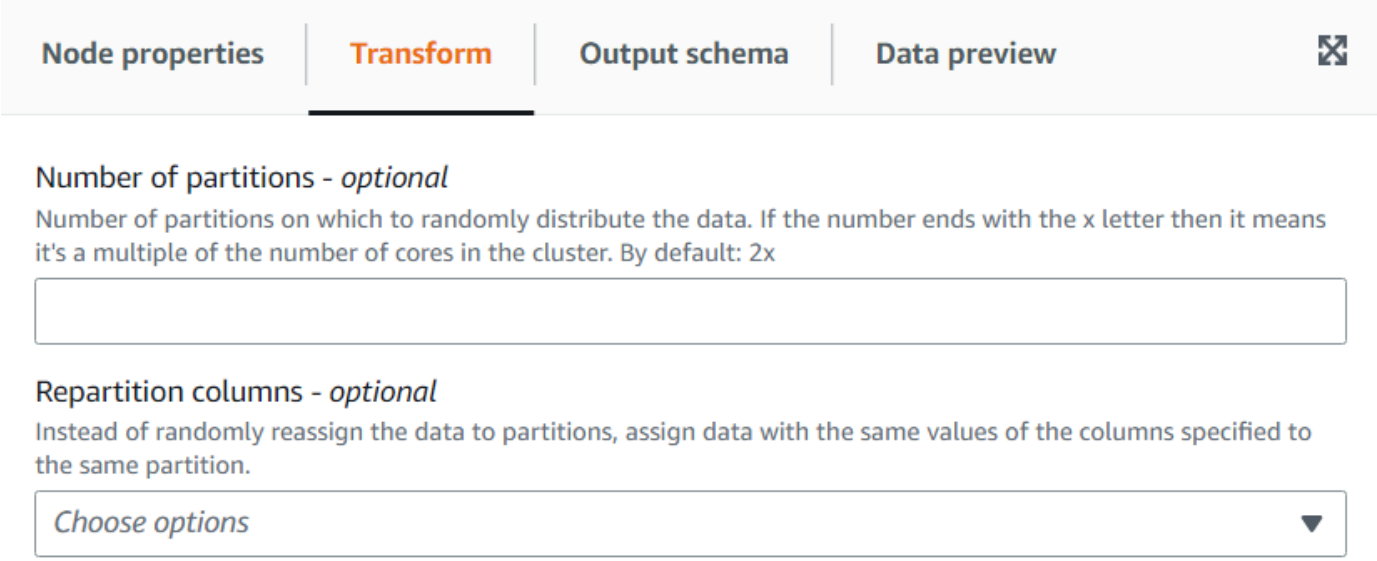



## Utilizzo della trasformazione Bilancia automaticamente elaborazione per ottimizzare il runtime

La trasformazione Bilancia automaticamente elaborazione ridistribuisce i dati tra i worker per migliorare le prestazioni. Ciò è utile nei casi in cui i dati non sono bilanciati o, poiché provengono dall'origine, non consentono un'elaborazione parallela sufficiente. Questo è comune quando l'origine è compressa con gzip o è JDBC. La ridistribuzione dei dati ha un costo prestazionale modesto, quindi l'ottimizzazione potrebbe non sempre compensare tale sforzo se i dati fossero già ben bilanciati. A un livello più basso, la trasformazione utilizza la ripartizione Apache Spark per riassegnare in modo casuale i dati tra una serie di partizioni ottimali per la capacità del cluster. Per gli utenti esperti, è

possibile inserire una serie di partizioni manualmente. Inoltre, può essere utilizzato per ottimizzare la scrittura di tabelle partizionate riorganizzando i dati in base a colonne specificate. Ciò si traduce in file di output più consolidati.

1. Apri il pannello Risorse, quindi scegli Bilancia automaticamente elaborazione per aggiungere una nuova trasformazione al diagramma di processo. Il nodo selezionato al momento dell'aggiunta del nodo ne sarà il nodo padre.
2. (Facoltativo) Nella scheda Proprietà del nodo, è possibile inserire un nome per il nodo nel diagramma del processo. Se non è già selezionato un nodo padre, scegliere un nodo dall'elenco Nodi padre da utilizzare come origine di input per la trasformazione.
3. (Facoltativo) Nella scheda Trasforma è possibile inserire un numero di partizioni. In generale, si consiglia di lasciare che sia il sistema a decidere questo valore, tuttavia è possibile regolare il moltiplicatore o inserire un valore specifico se è necessario controllarlo. Se intendi salvare i dati partizionati per colonne, puoi scegliere le stesse colonne come colonne di ripartizione. In questo modo ridurrà al minimo il numero di file su ciascuna partizione ed eviterà di avere molti file per partizione, il che ostacolerebbe le prestazioni degli strumenti che eseguono query su tali dati.




## Utilizzo della trasformazione Colonna derivata per combinare altre colonne

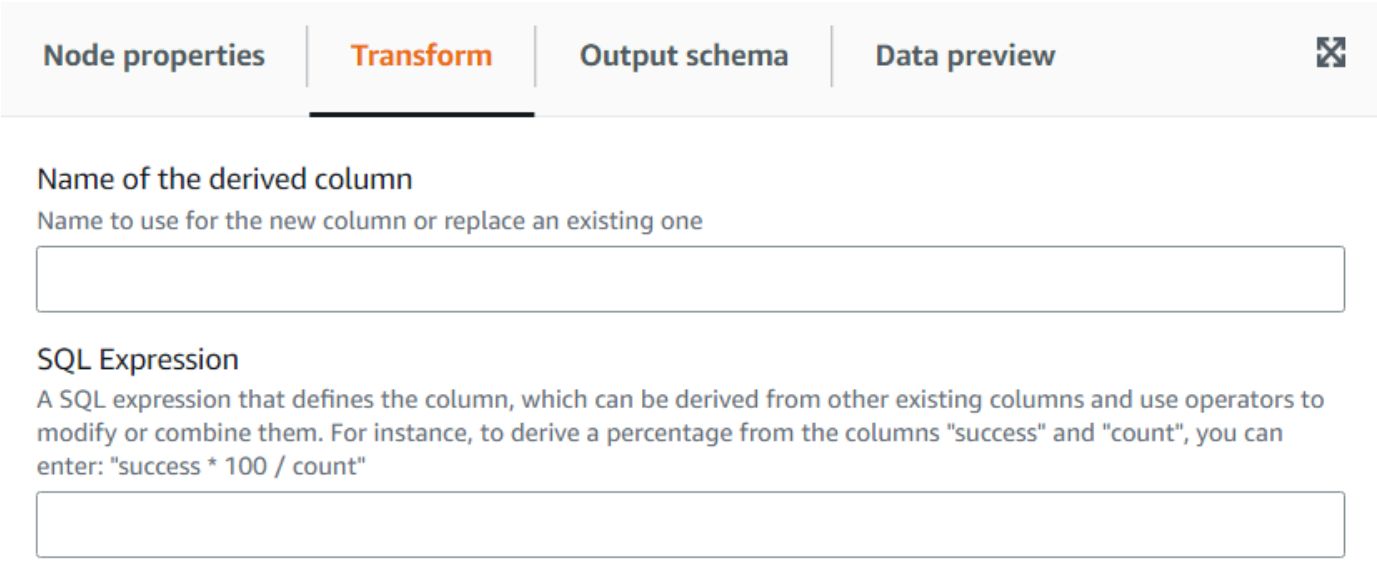
La trasformazione Colonna derivata consente di definire una nuova colonna basata su una formula matematica o un'espressione SQL in cui è possibile utilizzare altre colonne nei dati, oltre a costanti e valori letterali. Ad esempio, per ricavare una colonna "percentage" dalle colonne "success" e "count", puoi inserire l'espressione SQL: "success * 100/count || '%'".

Risultato dell'esempio:

| success | count | percentage |
|---|---|---|
| 14 | 100 | 14% |
| 6 | 20 | 3% |
| 3 | 40 | 7,5% |

Per aggiungere una trasformazione Colonna derivata:

1. Aprire il pannello Risorse, quindi scegliere Colonna derivata per aggiungere una nuova trasformazione al diagramma di processo. Il nodo selezionato al momento dell'aggiunta del nodo ne sarà il nodo padre.
2. (Facoltativo) Nella scheda Proprietà del nodo, è possibile inserire un nome per il nodo nel diagramma del processo. Se non è già selezionato un nodo padre, scegliere un nodo dall'elenco Nodi padre da utilizzare come origine di input per la trasformazione.
3. Nella scheda Trasforma, inserire il nome della colonna e l'espressione per il relativo contenuto.



## Utilizzo della trasformazione Ricerca per aggiungere dati corrispondenti da una tabella di catalogo

La trasformazione Ricerca consente di aggiungere colonne da una tabella di catalogo definita quando le chiavi corrispondono alle colonne di ricerca definite nei dati. Ciò equivale a eseguire un join esterno sinistro tra i dati e la tabella di ricerca, utilizzando come condizioni le colonne corrispondenti.

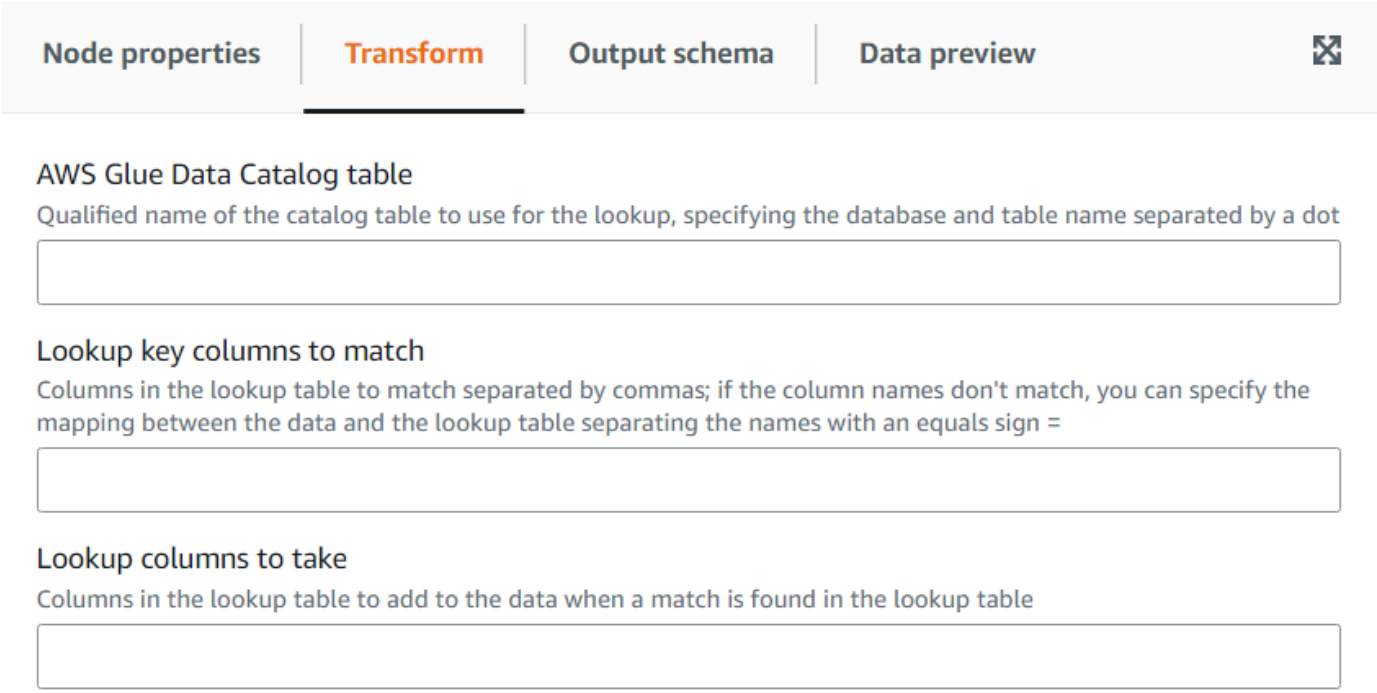
Per aggiungere una trasformazione Ricerca:

1. Apri il pannello Risorse, quindi scegli Ricerca per aggiungere una nuova trasformazione al diagramma di processo. Il nodo selezionato al momento dell'aggiunta del nodo ne sarà il nodo padre.
2. (Facoltativo) Nella scheda Proprietà del nodo, è possibile inserire un nome per il nodo nel diagramma del processo. Se non è già selezionato un nodo padre, scegliere un nodo dall'elenco Nodi padre da utilizzare come origine di input per la trasformazione.
3. Nella scheda Trasforma, inserisci il nome completo della tabella di catalogo da utilizzare per eseguire le ricerche. Ad esempio, se il tuo database è "mydb" e la tua tabella "mytable", inserisci "mydb.mytable". Inserisci quindi i criteri per trovare una corrispondenza nella tabella di ricerca, se la chiave di ricerca è composta. Inserisci l'elenco delle colonne chiave separate da virgole. Se una o più colonne chiave non hanno lo stesso nome, devi definire la mappatura delle corrispondenze.

   Ad esempio, se le colonne di dati sono "user_id" e "region" e nella tabella utenti le colonne corrispondenti sono denominate "id" e "region", nel campo Colonne da abbinare, inserisci: "user_id=id, region". Potresti anche utilizzare region=region, ma non è necessario poiché sono uguali.
4. Infine, inserisci le colonne da estrarre dalla riga corrispondente nella tabella di ricerca per incorporarle nei dati. Se non viene trovata alcuna corrispondenza, tali colonne verranno impostate su NULL.

> **Note**
>
> Sotto la trasformazione Ricerca, viene utilizzato un join a sinistra a fini di efficienza. Se la tabella di ricerca ha una chiave primaria composita, assicurati di impostare le colonne di corrispondenza in modo che includano tutte le colonne che fanno parte di tale chiave, così da ottenere una sola corrispondenza. Altrimenti, più righe nella tabella di ricerca

corrisponderanno, il che porterà ad aggiungere righe duplicate per ogni corrispondenza trovata.




## Utilizzo della trasformazione Espandi array o mappa in righe

La trasformazione Espandi consente di estrarre valori da una struttura nidificata in singole righe più facili da manipolare. Nel caso di un array, la trasformazione genererà una riga per ogni valore dell'array, replicando i valori per le altre colonne della riga. Nel caso di una mappa, la trasformazione genererà una riga per ogni voce con la chiave e il valore come colonne più ogni altra colonna presente nella riga.

Ad esempio, se abbiamo questo set di dati che ha una colonna di array "categoria" con più valori.

| product_id | category |
|---|---|
| 1 | [sport, inverno] |
| 2 | [giardino, attrezzi] |
| 3 | [videogiochi] |

| product_id | category |
|---|---|
| 4 | [gioco, gioco da tavolo, gioco di società] |
| 5 | [] |

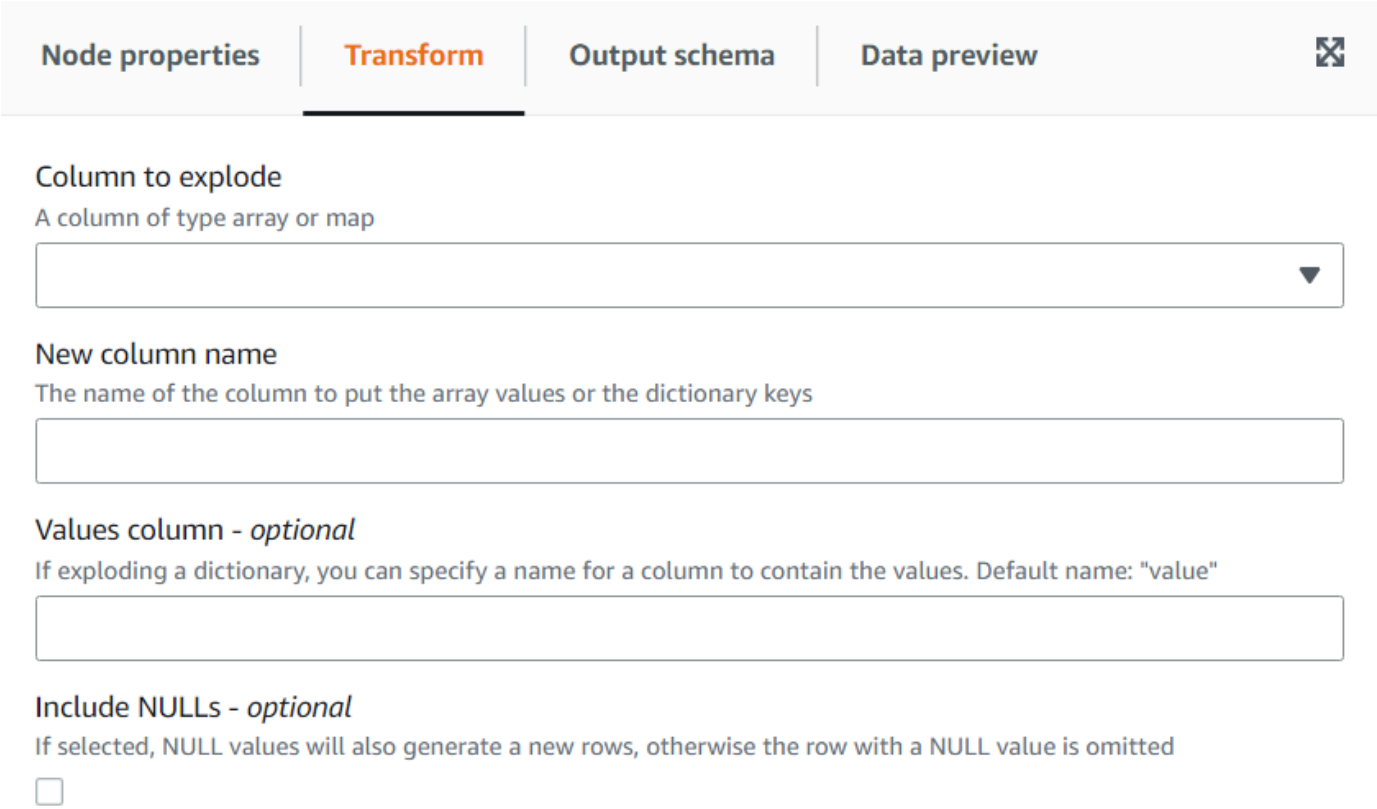
Se espandi la colonna "category" in una colonna con lo stesso nome, sovrascriverai la colonna. Puoi selezionare quello che desideri NULLs includere per ottenere quanto segue (ordinato a scopo illustrativo):

| product_id | category |
|---|---|
| 1 | sport |
| 1 | inverno |
| 2 | giardino |
| 2 | strumento |
| 3 | videogiochi |
| 4 | game |
| 4 | gioco da tavolo |
| 4 | gioco di società |
| 5 | |

Per aggiungere una trasformazione Espandi array o mappa in righe:

1. Apri il pannello Risorse, quindi scegli Espandi array o mappa in righe per aggiungere una nuova trasformazione al diagramma del processo. Il nodo selezionato al momento dell'aggiunta del nodo ne sarà il nodo padre.
2. (Facoltativo) Nella scheda Proprietà del nodo, è possibile inserire un nome per il nodo nel diagramma del processo. Se non è già selezionato un nodo padre, scegliere un nodo dall'elenco Nodi padre da utilizzare come origine di input per la trasformazione.

3. Nella scheda Trasforma, scegli la colonna da espandere (deve essere di tipo array o mappa). Quindi, specifica un nome per la colonna relativa agli elementi dell'array oppure i nomi delle colonne per chiavi e valori nel caso di espansione di una mappa.
4. (Facoltativo) Nella scheda Trasforma, per impostazione predefinita, se la colonna da espandere è NULL o ha una struttura vuota, verrà omessa nel set di dati espanso. Se vuoi mantenere la riga (con le nuove colonne come NULL), seleziona « NULLsIncludi».

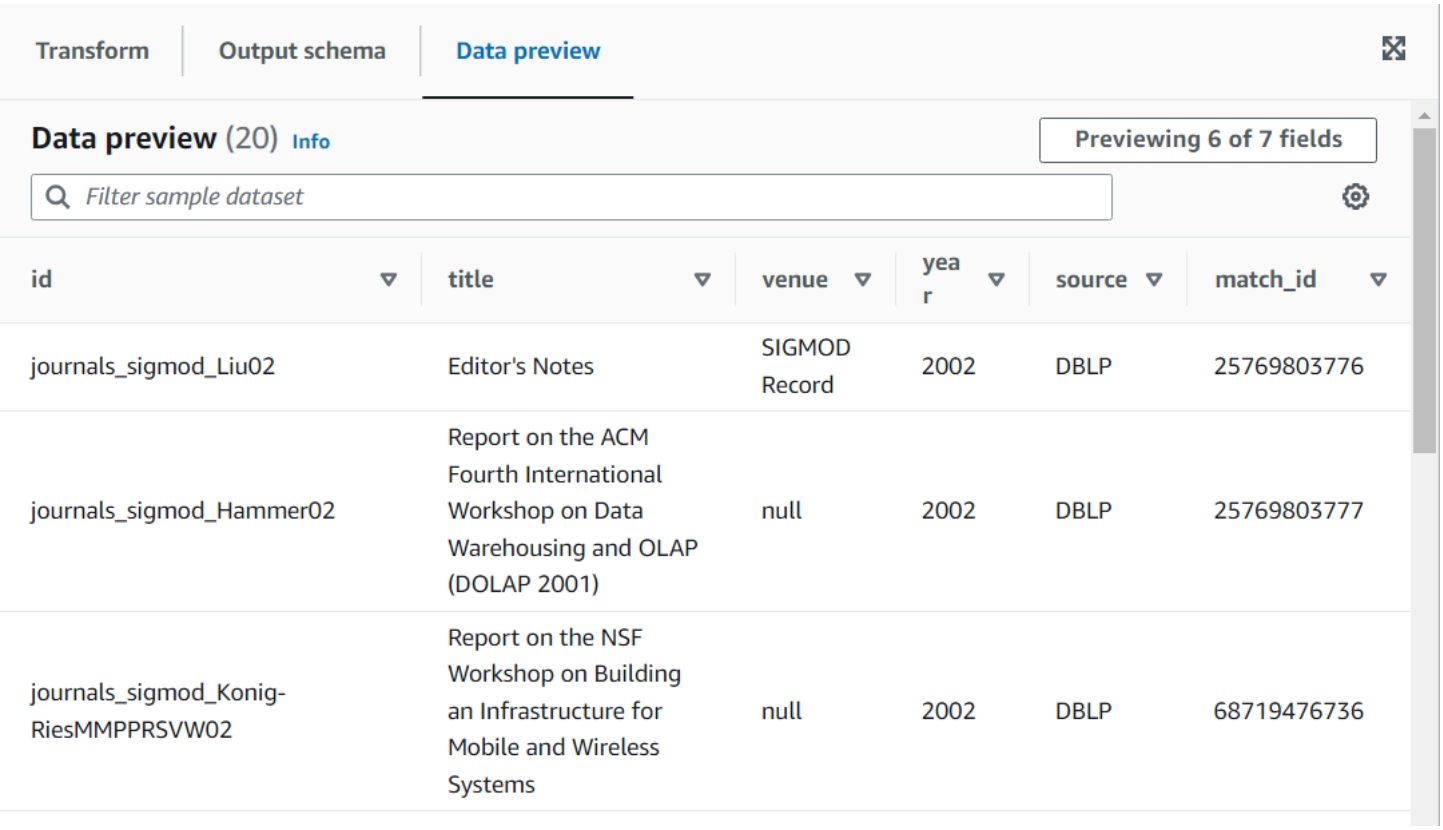



## Utilizzo della trasformazione Corrispondenza dei record per richiamare una trasformazione di classificazione dei dati esistente

Questa trasformazione richiama una trasformazione di classificazione dei dati di machine learning Corrispondenza dei record esistente.

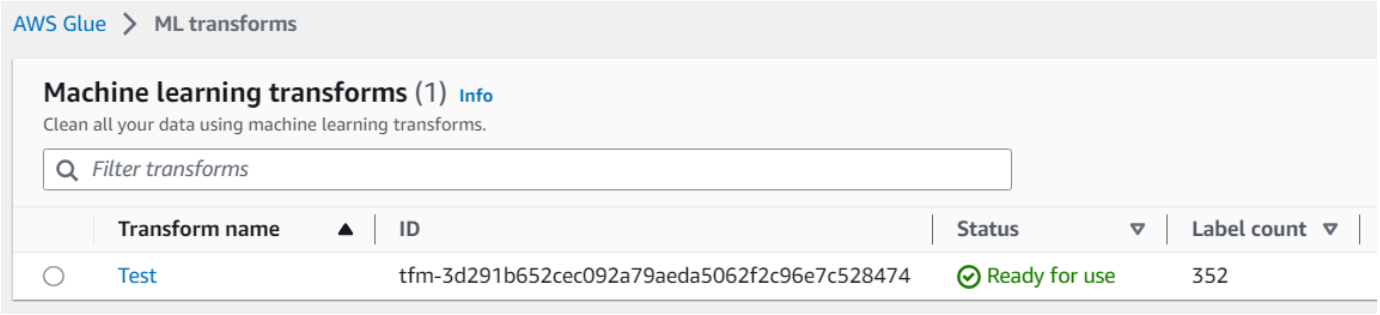
La trasformazione valuta i dati correnti rispetto al modello addestrato sulla base di etichette. Viene aggiunta una colonna "match_id" per assegnare ogni riga a un gruppo di elementi considerati equivalenti in base all'addestramento dell'algoritmo. Per ulteriori informazioni, vedere Record matching with Lake Formation FindMatches.

> **Note**
>
> La versione di AWS Glue utilizzata dal visual job deve corrispondere alla versione AWS Glue utilizzata per creare la trasformazione Record Matching.




Aggiunta di un nodo di trasformazione Corrispondenza dei record al diagramma di processo

1. Apri il pannello Risorse, quindi scegli Corrispondenza dei record per aggiungere una nuova trasformazione al diagramma del processo. Il nodo selezionato al momento dell'aggiunta del nodo ne sarà il nodo padre.
2. Nel pannello Proprietà del nodo, è possibile assegnare al nodo un nome nel diagramma del processo. Se non è già selezionato un nodo padre, scegli un nodo dall'elenco Node parents (Nodi padre) da utilizzare come origine di input per la trasformazione.
3. Nella scheda Trasforma, inserisci l'ID ottenuto dalla pagina delle Trasformazioni di Machine learning:




4. (Facoltativo) Nella scheda Trasforma, puoi selezionare l'opzione per aggiungere i punteggi di affidabilità. Al costo di un calcolo aggiuntivo, il modello stimerà un punteggio di affidabilità per ogni corrispondenza sotto forma di colonna aggiuntiva.

## Rimozione di righe nulle

Questa trasformazione rimuove dal set di dati le righe che hanno tutte le colonne come nulle. Inoltre, è possibile estendere questi criteri per includere anche i campi vuoti, in modo da mantenere le righe in cui almeno una colonna non è vuota.

Aggiunta di un nodo di trasformazione Rimuovi righe nulle al diagramma di processo

1. Apri il pannello Risorse, quindi scegli Rimuovi righe nulle per aggiungere una nuova trasformazione al diagramma del processo. Il nodo selezionato al momento dell'aggiunta del nodo ne sarà il nodo padre.
2. Nel pannello Proprietà del nodo, è possibile assegnare al nodo un nome nel diagramma del processo. Se non è già selezionato un nodo padre, scegli un nodo dall'elenco Node parents (Nodi padre) da utilizzare come origine di input per la trasformazione.
3. (Facoltativo) Nella scheda Trasforma, seleziona l'opzione Estesa se desideri che le righe siano, oltre che non nulle, anche non vuote; in questo modo le stringhe, gli array o le mappe vuote verranno considerati nulli ai fini di questa trasformazione.

## Analisi di una colonna di stringhe contenente dati JSON

Questa trasformazione analizza una colonna di stringhe contenente dati JSON e la converte in una struttura o in una colonna di array, a seconda che il JSON sia rispettivamente un oggetto o un array. Facoltativamente, puoi mantenere sia la colonna analizzata sia quella originale.

Lo schema JSON può essere fornito o dedotto (nel caso di oggetti JSON), con campionamento opzionale.

Aggiunta di un nodo di trasformazione Analizza colonna JSON al diagramma di processo

1. Apri il pannello Risorse, quindi scegli Analizza colonna JSON per aggiungere una nuova trasformazione al diagramma del processo. Il nodo selezionato al momento dell'aggiunta del nodo ne sarà il nodo padre.
2. Nel pannello Proprietà del nodo, è possibile assegnare al nodo un nome nel diagramma del processo. Se non è già selezionato un nodo padre, scegli un nodo dall'elenco Node parents (Nodi padre) da utilizzare come origine di input per la trasformazione.
3. Nella scheda Trasforma, seleziona la colonna contenente la stringa JSON.
4. (Facoltativo) Nella scheda Trasforma, inserisci lo schema seguito dai dati JSON utilizzando la sintassi SQL, ad esempio "field1 STRING, field2 INT" nel caso di un oggetto oppure "ARRAY<STRING>" nel caso di un array.

   Nel caso di un array, lo schema è richiesto, ma nel caso di un oggetto, se lo schema non è specificato, verrà dedotto utilizzando i dati. Per ridurre l'impatto dell'inferenza dello schema, specialmente su un set di dati di grandi dimensioni, puoi evitare di leggere l'intero dato due volte inserendo un Rapporto di campioni da utilizzare per dedurre lo schema. Se il valore è inferiore a 1, viene utilizzato il rapporto corrispondente di campioni casuali per dedurre lo schema. Se i dati sono affidabili e l'oggetto è coerente tra le righe, è possibile utilizzare un rapporto ridotto, ad esempio 0,1, per migliorare le prestazioni.
5. (Facoltativo) Nella scheda Trasforma, puoi inserire un nuovo nome di colonna se desideri mantenere sia la colonna di stringa originale sia la colonna analizzata.

## Estrazione di un percorso JSON

Questa trasformazione estrae nuove colonne da una colonna di stringhe JSON. Questa trasformazione è utile quando sono necessari solo pochi elementi di dati e non si desidera importare l'intero contenuto JSON nello schema della tabella.

Aggiunta di un nodo di trasformazione Estrai percorso JSON nel diagramma di processo

1. Apri il pannello Risorse, quindi scegli Estrai percorso JSON per aggiungere una nuova trasformazione al diagramma del processo. Il nodo selezionato al momento dell'aggiunta del nodo ne sarà il nodo padre.
2. Nel pannello Proprietà del nodo, è possibile assegnare al nodo un nome nel diagramma del processo. Se non è già selezionato un nodo padre, scegli un nodo dall'elenco Node parents (Nodi padre) da utilizzare come origine di input per la trasformazione.

3. Nella scheda Trasforma, seleziona la colonna contenente la stringa JSON. Inserisci una o più espressioni di percorso JSON separate da virgole, ognuna delle quali fa riferimento a come estrarre un valore dall'array o dall'oggetto JSON. Ad esempio, se la colonna JSON contenesse oggetti con le proprietà "prop_1" e "prop2", puoi estrarli entrambi specificando i nomi "prop_1, prop_2".

   Se il campo JSON contiene caratteri speciali, ad esempio per estrarre da questo JSON la proprietà `{"a. a": 1}`, puoi utilizzare il percorso `$['a. a']`. L'eccezione è la virgola perché è riservata a percorsi separati. Inserisci quindi i nomi di colonna corrispondenti per ogni percorso, separati da virgole.

4. (Facoltativo) Nella scheda Trasforma, puoi selezionare di eliminare la colonna JSON una volta estratta. Ciò ha senso quando non hai bisogno del resto dei dati JSON dopo aver estratto le parti necessarie.

## Estrazione di frammenti di stringa utilizzando un'espressione regolare

Questa trasformazione estrae frammenti di stringa utilizzando un'espressione regolare e crea a partire da essa una nuova colonna o anche più colonne, se si utilizzano gruppi di regex.

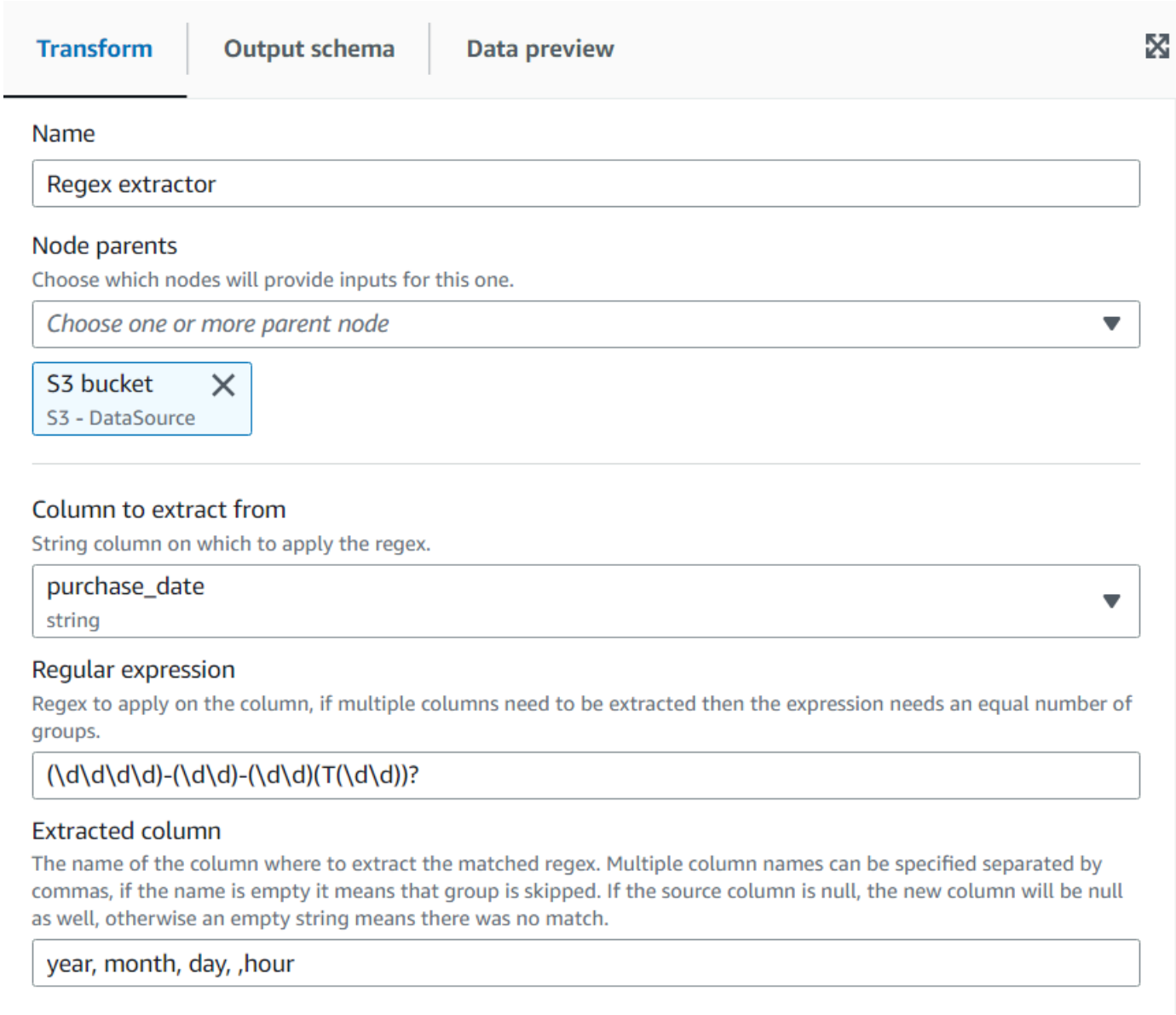
Aggiunta di un nodo di trasformazione Estrattore regex al diagramma di processo

1. Apri il pannello Risorse, quindi scegli Estrattore regex per aggiungere una nuova trasformazione al diagramma del processo. Il nodo selezionato al momento dell'aggiunta del nodo ne sarà il nodo padre.
2. Nel pannello Proprietà del nodo, è possibile assegnare al nodo un nome nel diagramma del processo. Se non è già selezionato un nodo padre, scegli un nodo dall'elenco Node parents (Nodi padre) da utilizzare come origine di input per la trasformazione.
3. Nella scheda Trasforma, inserisci l'espressione regolare e la colonna alla quale deve essere applicata. Quindi inserisci il nome della nuova colonna in cui archiviare la stringa corrispondente. La nuova colonna sarà nulla solo se la colonna di origine è nulla, mentre se l'espressione regolare non corrisponde la colonna sarà vuota.

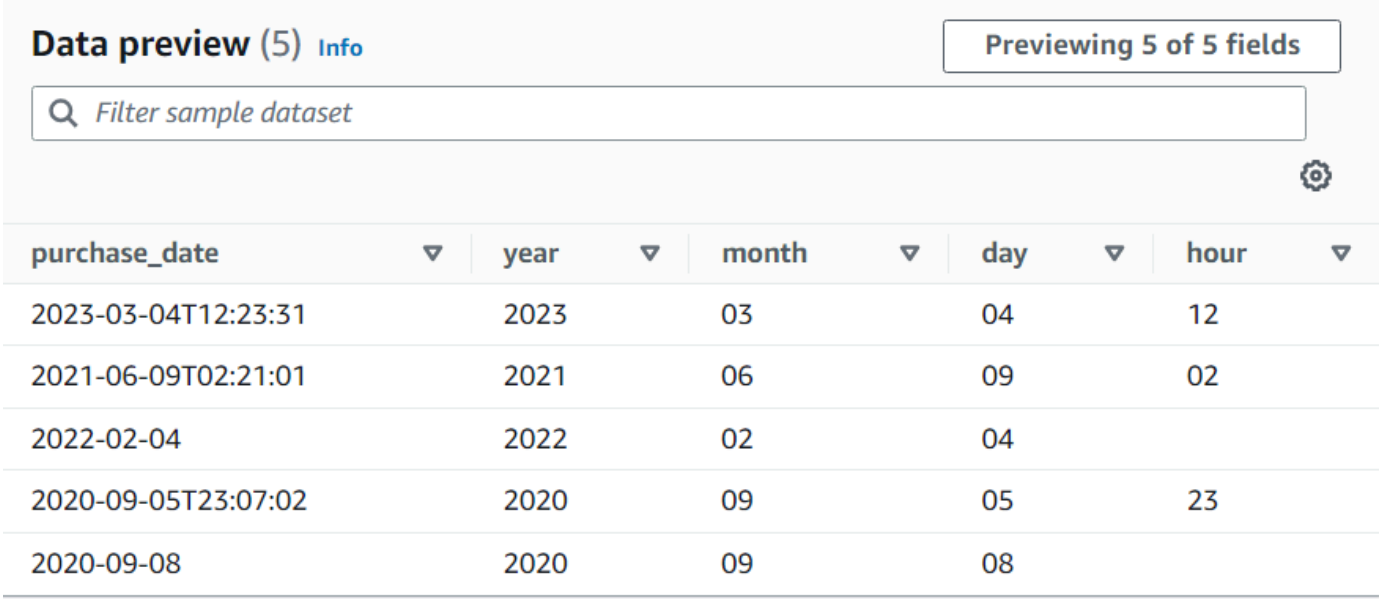
   Se l'espressione regolare utilizza gruppi, esiste un nome di colonna corrispondente separato da una virgola, ma è possibile saltare i gruppi lasciando vuoto il nome della colonna.

   Ad esempio, poniamo che tu abbia una colonna "purchase_date" con una stringa che utilizza formati di data ISO lunghi e brevi e voglia estrarre l'anno, il mese, il giorno e l'ora, se disponibili.

Nota che il gruppo delle ore è facoltativo, altrimenti, nelle righe in cui non è disponibile, tutti i gruppi estratti sarebbero stringhe vuote perché l'espressione regolare non corrisponde. In questo caso, non vogliamo che il gruppo renda facoltativo l'orario ma quello interno, quindi lasciamo il nome vuoto ed esso non verrà estratto (il gruppo includerebbe il carattere T).




Risultato dell'anteprima dei dati:




| purchase_date | year | month | day | hour |
|---|---|---|---|---|
| 2023-03-04T12:23:31 | 2023 | 03 | 04 | 12 |
| 2021-06-09T02:21:01 | 2021 | 06 | 09 | 02 |
| 2022-02-04 | 2022 | 02 | 04 | |
| 2020-09-05T23:07:02 | 2020 | 09 | 05 | 23 |
| 2020-09-08 | 2020 | 09 | 08 | |

## Creazione di una trasformazione personalizzata

Se devi eseguire trasformazioni più complicate sui dati o se vuoi aggiungere chiavi di proprietà dei dati al set di dati, puoi aggiungere una trasformazione Custom code al diagramma di processo. Il nodo Custom code permette di immettere uno script che esegue la trasformazione.

Quando si utilizza il codice personalizzato, è necessario utilizzare un editor di schemi per indicare le modifiche apportate all'output tramite il codice personalizzato. Quando modifichi lo schema, puoi eseguire le seguenti operazioni:

- Aggiungere o rimuovere chiavi di proprietà dei dati
- Modificare il tipo di dati delle chiavi di proprietà dei dati
- Modificare il nome delle chiavi di proprietà dei dati.
- Ristrutturare una chiave di proprietà nidificata

È necessario utilizzare una SelectFromCollectiontrasformazione per sceglierne una `DynamicFrame` dal risultato del nodo di trasformazione personalizzata prima di poter inviare l'output a una posizione di destinazione.

Usa i processi seguenti per aggiungere un nodo di trasformazione personalizzato al diagramma di processo.

Aggiunta di un nodo di trasformazione di codice personalizzato al diagramma di processo

Per aggiungere un nodo di trasformazione personalizzato al diagramma di processo

1. (Facoltativo) Apri il pannello Risorse, quindi scegli Trasformazione personalizzata per aggiungere una nuova trasformazione al diagramma del processo.
2. Nella scheda Node properties (Proprietà del nodo), inserisci un nome per il nodo nel diagramma del processo. Se non è già selezionato un nodo padre, o se desideri più input per la trasformazione personalizzata, scegli un nodo dall'elenco Node parents (Nodi padre) da utilizzare come origine di input per la trasformazione.

Immissione del codice per il nodo di trasformazione personalizzato

Puoi digitare o copiare il codice in un campo di input. Il processo utilizza questo codice per eseguire la trasformazione dei dati. Puoi fornire un frammento di codice in Python o Scala. Il codice richiede uno o più `DynamicFrames` come input e restituisce una raccolta di `DynamicFrames`.

Per inserire lo script per un nodo di trasformazione personalizzato

1. Con il nodo di trasformazione personalizzato selezionato nel diagramma di processo, scegli la casella Transform (Trasformazione).
2. Nel campo di immissione testo sotto l'intestazione Code block (Blocco di codice), incolla o immetti il codice per la trasformazione. Il codice utilizzato deve corrispondere al linguaggio specificato per il processo nella scheda Job details (Dettagli del processo).

   Quando si fa riferimento ai nodi di input nel codice, AWS Glue Studio assegna un nome ai `DynamicFrames` restituiti dai nodi del diagramma del processo in modo sequenziale in base all'ordine di creazione. Utilizza uno dei seguenti metodi di denominazione nel codice:

   - Generazione di codice classico: utilizza i nomi funzionali per fare riferimento ai nodi nel diagramma del processo.
     - Nodi di origine dati: `DataSource0`, `DataSource1`, `DataSource2` e così via.
     - Nodi di trasformazione : `Transform0`, `Transform1`, `Transform2` e così via.
   - Nuova generazione di codice: utilizza il nome specificato nella scheda Node properties (Proprietà del nodo) di un nodo, aggiunta con "_node1","_node2" e così via.
     Ad esempio, `S3bucket_node1`, `ApplyMapping_node2`, `S3bucket_node2`, `MyCustomNodeName_node1`.

Per ulteriori informazioni sul nuovo generatore di codice, consulta Generazione di codice dello script.

Gli esempi seguenti mostrano il formato del codice da inserire nella casella del codice:

Python

L'esempio seguente prende il primo `DynamicFrame` ricevuto, lo converte in un `DataFrame` per applicare il metodo di filtro nativo (conservando solo i registri che hanno più di 1000 voti), quindi prima di restituirlo lo converte di nuovo in un `DynamicFrame`.

```
def FilterHighVoteCounts (glueContext, dfc) -> DynamicFrameCollection:
    df = dfc.select(list(dfc.keys())[0]).toDF()
    df_filtered = df.filter(df["vote_count"] > 1000)
    dyf_filtered = DynamicFrame.fromDF(df_filtered, glueContext, "filter_votes")
    return(DynamicFrameCollection({"CustomTransform0": dyf_filtered}, glueContext))
```

Scala

L'esempio seguente prende il primo `DynamicFrame` ricevuto, lo converte in un `DataFrame` per applicare il metodo di filtro nativo (conservando solo i registri che hanno più di 1000 voti), quindi prima di restituirlo lo converte di nuovo in un `DynamicFrame`.

```
object FilterHighVoteCounts {
  def execute(glueContext : GlueContext, input : Seq[DynamicFrame]) :
 Seq[DynamicFrame] = {
    val frame = input(0).toDF()
    val filtered = DynamicFrame(frame.filter(frame("vote_count") > 1000),
 glueContext)
    Seq(filtered)
  }
}
```

Modifica dello schema in un nodo di trasformazione personalizzato

Quando si utilizza un nodo di trasformazione personalizzato, AWS Glue Studio non può dedurre automaticamente gli schemi di output creati dalla trasformazione. Devi utilizzare l'editor dello schema per descrivere le modifiche allo schema implementate dal codice di trasformazione personalizzato.

Un nodo di codice personalizzato può avere un numero qualsiasi di nodi padre, ognuno dei quali fornisce un `DynamicFrame` come input per il codice personalizzato. Un nodo di codice personalizzato restituisce una raccolta di `DynamicFrames`. Ogni `DynamicFrame` utilizzato come input ha uno schema associato. È necessario aggiungere uno schema che descriva ogni `DynamicFrame` restituito dal nodo di codice personalizzato.

> **Note**
>
> Quando imposti il tuo schema su una trasformazione personalizzata, AWS Glue Studio non eredita schemi dai nodi precedenti. Per aggiornare lo schema, seleziona il nodo di trasformazione personalizzata, quindi sceglie la scheda Data preview (anteprima dati. Una volta generata l'anteprima, scegli "Use Preview Schema" (usa schema di anteprima). Lo schema verrà quindi sostituito dallo schema utilizzando i dati di anteprima.

Per modificare gli schemi per un nodo di trasformazione personalizzato

1. Con il nodo di trasformazione personalizzato selezionato nel diagramma di processo, scegli la scheda Output schema (Schema di output) nel pannello dei dettagli del nodo.
2. Scegli Edit (Modifica) per apportare modifiche allo schema.

   Se disponi di chiavi di proprietà dei dati nidificate, ad esempio una matrice o un oggetto, puoi scegliere l'icona Expand-Rows (Espandi righe) ( ) in alto a destra di ogni pannello dello schema per espandere l'elenco delle chiavi di proprietà dei dati figlio. Dopo aver selezionato l'icona, questa si trasforma nell'icona Collapse-Rows (Comprimi righe) ( ), che puoi selezionare per comprimere l'elenco delle chiavi di proprietà figlio.
3. Modifica lo schema utilizzando le seguenti operazioni nella sezione a destra della pagina:

   - Per rinominare una chiave di proprietà, posiziona il cursore nella casella di testo Key (Chiave) per la chiave di proprietà, quindi immetti il nuovo nome.
   - Per modificare il tipo di dati per una chiave di proprietà dei dati, usa l'elenco per scegliere il nuovo tipo di dati per la chiave di proprietà.
   - Per aggiungere una nuova chiave di proprietà di livello superiore allo schema, scegli l'opzione Overflow

( ⋯ )
sulla sinistra del pulsante Cancel (Annulla), quindi scegli Add root key (Aggiungi chiave root).

- Per aggiungere una chiave di proprietà figlio allo schema, scegli l'icona Add-Key (Aggiungi chiave)
  associata alla chiave padre. Inserisci un nome per la chiave figlio e scegli il tipo di dati.
- Per rimuovere una colonna dallo schema, scegli l'icona Remove (Elimina)
  ( )
  all'estrema destra del nome della chiave.

4. Se il codice di trasformazione personalizzato utilizza più `DynamicFrames`, è possibile aggiungere schemi di output aggiuntivi.

   - Per aggiungere un nuovo schema vuoto, scegli l'icona Overflow
     ( ⋯ ),
     quindi scegli Add output schema (Aggiungi schema di output).
   - Per copiare uno schema esistente in un nuovo schema di output, assicurati che lo schema da copiare sia visualizzato nel selettore dello schema. Seleziona l'icona Overflow
     ( ⋯ ),
     quindi scegli Duplicate (Duplica).

   Se vuoi rimuovere uno schema di output, assicurati che lo schema da copiare sia visualizzato nel selettore dello schema. Seleziona l'icona Overflow
   ( ⋯ ),
   quindi scegli Delete (Elimina).
5. Aggiungi nuove chiavi radice al nuovo schema o modifica le chiavi duplicate.
6. Quando modifichi gli lo schemi di output, scegli il pulsante Apply (Applica) per salvare le modifiche e uscire dall'editor dello schema.

   Se non vuoi salvare le modifiche, seleziona il pulsante Cancel (Annulla).

## Configurare l'output della trasformazione personalizzata

Una trasformazione di codice personalizzata restituisce una raccolta di `DynamicFrames`, anche se nel set di risultati è presente solo un `DynamicFrame`.

Per elaborare l'output da un nodo di trasformazione personalizzato

1. Aggiungi un nodo di SelectFromCollectiontrasformazione, che ha il nodo di trasformazione personalizzato come nodo principale. Aggiorna questa trasformazione per indicare il set di dati da utilizzare. Per ulteriori informazioni, consulta Utilizzo SelectFromCollection per scegliere quale set di dati conservare.
2. Aggiungi SelectFromCollectiontrasformazioni aggiuntive al diagramma del lavoro se desideri utilizzarne altre `DynamicFrames` prodotte dal nodo di trasformazione personalizzato.

   Consideriamo uno scenario in cui aggiungi un nodo di trasformazione personalizzato per dividere un set di dati di volo in più set di dati, ma duplichi alcune delle chiavi di proprietà identificative in ciascuno schema di output, ad esempio la data di volo o il numero di volo. Si aggiunge un nodo di SelectFromCollectiontrasformazione per ogni schema di output, con il nodo di trasformazione personalizzato come elemento principale.
3. (Facoltativo) È quindi possibile utilizzare ogni nodo di SelectFromCollectiontrasformazione come input per altri nodi del processo o come genitore per un nodo di destinazione dei dati.

## Trasforma i dati con le trasformazioni visive personalizzate

Le trasformazioni visive personalizzate consentono di creare trasformazioni e renderle disponibili per l'uso nei processi AWS Glue Studio. Le trasformazioni visive personalizzate consentono agli sviluppatori ETL, che potrebbero non avere familiarità con la codifica, di cercare e utilizzare una libreria di trasformazioni in via di sviluppo tramite l'interfaccia AWS Glue Studio.

Puoi creare una trasformazione visiva personalizzata e caricarla su Amazon S3 per renderla disponibile all'uso tramite l'editor visivo in AWS Glue Studio e lavorare con questi processi.

Argomenti

- Nozioni di base sulle trasformazioni visive personalizzate
- Passaggio 1. Creazione di un file di configurazione JSON
- Passaggio 2. Implementazione della logica di trasformazione
- Fase 3. Convalidazione e risoluzione dei problemi delle trasformazioni visive personalizzate in AWS Glue Studio
- Passaggio 4. Aggiornamento delle trasformazioni visive personalizzate in base alle necessità
- Fase 5. Utilizzo della trasformazione visiva personalizzata in AWS Glue Studio
- Esempi di utilizzo

- Esempi di script visivi personalizzati
- Video

## Nozioni di base sulle trasformazioni visive personalizzate

Per creare una trasformazione visiva personalizzata, completa la seguente procedura.

- Passaggio 1. Creazione di un file di configurazione JSON
- Passaggio 2. Implementazione della logica di trasformazione
- Fase 3. Convalida della trasformazione visiva personalizzata
- Passaggio 4. Aggiornamento della trasformazione visiva personalizzata in base alle necessità
- Fase 5. Utilizzo della trasformazione visiva personalizzata in AWS Glue Studio

Inizia configurando il bucket Amazon S3 e continua con la Fase 1. Crea un file di configurazione JSON.

Prerequisiti

Le trasformazioni fornite dal cliente risiedono all'interno di un account cliente. AWS Quell'account possiede le trasformazioni e quindi dispone di tutte le autorizzazioni per visualizzarle (ricerca e uso), modificarle o eliminarle.

Per utilizzare una trasformazione personalizzata inAWS Glue Studio, dovrai creare e caricare due file nel bucket di asset Amazon S3 di quell'account: AWS

- File Python: contiene la funzione di trasformazione.
- File JSON: descrive la trasformazione. Questo è noto anche come file di configurazione necessario per definire la trasformazione.

Per accoppiare i file, utilizzate lo stesso nome per entrambi. Esempio:

- myTransform.json
- myTransform.py

Facoltativamente, puoi assegnare alla tua trasformazione visiva personalizzata un'icona personalizzata fornendo un file SVG contenente l'icona. Per accoppiare i file, utilizza lo stesso nome per l'icona:

- myTransform.svg

AWS Glue Studio li abbinerà automaticamente utilizzando i rispettivi nomi dei file. I nomi dei file non possono essere uguali per nessun modulo esistente.

Convenzione consigliata per il nome del file della trasformazione

AWS Glue Studio importerà il file come modulo (ad esempio, `import myTransform`) nello script del processo. Pertanto, il nome del file deve seguire le stesse regole di denominazione impostate per i nomi delle variabili python (identificatori). Nello specifico, devono iniziare con una lettera o un carattere di sottolineatura e poi essere composti interamente da lettere, cifre e trattini bassi. and/or

> **Note**
>
> Assicurati che il nome del file di trasformazione non sia in conflitto con i moduli python caricati esistenti (ad esempio, `sys, array, copy` ecc.) per evitare problemi di runtime imprevisti.

Configurazione del bucket Amazon S3

Le trasformazioni che crei vengono archiviate in Amazon S3 e sono di proprietà del tuo account. AWS È possibile creare nuove trasformazioni visive personalizzate semplicemente caricando i file (.json e .py) nella cartella delle risorse di Amazon S3 dove sono correntemente archiviati tutti gli script del processo (ad esempio, `s3://aws-glue-assets-<accountid>-<region>/transforms`). Se utilizzi un'icona personalizzata, carica anch'essa. Per impostazione predefinita, AWS Glue Studio leggerà tutti i file .json dalla cartella /transforms nello stesso bucket S3.

## Passaggio 1. Creazione di un file di configurazione JSON

Un file di configurazione JSON è necessario per definire e descrivere la trasformazione visiva personalizzata. Lo schema per il file di configurazione è il seguente.

Struttura dei file JSON

Campi

- `name: string`: (obbligatorio) il nome del sistema di trasformazione utilizzato per identificare le trasformazioni. Segui le stesse regole di denominazione impostate per i nomi delle variabili python (identificatori). In particolare, devono iniziare con una lettera o un carattere di sottolineatura e quindi essere composti interamente da lettere, cifre, and/or trattini bassi.
- `displayName: string`: (facoltativo) il nome della trasformazione visualizzata nell'editor di processi visivi di AWS Glue Studio. Se non viene specificato alcun `displayName`, come nome della trasformazione in AWS Glue Studio viene utilizzato `name`.
- `description: string`: (facoltativo) la descrizione della trasformazione viene visualizzata in AWS Glue Studio ed è ricercabile.
- `functionName: string`: (obbligatorio) il nome della funzione Python viene utilizzato per identificare la funzione da chiamare nello script Python.
- `path: string`: (facoltativo) il percorso completo di Amazon S3 del file sorgente di Python. Se non viene specificato, AWS Glue utilizza il nome del file corrispondente per accoppiare i file .json e .py. Ad esempio, il nome del file JSON, `myTransform.json`, verrà associato al file Python, `myTransform.py`, nella stessa posizione di Amazon S3.
- `parameters: Array of TransformParameter object`: (facoltativo) l'elenco dei parametri da visualizzare quando li si configura nell'editor visivo di AWS Glue Studio.

TransformParameter campi

- `name: string`: (obbligatorio) il nome del parametro che verrà passato alla funzione python come argomento denominato nello script del processo. Segui le stesse regole di denominazione impostate per i nomi delle variabili python (identificatori). In particolare, devono iniziare con una lettera o un carattere di sottolineatura e quindi essere composti interamente da lettere, and/or cifre e caratteri di sottolineatura.
- `displayName: string`: (facoltativo) il nome della trasformazione visualizzata nell'editor di processi visivi di AWS Glue Studio. Se non viene specificato alcun `displayName`, come nome della trasformazione in AWS Glue Studio viene utilizzato `name`.
- `type: string`: (obbligatorio) il tipo di parametro che accetta i tipi di dati Python comuni. I valori validi sono 'str' | 'int' | 'float' | 'list' | 'bool'.
- `isOptional: boolean`: (facoltativo) determina se il parametro è facoltativo. Per impostazione predefinita, tutti i parametri sono obbligatori.
- `description: string`: (facoltativo) viene visualizzata una descrizione in AWS Glue Studio per aiutare l'utente a configurare il parametro di trasformazione.

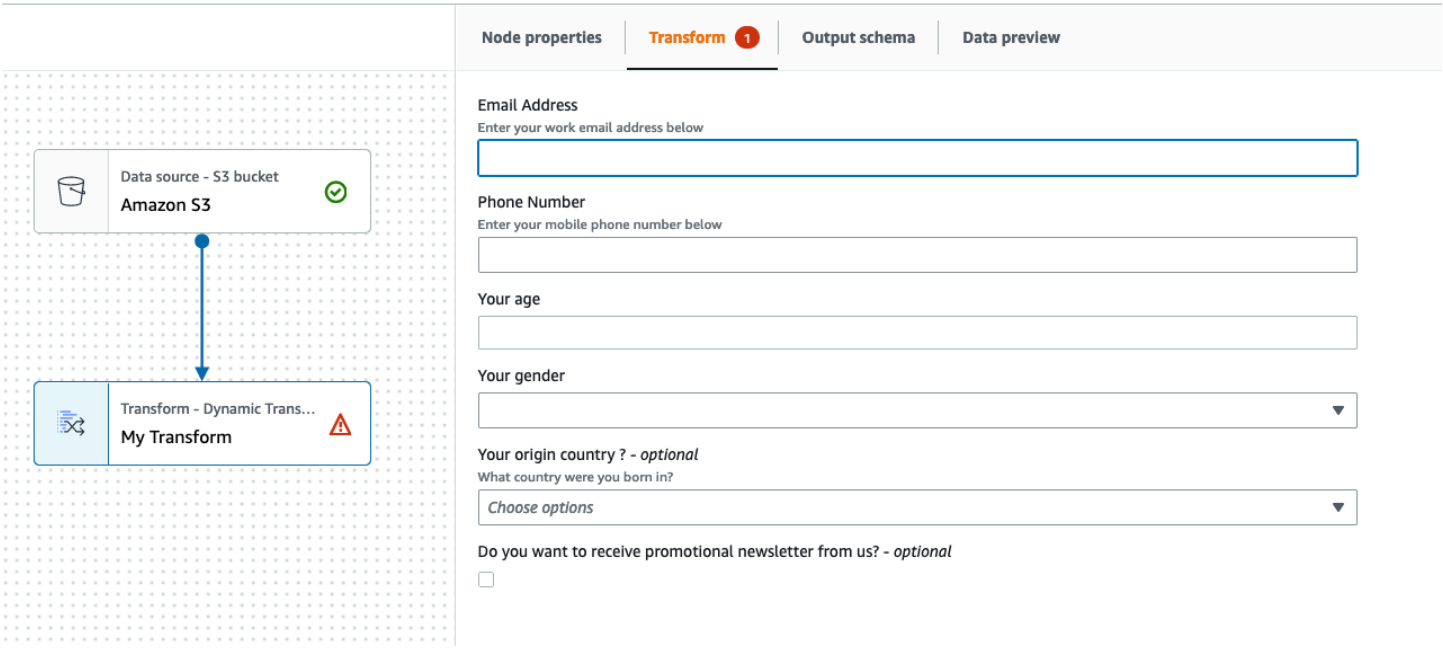
- `validationType: string`: (facoltativo) definisce il modo in cui questo parametro viene convalidato. Al momento, supporta solo le espressioni regolari. Per impostazione predefinita, il tipo di convalida è impostato su `RegularExpression`.
- `validationRule: string`: (facoltativo) l'espressione regolare utilizzata per convalidare l'input del modulo prima dell'invio quando `validationType` è impostato su `RegularExpression`. La sintassi delle espressioni regolari deve essere compatibile con le specifiche di Ecmascript. RegExp
- `validationMessage: string`: (facoltativo) il messaggio da visualizzare quando la convalida non riesce.
- `listOptions: An array of TransformParameterListOption object` OPPURE una `string` o il valore di stringa "column": (facoltativo) le opzioni da visualizzare nel controllo Select o Multiselect dell'interfaccia utente. Accettazione di un elenco di valori separati da virgole o di un oggetto JSON fortemente tipizzato di tipo `TransformParameterListOption`. Può anche compilare dinamicamente l'elenco di colonne dello schema del nodo padre specificando il valore di stringa "column".
- `listType: string`: (facoltativo) definisci i tipi di opzioni per type = 'list'. I valori validi sono 'str' | 'int' | 'float' | 'list' | 'bool'. Tipo di parametro che accetta i tipi di dati Python comuni.

TransformParameterListOption campi

- `value: string | int | float | bool`: (obbligatorio) il valore dell'opzione.
- `label: string` (facoltativo) l'etichetta dell'opzione visualizzata nel menu a discesa di selezione.

Trasformazione dei parametri in AWS Glue Studio

Per impostazione predefinita, i parametri sono obbligatori a meno che non siano contrassegnati come `isOptional` nel file .json. In AWS Glue Studio, i parametri vengono visualizzati nella scheda Transform (Trasforma). L'esempio mostra parametri definiti dall'utente come indirizzo e-mail, numero di telefono, età, sesso e paese di origine.

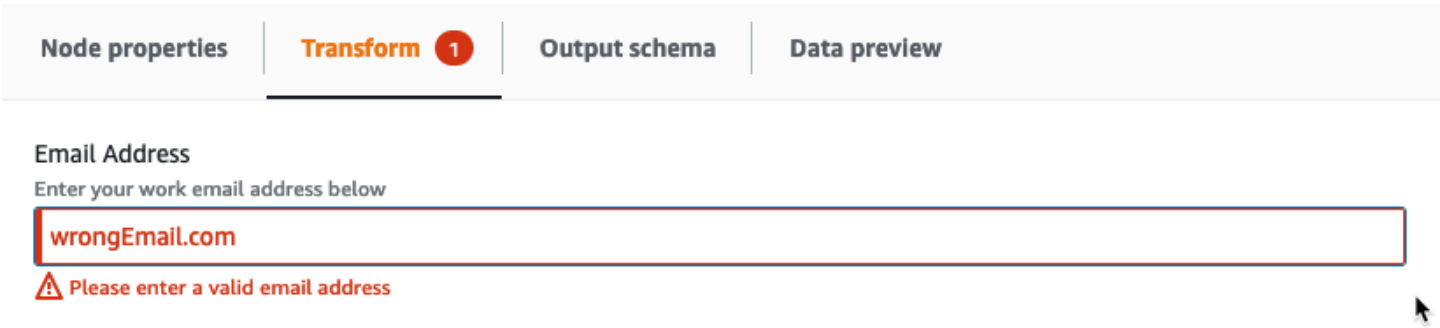



È possibile applicare alcune convalide in AWS Glue Studio utilizzando le espressioni regolari nel file .json specificando il parametro `validationRule` e specificando un messaggio di convalida in `validationMessage`.

```
    "validationRule": "^\\(?(\\d{3})\\)?[- ]?(\\d{3})[- ]?(\\d{4})$",
    "validationMessage": "Please enter a valid US number"
```

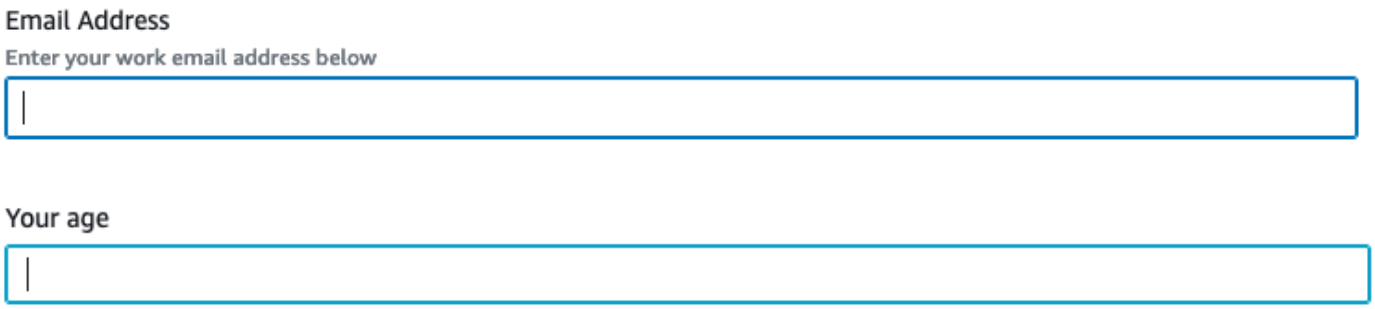
> **Note**
>
> Poiché la convalida avviene nel browser, la sintassi delle espressioni regolari deve essere compatibile con le specifiche di RegExp Ecmascript. La sintassi di Python non è supportata per queste espressioni regolari.

L'aggiunta della convalida impedirà all'utente di salvare il processo con un input utente errato. AWS Glue Studio visualizza il messaggio di convalida come mostrato nell'esempio:

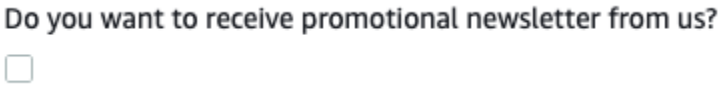



I parametri vengono visualizzati in AWS Glue Studio in base alla loro configurazione.

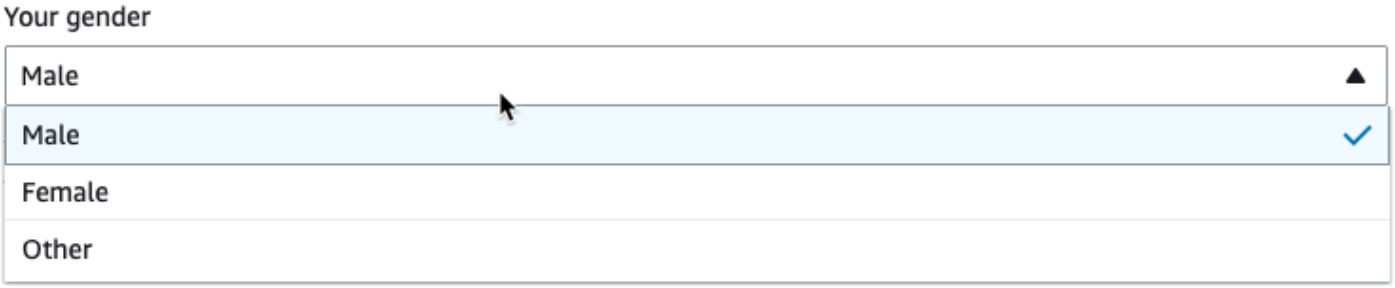
- Quando `type` è `str`, `int` o `float`, viene visualizzato un campo di immissione di testo. Ad esempio, la schermata mostra i campi di input per i parametri "Indirizzo e-mail" ed "Età".



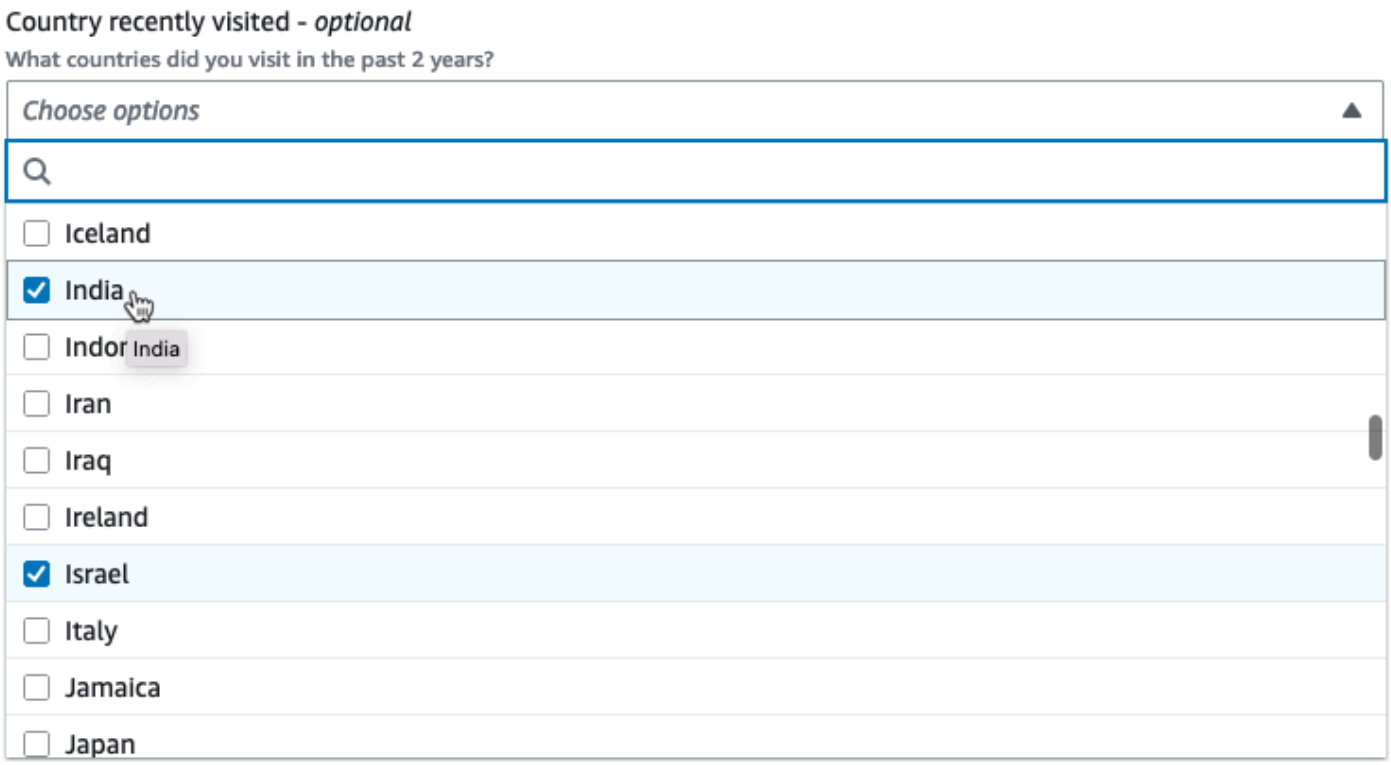

- Quando `type` è `bool`, viene visualizzata una casella di controllo.




- Quando `type` è `str` e viene fornito `listOptions`, viene visualizzato un unico elenco di selezione.



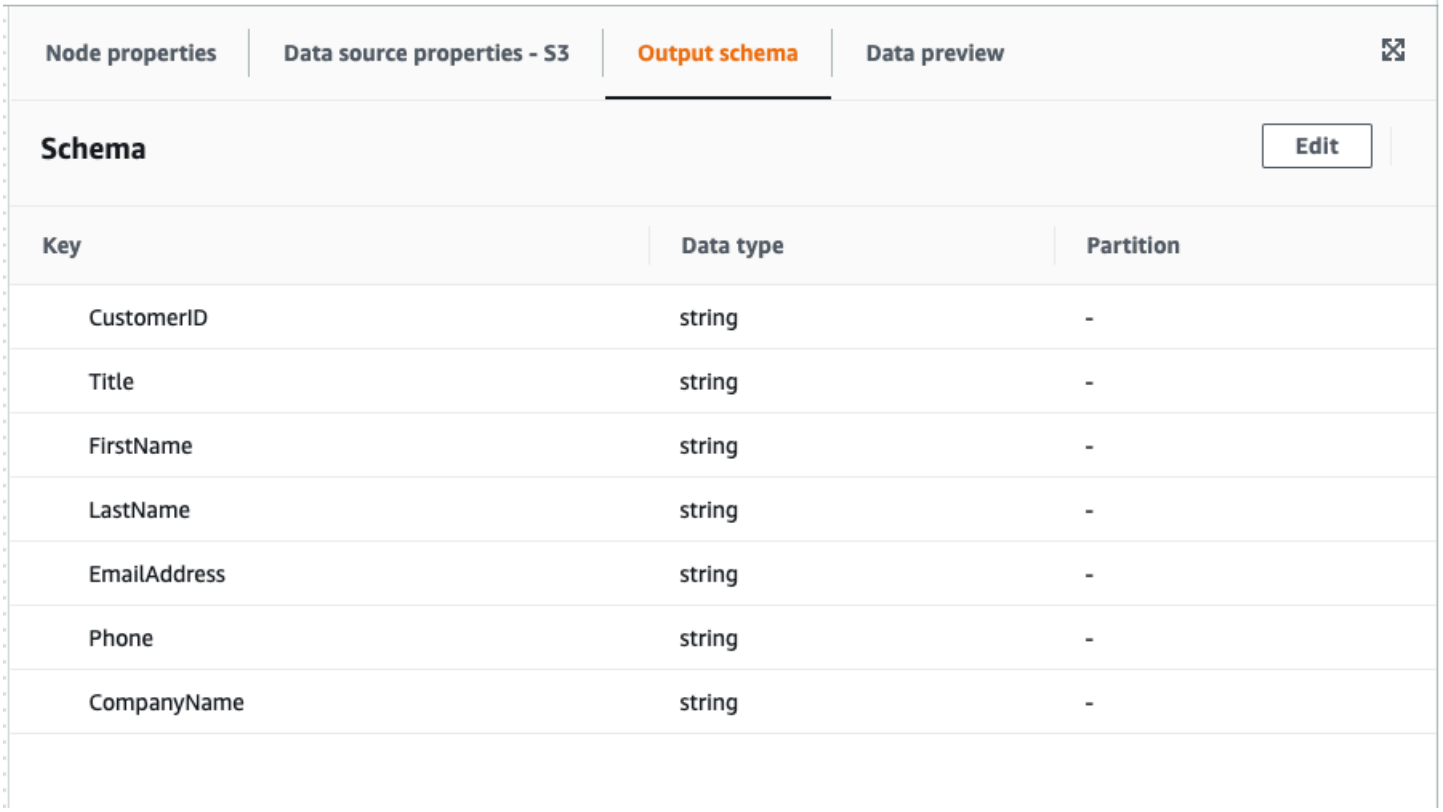

- Quando `type` è `list` e sono forniti `listOptions` e `listType`, viene visualizzato un elenco a selezione multipla.




Visualizzazione di un selettore di colonna come parametro

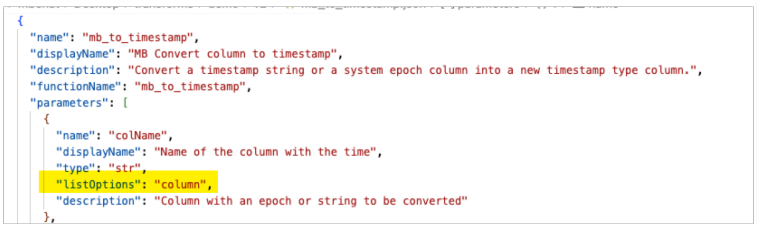
Se la configurazione richiede all'utente di scegliere una colonna dallo schema, è possibile visualizzare un selettore di colonna in modo che l'utente non debba digitarne il nome. Impostando il campo `listOptions` su "column", AWS Glue Studio visualizza dinamicamente un selettore di colonne basato sullo schema di output del nodo principale. AWS Glue Studio può visualizzare un selettore a colonna singola o multipla.

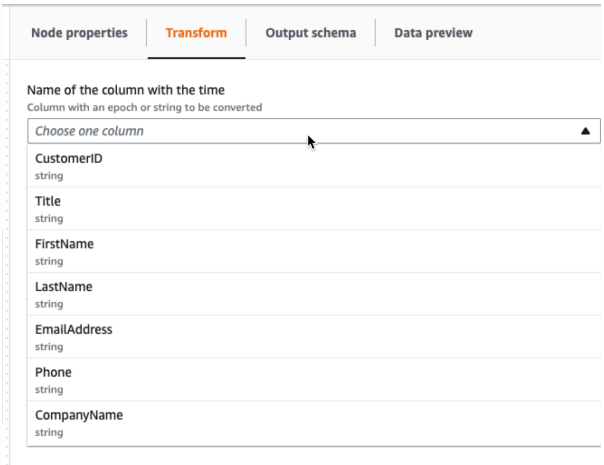
L'esempio seguente utilizza lo schema:




Per definire il parametro Trasformazione visiva personalizzata per visualizzare una singola colonna:

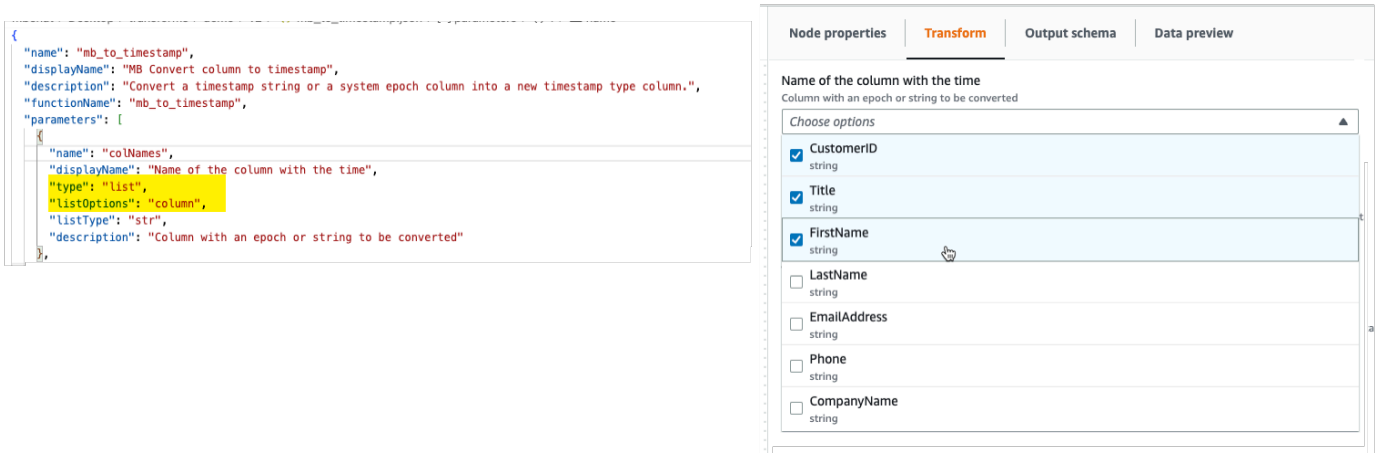
1. Nel file JSON, per l'oggetto `parameters`, imposta il valore di `listOptions` su "column". Ciò consente a un utente di scegliere una colonna da un elenco di selezione in AWS Glue Studio.




2. È inoltre possibile consentire la selezione di più colonne definendo il parametro come:

   - `listOptions: "column"`
   - `type: "list"`




## Passaggio 2. Implementazione della logica di trasformazione

> **Note**
>
> Le trasformazioni visive personalizzate supportano solo gli script Python. Scala non è supportato.

Per aggiungere il codice che implementa la funzione definita dal file di configurazione .json, si consiglia di posizionare il file Python nella stessa posizione del file .json, con lo stesso nome ma con l'estensione ".py". AWS Glue Studio accoppia automaticamente i file .json e .py in modo che non sia necessario specificare il percorso del file Python nel file di configurazione.

Nel file Python, aggiungi la funzione dichiarata, con i parametri denominati configurati e registrala per l'uso in `DynamicFrame`. Di seguito è riportato un esempio di un file Python:

```
from awsglue import DynamicFrame

# self refers to the DynamicFrame to transform,
# the parameter names must match the ones defined in the config
# if it's optional, need to provide a default value
def myTransform(self, email, phone, age=None, gender="",
                      country="", promotion=False):
   resulting_dynf = # do some transformation on self
   return resulting_dynf

DynamicFrame.myTransform = myTransform
```

Si consiglia di utilizzare un notebook AWS Glue per sviluppare e testare il codice python in modo più rapido. consulta, Nozioni di base sui notebook in AWS Glue Studio.

Per illustrare come implementare la logica della trasformazione, la trasformazione visiva personalizzata nell'esempio seguente è una trasformazione per filtrare i dati in entrata e conservare solo i dati relativi a uno specifico stato degli Stati Uniti. Il file .json contiene il parametro per `functionName` come `custom_filter_state` e due argomenti ("state" e "colName" con tipo "str").

Il file di configurazione .json di esempio è:

```
{
"name": "custom_filter_state",
"displayName": "Filter State",
"description": "A simple example to filter the data to keep only the state indicated.",
"functionName": "custom_filter_state",
"parameters": [
   {
    "name": "colName",
    "displayName": "Column name",
    "type": "str",
    "description": "Name of the column in the data that holds the state postal code"
   },
   {
    "name": "state",
    "displayName": "State postal code",
    "type": "str",
    "description": "The postal code of the state whole rows to keep"
   }
  ]
}
```

Implementazione dello script complementare in Python

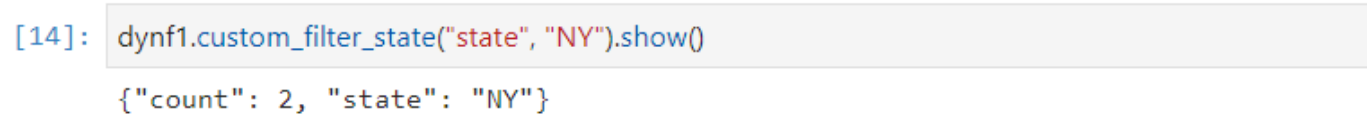
1. Avvia un notebook AWS Glue ed esegui la cella iniziale fornita per l'avvio della sessione. L'esecuzione della cella iniziale crea i componenti di base necessari.
2. Crea una funzione che esegua il filtraggio come descritto nell'esempio e registrala in `DynamicFrame`. Copia il codice seguente e incollalo in una cella del notebook AWS Glue.

```
from awsglue import DynamicFrame

def custom_filter_state(self, colName, state):
    return self.filter(lambda row: row[colName] == state)

DynamicFrame.custom_filter_state = custom_filter_state
```

3. Crea o carica dati di esempio per testare il codice nella stessa cella o in una nuova cella. Se aggiungi i dati di esempio in una nuova cella, non dimenticare di eseguire la cella. Esempio:

```
# A few of rows of sample data to test
data_sample = [
    {"state": "CA", "count": 4},
    {"state": "NY", "count": 2},
    {"state": "WA", "count": 3}
]
df1 = glueContext.sparkSession.sparkContext.parallelize(data_sample).toDF()
dynf1 = DynamicFrame.fromDF(df1, glueContext, None)
```

4. Esegui test per convalidare "custom_filter_state" con diversi argomenti:




5. Dopo aver eseguito diversi test, salva il codice con l'estensione .py e assegna al file .py un nome che rispecchi il nome del file .json. I file .py e .json devono trovarsi nella stessa cartella di trasformazione.

   Copia il codice seguente e incollalo in un file, quindi rinominalo con l'estensione .py.

```
from awsglue import DynamicFrame

def custom_filter_state(self, colName, state):
    return self.filter(lambda row: row[colName] == state)

DynamicFrame.custom_filter_state = custom_filter_state
```

6. In AWS Glue Studio, apri un processo visivo e aggiungi una trasformazione selezionandola tra quelle disponibili nell'elenco Transforms (Trasformazioni).

   Per riutilizzare questa trasformazione in un codice script Python, aggiungi il percorso Amazon S3 al file .py nel processo nel "Percorso dei file di riferimento" e nello script, importa il nome del file python (senza estensione) aggiungendolo all'inizio del file. Ad esempio: `import` <nome del file (senza estensione)>

## Fase 3. Convalidazione e risoluzione dei problemi delle trasformazioni visive personalizzate in AWS Glue Studio

AWS Glue Studio convalida il file di configurazione JSON prima che le trasformazioni visive personalizzate vengano caricate in AWS Glue Studio. La convalida include:

- Presenza di campi obbligatori
- Convalida del formato JSON
- Parametri non corretti o non validi
- Presenza dei file .py e .json nello stesso percorso Amazon S3
- Nomi di file corrispondenti per i file .py e .json

Se la convalida ha esito positivo, la trasformazione viene riportata tra quelle disponibili nell'elenco Actions (Operazioni) nell'editor visivo. Se è stata fornita un'icona personalizzata, dovrebbe essere visibile accanto a Operazione.

Se la convalida non riesce, AWS Glue Studio non carica la trasformazione visiva personalizzata.

## Passaggio 4. Aggiornamento delle trasformazioni visive personalizzate in base alle necessità

Una volta creato e utilizzato, lo script di trasformazione può essere aggiornato purché la trasformazione segua la definizione json corrispondente:

- Il nome usato quando si assegna DynamicFrame troppo corrisponde al `functionName` json.
- Gli argomenti della funzione devono essere definiti nel file .json come descritto in Passaggio 1. Creazione di un file di configurazione JSON.
- Il percorso Amazon S3 del file Python non può cambiare, poiché i processi dipendono direttamente da esso.

> **Note**
>
> Se è necessario apportare aggiornamenti, assicurati che lo script e il file .json siano costantemente aggiornati e che tutti i processi visivi vengano nuovamente salvati correttamente con la nuova trasformazione. Se i processi visivi non vengono salvati dopo aver effettuato gli aggiornamenti, gli aggiornamenti non saranno applicati e convalidati. Se il file di script Python viene rinominato o non posizionato insieme al file .json, è necessario specificare il percorso completo nel file .json.

Icona personalizzata

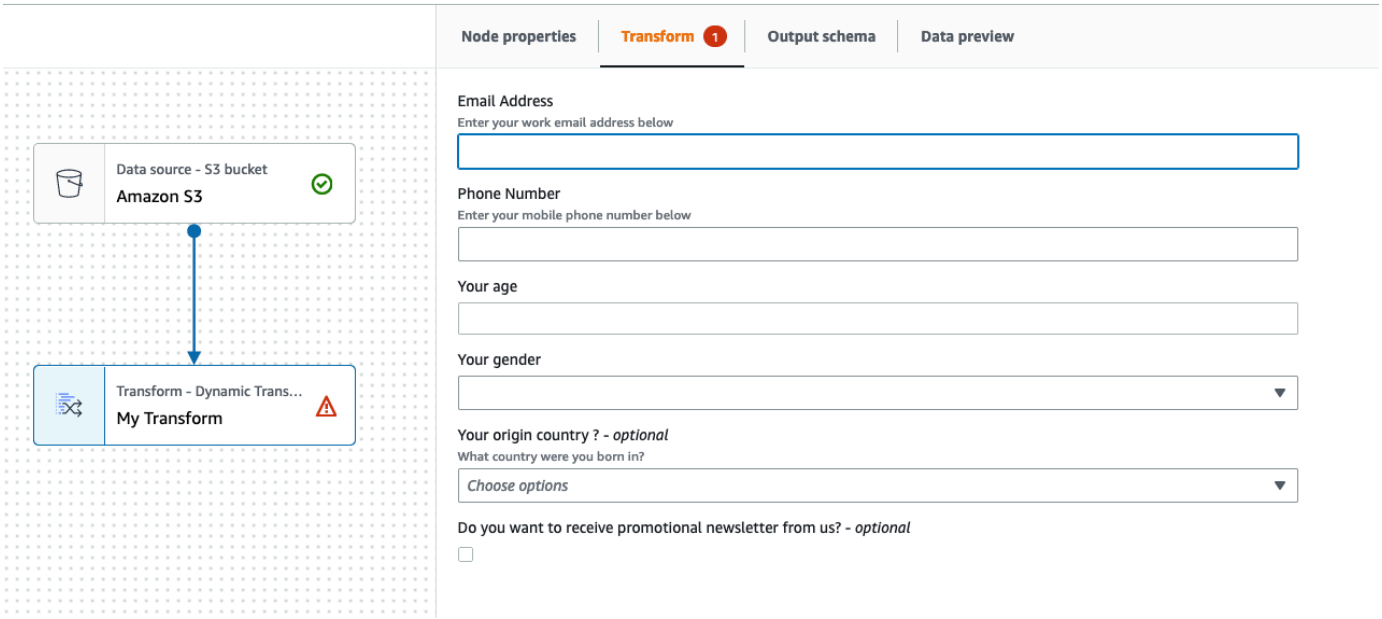
Se ritieni che l'icona predefinita per l'Operazione non la distingue visivamente come parte dei flussi di lavoro, puoi fornire un'icona personalizzata, come descritto nella sezione the section called "Nozioni di base sulle trasformazioni visive personalizzate". È possibile aggiornare l'icona aggiornando il file SVG corrispondente ospitato in Amazon S3.

Per ottenere risultati ottimali, progetta l'immagine in modo che venga visualizzata a 32x32 pixel seguendo le linee guida del Cloudscape Design System. Per ulteriori informazioni sulle linee guida di Cloudscape, consulta la documentazione di Cloudscape.

## Fase 5. Utilizzo della trasformazione visiva personalizzata in AWS Glue Studio

Per utilizzare una trasformazione visiva personalizzata in AWS Glue Studio, si devono caricare i file di configurazione e di origine, quindi selezionare la trasformazione dal menu Action (Operazione). Tutti i parametri che richiedono valori o un input sono disponibili nella scheda Transform (Trasforma).

1. Carica i due file (file di origine Python e file di configurazione JSON) nella cartella delle risorse di Amazon S3 in cui sono archiviati gli script di processo. Per impostazione predefinita, AWS Glue estrarrà tutti i file JSON dalla cartella /transforms nello stesso bucket Amazon S3.
2. Dal menu Action (Operazione), scegli la trasformazione visiva personalizzata. Viene denominato con la trasformazione `displayName` o il nome specificato nel file di configurazione .json.
3. Immetti i valori per tutti i parametri configurati nel file di configurazione.




## Esempi di utilizzo

Di seguito è riportato un esempio di tutti i parametri possibili in un file di configurazione .json.

```
{
  "name": "MyTransform",
  "displayName": "My Transform",
  "description": "This transform description will be displayed in UI",
  "functionName": "myTransform",
  "parameters": [
      {
      "name": "email",
      "displayName": "Email Address",
      "type": "str",
      "description": "Enter your work email address below",
      "validationType": "RegularExpression",
      "validationRule": "^\\w+([\\.-]?\\w+)*@\\w+([\\.-]?\\w+)*(\\.\\w{2,3})+$",
      "validationMessage": "Please enter a valid email address"
    },
    {
      "name": "phone",
      "displayName": "Phone Number",
      "type": "str",
      "description": "Enter your mobile phone number below",
      "validationRule": "^\\(?(\\d{3})\\)?[- ]?(\\d{3})[- ]?(\\d{4})$",
      "validationMessage": "Please enter a valid US number"
```

```
    },
    {
      "name": "age",
      "displayName": "Your age",
      "type": "int",
      "isOptional": true
    },
    {
      "name": "gender",
      "displayName": "Your gender",
      "type": "str",
      "listOptions": [
            {"label": "Male", "value": "male"},
            {"label": "Female", "value": "female"},
            {"label": "Other", "value": "other"}
        ],
      "isOptional": true
    },
    {
      "name": "country",
      "displayName": "Your origin country ?",
      "type": "list",
      "listOptions": "Afghanistan,Albania,Algeria,American
 Samoa,Andorra,Angola,Anguilla,Antarctica,Antigua and
 Barbuda,Argentina,Armenia,Aruba,Australia,Austria,Azerbaijan,Bahamas,Bahrain,Bangladesh,Barbad
 and Herzegovina,Botswana,Bouvet Island,Brazil,British Indian Ocean Territory,Brunei
 Darussalam,Bulgaria,Burkina Faso,Burundi,Cambodia,Cameroon,Canada,Cape
 Verde,Cayman Islands,Central African Republic,Chad,Chile,China,Christmas
 Island,Cocos (Keeling Islands),Colombia,Comoros,Congo,Cook Islands,Costa
 Rica,Cote D'Ivoire (Ivory Coast),Croatia (Hrvatska,Cuba,Cyprus,Czech
 Republic,Denmark,Djibouti,Dominica,Dominican Republic,East Timor,Ecuador,Egypt,El
 Salvador,Equatorial Guinea,Eritrea,Estonia,Ethiopia,Falkland Islands (Malvinas),Faroe
 Islands,Fiji,Finland,France,France,Metropolitan,French Guiana,French Polynesia,French
 Southern
 Territories,Gabon,Gambia,Georgia,Germany,Ghana,Gibraltar,Greece,Greenland,Grenada,Guadeloupe,G
Bissau,Guyana,Haiti,Heard and McDonald Islands,Honduras,Hong
 Kong,Hungary,Iceland,India,Indonesia,Iran,Iraq,Ireland,Israel,Italy,Jamaica,Japan,Jordan,Kazak
 (North),Korea
 (South),Kuwait,Kyrgyzstan,Laos,Latvia,Lebanon,Lesotho,Liberia,Libya,Liechtenstein,Lithuania,Lu
 Islands,Martinique,Mauritania,Mauritius,Mayotte,Mexico,Micronesia,Moldova,Monaco,Mongolia,Mont
 Antilles,New Caledonia,New Zealand,Nicaragua,Niger,Nigeria,Niue,Norfolk
 Island,Northern Mariana Islands,Norway,Oman,Pakistan,Palau,Panama,Papua
 New Guinea,Paraguay,Peru,Philippines,Pitcairn,Poland,Portugal,Puerto
 Rico,Qatar,Reunion,Romania,Russian Federation,Rwanda,Saint Kitts and Nevis,Saint
```

```
 Lucia,Saint Vincent and The Grenadines,Samoa,San Marino,Sao Tome and Principe,Saudi
 Arabia,Senegal,Seychelles,Sierra Leone,Singapore,Slovak Republic,Slovenia,Solomon
 Islands,Somalia,South Africa,S. Georgia and S. Sandwich Isls.,Spain,Sri
 Lanka,St. Helena,St. Pierre and Miquelon,Sudan,Suriname,Svalbard and Jan Mayen
 Islands,Swaziland,Sweden,Switzerland,Syria,Tajikistan,Tanzania,Thailand,Togo,Tokelau,Tonga,Tri
 and Tobago,Tunisia,Turkey,Turkmenistan,Turks and Caicos
 Islands,Tuvalu,Uganda,Ukraine,United Arab Emirates,United Kingdom
 (Britain / UK),United States of America (USA),US Minor Outlying
 Islands,Uruguay,Uzbekistan,Vanuatu,Vatican City State (Holy See),Venezuela,Viet
 Nam,Virgin Islands (British),Virgin Islands (US),Wallis and Futuna Islands,Western
 Sahara,Yemen,Yugoslavia,Zaire,Zambia,Zimbabwe",
      "description": "What country were you born in?",
      "listType": "str",
      "isOptional": true
    },
    {
      "name": "promotion",
      "displayName": "Do you want to receive promotional newsletter from us?",
      "type": "bool",
      "isOptional": true
    }
  ]
}
```

## Esempi di script visivi personalizzati

Gli esempi seguenti eseguono trasformazioni equivalenti. Tuttavia, il secondo esempio (SparkSQL) è il più pulito ed efficiente, seguito dall'UDF di pandas e infine dalla mappatura a basso livello nel primo esempio. L'esempio seguente è un esempio completo di una semplice trasformazione per sommare due colonne:

```
from awsglue import DynamicFrame

# You can have other auxiliary variables, functions or classes on this file, it won't
 affect the runtime
def record_sum(rec, col1, col2, resultCol):
    rec[resultCol] = rec[col1] + rec[col2]
    return rec


# The number and name of arguments must match the definition on json config file
# (expect self which is the current DynamicFrame to transform
```

```
# If an argument is optional, you need to define a default value here
#  (resultCol in this example is an optional argument)
def custom_add_columns(self, col1, col2, resultCol="result"):
    # The mapping will alter the columns order, which could be important
    fields = [field.name for field in self.schema()]
    if resultCol not in fields:
        # If it's a new column put it at the end
        fields.append(resultCol)
    return self.map(lambda record: record_sum(record, col1, col2,
 resultCol)).select_fields(paths=fields)


# The name we assign on DynamicFrame must match the configured "functionName"
DynamicFrame.custom_add_columns = custom_add_columns
```

L'esempio seguente è una trasformazione equivalente che sfrutta l'API SparkSQL.

```
from awsglue import DynamicFrame

# The number and name of arguments must match the definition on json config file
# (expect self which is the current DynamicFrame to transform
# If an argument is optional, you need to define a default value here
#  (resultCol in this example is an optional argument)
def custom_add_columns(self, col1, col2, resultCol="result"):
    df = self.toDF()
    return DynamicFrame.fromDF(
        df.withColumn(resultCol, df[col1] + df[col2]) # This is the conversion logic
        , self.glue_ctx, self.name)


# The name we assign on DynamicFrame must match the configured "functionName"
DynamicFrame.custom_add_columns = custom_add_columns
```

L'esempio seguente utilizza le stesse trasformazioni ma utilizzando un'UDF di pandas, che è più efficiente rispetto all'utilizzo di un'UDF semplice. Per ulteriori informazioni sulla scrittura dei panda, UDFs vedere: Documentazione SQL di Apache Spark.

```
from awsglue import DynamicFrame
import pandas as pd
from pyspark.sql.functions import pandas_udf
```

```
# The number and name of arguments must match the definition on json config file
# (expect self which is the current DynamicFrame to transform
# If an argument is optional, you need to define a default value here
#  (resultCol in this example is an optional argument)
def custom_add_columns(self, col1, col2, resultCol="result"):
    @pandas_udf("integer")  # We need to declare the type of the result column
    def add_columns(value1: pd.Series, value2: pd.Series) # pd.Series:
        return value1 + value2

    df = self.toDF()
    return DynamicFrame.fromDF(
        df.withColumn(resultCol, add_columns(col1, col2)) # This is the conversion
 logic
        , self.glue_ctx, self.name)

# The name we assign on DynamicFrame must match the configured "functionName"
DynamicFrame.custom_add_columns = custom_add_columns
```

### Video

Il video seguente fornisce un'introduzione alle trasformazioni visive personalizzate e dimostra come utilizzarle.

## Utilizzo dei framework data lake con AWS Glue Studio

### Panoramica di

I framework di data lake open source semplificano l'elaborazione incrementale dei dati per i file archiviati in data lake basati su Amazon S3. AWS Glue 3.0 e versioni successive supportano i seguenti framework di data lake open source:

- Apache Hudi
- Linux Foundation Delta Lake
- Apache Iceberg

A partire da AWS Glue 4.0, AWS Glue fornisce il supporto nativo per questi framework in modo che sia possibile leggere e scrivere i dati archiviati in Amazon S3 in modo coerente dal punto di vista transazionale. Non è necessario installare un connettore separato o completare passaggi di configurazione aggiuntivi per utilizzare questi framework nei processi di AWS Glue.

I framework Data Lake possono essere utilizzati come origine o destinazione all'interno di AWS Glue Studio tramite i processi dell'editor di script Spark. Per ulteriori informazioni sull'utilizzo di Apache Hudi, Apache Iceberg e Delta Lake, consulta Utilizzo di framework di data lake con processi AWS Glue ETL.

## Creazione di formati di tabelle aperte da una fonte di AWS Glue streaming

AWS Glue i lavori ETL in streaming consumano continuamente dati provenienti da fonti di streaming, puliscono e trasformano i dati in corso e li rendono disponibili per l'analisi in pochi secondi.

AWS offre un'ampia selezione di servizi per soddisfare le tue esigenze. Un servizio di replica del AWS database come Database Migration Service può replicare i dati dai sistemi di origine su Amazon S3, che di solito ospita il livello di storage del data lake. Sebbene sia semplice applicare gli aggiornamenti su un sistema di gestione di database relazionale (RDBMS) che supporta un'applicazione di origine online, è difficile applicare questo processo CDC sui data lake. I framework di gestione dei dati open-source semplificano l'elaborazione incrementale dei dati e lo sviluppo di pipeline di dati e sono una buona opzione per risolvere questo problema.

Per ulteriori informazioni, consulta:

- Crea un data lake transazionale basato su Apache HUDI utilizzando near-real-time Streaming AWS Glue
- Crea un data lake Apache Iceberg allineato al GDPR in tempo reale

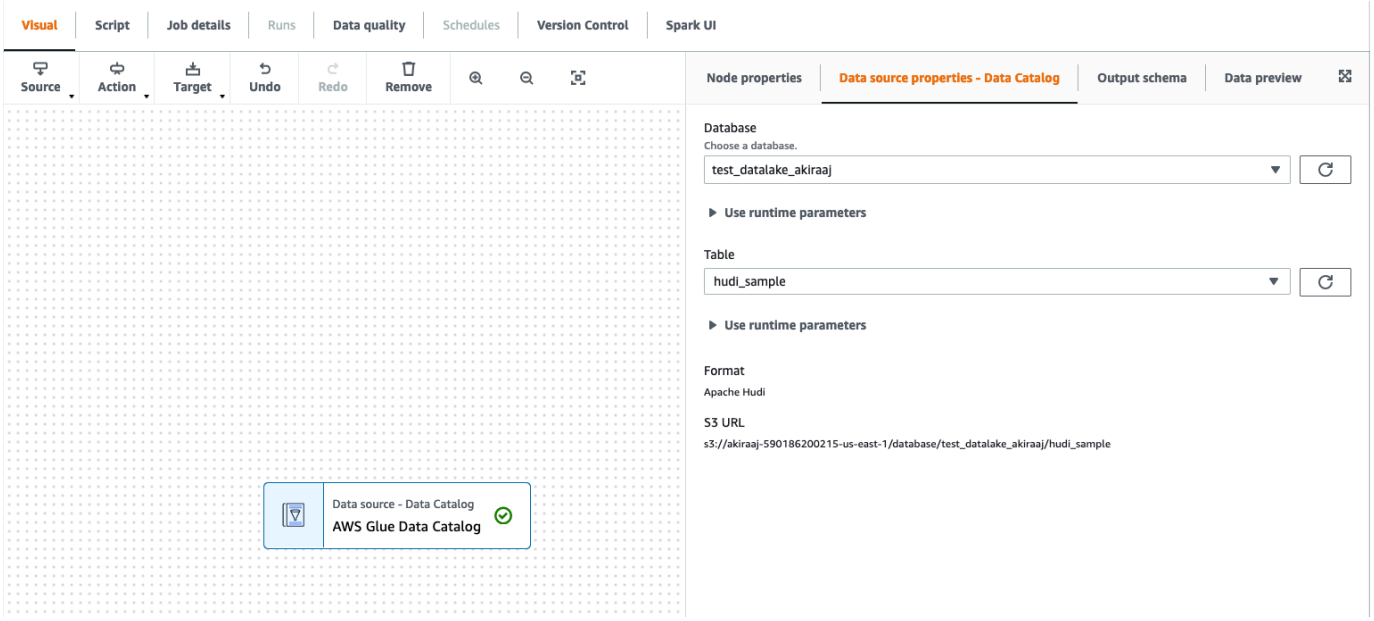
## Utilizzo del framework Hudi in AWS Glue Studio

Quando crei o modifichi un processo, AWS Glue Studio aggiunge automaticamente le librerie Hudi corrispondenti a seconda della versione di AWS Glue che stai utilizzando. Per ulteriori informazioni, consulta Utilizzo del framework di Hudi in AWS Glue.

Utilizzo del framework di Apache Hudi nelle origini dati del catalogo dati

Per aggiungere un formato di origine dati di Hudi a un processo:

1. Dal menu Origine, scegli Catalogo dati AWS Glue Studio.
2. Nella scheda Proprietà dell'origine dati, scegli un database e una tabella.
3. AWS Glue Studio mostra il tipo di formato come Apache Hudi e l'URL di Amazon S3.



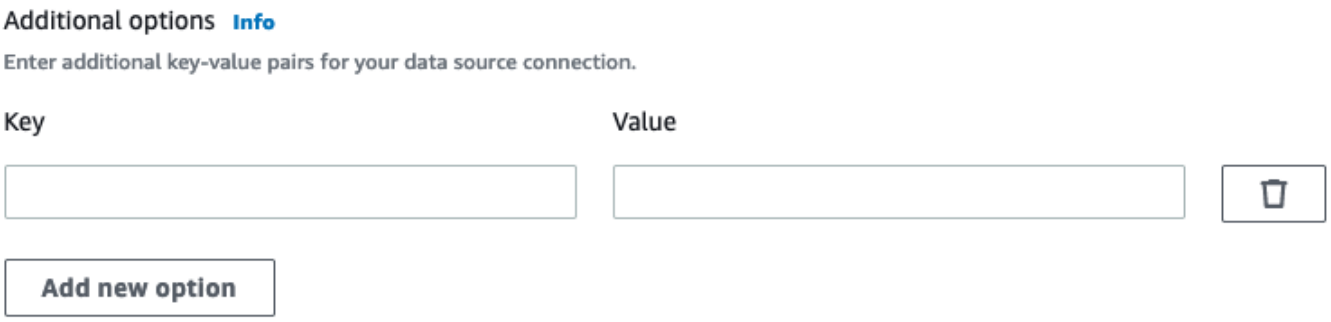

Utilizzo del framework di Hudi nelle origini dati di Amazon S3

1. Dal menu Origine, scegli Amazon S3.
2. Se scegli la tabella del catalogo dati come tipo di origine di Amazon S3, scegli un database e una tabella.
3. AWS Glue Studio mostra il formato come Apache Hudi e l'URL di Amazon S3.
4. Se scegli la posizione Amazon S3 come tipo di origine Amazon S3, scegli l'URL di Amazon S3 facendo clic su Sfoglia Amazon S3.
5. In Formato dati, seleziona Apache Hudi.

> **Note**
>
> Se AWS Glue Studio non riesce a inferire lo schema dalla cartella o dal file Amazon S3 che hai selezionato, scegli Opzioni aggiuntive per selezionare una nuova cartella o un nuovo file.
> In Opzioni aggiuntive, scegli tra le seguenti opzioni in Inferenza dello schema:
>
> - Lascia che AWS Glue Studio scelga automaticamente un file di esempio: AWS Glue Studio sceglierà un file di esempio nella posizione di Amazon S3 in modo da poter inferire lo schema. Nel campo File con campionatura automatica, puoi visualizzare il file che è stato selezionato automaticamente.

- Scegli un file di esempio da Amazon S3: scegli il file Amazon S3 da utilizzare facendo clic su Sfoglia Amazon S3.

6. Fai clic su Inferisci schema. A questo punto potrai visualizzare lo schema di output facendo clic sulla scheda Schema di output.
7. Scegli Opzioni aggiuntive per inserire una coppia chiave-valore.



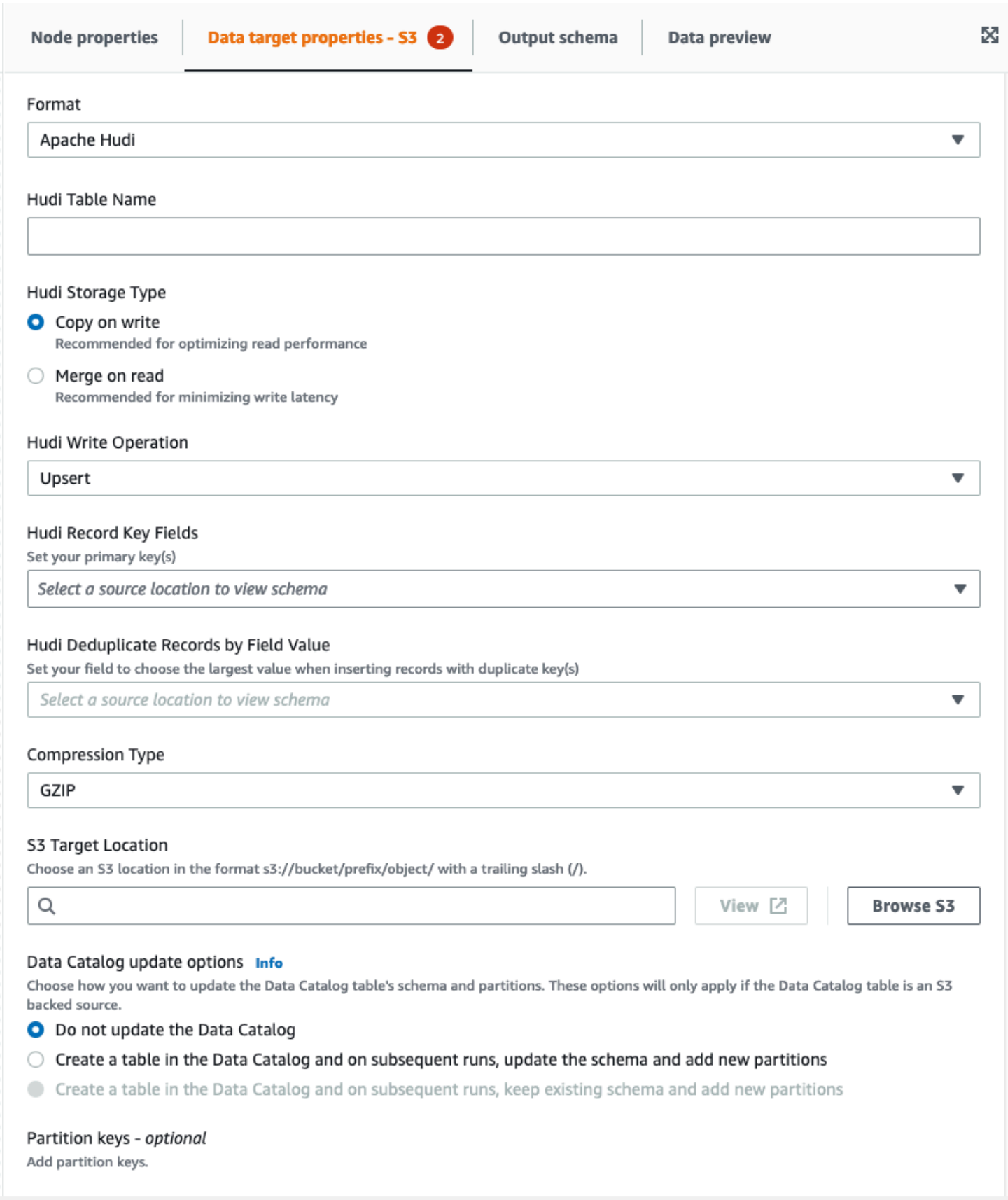

Utilizzo del framework di Apache Hudi nelle destinazioni dei dati

Utilizzo del framework di Apache Hudi nelle destinazioni dei dati del catalogo dati

1. Dal menu Destinazione, scegli Catalogo dati AWS Glue Studio.
2. Nella scheda Proprietà dell'origine dati, scegli un database e una tabella.
3. AWS Glue Studio mostra il tipo di formato come Apache Hudi e l'URL di Amazon S3.

Utilizzo del framework di Apache Hudi nelle destinazioni dei dati di Amazon S3

Inserisci valori o scegli tra le opzioni disponibili per configurare il formato di Apache Hudi. Per ulteriori informazioni su Apache Hudi, consulta la documentazione di Apache Hudi.


Node properties
Data target properties - S3 2
Output schema
Data preview
Format
Apache Hudi
Hudi Table Name
Hudi Storage Type
Copy on write
Recommended for optimizing read performance
Merge on read
Recommended for minimizing write latency
Hudi Write Operation
Upsert
Hudi Record Key Fields
Set your primary key(s)
Select a source location to view schema
Hudi Deduplicate Records by Field Value
Set your field to choose the largest value when inserting records with duplicate key(s)
Select a source location to view schema
Compression Type
GZIP
S3 Target Location
Choose an S3 location in the format s3://bucket/prefix/object/ with a trailing slash (/).
View
Browse S3
Data Catalog update options Info
Choose how you want to update the Data Catalog table's schema and partitions. These options will only apply if the Data Catalog table is an S3 backed source.
Do not update the Data Catalog
Create a table in the Data Catalog and on subsequent runs, update the schema and add new partitions
Create a table in the Data Catalog and on subsequent runs, keep existing schema and add new partitions
Partition keys - optional
Add partition keys.

- Nome tabella Hudi: questo è il nome della tua tabella Hudi.
- Tipo di archiviazione Hudi - scegli tra due opzioni:
  - Copia in scrittura: consigliata per ottimizzare le prestazioni di lettura. È il tipo di archiviazione di Hudi predefinito. Ogni aggiornamento crea una nuova versione dei file durante una scrittura.
  - Unisci in lettura: consigliata per ridurre al minimo la latenza di scrittura. Gli aggiornamenti vengono registrati nei file delta basati su righe e vengono compattati come necessario per creare nuove versioni dei file colonnari.
- Operazione di scrittura di Hudi - scegli una delle seguenti opzioni:
  - Upsert: questa è l'operazione predefinita in cui i record di input vengono prima contrassegnati come inserimenti o aggiornamenti cercando l'indice. Consigliata laddove stai aggiornando dati esistenti.
  - Inserimento: inserisce i record ma non verifica i record esistenti e può generare duplicati.
  - Inserimento in blocco: consente di inserire record ed è consigliato per grandi quantità di dati.
- Campi chiave record Hudi: utilizza la barra di ricerca per cercare e scegliere le chiavi record primarie. I record in Hudi sono identificati da una chiave primaria che è una coppia composta da chiave di record e percorso di partizione a cui appartiene il record.
- Campo di precombinazione Hudi: questo è il campo utilizzato in "preCombining" prima della scrittura effettiva. Quando due record hanno lo stesso valore chiave, AWS Glue Studio seleziona quello con il valore più grande per il campo di precombinazione. Imposta un campo con valore incrementale (ad esempio updated_at) a cui appartiene.
- Tipo di compressione: scegli una delle opzioni per il tipo di compressione: Uncompressed, GZIP, LZO o Snappy.
- Posizione di destinazione di Amazon S3: scegli la posizione di destinazione di Amazon S3 facendo clic su Sfoglia S3.
- Opzioni di aggiornamento del catalogo dati - scegli una delle seguenti opzioni:
  - Do not update the Data Catalog (Non aggiornare il catalogo dati): (impostazione predefinita) scegli questa opzione se non vuoi che il processo aggiorni il catalogo dati, anche se lo schema viene modificato o sono aggiunte nuove partizioni.
  - Crea una tabella nel catalogo dati e, nelle esecuzioni successive, aggiorna lo schema e aggiungi nuove partizioni: se scegli questa opzione, il processo crea la tabella nel catalogo dati alla prima esecuzione. Nelle successive esecuzioni del processo, questo aggiorna la tabella del catalogo dati se lo schema viene modificato o sono aggiunte nuove partizioni.

    Devi inoltre selezionare un database dal catalogo dati e inserire un nome di tabella.

- Create a table in the Data Catalog and on subsequent runs, keep existing schema and add new partitions (Crea una tabella nel catalogo dati e, nelle esecuzioni successive, mantieni lo schema esistente e aggiungi nuove partizioni): se scegli questa opzione, il processo crea la tabella nel catalogo dati alla prima esecuzione. Nelle successive esecuzioni del processo, questo aggiorna la tabella del catalogo dati solo per aggiungere nuove partizioni.

  Devi inoltre selezionare un database dal catalogo dati e inserire un nome di tabella.

- Partition keys (Chiavi di partizione): scegli quali colonne utilizzare come chiavi di partizionamento nell'output. Per aggiungere altre chiavi di partizione, scegli Add a partition key (Aggiungi una chiave di partizione).
- Opzioni aggiuntive: inserisci una coppia chiave-valore, se necessario.

### Generazione di codice tramite AWS Glue Studio

Quando il processo viene salvato, i seguenti parametri di processo vengono aggiunti al processo se viene rilevata un'origine o una destinazione Hudi:

- `--datalake-formats`: un elenco distinto di formati di data lake rilevati nel processo visivo (direttamente scegliendo un "Formato" o indirettamente selezionando una tabella di catalogo supportata da un data lake).
- `--conf` : generato in base al valore di `--datalake-formats`. Ad esempio, se il valore per `--datalake-formats` è "hudi", AWS Glue genera un valore di `spark.serializer=org.apache.spark.serializer.KryoSerializer –conf spark.sql.hive.convertMetastoreParquet=false` per questo parametro.

### Sostituzione delle librerie fornite da AWS Glue

Per utilizzare una versione di Hudi che AWS Glue non supporta, puoi specificare i tuoi file JAR della libreria di Hudi. Per usare il tuo file JAR:

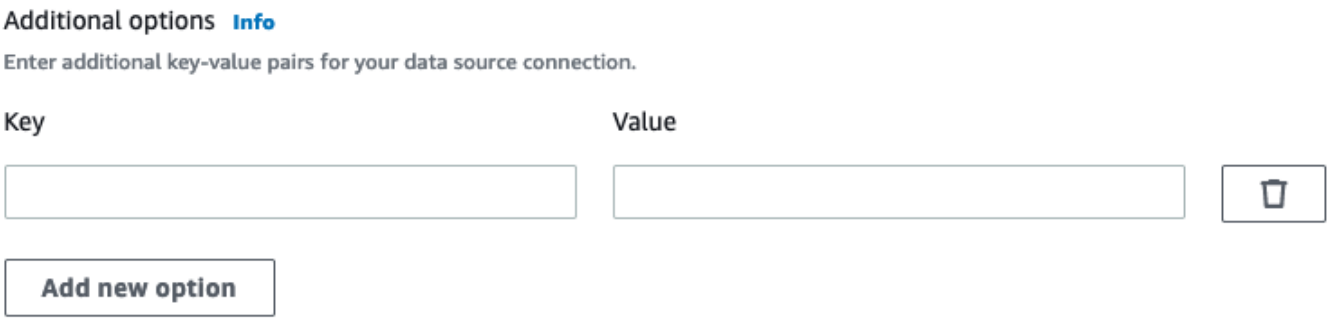
- utilizza il parametro del processo `--extra-jars`. Ad esempio, `'--extra-jars': 's3pathtojarfile.jar'`. Per ulteriori informazioni, consulta i parametri del processo AWS Glue.
- Non includere `hudi` come valore per il parametro del processo `--datalake-formats`. L'immissione di una stringa vuota come valore garantisce che nessuna libreria di data lake venga fornita automaticamente da AWS Glue. Per ulteriori informazioni, consulta Utilizzo del framework di Hudi in AWS Glue.

# Utilizzo del framework Delta Lake in AWS Glue Studio

## Utilizzo del framework Delta Lake in origini dati

### Utilizzo del framework Delta Lake in origini dati Amazon S3

1. Dal menu Origine, scegli Amazon S3.
2. Se scegli la tabella del catalogo dati come tipo di origine di Amazon S3, scegli un database e una tabella.
3. AWS Glue Studio mostra il formato come Delta Lake e l'URL di Amazon S3.
4. Scegli Opzioni aggiuntive per inserire una coppia chiave-valore. Ad esempio, una coppia chiave-valore potrebbe essere: chiave: timestampAsOf e valore: 2023-02-24 14:16:18.




5. Se scegli la posizione Amazon S3 come tipo di origine Amazon S3, scegli l'URL di Amazon S3 facendo clic su Sfoglia Amazon S3.
6. In Formato data, scegli Delta Lake.

> **Note**
>
> Se AWS Glue Studio non riesce a inferire lo schema dalla cartella o dal file Amazon S3 che hai selezionato, scegli Opzioni aggiuntive per selezionare una nuova cartella o un nuovo file.
> In Opzioni aggiuntive, scegli tra le seguenti opzioni in Inferenza dello schema:
>
> - Lascia che AWS Glue Studio scelga automaticamente un file di esempio: AWS Glue Studio sceglierà un file di esempio nella posizione di Amazon S3 in modo da poter inferire lo schema. Nel campo File con campionatura automatica, puoi visualizzare il file che è stato selezionato automaticamente.

- Scegli un file di esempio da Amazon S3: scegli il file Amazon S3 da utilizzare facendo clic su Sfoglia Amazon S3.

7. Fai clic su Inferisci schema. A questo punto potrai visualizzare lo schema di output facendo clic sulla scheda Schema di output.

Utilizzo del framework Delta Lake in origini dati Catalogo dati

1. Dal menu Origine, scegli Catalogo dati AWS Glue Studio.
2. Nella scheda Proprietà dell'origine dati, scegli un database e una tabella.
3. AWS Glue Studio mostra il tipo di formato come Delta Lake e l'URL di Amazon S3.

> **Note**
>
> Se la tua origine Delta Lake non è ancora registrata come tabella del catalogo dati AWS Glue, hai due opzioni:
>
> 1. Creare un crawler AWS Glue per l'archivio dati Delta Lake. Per ulteriori informazioni, consulta Come specificare le opzioni di configurazione per un archivio dati Delta Lake.
> 2. Utilizzare un'origine dati Amazon S3 per selezionare la tua origine dati Delta Lake. Per informazioni, consulta Utilizzo del framework Delta Lake in origini dati Amazon S3.

Utilizzo dei formati Delta Lake negli obiettivi dei dati

Utilizzo dei formati Delta Lake negli obiettivi dei dati del Catalogo dati

1. Dal menu Destinazione, scegli Catalogo dati AWS Glue Studio.
2. Nella scheda Proprietà dell'origine dati, scegli un database e una tabella.
3. AWS Glue Studio mostra il tipo di formato come Delta Lake e l'URL di Amazon S3.

Utilizzo dei formati Delta Lake nelle origini dati di Amazon S3

Inserisci valori o scegli tra le opzioni disponibili per configurare il formato di Delta Lake.

- Tipo di compressione: scegli una delle opzioni per il tipo di compressione: Uncompressed o Snappy.

- Posizione di destinazione di Amazon S3: scegli la posizione di destinazione di Amazon S3 facendo clic su Sfoglia S3.
- Opzioni di aggiornamento del Catalogo dati: l'aggiornamento del Catalogo dati non è supportato per questo formato nell'editor visivo di Glue Studio.
  - Do not update the Data Catalog (Non aggiornare il catalogo dati): (impostazione predefinita) scegli questa opzione se non vuoi che il processo aggiorni il catalogo dati, anche se lo schema viene modificato o sono aggiunte nuove partizioni.
  - Per aggiornare il Catalogo dati dopo l'esecuzione del processo AWS Glue, esegui o pianifica un crawler AWS Glue. Per ulteriori informazioni, consulta Come specificare le opzioni di configurazione per un archivio dati Delta Lake.
- Chiavi di partizione: scegli quali colonne utilizzare come chiavi di partizionamento nell'output. Per aggiungere altre chiavi di partizione, scegli Add a partition key (Aggiungi una chiave di partizione).
- Facoltativamente, scegli Opzioni aggiuntive per inserire una coppia chiave-valore. Ad esempio, una coppia chiave-valore potrebbe essere: chiave: e valore: 2023-02-24 14:16:18. timestampAsOf

## Utilizzo del framework Apache Iceberg in AWS Glue Studio

### Utilizzo del framework Apache Iceberg nelle destinazioni dati

### Utilizzo del framework Apache Iceberg nelle destinazioni dati di Catalogo dati

1. Dal menu Destinazione, scegli Catalogo dati AWS Glue Studio.
2. Nella scheda Proprietà dell'origine dati, scegli un database e una tabella.
3. AWS Glue Studio mostra il tipo di formato come Apache Iceberg e l'URL di Amazon S3.

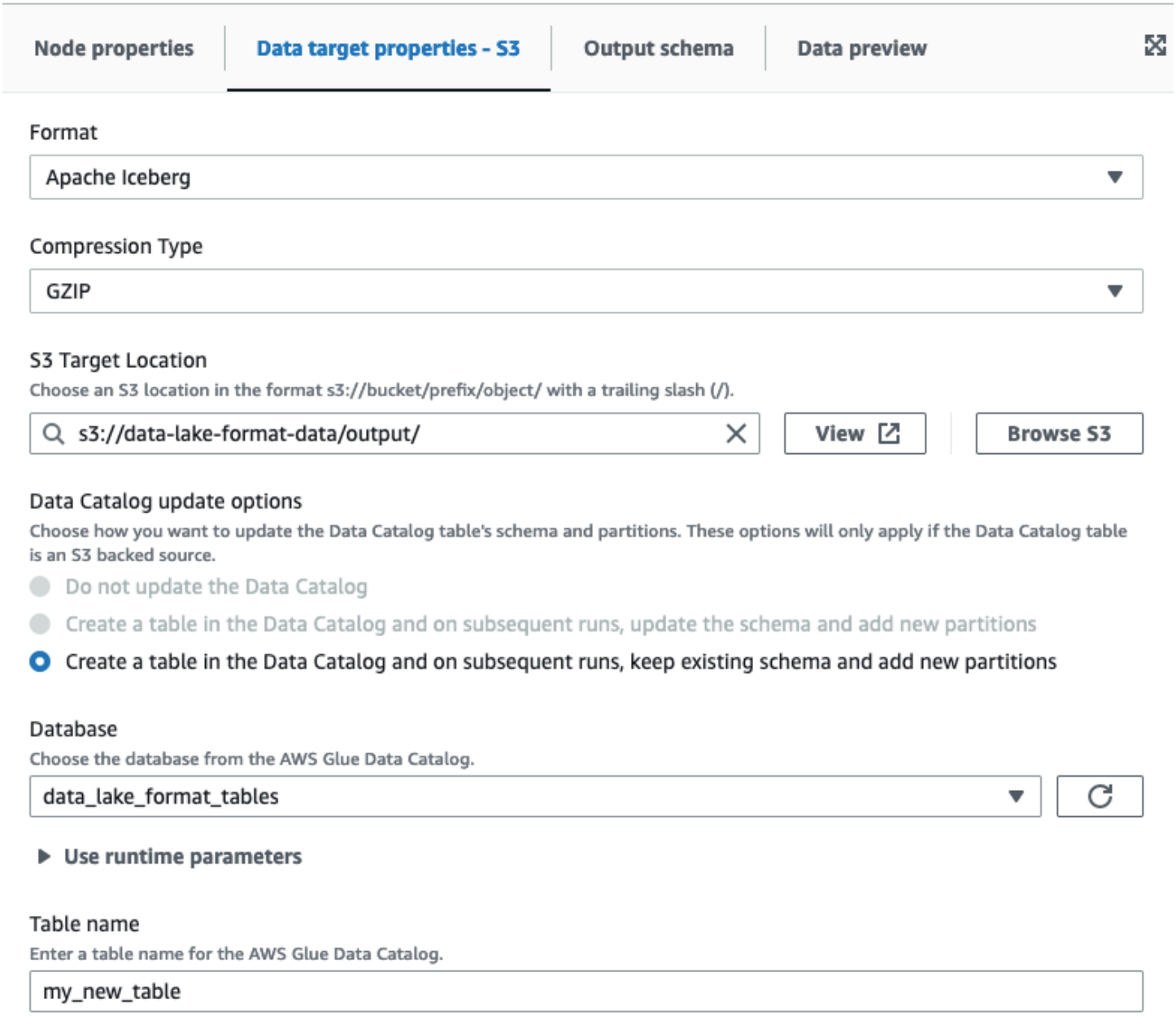
### Utilizzo del framework Apache Iceberg nelle destinazioni dati di Amazon S3

Inserisci valori o scegli tra le opzioni disponibili per configurare il formato di Apache Iceberg.

- Formato: scegli Apache Iceberg dal menu a discesa.
- Posizione di destinazione di Amazon S3: scegli la posizione di destinazione di Amazon S3 facendo clic su Sfoglia S3.
- Opzioni di aggiornamento del Catalogo dati: è necessario selezionare Crea una tabella in Catalogo dati e, nelle esecuzioni successive, mantieni lo schema esistente e aggiungi nuove partizioni per procedere. La scrittura di una nuova tabella Iceberg utilizzando AWS Glue richiede che Data

Catalog venga configurato come catalogo per la tabella Iceberg. Per aggiornare una tabella Iceberg esistente che è stata registrata in Data Catalog, scegli Data Catalog come destinazione.

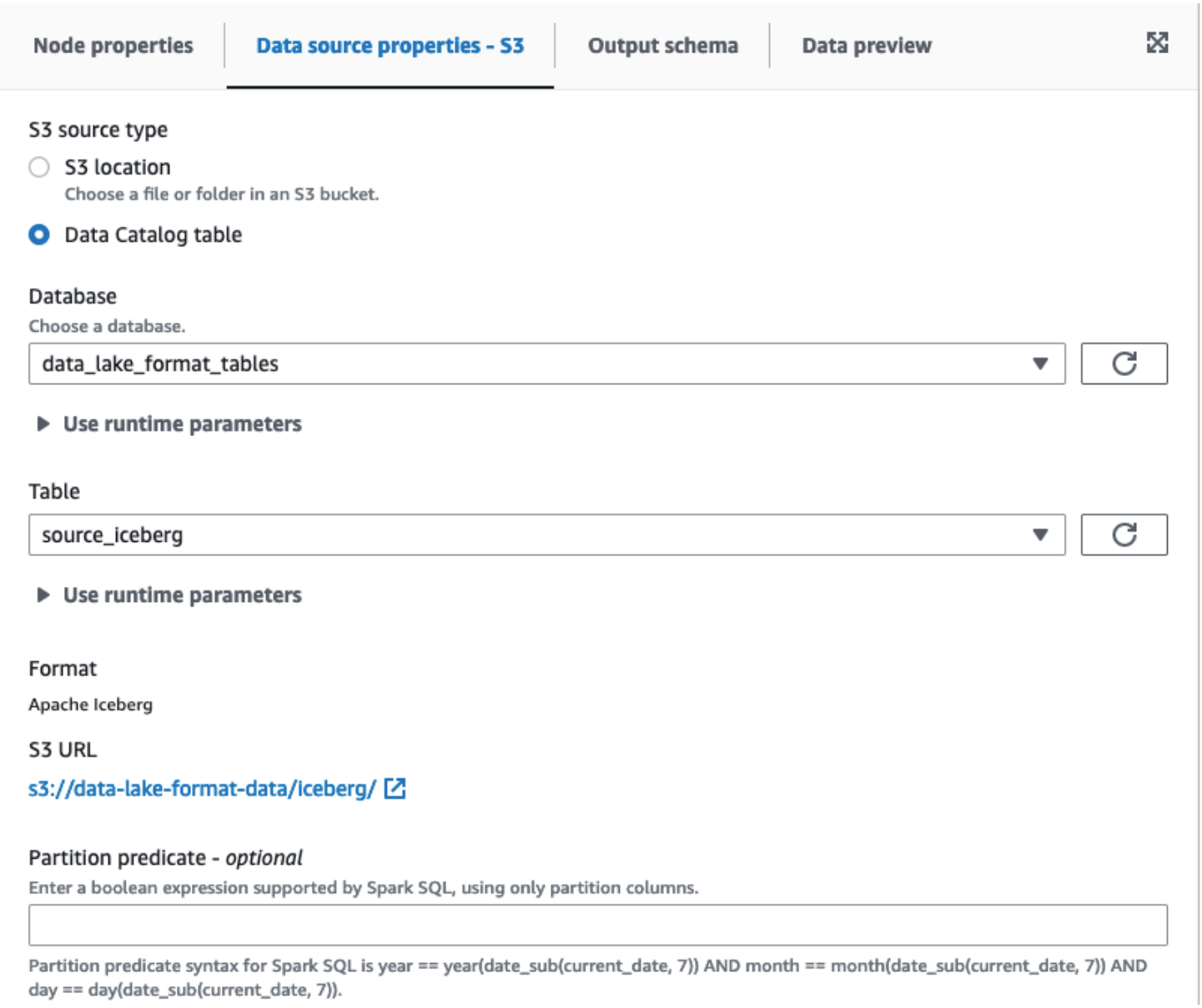
- Database: scegli il database dal Data Catalog.
- Nome tabella: inserisci un nome univoco per la tabella. I nomi delle tabelle di Apache Iceberg devono essere tutti in minuscolo. Utilizza i caratteri di sottolineatura, se necessario, poiché gli spazi non sono consentiti. Ad esempio "data_lake_format_tables".



Utilizzo del framework Apache Iceberg nelle origini dati di Amazon S3

Utilizzo del framework Apache Iceberg nelle origini dati di Catalogo dati

1. Dal menu Origine, scegli Catalogo dati AWS Glue Studio.
2. Nella scheda Proprietà dell'origine dati, scegli un database e una tabella.
3. AWS Glue Studio mostra il tipo di formato come Apache Iceberg e l'URL di Amazon S3.



Utilizzo del framework Apache Iceberg nelle origini dati di Amazon S3

Apache Iceberg non è disponibile come opzione dati per i nodi origine di Amazon S3 in AWS Glue Studio.

# Connessione alle origini dati tramite processi ETL visivi

Durante la creazione di un nuovo processo, è possibile utilizzare le connessioni per connettersi ai dati durante la modifica dei processi visivi ETL in AWS Glue. È possibile farlo aggiungendo nodi di origine che utilizzano connettori per leggere i dati e nodi di destinazione per specificare la posizione in cui scrivere i dati.

Argomenti


- Modifica delle proprietà di un nodo di origine dati
- Utilizzo delle tabelle del catalogo dati per l'origine dati
- Utilizzo di un connettore per l'origine dati
- Utilizzo di file in Amazon S3 per l'origine dati
- Utilizzo di un'origine dati di streaming
- Riferimenti


## Modifica delle proprietà di un nodo di origine dati

Per specificare le proprietà di origine dati, è innanzitutto necessario scegliere un nodo di origine dati nel diagramma del processo. Quindi, sul lato destro nel pannello dei dettagli del nodo, puoi configurare le proprietà del nodo.

Per modificare le proprietà di un nodo di origine dati

1. Vai all'editor visivo per un processo nuovo o salvato.
2. Scegli un nodo di origine dati nel diagramma del processo.
3. Seleziona Node properties (Proprietà del nodo) nel pannello dei dettagli del nodo, quindi inserisci le seguenti informazioni:

    - Name (Nome): (facoltativo) immetti un nome da associare al nodo nel diagramma del processo. Questo nome deve essere univoco tra tutti i nodi per questo processo.

- Node type (Tipo di nodo): il tipo di nodo determina l'azione eseguita dal nodo. Nell'elenco delle opzioni per Node type (Tipo di nodo), scegli uno dei valori elencati sotto l'intestazione Data source (Origine dati).

4. Configura le informazioni di Data source properties (Proprietà dell'origine dati). Per ulteriori informazioni, consulta le sezioni seguenti:

   - Utilizzo delle tabelle del catalogo dati per l'origine dati
   - Utilizzo di un connettore per l'origine dati
   - Utilizzo di file in Amazon S3 per l'origine dati
   - Utilizzo di un'origine dati di streaming

5. (Facoltativo) Dopo aver configurato le proprietà del nodo e dell'origine dati, puoi visualizzare lo schema per l'origine dati scegliendo la scheda Output schema (Schema di output) nel pannello dei dettagli del nodo. La prima volta che si sceglie questa scheda per qualsiasi nodo del processo, viene richiesto di fornire un ruolo IAM per accedere ai dati. Se non è stato specificato un ruolo IAM nella scheda Job details (Dettagli del processo), viene richiesto di immettere un ruolo IAM a questo punto.
6. (Facoltativo) Dopo aver configurato le proprietà del nodo e dell'origine dati, puoi visualizzare il set di dati dall'origine dati scegliendo la scheda Data preview (Anteprima dei dati) nel pannello dei dettagli del nodo. La prima volta che si sceglie questa scheda per qualsiasi nodo del processo, viene richiesto di fornire un ruolo IAM per accedere ai dati. Esiste un costo per l'utilizzo di questa caratteristica e la fatturazione inizia non appena si fornisce un ruolo IAM.

## Utilizzo delle tabelle del catalogo dati per l'origine dati

Per tutte le origini dati ad eccezione di Amazon S3 e dei connettori, è necessario che esista una tabella in AWS Glue Data Catalog per il tipo di origine scelto. AWS Glue non crea la tabella in Data Catalog.

Per configurare un nodo di origine dati basato su una tabella del catalogo dati

1. Vai all'editor visivo per un processo nuovo o salvato.
2. Scegli un nodo di origine dati nel diagramma del processo.
3. Seleziona la scheda Data source properties (Proprietà dell'origine dati), quindi immetti le informazioni riportate di seguito:

- S3 source type (Tipo di origine S3): (solo per origini dati Amazon S3) scegli l'opzione Select a Catalog table (Seleziona una tabella del catalogo) per utilizzare una tabella di AWS Glue Data Catalog.
- Database: scegli il database nel catalogo dati contenente la tabella di origine da utilizzare per questo processo. Puoi utilizzare il campo di ricerca per cercare un database per nome.
- Table (Tabella): scegli dall'elenco la tabella associata ai dati di origine. Questa tabella deve esistere già in AWS Glue Data Catalog. Puoi utilizzare il campo di ricerca per cercare una tabella per nome.
- Partition predicate (Predicato di partizione): (solo per origini dati Amazon S3) inserisci un'espressione booleana basata su Spark SQL che includa solo le colonne di partizionamento. Ad esempio: `"(year=='2020' and month=='04')"`
- Temporary directory (Directory temporanea): (solo per le origini dati Amazon Redshift) inserisci un percorso per la posizione di una directory di processo in Amazon S3 in cui il processo ETL può scrivere risultati intermedi temporanei.
- Role associated with the cluster (Ruolo associato al cluster): (solo per le origini dati Amazon Redshift) inserisci un ruolo da utilizzare per il processo ETL che contiene le autorizzazioni per i cluster Amazon Redshift . Per ulteriori informazioni, consulta the section called “Autorizzazioni origine dati e destinazione dati”.

## Utilizzo di un connettore per l'origine dati

Se per Node type (Tipo di nodo) selezioni un connettore, segui le istruzioni in Creazione di processi con connettori personalizzati per completare la configurazione delle proprietà dell'origine dati.

## Utilizzo di file in Amazon S3 per l'origine dati

Se scegli Amazon S3 come origine dati, puoi scegliere:

- Un database e una tabella del catalogo dati.
- Un bucket, una cartella o un file in Amazon S3.

Se utilizzi un bucket Amazon S3 come origine dati, AWS Glue rileva lo schema dei dati nella posizione specificata da uno dei file o utilizzando il file specificato come file di esempio. Il rilevamento dello schema si verifica quando si utilizza il pulsante Infer schema (Deduci schema). Se modifichi

la posizione di Amazon S3 o il file di esempio, devi selezionare nuovamente Infer schema (Deduci schema) per eseguire il rilevamento dello schema utilizzando le nuove informazioni.

Per configurare un nodo origine dati che legge direttamente dai file in Amazon S3

1. Vai all'editor visivo per un processo nuovo o salvato.
2. Scegli un nodo di origine dati nel diagramma del processo per un'origine Amazon S3.
3. Seleziona la scheda Data source properties (Proprietà dell'origine dati), quindi immetti le informazioni riportate di seguito:

   - S3 source type (Tipo di origine S3): (solo per origini dati Amazon S3) scegli l'opzione S3 location (Posizione S3).
   - S3 URL (URL S3): inserisci il percorso del bucket, della cartella o del file Amazon S3 che contiene i dati per il processo. Puoi scegliere Browse S3 (Sfoglia S3) per selezionare il percorso dalle posizioni disponibili per il tuo account.
   - Recursive (Ricorsiva): scegli questa opzione se vuoi che AWS Glue legga i dati dai file nelle cartelle figlio nella posizione S3.

     Se le cartelle figlio contengono dati partizionati, AWS Glue non aggiunge le informazioni di partizione specificate nei nomi delle cartelle al Data Catalog. Considera, ad esempio, le seguenti cartelle in Amazon S3:

     ```
     S3://sales/year=2019/month=Jan/day=1
     S3://sales/year=2019/month=Jan/day=2
     ```

     Scegliendo Recursive (Ricorsiva) e selezionando `sales` come posizione S3, AWS Glue legge i dati in tutte le cartelle figlio, ma non crea partizioni per anno, mese o giorno.
   - Data format (Formato dei dati): scegli il formato in cui sono memorizzati i dati. Puoi scegliere JSON, CSV o Parquet. Il valore selezionato indica al processo AWS Glue come leggere i dati dal file di origine.

     > **Note**
     >
     > Se non selezioni il formato dei dati corretto, AWS Glue potrebbe dedurre lo schema correttamente, ma il processo non sarà in grado di analizzare correttamente i dati dal file di origine.

Puoi immettere opzioni di configurazione aggiuntive, a seconda del formato scelto.

- JSON (notazione di JavaScript oggetti)
  - JsonPath: Immettete un percorso JSON che punti a un oggetto utilizzato per definire uno schema di tabella. Le espressioni di percorso JSON fanno sempre riferimento a una struttura JSON nello stesso modo in cui le XPath espressioni vengono utilizzate in combinazione con un documento XML. L' "oggetto membro root" nel percorso JSON è sempre indicato come $, anche se si tratta di un oggetto o di una matrice. È possibile scrivere il percorso JSON in notazione punto o in notazione parentesi.

    Per ulteriori informazioni sul percorso JSON, consultate il JsonPathsito Web. GitHub
  - Records in source files can span multiple lines (I registri nei file di origine possono estendersi su più righe): seleziona questa opzione se un singolo registro può estendersi su più righe nel file CSV.
- CSV (valori separati da virgola)
  - Delimiter (Delimitatore): immetti un carattere per indicare il separatore di ogni voce di colonna nella riga, ad esempio `;` o `,`.
  - Escape character (Carattere di escape): immetti un carattere utilizzato come carattere di escape. Questo carattere indica che il carattere che segue immediatamente il carattere di escape deve essere preso alla lettera e non deve essere interpretato come un delimitatore.
  - Quote character (Carattere virgolette): immetti il carattere utilizzato per raggruppare stringhe separate in un singolo valore. Ad esempio, devi scegliere Double quote (") (virgolette doppie ["]) se nel file CSV sono presenti valori `"This is a single value"`.
  - Records in source files can span multiple lines (I registri nei file di origine possono estendersi su più righe): seleziona questa opzione se un singolo registro può estendersi su più righe nel file CSV.
  - First line of source file contains column headers (La prima riga del file di origine contiene le intestazioni di colonna): scegli questa opzione se la prima riga del file CSV contiene intestazioni di colonna anziché dati.
- Parquet (storage a colonne Apache Parquet)

  Non ci sono impostazioni aggiuntive da configurare per i dati memorizzati in formato Parquet.
- Apache Hudi

Non ci sono impostazioni aggiuntive da configurare per i dati memorizzati in formato Apache Hudi.

- Delta Lake

  Non ci sono impostazioni aggiuntive da configurare per i dati memorizzati in formato Delta Lake.

- Excel

  Non ci sono impostazioni aggiuntive da configurare per i dati memorizzati in formato Excel.

- Partition predicate (Predicato di partizione): per partizionare i dati letti dall'origine dati, inserisci un'espressione booleana basata su Spark SQL che includa solo le colonne di partizionamento. Ad esempio: `"(year=='2020' and month=='04')"`
- Advanced options (Opzioni avanzate): espandi questa sezione se vuoi che AWS Glue rilevi lo schema dei dati in base a un file specifico.
  - Schema inference (Deduzione dello schema): seleziona l'opzione Choose a sample file from S3 (Scegli un file di esempio da S3) se vuoi utilizzare un file specifico invece di lasciare che AWS Glue scelga un file. La deduzione dello schema non è disponibile per l'origine Excel.
  - Auto-sampled file (File con campionatura automatica): inserisci il percorso del file in Amazon S3 da utilizzare per dedurre lo schema.

  Se stai apportando modifiche a un nodo dell'origine dati e al file di esempio selezionato, scegli Reload schema (Ricarica schema) per rilevare lo schema utilizzando il nuovo file di esempio.

4. Seleziona il pulsante Infer schema (Seleziona schema) per rilevare lo schema dai file di origine in Amazon S3. Se modifichi la posizione di Amazon S3 o il file di esempio, devi selezionare nuovamente Infer schema (Deduci schema) per rilevare lo schema utilizzando le nuove informazioni.

## Utilizzo di un'origine dati di streaming

È possibile creare processi in streaming di estrazione, trasformazione e caricamento (ETL) che vengono eseguiti continuamente e consumano dati da origini di streaming in Amazon Kinesis Data Streams, Apache Kafka e Amazon Streaming gestito per Apache Kafka (Amazon MSK).

Per configurare le proprietà per un'origine dati di streaming

1. Vai all'editor grafico visivo per un processo nuovo o salvato.

2. Scegli un nodo origine dati nel grafico per Kafka o Kinesis Data Streams.
3. Seleziona la scheda Data source properties (Proprietà dell'origine dati), quindi immetti le informazioni riportate di seguito:

   Kinesis

   - Kinesis source type (Tipo sorgente Kinesis): scegli l'opzione Stream details (Dettagli streaming) per utilizzare l'accesso diretto alla sorgente di streaming o Data Catalog table (Tabella Data Catalog) per utilizzare invece le informazioni archiviate in questa posizione.

     Se scegli Stream details (Dettagli streaming), specifica le seguenti informazioni aggiuntive.

     - Posizione del flusso di dati: scegli se il flusso di dati è associato all'utente corrente o se è associato a un altro utente.
     - Regione: scegli Regione AWS dove esiste lo stream. Queste informazioni vengono utilizzate per costruire l'ARN per l'accesso al flusso di dati.
     - Stream ARN (ARN del flusso di dati): l'Amazon Resource Name (ARN) per l'endpoint del flusso di dati Kinesis. Se il flusso di dati si trova nell'account corrente, è possibile selezionarne il nome dall'elenco a discesa. Puoi utilizzare il campo di ricerca per cercare un flusso dei dati per nome o per ARN.
     - Data format (Formato dei dati): scegli il formato utilizzato dal flusso di dati dall'elenco.

       AWS Glue rileva automaticamente lo schema dai dati in streaming.

     Se scegli Data Catalog table (Tabella Data Catalog), specifica le seguenti informazioni aggiuntive.

     - Database: (facoltativo) scegli il database nel Data Catalog di AWS Glue che contiene la tabella associata all'origine dati in streaming. Puoi utilizzare il campo di ricerca per cercare un database per nome.
     - Table (Tabella): (facoltativo) scegli dall'elenco la tabella associata ai dati di origine. Questa tabella deve esistere già nel AWS Glue Data Catalog. Puoi utilizzare il campo di ricerca per cercare una tabella per nome.
     - Detect schema (Rileva schema): scegli questa opzione per permettere ad AWS Glue Glue Studio di rilevare lo schema dai dati di streaming, anziché archiviare le informazioni sullo schema in una tabella di Data Catalog. Se scegli l'opzione Stream details (Dettagli streaming), questa opzione è abilitata automaticamente.

- Starting position (Posizione di inizio): per impostazione predefinita, il processo ETL utilizza l'opzione Earliest (Primo), il che significa che legge i dati a partire dal registro più vecchio disponibile nel flusso di dati. Puoi invece scegliere Latest (Più recente), che indica che il processo ETL dovrebbe iniziare a leggere subito dopo il registro più recente nel flusso di dati.
- Window size (Dimensione finestra): per impostazione predefinita, il processo ETL elabora e scrive i dati in finestre di 100 secondi. Ciò consente di elaborare i dati in modo efficiente e di eseguire aggregazioni su dati che arrivano più tardi del previsto. Puoi modificare questa dimensione della finestra per aumentare la tempestività o la precisione dell'aggregazione.

  AWS Glue i lavori di streaming utilizzano i checkpoint anziché i segnalibri di lavoro per tenere traccia dei dati che sono stati letti.
- Connection options (Opzioni di connessione): espandi questa sezione per aggiungere coppie chiave-valore per specificare opzioni di connessione aggiuntive. Per informazioni sulle opzioni che è possibile specificare qui, consulta "connectionType": "kinesis" nella Guida per gli sviluppatori di AWS Glue .

Kafka

- Apache Kafka source (Origine Apache Kafka): scegli l'opzione Stream details (Dettagli streaming) per utilizzare l'accesso diretto alla sorgente di streaming o Data Catalog table (Tabella Data Catalog) per utilizzare invece le informazioni archiviate in questa posizione.

  Se scegli Data Catalog table (Tabella Data Catalog), specifica le seguenti informazioni aggiuntive.

  - Database: (facoltativo) scegli il database nel Data Catalog di AWS Glue che contiene la tabella associata all'origine dati in streaming. Puoi utilizzare il campo di ricerca per cercare un database per nome.
  - Table (Tabella): (facoltativo) scegli dall'elenco la tabella associata ai dati di origine. Questa tabella deve esistere già nel AWS Glue Data Catalog. Puoi utilizzare il campo di ricerca per cercare una tabella per nome.
  - Detect schema (Rileva schema): scegli questa opzione per permettere ad AWS Glue di rilevare lo schema dai dati di streaming, anziché archiviare le informazioni sullo schema in una tabella di Data Catalog. Se scegli l'opzione Stream details (Dettagli streaming), questa opzione è abilitata automaticamente.

Se scegli Stream details (Dettagli streaming), specifica le seguenti informazioni aggiuntive.

- Connection name (Nome della connessione): scegli la connessione AWS Glue che contiene le informazioni di accesso e autenticazione per il flusso dei dati Kafka. È necessario utilizzare una connessione con le origini dati in streaming di Kafka. Se non esiste una connessione, per creare una connessione per il flusso di dati Kafka è possibile utilizzare la console AWS Glue.
- Topic name (Nome argomento): inserisci il nome dell'argomento da cui leggere.
- Data format (Formato dei dati): scegli il formato da utilizzare durante la lettura dei dati dal flusso di eventi Kafka.

- Starting position (Posizione di inizio): per impostazione predefinita, il processo ETL utilizza l'opzione Earliest (Primo), il che significa che legge i dati a partire dal registro più vecchio disponibile nel flusso di dati. Puoi invece scegliere Latest (Più recente), che indica che il processo ETL dovrebbe iniziare a leggere subito dopo il registro più recente nel flusso di dati.
- Window size (Dimensione finestra): per impostazione predefinita, il processo ETL elabora e scrive i dati in finestre di 100 secondi. Ciò consente di elaborare i dati in modo efficiente e di eseguire aggregazioni su dati che arrivano più tardi del previsto. Puoi modificare questa dimensione della finestra per aumentare la tempestività o la precisione dell'aggregazione.

  I processi di streaming AWS Glue utilizzano i checkpoint anziché i segnalibri di processo per tenere traccia dei dati letti.
- Connection options (Opzioni di connessione): espandi questa sezione per aggiungere coppie chiave-valore per specificare opzioni di connessione aggiuntive. Per informazioni sulle opzioni che è possibile specificare qui, consulta "connectionType": "kafka" nella Guida per gli sviluppatori di AWS Glue .

> **Note**
>
> Le anteprime dei dati non sono attualmente supportate per le origini dati di streaming.

## Riferimenti

### Best practice

- Crea una pipeline di servizi ETL per caricare i dati in modo incrementale da Amazon S3 all'utilizzo Amazon RedshiftAWS Glue

Programmazione ETL

- Tipi e opzioni di connessione per ETL in AWS Glue
- Valori di connectionType JDBC
- Opzioni avanzate per lo spostamento dei dati da e verso Amazon Redshift

## Configurazione dei nodi di destinazione dati

La destinazione dati è la posizione in cui il processo scrive i dati trasformati.

### Panoramica delle opzioni di destinazione dati

La destinazione dati (chiamata anche sink dei dati) può essere:

- S3 – Il processo scrive i dati in un file nella posizione Amazon S3 scelta e nel formato specificato.

  Se configuri le colonne di partizione per la destinazione dati, il processo scrive il set di dati su Amazon S3 in directory basate sulla chiave di partizione.
- AWS Glue Data Catalog – Il processo utilizza le informazioni associate alla tabella nel catalogo dati per scrivere i dati di output in una posizione di destinazione.

  Puoi creare la tabella manualmente o con il crawler. Puoi utilizzare anche modelli AWS CloudFormation per creare tabelle nel catalogo dati.
- Un connettore – un connettore è una parte di codice che facilita la comunicazione tra l'archivio dati e AWS Glue. Il processo utilizza il connettore e la connessione associata per scrivere i dati di output in una posizione di destinazione. Puoi abbonarti a un connettore offerto in Marketplace AWS oppure puoi creare il tuo connettore personalizzato. Per ulteriori informazioni, consulta Aggiunta di connettori a AWS Glue Studio

Puoi scegliere di aggiornare il catalogo dati quando il tuo processo scrive in una destinazione dati Amazon S3. Anziché richiedere a un crawler di aggiornare il catalogo dati quando lo schema o le partizioni cambiano, questa opzione semplifica l'aggiornamento delle tabelle. Questa opzione semplifica il processo che rende disponibili i dati per l'analisi aggiungendo facoltativamente nuove

tabelle al catalogo dati, aggiornando le partizioni di tabella e aggiornando lo schema delle tabelle direttamente dal processo.

## Modifica del nodo di destinazione dati

La destinazione dati è la posizione in cui il processo scrive i dati trasformati.

Per aggiungere o configurare un nodo di destinazione dati nel diagramma di processo

1. (Facoltativo) Se devi aggiungere un nodo di destinazione, scegli Target (Destinazione) nella barra degli strumenti nella parte superiore dell'editor visivo, quindi scegli S3 o Glue Data Catalog.

   - Se scegli S3 per la destinazione, il processo scrive il set di dati in uno o più file nella posizione Amazon S3 specificata.
   - Se scegli AWS Glue Data Catalog per la destinazione, il processo scrive in una posizione descritta dalla tabella selezionata dal catalogo dati.
2. Scegli un nodo di destinazione dati nel diagramma del processo. Quando scegli un nodo, il pannello dei dettagli del nodo viene visualizzato sul lato destro della pagina.
3. Seleziona la scheda Node properties (Proprietà del nodo), quindi inserisci le informazioni riportate di seguito:

   - Name (Nome): inserisci un nome da associare al nodo nel diagramma del processo.
   - Node type (Tipo di nodo): dovrebbe essere già selezionato un valore, ma è possibile modificarlo in base alle necessità.
   - Node parents (Nodi padre): il nodo padre è il nodo nel diagramma del processo che fornisce i dati di output da scrivere nella posizione di destinazione. Per un diagramma di processo precompilato, il nodo di destinazione deve già avere il nodo padre selezionato. Se non è visualizzato alcun nodo padre, scegline uno dall'elenco.

     Un nodo di destinazione ha un singolo nodo padre.
4. Configura le informazioni di Data target properties (Proprietà della destinazione dati). Per ulteriori informazioni, consulta le sezioni seguenti:

   - Uso di Amazon S3 per la destinazione dati
   - Utilizzo delle tabelle del catalogo dati per la destinazione dati
   - Utilizzo di un connettore per la destinazione dati

5. (Facoltativo) Dopo aver configurato le proprietà del nodo di destinazione dati, puoi visualizzare lo schema di output per i dati scegliendo la scheda Output schema (Schema di output) nel pannello dei dettagli del nodo. La prima volta che si sceglie questa scheda per qualsiasi nodo del processo, viene richiesto di fornire un ruolo IAM per accedere ai dati. Se non è stato specificato un ruolo IAM nella scheda Job details (Dettagli del processo), viene richiesto di immettere un ruolo IAM a questo punto.

### Uso di Amazon S3 per la destinazione dati

Per tutte le origini dati ad eccezione di Amazon S3 e dei connettori, è necessario che esista una tabella in AWS Glue Data Catalog per il tipo di origine scelto. AWS Glue Studio non crea la tabella in Data Catalog.

Per configurare un nodo di destinazione dati che scrive su Amazon S3

1. Vai all'editor visivo per un processo nuovo o salvato.
2. Scegli un nodo di origine dati nel diagramma del processo.
3. Seleziona la scheda Data source properties (Proprietà dell'origine dati), quindi immetti le informazioni riportate di seguito:

   - Format (Formato): Scegli un formato dall'elenco. I tipi di formato disponibili per i risultati dei dati sono:
     - JSON: notazione JavaScript degli oggetti.
     - CSV: valori separati da virgola.
     - Avro: Apache Avro JSON binario.
     - Parquet: un tipo personalizzato di writer Parquet ottimizzato per `DynamicFrames`come formato dei dati. Anziché richiedere uno schema precalcolato per i dati, calcola e modifica lo schema in modo dinamico.
     - ORC: formato Apache Optimized Row Columnar (ORC).
     - Apache Hudi: un framework di archiviazione di data lake open source che semplifica l'elaborazione incrementale dei dati e lo sviluppo di pipeline di dati.
     - Apache Iceberg: un formato di tabella ad alte prestazioni che funziona proprio come una tabella SQL.
     - Delta Lake: un framework di archiviazione di data lake open source che consente di eseguire transazioni ACID, scalare la gestione dei metadati e unificare lo streaming e l'elaborazione dei dati in batch.

- XML: Extensible Markup Language (XML).
- Tableau Hyper: tecnologia del motore di dati in memoria di Tableau.

Per ulteriori informazioni su queste opzioni di formato, consulta Opzioni di formato per gli input e output ETL in AWS Glue nella Guida per gli sviluppatori di AWS Glue .

- Compression Type: puoi scegliere di comprimere i dati usando i tipi di file `CSV`, `JSON` o `Parquet`. L'impostazione predefinita non è alcuna compressione, oppure None (Nessuna).

| Tipo di file | Compressioni |
|---|---|
| JSON/CSV/XML | GZIP, BROTLI BZIP2, DEFLATE, Snappy LZ4 |
| Parquet | Snappy, LZO, BROTLI, GZIP LZ4 |
| ORC | Snappy, ZLIB, Uncompressed, LZO |
| Avro | GZIP, BROTLI, DEFLATE, BZIP2 Snappy LZ4 |
| Delta Lake | GZIP, BROTLI, DEFLATE, Snappy BZIP2 LZ4 |
| Apache Hudi | GZIP, LZO, Snappy |
| Apache Iceberg | GZIP, LZO, Snappy |
| Tableau Hyper | Nessuno |

- S3 Target Location (Posizione di destinazione S3): il bucket Amazon S3 e la posizione per l'output dei dati. Puoi selezionare il pulsante Browse S3 (Sfoglia S3) per visualizzare i bucket Amazon S3 a cui hai accesso e sceglierne uno come destinazione.
- Opzioni per l'aggiornamento del catalogo dati
  - Do not update the Data Catalog (Non aggiornare il catalogo dati): (impostazione predefinita) scegli questa opzione se non vuoi che il processo aggiorni il catalogo dati, anche se lo schema viene modificato o sono aggiunte nuove partizioni.
  - Create a table in the Data Catalog and on subsequent runs, update the schema and add new partitions (Crea una tabella nel catalogo dati e, nelle esecuzioni successive, aggiorna

lo schema e aggiungi nuove partizioni): se scegli questa opzione, il processo crea la tabella nel catalogo dati alla prima esecuzione. Nelle successive esecuzioni del processo, questo aggiorna la tabella del catalogo dati se lo schema viene modificato o sono aggiunte nuove partizioni.

Devi inoltre selezionare un database dal catalogo dati e inserire un nome di tabella.

- Create a table in the Data Catalog and on subsequent runs, keep existing schema and add new partitions (Crea una tabella nel catalogo dati e, nelle esecuzioni successive, mantieni lo schema esistente e aggiungi nuove partizioni): se scegli questa opzione, il processo crea la tabella nel catalogo dati alla prima esecuzione. Nelle successive esecuzioni del processo, questo aggiorna la tabella del catalogo dati solo per aggiungere nuove partizioni.

  Devi inoltre selezionare un database dal catalogo dati e inserire un nome di tabella.

- Partizionamento del file: scegli il tipo di partizionamento in cui vuoi salvare l'output.
  - Generazione automatica dei file (consigliato): è il valore predefinito per il numero di file generati.
  - Output di file multipli: specifica il numero di file di output che desideri. Per prestazioni ottimali, usa il valore predefinito generato automaticamente per il numero di file.
- Partition keys (Chiavi di partizione): scegli quali colonne utilizzare come chiavi di partizionamento nell'output. Per aggiungere altre chiavi di partizione, scegli Add a partition key (Aggiungi una chiave di partizione).

Il partizionamento dei file non è supportato per Tableau Hyper come formato di destinazione.

### Utilizzo delle tabelle del catalogo dati per la destinazione dati

Per tutte le origini dati ad eccezione di Amazon S3 e dei connettori, è necessario che esista una tabella in AWS Glue Data Catalog per il tipo di destinazione scelto. AWS Glue Studio non crea la tabella in Data Catalog.

Per configurare le proprietà dei dati per una destinazione che utilizza una tabella del catalogo dati

1. Vai all'editor visivo per un processo nuovo o salvato.
2. Scegli un nodo di destinazione dati nel diagramma del processo.
3. Seleziona la scheda Data target properties (Proprietà della destinazione dati), quindi inserisci le informazioni riportate di seguito:

- Database: scegli dall'elenco il database che contiene la tabella da utilizzare come destinazione. Questo database deve esistere già nel catalogo dati.
- Table (Tabella): scegli la tabella che definisce lo schema dei dati di output dall'elenco. Questa tabella deve esistere già nel catalogo dati.

  Una tabella nel catalogo dati contiene i nomi delle colonne, le definizioni dei tipi di dati, le informazioni sulle partizioni e altri metadati su un set di dati di destinazione. Il processo scrive in una posizione descritta da questa tabella nel catalogo dati.

  Per ulteriori informazioni sulla creazione di tabelle nel catalogo dati, consulta Definizione di tabelle nel Catalogo dati nella Guida per gli sviluppatori di AWS Glue .
- Opzioni per l'aggiornamento del catalogo dati
  - Do not change table definition (Non modificare la definizione della tabella): (impostazione predefinita) scegli questa opzione se non vuoi che il processo aggiorni il catalogo dati, anche se lo schema viene modificato o sono aggiunte nuove partizioni.
  - Update schema and add new partitions (Aggiorna lo schema e aggiungi nuove partizioni): se scegli questa opzione, il processo aggiorna la tabella del catalogo dati se lo schema viene modificato o sono aggiunte nuove partizioni.
  - Keep existing schema and add new partitions (Mantieni lo schema esistente e aggiungi nuove partizioni): se scegli questa opzione, il processo aggiorna la tabella del catalogo dati solo per aggiungere nuove partizioni.
  - Partition keys (Chiavi di partizione): scegli quali colonne utilizzare come chiavi di partizionamento nell'output. Per aggiungere altre chiavi di partizione, scegli Add a partition key (Aggiungi una chiave di partizione).

Utilizzo di un connettore per la destinazione dati

Se per Node type (Tipo di nodo) selezioni un connettore, segui le istruzioni in Creazione di processi con connettori personalizzati per completare la configurazione delle proprietà della destinazione dati.

## Modifica o caricamento di uno script del processo

Utilizzo dell'editor visivo AWS Glue Studio per modificare lo script del processo o caricare il proprio script.

È possibile utilizzare l'editor visivo per modificare i nodi di processo solo se i processi sono stati creati con AWS Glue Studio. Se il processo è stato creato utilizzando la console AWS Glue, tramite i comandi API o con l'interfaccia a riga di comando (CLI), puoi utilizzare l'editor di script in AWS Glue Studio per modificare lo script del processo, i parametri e la pianificazione. Puoi anche modificare lo script per un processo creato in AWS Glue Studio convertendo il processo in modalità solo script.

Per modificare lo script del processo o caricare il proprio script

1. Se crei un nuovo processo, nella pagina Jobs (Processi), seleziona l'opzione Spark script editor (Editor di script Spark) per creare un processo Spark o scegli l'opzione Python Shell script editor (Editor di script shell Python) per creare un processo shell Python. Puoi scrivere un nuovo script o caricare uno script esistente. Se scegli Spark script editor (Editor di script Spark), puoi scrivere o caricare uno script Scala o Python. Se scegli Python Shell script editor (Editor di script shell Python), puoi scrivere o caricare solo uno script Python.

   Dopo aver selezionato l'opzione per creare un nuovo processo, nella sezione Options (Opzioni) che appare, puoi scegliere di iniziare con uno script di inizio (Create a new script with boilerplate code [Crea un nuovo script con codice boilerplate]), oppure puoi caricare un file locale da utilizzare come script del processo.

   Se hai scelto Spark script editor (Editor di script Spark), puoi caricare un file script Python o Scala. Gli script Scala devono avere l'estensione di file `.scala`. Gli script Python devono essere riconosciuti come file di tipo Python. Se hai scelto Python Shell script editor (Editor di script shell Python), puoi caricare solo file di script Python.

   Una volta completate le scelte, seleziona Create (Crea) per creare il processo e aprire l'editor visivo.
2. Vai all'editor di processo visivo per il processo nuovo o salvato, quindi seleziona la scheda Script.
3. Se non hai creato un nuovo processo utilizzando una delle opzioni dell'editor di script e non hai mai modificato lo script per un processo esistente, la scheda Script mostra l'intestazione Script (Locked) (Script [bloccato]). Ciò significa che l'editor di script è in modalità di sola lettura. Scegli Edit script (Modifica script) per sbloccare lo script per la modifica.

   Per rendere lo script modificabile, AWS Glue Studio converte il processo da processo visivo a processo solo script. Sbloccando lo script per la modifica, non puoi più utilizzare l'editor visivo per questo processo dopo averlo salvato.

Nella finestra di conferma, scegli Confirm (Conferma) per continuare o Cancel (Annulla) per mantenere il processo disponibile per la modifica visiva.

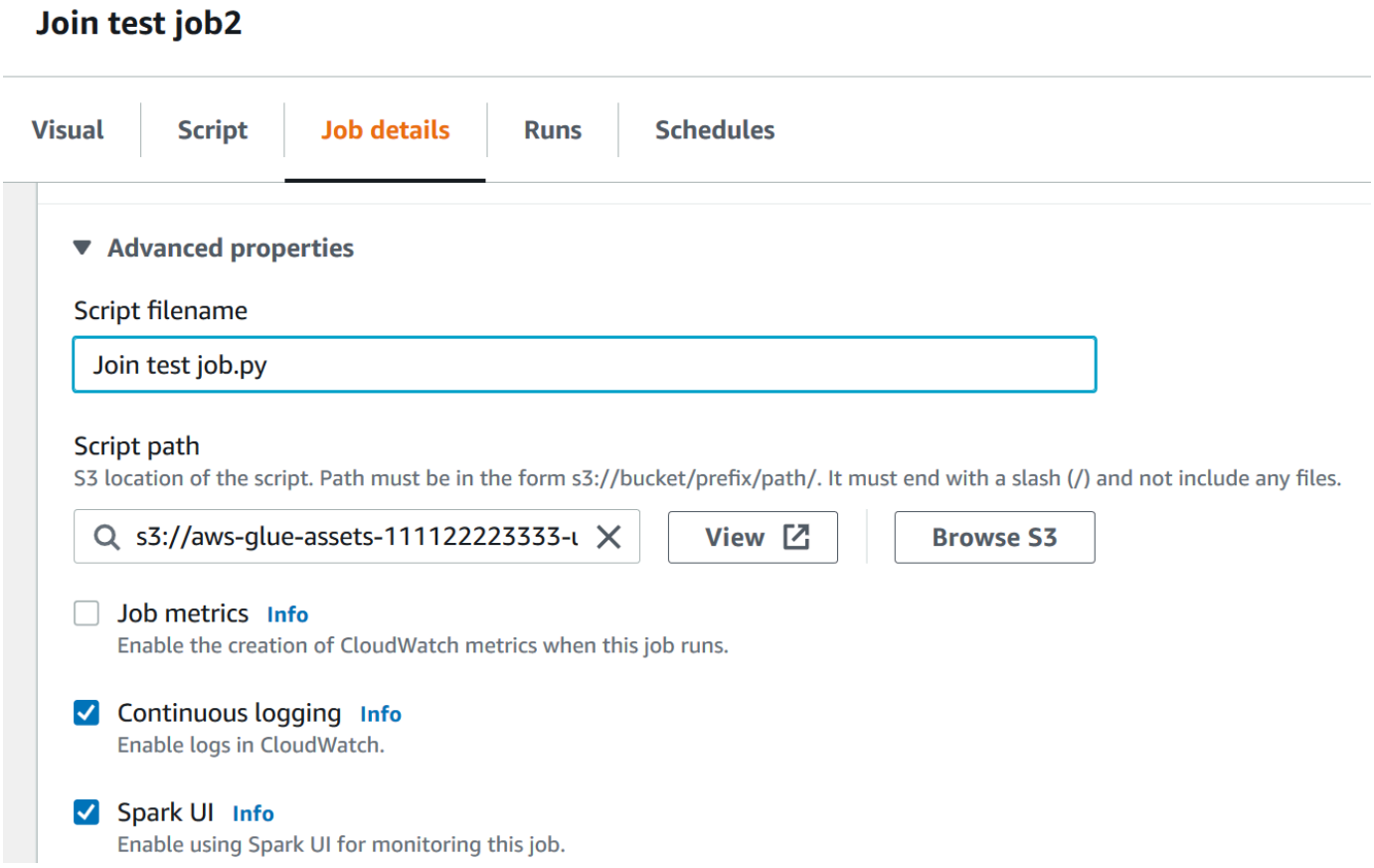
Scegliendo Confirm (Conferma), la scheda Visual (Visivo) non viene più mostrata nell'editor. Puoi utilizzare AWS Glue Studio per modificare lo script utilizzando l'editor di script, modificare i dettagli o la pianificazione del processo o visualizzarne le esecuzioni.

> **Note**
>
> Fino a quando non salvi il processo, la conversione in un processo solo script non è permanente. Se aggiorni la pagina Web della console o chiudi il processo prima di salvarlo e lo riapri nell'editor visivo, potrai ancora modificare i singoli nodi nell'editor visivo.

4. Modifica lo script in base alle esigenze.

   Dopo aver modificato lo script, seleziona Save (Salva) per salvare il processo e convertirlo in modo permanente da visivo a solo script.

5. (Facoltativo) Puoi scaricare lo script dalla console AWS Glue Studio selezionando il pulsante Download (Scarica) nella scheda Script. Selezionando questo pulsante, si apre una nuova finestra del browser che mostra lo script dalla sua posizione in Amazon S3. I parametri Script filename (Nome del file di script) e Script path (Percorso dello script) nella scheda del processo Job details (Dettagli del processo) determinano il nome e la posizione del file di script in Amazon S3.




Quando salvi il processo, AWS Glue salva lo script del processo nella posizione specificata da questi campi. Se modifichi il file di script in questa posizione all'interno di Amazon S3, AWS Glue Studio caricherà lo script modificato alla successiva modifica del processo.

## Creazione e modifica di script Scala in AWS Glue Studio

Quando scegli l'editor di script per la creazione di un processo, per impostazione predefinita, il linguaggio di programmazione dei processi è impostato su `Python 3`. Se scegli di scrivere un nuovo script invece di caricarne uno, AWS Glue Studio avvia un nuovo script con testo boilerplate scritto in Python. Se invece vuoi scrivere uno script Scala, devi prima configurare l'editor di script per utilizzare Scala.

> **Note**
>
> Se scegli Scala come linguaggio di programmazione per il processo e usi l'editor visivo per progettare il processo, lo script del processo generato viene scritto in Scala e non sono necessarie ulteriori azioni.

Come scrivere un nuovo script Scala in AWS Glue Studio

1. Crea un nuovo processo scegliendo l'opzione Spark script editor (Editor di script Spark).
2. Sotto Options (Opzioni), scegli Create a new script with boilerplate code (Crea un nuovo script con codice boilerplate).
3. Seleziona Job details (Dettagli del processo) e imposta Language (Linguaggio) su `Scala` (invece di `Python 3`).

   > **Note**
   >
   > La proprietà Type (Tipo) per il processo viene automaticamente impostata su `Spark` quando scegli l'opzione Spark script editor (Editor di script Spark) per creare un processo.

4. Seleziona la scheda Script.
5. Rimuovi il testo boilerplate Python. Puoi sostituirlo con il seguente testo boilerplate Scala.

```
import com.amazonaws.services.glue.{DynamicRecord, GlueContext}
import org.apache.spark.SparkContext
import com.amazonaws.services.glue.util.JsonOptions
import com.amazonaws.services.glue.util.GlueArgParser
import com.amazonaws.services.glue.util.Job

object MyScript {
  def main(args: Array[String]): Unit = {
    val sc: SparkContext = new SparkContext()
    val glueContext: GlueContext = new GlueContext(sc)

    }
}
```

6. Scrivi lo script del processo Scala nell'editor. Aggiungi ulteriori istruzioni `import` in base alle esigenze.

## Creazione e modifica di processi shell Python in AWS Glue Studio

Scegliendo l'editor di script shell Python per la creazione di un processo, puoi caricare uno script Python esistente o scriverne uno nuovo. Se scegli di scrivere un nuovo script, il codice boilerplate viene aggiunto al nuovo script del processo Python.

Per creare un nuovo processo shell Python

Fai riferimento alle istruzioni riportate in Avvio di processi in AWS Glue Studio.

Le proprietà del processo supportate per i processi shell Python non sono le stesse supportate per i processi Spark. Nell'elenco seguente vengono descritte le modifiche ai parametri di processo disponibili per i processi shell Python nella scheda Job details (Dettagli del processo).

- La proprietà Type (Tipo) per il processo viene automaticamente impostata su `Python Shell` e non può essere modificata.
- Invece di Language (Linguaggio), è presente la proprietà Python version (Versione di Python) per il processo. Al momento, i processi shell Python creati in AWS Glue Studio utilizzano Python 3.6.
- La proprietà Glue version (Versione Glue) non è disponibile, perché non applicabile ai processi shell Python.
- Invece di Worker type (Tipo di worker) e Number of workers (Numero di worker), è mostrata la proprietà Data processing units (Unità di elaborazione dati). Questa proprietà del lavoro determina quante unità di elaborazione dati (DPUs) vengono utilizzate dalla shell Python durante l'esecuzione del lavoro.
- La proprietà Job bookmark (Segnalibro del processo) non è disponibile, perché non è supportata per i processi shell Python.
- Sotto Advanced properties (Proprietà avanzate), le seguenti proprietà non sono disponibili per i processi shell Python.
  - Parametri del processo
  - Registrazione continua
  - Spark UI (Interfaccia utente di Spark) e Spark UI logs path (Percorso dei log dell'interfaccia utente Spark)
  - Dependent jars path (Percorso file .jar dipendente), sotto la voce Libraries (Librerie).

## Modifica dei nodi padre per un nodo nel diagramma del processo

Puoi modificare i nodi padre di un nodo per spostare i nodi all'interno del diagramma del processo o per modificare un'origine dati per un nodo.

Per modificare il nodo padre

1. Scegli il nodo nel diagramma del processo da modificare.

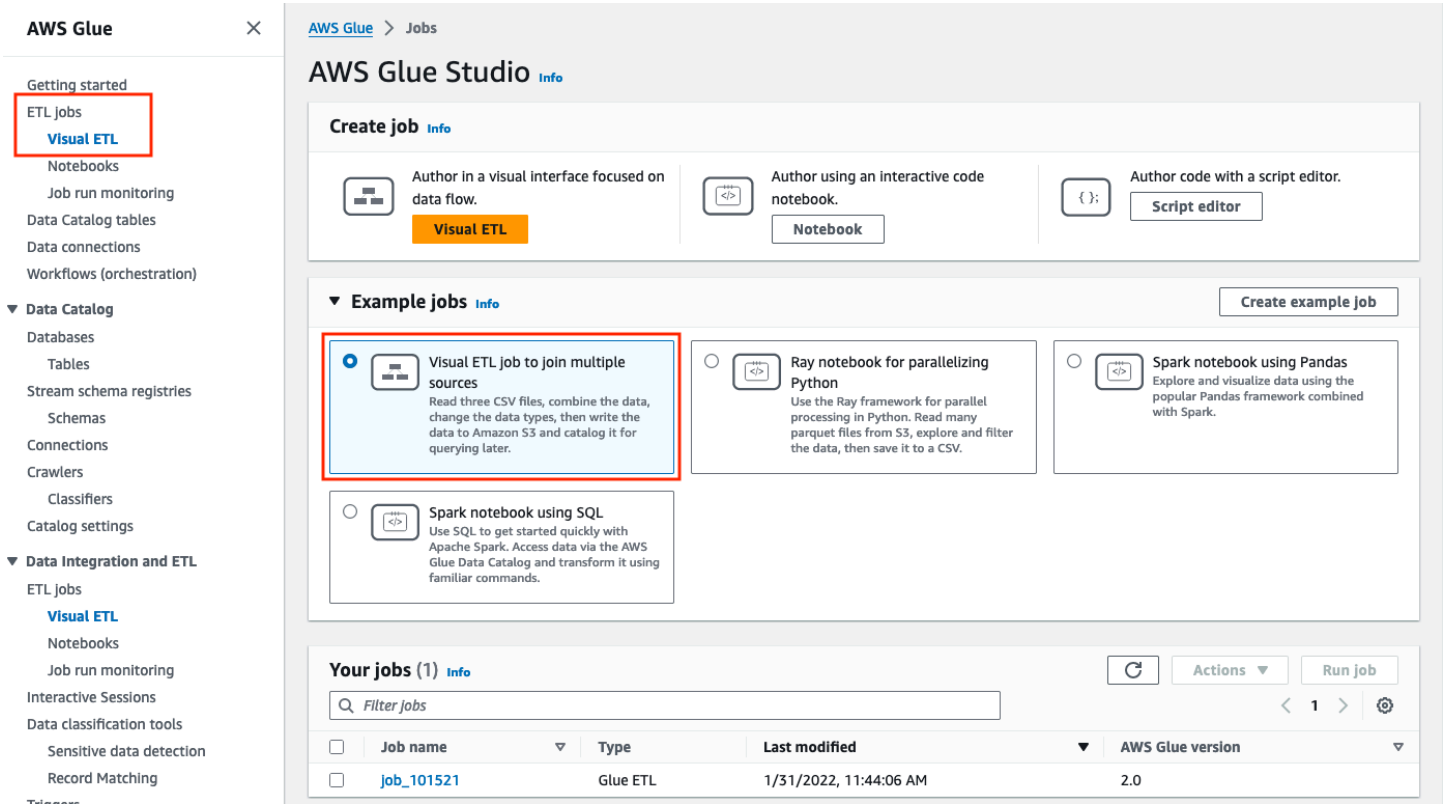
2. Nel pannello dei dettagli del nodo, nella scheda Node properties (Proprietà del nodo), sotto l'intestazione Node parents (Nodi padre), rimuovi l'attuale padre per il nodo.
3. Scegli un nuovo nodo padre dall'elenco.
4. Modifica le altre proprietà del nodo in base alle esigenze in modo che corrispondano al nodo padre appena selezionato.

Se hai modificato un nodo per errore, puoi utilizzare il pulsante Undo (Annulla) nella barra degli strumenti per invertire l'operazione.

## Eliminazione di nodi dal diagramma del processo

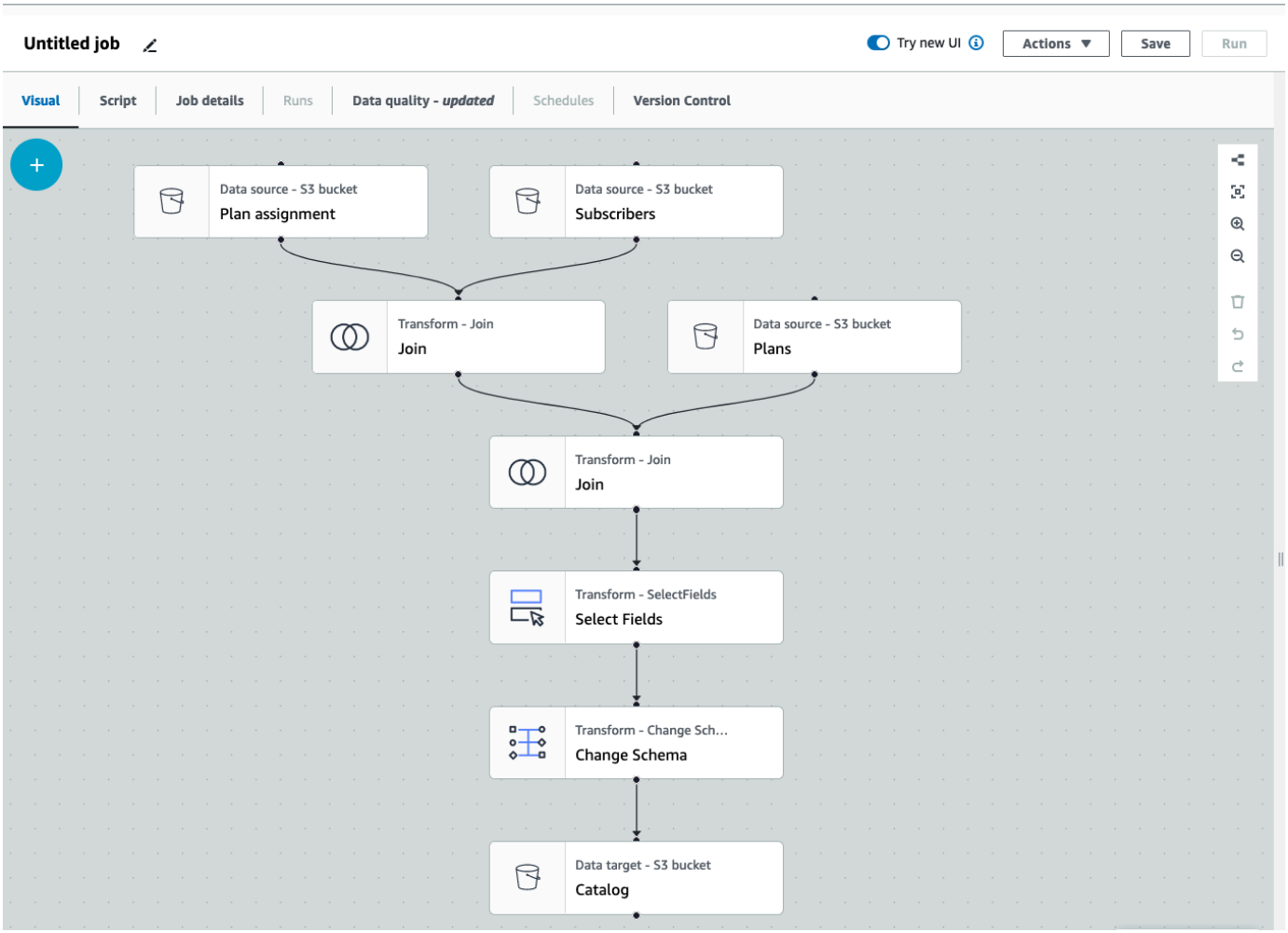
Quando si lavora con i processi ETL visivo, è possibile rimuovere i nodi dall'area di lavoro senza dover aggiungere nuovamente o ristrutturare i nodi collegati al nodo rimosso.

Nell'esempio seguente, puoi seguire la procedura selezionando Processo ETL > ETL visivo, quindi andando su Processi di esempio e scegliendo Processo ETL visivo per eseguire il join di più origini. Seleziona Crea processo di esempio per creare un processo e segui la procedura riportata di seguito.



Per rimuovere un nodo dall'area di lavoro

1. Dalla AWS Glue console, scegli Visual ETL dal menu di navigazione e scegli un lavoro esistente. L'area di lavoro del processo mostra l'esempio di processo illustrato di seguito.

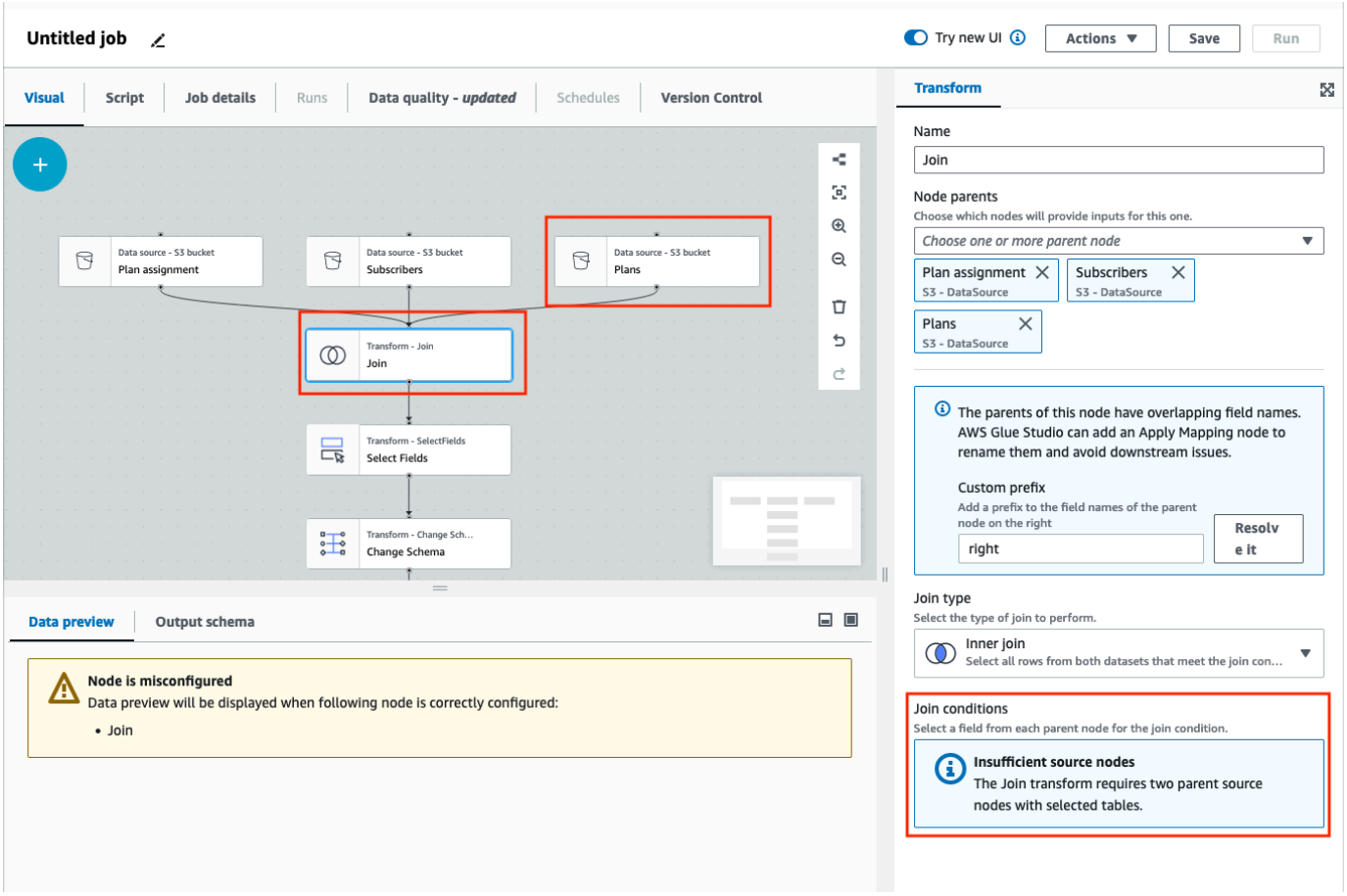



2. Scegli il nodo da rimuovere. L'area di lavoro ingrandirà il nodo. Nella barra degli strumenti sul lato destro dell'area di lavoro, seleziona l'icona del Cestino. In questo modo il nodo verrà rimosso e qualsiasi altro nodo collegato a esso si sposterà per prendere il suo posto nel flusso di lavoro. In questo esempio, il primo nodo Join (Unisci) è stato eliminato dall'area di lavoro.

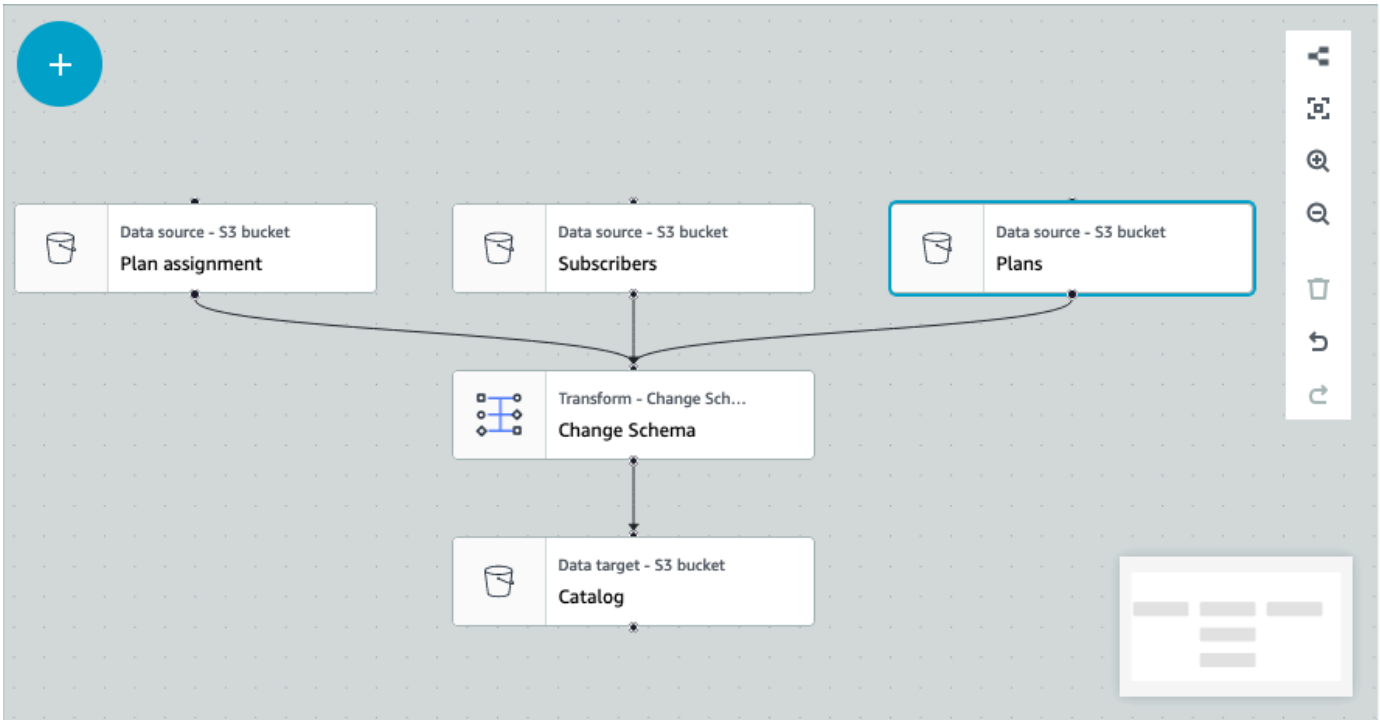
   Se elimini un nodo dal flusso di lavoro, AWS Glue riorganizzerà i nodi in modo che siano organizzati in modo da non creare un flusso di lavoro non valido. Potrebbe essere comunque necessario correggere la configurazione di un nodo.

   Nell'esempio, è stato rimosso il nodo Join (Unisci) sotto il nodo Subscribers (Abbonati). Di conseguenza, il nodo sorgente Plans (Piani) è stato spostato al livello superiore ed è ancora connesso al nodo secondario Join. Adesso il nodo Join (Unisci) richiede una configurazione

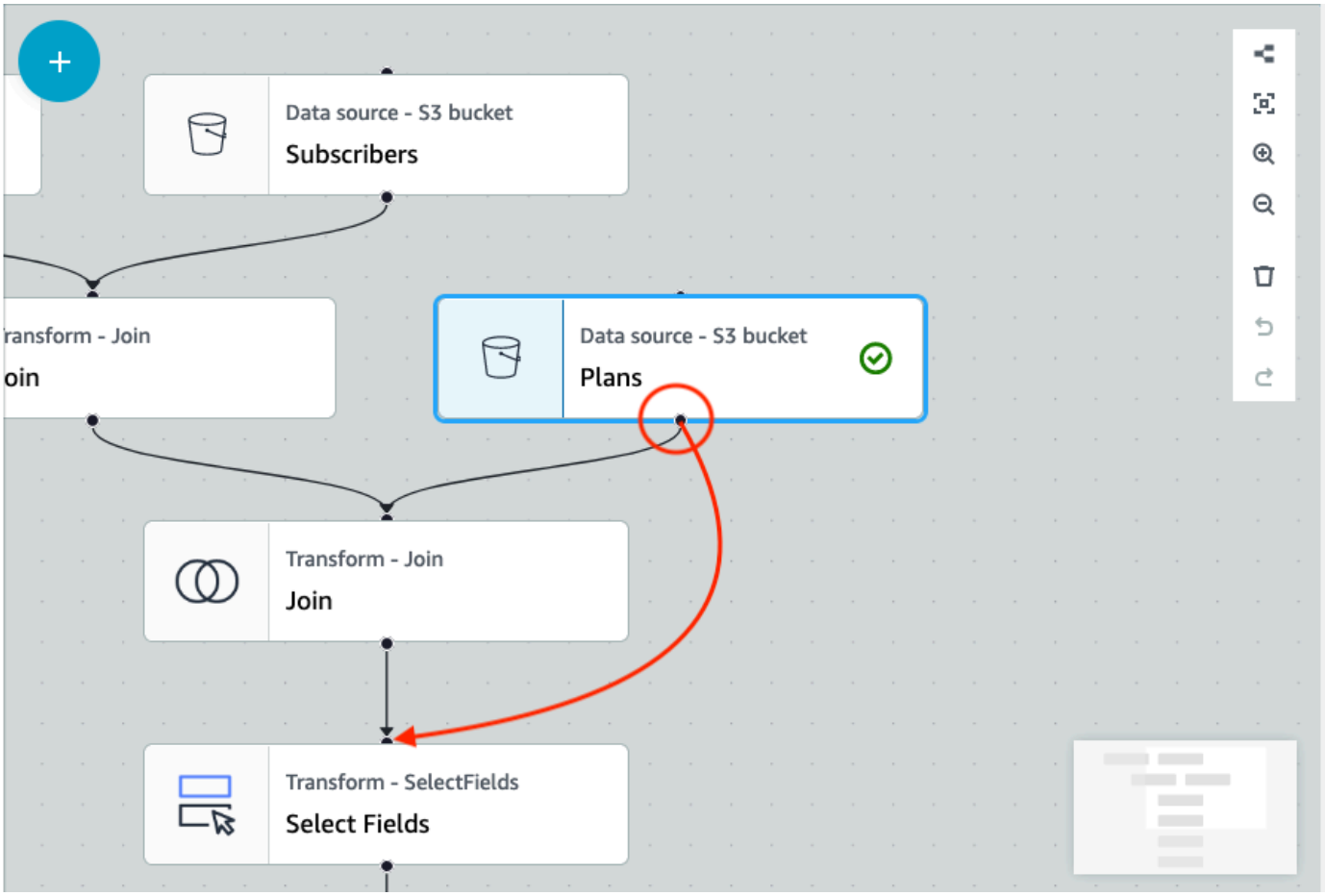
aggiuntiva in quanto Join richiede due nodi di origine con tabelle selezionate. La scheda Transform (Trasforma) a destra dell'area di lavoro mostra il requisito mancante Join conditions (Unisci condizioni).




3. Elimina il secondo nodo Join (Unisci) e il nodo Select Fields (Seleziona campi). Una volta eliminati i nodi, il flusso di lavoro apparirà come nell'esempio riportato di seguito.




4. Per modificare le connessioni dei nodi, fai clic sul pallino del nodo e trascina la connessione su un nuovo nodo. Ciò consentirà di eliminare i nodi e di riorganizzarli in un flusso logico. Nell'esempio, viene creata una nuova connessione facendo clic sulla maniglia nel nodo Plans (Piani) e trascinando la connessione sul nodo Join (Unisci), come indicato dalla freccia rossa.




5. Se devi annullare un'azione, seleziona l'icona Annulla direttamente sotto l'icona del Cestino nella barra degli strumenti sul lato destro dell'area di lavoro.

## Aggiungendo parametri di origine e destinazione al nodo AWS Glue Data Catalog

AWS Glue Studio consente di parametrizzare i processi visivi. Poiché i nomi delle tabelle di catalogo nell'ambiente di produzione e sviluppo possono essere diversi, è possibile definire e selezionare i parametri di runtime per database e tabelle che verranno eseguiti durante l'esecuzione del processo.

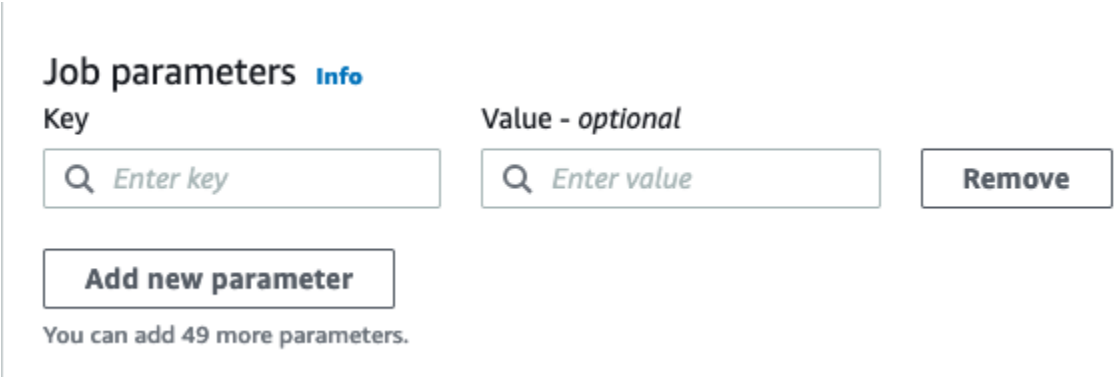
La parametrizzazione dei processi consente di parametrizzare origini e destinazioni e di salvare tali parametri sul processo quando usi il nodo AWS Glue Data Catalog. Quando si specificano origini e destinazioni come parametri, si abilita la riutilizzabilità dei processi, in particolare quando si utilizza lo stesso processo in più ambienti. Questo è utile quando si promuove il codice negli ambienti di implementazione, risparmiando tempo e fatica nella gestione delle origini e delle destinazioni. Inoltre,

i parametri personalizzati specificati sostituiranno qualsiasi argomento predefinito per esecuzioni specifiche di processi AWS Glue.

Aggiungere parametri di origine e destinazione

Sia che si stia usando il nodo AWS Glue Data Catalog come origine o destinazione, puoi definire i parametri di runtime nella sezione Proprietà avanzate nella scheda Dettagli del processo.

1. Scegli il nodo AWS Glue Data Catalog come nodo di origine o nodo di destinazione.
2. Seleziona la scheda Job details (Dettagli del processo).
3. Scegli Proprietà avanzate.
4. Nella sezione Parametri del processo, inserisci un valore chiave. Ad esempio,`--db.source` sarebbe il parametro per un database di origine. Puoi inserire qualsiasi nome come chiave, purché il nome della chiave sia seguito da "trattino trattino".

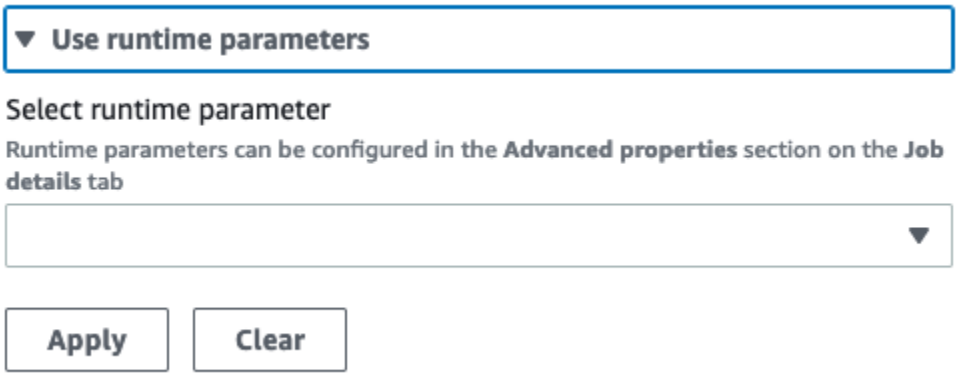



5. Inserire il valore. Ad esempio, `databasename` sarebbe il valore per la parametrizzazione del database.
6. Scegli Aggiunta nuovo parametro se si desidera aggiungere altri parametri. Sono consentiti fino a 50 parametri. Una volta definita la coppia di valori chiave, puoi utilizzare il parametro nel nodo AWS Glue Data Catalog.

Selezionare un parametro di runtime

> **Note**
>
> Il processo di selezione dei parametri di runtime per database e tabelle è lo stesso indipendentemente dal fatto che il nodo AWS Glue Data Catalog sia l'origine o la destinazione.

1. Scegli il nodo AWS Glue Data Catalog come nodo di origine o nodo di destinazione.
2. Nella scheda Proprietà dell'origine dei dati: Data Catalog , sotto Database, scegli Usa parametri di runtime.




3. Scegli un parametro dal menu a discesa. Ad esempio, quando si seleziona un parametro definito per un database di origine, il database verrà inserito automaticamente nel menu a discesa del database quando si sceglie Applicazione.
4. Nella sezione Tabella, scegli un parametro già definito come tabella di origine. Quando si sceglie Applicazione, la tabella viene inserita automaticamente come tabella da utilizzare.
5. Quando si salva e si esegue il processo, AWS Glue Studio farà riferimento ai parametri selezionati durante l'esecuzione del processo.

## Utilizzo dei sistemi di controllo delle versioni Git in AWS Glue

> **Note**
>
> I notebook non sono attualmente supportati per il controllo della versione in AWS Glue Studio. Tuttavia, è supportato il controllo della versione per gli script del processo AWS Glue e i processi ETL visivi.

Se disponi di repository remoti e desideri gestire i tuoi AWS Glue lavori utilizzando i tuoi repository, puoi utilizzare AWS Glue Studio o AWS CLI per sincronizzare le modifiche ai tuoi repository e ai tuoi lavori in. AWS Glue Quando si sincronizzano le modifiche in questo modo, si sta inviando il processo da AWS Glue Studio al repository o lo si sta estraendo dal repository verso AWS Glue Studio.

Grazie all'integrazione di Git in AWS Glue Studio, è possibile:

- Integrazione con i sistemi di controllo delle versioni Git AWS CodeCommit, come, GitHub GitLab, e Bitbucket
- Modifica i processi AWS Glue in AWS Glue Studio indipendentemente dal fatto che utilizzi processi visivi o processi di script e li sincronizzi con un repository
- Parametrizza le origini e le destinazioni nei processi
- Estrai i processi da un repository e modificali in AWS Glue Studio
- Metti alla prova i lavori passando dalle filiali alle filiali and/or utilizzando flussi di lavoro multifiliale in AWS Glue Studio
- Scarica file da un repository e carica processi in AWS Glue Studio per la creazione di processi tra account
- Utilizza lo strumento di automazione che preferisci (ad esempio, Jenkins, ecc.) AWS CodeDeploy

Questo video dimostra come integrare AWS Glue con Git e creare una pipeline di codice continua e collaborativa.

## autorizzazioni IAM

Verifica che il processo abbia una delle seguenti autorizzazioni IAM. Per ulteriori informazioni su come configurare le autorizzazioni IAM, consulta Impostare le autorizzazioni IAM per AWS Glue Studio.

- `AWSGlueServiceRole`
- `AWSGlueConsoleFullAccess`

Come minimo, sono necessarie le seguenti operazioni per l'integrazione con Git:

- `glue:UpdateJobFromSourceControl` — essere in grado di aggiornare AWS Glue con un processo presente in un sistema di controllo delle versioni
- `glue:UpdateSourceControlFromJob` — essere in grado di aggiornare il sistema di controllo delle versioni con un processo archiviato in AWS Glue
- `s3:GetObject` — essere in grado di recuperare lo script per il processo mentre si invia al sistema di controllo delle versioni
- `s3:PutObject` — essere in grado di aggiornare lo script quando si estrae un processo da un sistema di controllo dell'origine

## Prerequisiti

Per poter inviare i processi a un repository di controllo del codice sorgente, avrai bisogno di:

- un repository che è già stato creato dall'amministratore
- un ramo nel repository
- un token di accesso personale (per Bitbucket, questo è il Repository Access Token)
- il nome utente del proprietario del repository
- impostare le autorizzazioni nel repository per consentire ad AWS Glue Studio la lettura e la scrittura nel repository
  - GitLab— imposta gli ambiti dei token su api, read_repository e write_repository
  - Bitbucket: imposta le autorizzazioni su:
    - Iscrizione a Workspace: lettura e scrittura
    - Progetti: scrittura, lettura e amministratore
    - Repository: lettura, scrittura, eliminazione e amministratore

> **Note**
>
> Durante l'utilizzo AWS CodeCommit, non sono necessari il token di accesso personale e il proprietario del repository. Consulta Introduzione a Git e AWS CodeCommit.

Utilizzo dei processi dal tuo repository di controllo del codice sorgente in AWS Glue Studio

Per poter estrarre un processo dal repository per il controllo di origine che non è presente in AWS Glue Studio e utilizzare quel processo in AWS Glue Studio, i prerequisiti dipenderanno dal tipo di processo.

Per i processi visivi:

- sono necessari una cartella e un file JSON della definizione del processo che corrisponda al nome del processo

  Ad esempio, vedi la definizione del processo di seguito. Il ramo nel repository dovrebbe contenere un percorso `my-visual-job/my-visual-job.json` dove sia la cartella che il file JSON corrispondono al nome del processo

```
{
  "name" : "my-visual-job",
  "description" : "",
  "role" : "arn:aws:iam::aws_account_id:role/Rolename",
  "command" : {
    "name" : "glueetl",
    "scriptLocation" : "s3://foldername/scripts/my-visual-job.py",
    "pythonVersion" : "3"
  },
  "codeGenConfigurationNodes" : "{\"node-nodeID\":{\"S3CsvSource\":
{\"AdditionalOptions\":{\"EnableSamplePath\":false,\"SamplePath\":\"s3://notebook-
test-input/netflix_titles.csv\"},\"Escaper\":\"\",\"Exclusions\":[],\"Name\":\"Amazon
 S3\",\"OptimizePerformance\":false,\"OutputSchemas\":[{\"Columns\":[{\"Name\":
\"show_id\",\"Type\":\"string\"},{\"Name\":\"type\",\"Type\":\"string\"},{\"Name\":
\"title\",\"Type\":\"choice\"},{\"Name\":\"director\",\"Type\":\"string\"},{\"Name\":
\"cast\",\"Type\":\"string\"},{\"Name\":\"country\",\"Type\":\"string\"},{\"Name\":
\"date_added\",\"Type\":\"string\"},{\"Name\":\"release_year\",\"Type\":\"bigint\"},
{\"Name\":\"rating\",\"Type\":\"string\"},{\"Name\":\"duration\",\"Type\":\"string
\"},{\"Name\":\"listed_in\",\"Type\":\"string\"},{\"Name\":\"description\",\"Type
\":\"string\"}]}],\"Paths\":[\"s3://dalamgir-notebook-test-input/netflix_titles.csv
\"],\"QuoteChar\":\"quote\",\"Recurse\":true,\"Separator\":\"comma\",\"WithHeader
\":true}}}"
}
```

Per i processi di script:

- hai bisogno di una cartella, un file JSON con la definizione del processo e lo script
- la cartella e il file JSON devono corrispondere al nome del processo. Il nome dello script deve corrispondere alla `scriptLocation` nella definizione del processo insieme all'estensione del file

  Ad esempio, nella definizione del processo riportata di seguito, il ramo nel repository deve contenere un percorso `my-script-job/my-script-job.json` e `my-script-job/my-script-job.py`. Il nome dello script deve corrispondere al nome nella `scriptLocation` inclusa l'estensione dello script

```
{
  "name" : "my-script-job",
  "description" : "",
  "role" : "arn:aws:iam::aws_account_id:role/Rolename",
```

```
  "command" : {
    "name" : "glueetl",
    "scriptLocation" : "s3://foldername/scripts/my-script-job.py",
    "pythonVersion" : "3"
  }
}
```

## Limitazioni

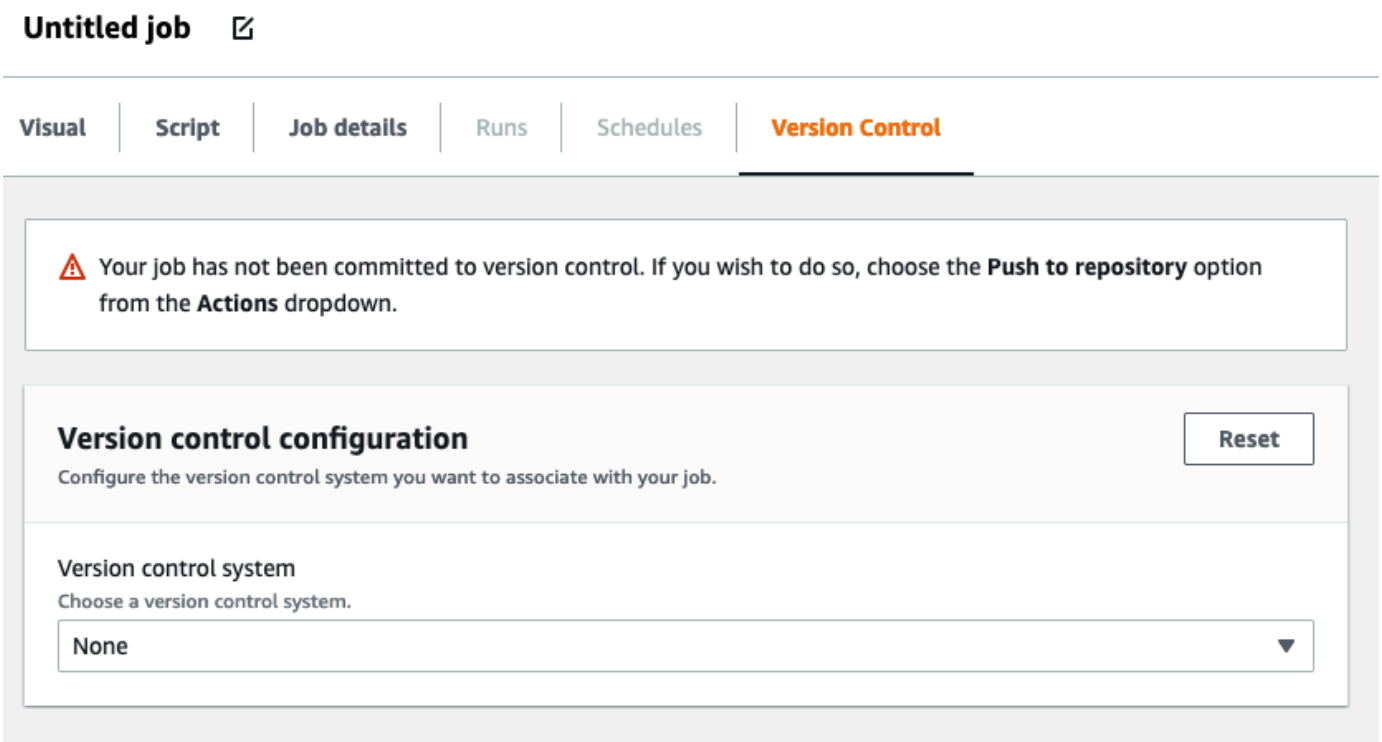
- AWS Glueattualmente non supporta i gruppi pushing/pulling GitLab-Groups.

## Collegamento dei repository di controllo delle versioni a AWS Glue

È possibile inserire i dettagli del repository di controllo delle versioni e gestirli nella scheda Controllo delle versioni nell'editor dei processi AWS Glue Studio. Per integrarlo con il repository Git, bisogna connettersi al repository ogni volta che si accede a AWS Glue Studio.

Per connettere un sistema di controllo della versione Git:

1. In AWS Glue Studio, inizia un nuovo processo e scegli la scheda Controllo delle versioni.



2. In Sistema di controllo delle versioni, scegli Git Service tra le opzioni disponibili facendo clic sul menu a discesa.

   - AWS CodeCommit
   - GitHub
   - GitLab
   - Bitbucket

3. A seconda del sistema di controllo delle versioni Git scelto, si avranno diversi campi da completare.

   Per AWS CodeCommit:

   Completa la configurazione del repository selezionando il repository e il ramo per il processo:

   - Repository: se hai configurato dei repository in AWS CodeCommit, seleziona il repository dal menu a discesa. I repository verranno inseriti automaticamente nell'elenco
   - Ramo — seleziona il ramo dal menu a discesa
   - Cartella — facoltativo: inserisci il nome della cartella in cui salvare il processo. Se lasciato vuoto, viene creata automaticamente una cartella. Il nome predefinito della cartella è il nome del processo

   Per: GitHub

   Completa la GitHub configurazione completando i campi:

   - Token di accesso personale: si tratta del token fornito dal GitHub repository. Per ulteriori informazioni sui token di accesso personali, consulta Docs GitHub
   - Proprietario del repository: è il proprietario del repository. GitHub

   Completa la configurazione del repository selezionando il repository e il ramo da. GitHub

   - Repository: se hai configurato dei repository in GitHub, seleziona il repository dal menu a discesa. I repository verranno inseriti automaticamente nell'elenco
   - Ramo — seleziona il ramo dal menu a discesa

- Cartella — facoltativo: inserisci il nome della cartella in cui salvare il processo. Se lasciato vuoto, viene creata automaticamente una cartella. Il nome predefinito della cartella è il nome del processo

Per: GitLab

> **Note**
>
> AWS Glueattualmente non supporta pushing/pulling i GitLabgruppi -Groups.

- Token di accesso personale: questo è il token fornito dal GitLab repository. Per ulteriori informazioni sui token di accesso personali, consulta Token di accesso GitLab personali
- Proprietario del repository: si tratta del proprietario del repository. GitLab

Completa la configurazione del repository selezionando il repository e il ramo da. GitLab

- Repository: se hai configurato dei repository in GitLab, seleziona il repository dal menu a discesa. I repository verranno inseriti automaticamente nell'elenco
- Ramo — seleziona il ramo dal menu a discesa
- Cartella — facoltativo: inserisci il nome della cartella in cui salvare il processo. Se lasciato vuoto, viene creata automaticamente una cartella. Il nome predefinito della cartella è il nome del processo

Per Bitbucket:

- Password dell'app: Bitbucket usa le password dell'app e non i token di accesso al repository. Per maggiori informazioni sulle password delle app, consulta Password delle app.
- Proprietario del repository: questo è il proprietario del repository Bitbucket. In Bitbucket, il proprietario è il creatore del repository.

Completa la configurazione del repository selezionando il workspace, il repository, il ramo e la cartella da Bitbucket.

- Workspace: se sono stati configurati i workspace in Bitbucket, selezionare il workspace dal menu a discesa. I workspace vengono popolati automaticamente
- Repository: se sono stati configurati i repository in Bitbucket, selezionare il repository dal menu a discesa. I repository vengono popolati automaticamente
- Ramo: seleziona il ramo dal menu a discesa. I tuoi rami vengono popolati automaticamente
- Cartella — facoltativo: inserisci il nome della cartella in cui salvare il processo. Se lasciato vuoto, viene creata automaticamente una cartella con il nome del processo.

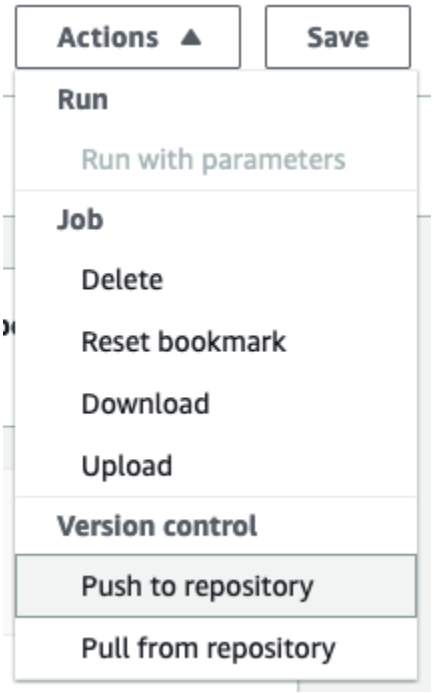
4. Scegli Save (Salva) nella parte superiore del processo AWS Glue Studio

## Invio dei processi AWS Glue al repository di origine

Dopo aver inserito i dettagli del sistema di controllo delle versioni, puoi modificare i processi in AWS Glue Studio e inviarli al repository di origine. Se non conosci i concetti di Git come inviare ed estrarre, guarda questo tutorial Nozioni di base su Git e AWS CodeCommit.

Per inviare il processo a un repository, devi inserire i dettagli del sistema di controllo delle versioni e salvare il processo.

1. Nel processoAWS Glue Studio, scegli Operazioni. Questo aprirà opzioni di menu aggiuntive.




2. Scegli Invia a repository.

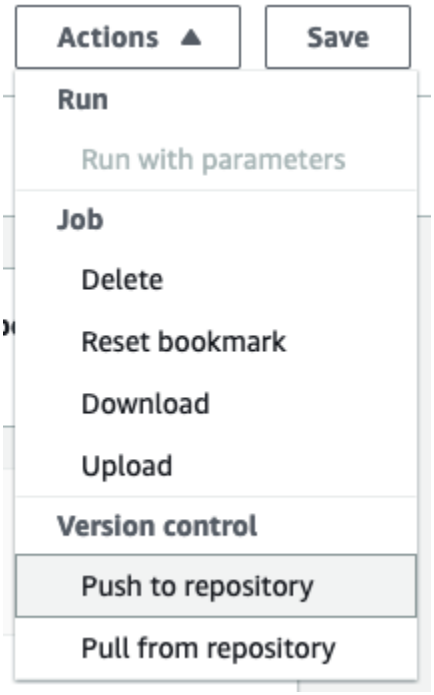
Questa operazione salverà il processo. Quando si esegue l'invio al repository,AWS Glue Studio invia l'ultima modifica salvata. Se il processo nel repository è stato modificato da te o da un altro utente e non è sincronizzato con il processo in AWS Glue Studio, quando si invia il processo da AWS Glue Studio, il processo nel repository viene sovrascritto con il processo salvato in AWS Glue Studio.

3. Scegli Conferma per completare l'operazione. Questo crea un nuovo commit nel repository. Se lo stai utilizzando AWS CodeCommit, un messaggio di conferma mostrerà un link all'ultimo commit on. AWS CodeCommit

## Estrarre processi AWS Glue dal repository di origine

Dopo aver inserito i dettagli del repository Git nella scheda Controllo delle versioni, puoi anche estrarre i processi dal repository e modificarli in AWS Glue Studio.

1. Nel processo AWS Glue Studio, scegli Operazioni. Questo aprirà opzioni di menu aggiuntive.




2. Scegli Estrai dal repository.
3. Scegli Conferma. Questo prende il commit più recente dal repository e aggiorna il processo in AWS Glue Studio.
4. Modificare il processo in AWS Glue Studio. Se apporti delle modifiche, puoi sincronizzare il processo con il repository scegliendo Invia a repository dal menu a discesa Operazioni.

# Creazione di codice con notebook AWS Glue Studio

I data engineer possono creare processi AWS Glue più velocemente e più facilmente rispetto a prima utilizzando l'interfaccia interattiva per notebook in AWS Glue Studio o le sessioni interattive in AWS Glue.

## Limitazioni

- AWS Glue Studio i notebook non supportano Scala.

Argomenti


- Panoramica sull'utilizzo dei notebook
- Creazione di un processo ETL utilizzando i notebook in AWS Glue Studio
- Componenti per l'editor del notebook
- Salvataggio del notebook e dello script del processo
- Gestione delle sessioni di notebook
- Utilizzo di Amazon Q Developer con AWS Glue Studio notebook


## Panoramica sull'utilizzo dei notebook

AWS Glue Studio permette di creare processi in modo interattivo in un'interfaccia notebook basata su Jupyter Notebooks. Attraverso i notebook in AWS Glue Studio, è possibile modificare gli script di processo e visualizzare l'output senza dover eseguire un processo completo e modificare il codice di integrazione dei dati e visualizzare l'output senza dover eseguire un processo completo. Inoltre, è possibile aggiungere markdown e salvare i notebook come file con estensione .ipynb e script di processo. È possibile avviare un notebook senza installare software localmente o gestire server. Quando hai terminato di lavorare con il codice, AWS Glue Studio può convertire il notebook in un processo Glue con un semplice clic.

Alcuni dei vantaggi derivanti dall'utilizzo dei notebook sono:

- Nessun cluster di cui effettuare il provisioning o da gestire
- Nessun cluster inattivo da pagare
- Nessuna configurazione iniziale richiesta
- Non è richiesta l'installazione di notebook Jupyter

- Lo stesso runtime/platform di AWS Glue ETL

All'avvio di un notebook tramite AWS Glue Studio, tutti i passaggi di configurazione vengono eseguiti per te, in modo che tu possa esplorare i dati e iniziare a sviluppare lo script del processo dopo pochi secondi. AWS Glue Studio configura un notebook Jupyter con il kernel AWS Glue Jupyter. Non è necessario configurare VPCs connessioni di rete o endpoint di sviluppo per utilizzare questo notebook.

Per creare processi utilizzando l'interfaccia notebook:

- configura le autorizzazioni IAM necessarie
- avvia una sessione notebook per creare un processo
- scrivi codice nelle celle del notebook
- esegui e testa il codice per visualizzare l'output
- salva il processo

Dopo aver salvato il notebook, questo sarà un processo AWS Glue completo. È possibile gestire tutti gli aspetti del processo, ad esempio la pianificazione delle esecuzioni e l'impostazione dei parametri del processo e la visualizzazione della cronologia dell'esecuzione del processo direttamente accanto al notebook.

## Creazione di un processo ETL utilizzando i notebook in AWS Glue Studio

Per iniziare a utilizzare i notebook nella console AWS Glue Studio

1. Allega AWS Identity and Access Management le policy all'AWS Glue Studioutente e crea un ruolo IAM per il tuo job e notebook ETL.
2. Configura la sicurezza IAM aggiuntiva per notebook, come descritto in Concessione di autorizzazioni per il ruolo IAM.
3. Apri la console AWS Glue Studio all'indirizzo https://console.aws.amazon.com/gluestudio/.

> **Note**
>
> Verifica che il tuo browser non blocchi i cookie di terzi. Qualsiasi browser che blocca i cookie di terze parti per impostazione predefinita o abilitata dall'utente impedirà l'avvio di notebook. Per ulteriori informazioni sulla gestione dei cookie, consulta:

- Chrome
- Firefox
- Safari

4. Scegli il link Jobs (Processi) nel menu di navigazione a sinistra.
5. Scegli Notebook Jupyter e quindi Create (Crea) per avviare una nuova sessione del notebook.
6. Nella pagina Create job in Jupyter notebook (Crea processo nel notebook Jupyter), specifica il nome del processo, il ruolo IAM da utilizzare. Scegli Create job (Crea processo).

   Dopo un breve periodo di tempo, viene visualizzato l'editor del notebook.
7. Dopo aver aggiunto il codice, è necessario eseguire la cella per avviare una sessione. Esistono diversi modi per eseguire la cella:

   - Premi il pulsante play.
   - Utilizza la scelta rapida da tastiera:
     - Su MacOS, Command + Invio per eseguire la cella.
     - Su Windows, Maius + Invio per eseguire la cella.

   Per informazioni sulla scrittura di codice utilizzando un'interfaccia per notebook Jupyter, vedi la Documentazione utente di Jupyter Notebook .
8. Per testare lo script, esegui l'intero script o le singole celle. Qualsiasi output di comando verrà visualizzato nell'area sotto la cella.
9. Dopo aver completato lo sviluppo del notebook, è possibile salvare il processo e quindi eseguirlo. Lo script è disponibile nella tabella Script. Tutte i magic aggiunti al taccuino verranno rimossi e non verranno salvati come parte dello script del processo AWS Glue. AWS Glue Studio aggiungerà automaticamente un `job.commit()` fino alla fine dello script generato dal contenuto di notebook.

   Per informazioni su come creare i processi, consulta Avviare un'esecuzione del processo.

## Componenti per l'editor del notebook

L'interfaccia dell'editor del notebook ha le seguenti sezioni principali.

- Interfaccia notebook (pannello principale) e barra degli strumenti
- Schede di modifica dei processi

### L'editor del notebook

L'editor del notebook AWS Glue Studio si basa sull'applicazione Jupyter Notebook. L'interfaccia del notebook AWS Glue Studio è simile a quella fornita da Juypter Notebook, descritta nella sezione Interfaccia utente per notebook. Il notebook utilizzato dalle sessioni interattive è un Jupyter Notebook.

Anche se il notebook AWS Glue Studio è simile a Juptyer Notebook, si differenzia in alcuni modi chiave:

- attualmente, il notebook AWS Glue Studio non può installare le estensioni
- non è possibile utilizzare più schede; esiste una relazione 1:1 tra un processo e un notebook
- il notebook AWS Glue Studio non ha lo stesso menu di file superiore presente in Jupyter Notebook
- attualmente, il notebook AWS Glue Studio funziona solo con il kernel AWS Glue Nota che non è possibile aggiornare il kernel da solo.

### schede di modifica dei processi AWS Glue Studio

Le schede utilizzate per interagire con il processo ETL si trovano nella parte superiore della pagina del notebook. Sono simili alle schede che appaiono nell'editor dei processi visivi di AWS Glue Studio ed eseguono le stesse operazioni.

- Notebook: utilizza questa scheda per visualizzare lo script del processo utilizzando l'interfaccia del notebook.
- Job details (Dettagli processo): configura l'ambiente e le proprietà per l'esecuzione del processo.
- Runs (Esecuzioni): visualizza le informazioni sulle precedenti esecuzioni di questo processo.
- Schedules (Piani): configura una pianificazione per l'esecuzione del processo in momenti specifici.

## Salvataggio del notebook e dello script del processo

È possibile salvare il notebook e lo script del processo che si sta creando in qualsiasi momento. Basta scegliere il pulsante Save (Salva) nell'angolo in alto a destra, come se stessi utilizzando l'editor visivo o di script.

Quando scegli Save (Salva), il file del notebook viene salvato nelle posizioni predefinite:

- Per impostazione predefinita, lo script del processo viene salvato nella posizione Amazon S3 indicata nella posizione Amazon S3 indicata nella scheda Job Details (Dettagli del processo), in Advanced properties (Proprietà avanzate), in Script path (Percorso script) della proprietà dei dettagli del processo. Gli script del processo vengono salvati in una sottocartella denominata `Scripts`.
- Per impostazione predefinita, il file del notebook (`.ipynb`) viene salvato nella posizione Amazon S3 indicata nella scheda Job Details (Dettagli del processo), in Advanced properties (Proprietà avanzate) in Script path (Percorso script) dei dettagli del processo. I file del notebook vengono salvati in una sottocartella denominata `Notebooks`.

> **Note**
>
> Quando salvi il processo, lo script contiene solo le celle di codice del notebook. Le celle Markdown e i magic non sono inclusi nello script del processo. Tuttavia, il file `.ipynb` conterrà qualsiasi markdown e magic.

Dopo aver salvato il processo, è possibile eseguirlo utilizzando lo script creato nel notebook.

## Gestione delle sessioni di notebook

I notebook in AWS Glue Studio si basano sulla caratteristica di sessioni interattive di AWS Glue. Per l'utilizzo delle sessioni interattive, è previsto un costo. Per gestire i costi, puoi monitorare le sessioni create per il tuo account e configurare le impostazioni di default per tutte le sessioni.

### Modifica del timeout di default per tutte le sessioni del notebook

Per impostazione predefinita, se il notebook AWS Glue Studio sottoposto a provisioning è stato avviato e non è stata eseguita alcuna cella, scadrà dopo 12 ore. Tale timeout non è configurabile e non comporta alcun costo aggiuntivo.

Dopo aver eseguito una cella, verrà avviata una sessione interattiva che scade dopo 48 ore. Questo timeout può essere configurato passando un magic `%idle_timeout` prima di eseguire una cella.

Come modificare il timeout di sessione predefinito per i notebook in AWS Glue Studio

1. Nel notebook, inserisci il magic `%idle_timeout` in una cella e specifica il valore di timeout in minuti.
2. Ad esempio: `%idle_timeout 15` cambierà il timeout di default a 15 minuti. Se la sessione non viene utilizzata entro 15 minuti, viene automaticamente interrotta.

## Installazione di moduli Python aggiuntivi

Se desideri installare moduli aggiuntivi per la tua sessione usando pip, puoi farlo utilizzando `%additional_python_modules` per aggiungerli alla sessione:

```
%additional_python_modules awswrangler, s3://amzn-s3-demo-bucket/mymodule.whl
```

Tutti gli argomenti di additional_python_modules vengono passati a `pip3 install -m <>`

Per visualizzare un elenco di moduli Python disponibili, consulta la pagina Using Python libraries with AWS Glue.

## Modifica della configurazione di AWS Glue

Puoi usare i magic per controllare i valori di configurazione del processo AWS Glue. Se si desidera modificare un valore di configurazione del processo, è necessario utilizzare i magic corretti nel notebook. Consulta la pagina Magics supported by AWS Glue interactive sessions for Jupyter.

> **Note**
>
> La sovrascrittura delle proprietà per una sessione in esecuzione non è più disponibile. Per modificare le configurazioni della sessione, interrompere la sessione, impostare le nuove configurazioni e quindi iniziare una nuova sessione.

AWS Glue supporta diversi tipi di dipendenti. È possibile impostare il tipo di dipendente con `%worker_type`. Ad esempio: `%worker_type G.2X` . I tipi di worker disponibili includono G.1X, G.2X, G.4X, G.8X, G.12X, G.16X, R.1X, R.2X, R.4X e R.8X. Il valore di default è G.1X.

Inoltre, è possibile specificare il numero di dipendenti con `%number_of_workers`. Ad esempio, per specificare 40 dipendenti: `%number_of_workers 40`.

Per ulteriori informazioni, consulta Definizione delle proprietà del processo

### Arresto di una sessione di notebook

Per interrompere una sessione del notebook, usa il magic `%stop_session`.

Se ti allontani dal notebook nella AWS console, riceverai un messaggio di avviso in cui puoi scegliere di interrompere la sessione.

## Utilizzo di Amazon Q Developer con AWS Glue Studio notebook

AWS Glue Studio permette di creare processi in modo interattivo in un'interfaccia notebook basata su Jupyter Notebooks. L'utilizzo di Amazon Q Developer migliora l'esperienza di creazione all'interno dei notebook AWS Glue Studio.

L'estensione Amazon Q Developer supporta la scrittura di codice generando suggerimenti sul codice e consigliando miglioramenti relativi ai problemi di codice. Amazon Q Developer supporta sia Python che Scala, i due linguaggi utilizzati per la codifica degli script ETL per i processi Spark in AWS Glue Studio notebook.

### Cos'è Amazon Q Developer?

Amazon Q Developer è un servizio basato sul machine learning che aiuta a migliorare la produttività degli sviluppatori. A tale scopo, Amazon Q Developer genera suggerimenti sul codice basati sui commenti degli sviluppatori in linguaggio naturale e sul codice nell'IDE. Il servizio si integra con Amazon SageMaker AI Studio JupyterLab, istanze Amazon SageMaker AI notebook e altri ambienti di sviluppo integrati ()IDEs.

Per ulteriori informazioni, consulta Utilizzo di Amazon Q Developer con AWS Glue Studio.

# Stati di esecuzione dei processi AWS Glue sulla console

Puoi visualizzare lo stato di un processo di estrazione, trasformazione e caricamento (ETL) AWS Glue mentre è in esecuzione o una volta arrestato. Puoi visualizzare lo stato tramite console AWS Glue.

## Accesso al pannello di controllo di monitoraggio dei processi

Si accede alla dashboard di monitoraggio dei processi scegliendo il link Monitoraggio dell'esecuzione dei processi nel pannello di navigazione AWS Glue della sezione Processi ETL.

## Panoramica del pannello di controllo di monitoraggio dei processi

Il pannello di controllo di monitoraggio dei processi fornisce un riepilogo generale delle esecuzioni del processo, con i totali per i processi con lo stato di In esecuzione, Annullato, Riuscito oppure Non riuscito. I riquadri aggiuntivi forniscono il tasso di successo complessivo dell'esecuzione del processo, l'utilizzo stimato della DPU per i processi, una suddivisione dei conteggi dello stato del processo per tipo di processo, per tipo di worker e per giorno.

I grafici nei riquadri sono interattivi. È possibile scegliere qualsiasi blocco in un grafico per eseguire un filtro che visualizzi solo quei processi nella tabella Esecuzioni del processo nella parte inferiore della pagina.

Per modificare l'intervallo di date delle informazioni visualizzate in questa pagina, utilizza il selettore Intervallo date. Quando si modifica l'intervallo di date, i riquadri delle informazioni vengono adattati per visualizzare i valori per il numero di giorni specificato prima della data attuale. È possibile anche usare un intervallo di date specifico scegliendo Personalizzato dal selettore dell'intervallo di date.

## Visualizzazione esecuzioni dei processi

> **Note**
>
> La cronologia di esecuzione dei processi è accessibile per 90 giorni per flusso di lavoro ed esecuzione del processo.

L'elenco delle risorse Esecuzioni dei processi mostra i processi per l'intervallo di date specificato e i filtri.

È possibile filtrare i processi in base a criteri aggiuntivi, ad esempio lo stato, il tipo di worker, il tipo di processo e il nome del processo. Nella casella filtro nella parte superiore della tabella è possibile inserire il testo da utilizzare come filtro. Durante l'inserimento del testo, i risultati della tabella vengono aggiornati con righe contenenti testo corrispondente.

È possibile visualizzare un sottoinsieme dei processi scegliendo gli elementi dai grafici nel pannello di controllo di monitoraggio del processo. Ad esempio, se si sceglie il numero di processi in esecuzione

nella finestra Riepilogo delle esecuzioni, l'elenco Esecuzioni dei processi visualizza solo i processi che hanno attualmente lo stato `Running`. Se si sceglie una delle barre nel grafico a barre Analisi del tipo di worker, nell'elenco Esecuzioni dei processi vengono mostrate solo le esecuzioni del processo con il tipo e lo stato corrispondenti.

L'elenco delle risorse Esecuzioni dei processi mostra i dettagli delle esecuzioni del processo. È possibile ordinare le righe della tabella scegliendo un'intestazione di colonna. La tabella contiene le informazioni seguenti:

| Proprietà | Description |
| --- | --- |
| Nome del lavoro | Il nome del processo . |
| Tipo | Il tipo di ambiente per il processo:<br>• ETL Glue: esegue in un ambiente Apache Spark gestito da AWS Glue.<br>• Streaming Glue: esegue in un ambiente Apache Spark ed esegue ETL sui flussi di dati.<br>• Shell Python: esegue gli script di Python come una shell. |
| Ora di inizio | La data e ora in cui questa esecuzione di processo è stata avviata. |
| Ora di fine | La data e ora in cui questa elaborazione di processo è stata completata. |
| Stato di esecuzione | Lo stato attuale del processo eseguito. I valori possono essere:<br>• `STARTING`<br>• `RUNNING`<br>• `STOPPING`<br>• `STOPPED`<br>• `SUCCEEDED`<br>• `FAILED` |

| Proprietà | Description |
| --- | --- |
| | • `TIMEOUT` |
| Tempo di esecuzione | Quantità di tempo durante la quale l'esecuzione dell'attività ha utilizzato le risorse. |
| Capacity | Il numero di unità di elaborazione AWS Glue dati (DPUs) che sono state allocate per questa esecuzione del processo. Per ulteriori informazioni sulla pianificazione della capacità, consulta Monitoraggio per la pianificazione della capacità DPU nella Guida per gli sviluppat ori di AWS Glue. |

| Proprietà | Description |
| --- | --- |
| Tipo di lavoratore | Il tipo di worker predefinito allocato quando è stato eseguito il processo. I valori possono essere `G.1X`, `G.2X`, `G.4X` o `G.8X`.<br><br>• **`G.1X`**: quando si sceglie questo tipo, si fornisce anche un valore per Numero di worker. Ogni worker esegue il mapping su 1 DPU (4 vCPUs, 16 GB di memoria) con disco da 84 GB (circa 34 GB gratuiti). Consiglia mo questo tipo di lavoratore per lavori che richiedono molta memoria. Questa è l'impostazione predefinita per Tipo di worker per la versione AWS Glue 2.0 o successive.<br>• **`G.2X`**: quando si sceglie questo tipo, si fornisce anche un valore per Numero di worker. Ogni worker esegue il mapping su 2 DPU (8 VCPUs, 32 GB di memoria) con disco da 128 GB (circa 77 GB gratuiti) . Suggeriamo questo tipo di dipendente per i processi ad alto consumo di memoria e per i processi che eseguono trasformazioni machine learning.<br>• **`G.4X`**: quando si sceglie questo tipo, si fornisce anche un valore per Numero di worker. Ogni worker esegue il mapping a 4 DPU (16 vCPUs, 64 GB di memoria) con disco da 256 GB (circa 235 GB gratuiti). Questi tipi di worker sono raccomandati per i processi i cui carichi di lavoro contengono trasformazioni, aggregazioni, join e query con i requisiti più elevati. Questo tipo di worker è disponibile solo per i processi ETL di AWS Glue Spark versione 3.0 o successiva nelle seguenti Regioni AWS : Stati Uniti orientali |

| Proprietà | Description |
| --- | --- |
| | (Ohio), Stati Uniti orientali (Virginia settentrionale), Stati Uniti occidentali (Oregon), Asia Pacifico (Singapore), Asia Pacifico (Sydney), Asia Pacifico (Tokyo), Canada (Centrale), Europa (Francoforte), Europa (Irlanda) ed Europa (Stoccolma).<br>• **`G.8X`**: quando si sceglie questo tipo, si fornisce anche un valore per Numero di worker. Ogni worker esegue il mapping a 8 DPU (32 vCPUs, 128 GB di memoria) con disco da 512 GB (circa 487 GB gratuiti). Questi tipi di worker sono raccomandati per i processi i cui carichi di lavoro contengono trasformazioni, aggregazioni, join e query con i requisiti più elevati. Questo tipo di worker è disponibile solo per i job Spark ETL AWS Glue versione 3.0 o successiva, nelle stesse AWS regioni supportate per il tipo di lavoratore. `G.4X` |
| Ore DPU | Il numero stimato di dati DPUs utilizzati per l'esecuzione del job. Una DPU è una misura relativa della potenza di elaborazione. DPUs vengono utilizzati per determinare il costo di esecuzione del lavoro. Per ulteriori informazioni, consultare la pagina dei prezzi di AWS Glue. |

È possibile scegliere qualsiasi processo eseguito nell'elenco e visualizzare informazioni aggiuntive. Scegliere un'esecuzione del processo, quindi eseguire una delle operazioni seguenti:

- Scegliere il menu Operazioni e l'opzione Visualizza processo per visualizzare il processo nell'editor visivo.

- Scegliere il menu Operazioni e l'opzione Interrompi esecuzione per interrompere l'esecuzione corrente del processo.
- Scegli il pulsante Visualizza CloudWatch registri per visualizzare i registri di esecuzione del processo per quel processo.
- Scegliere Visualizza dettagli per visualizzare la pagina dei dettagli dell'esecuzione.

## Visualizzazione dei log di esecuzione del processo

Puoi visualizzare i log del processo in diversi modi:

- Nella pagina Monitoraggio, nella tabella Job run, scegli un job run, quindi scegli View CloudWatch logs.
- Nell'editor visivo dei processi, nella scheda Esecuzioni per un processo, scegliere i collegamenti ipertestuali per visualizzare i log:
  - Log: collega ai log dei processi di Apache Spark scritti quando la registrazione continua è abilitata per l'esecuzione di un processo. Quando si sceglie questo collegamento, si accede ai Amazon CloudWatch log del gruppo di `/aws-glue/jobs/logs-v2` log. Per impostazione predefinita, i log escludono i messaggi di log di heartbeat inutili di driver o executor Apache Spark e Apache Hadoop YARN. Per ulteriori informazioni sulla registrazione continua, consultare Registrazione continua per processi di AWS Glue nella Guida per gli sviluppatori di AWS Glue.
  - Log di errore: collega ai log scritti in `stderr` per questa esecuzione di processo. Quando si sceglie questo collegamento, si accede ai log Amazon CloudWatch nel gruppo di log `/aws-glue/jobs/error`. Questi log possono essere utilizzati per visualizzare i dettagli su tutti gli errori riscontrati durante l'esecuzione del processo.
  - Log di output: collega ai log scritti in `stdout` per questa esecuzione del processo. Quando si sceglie questo collegamento, si accede ai log Amazon CloudWatch nel gruppo di log `/aws-glue/jobs/output`. Qui è possibile visualizzare i log per vedere tutti i dettagli sulle tabelle create in AWS Glue Data Catalog ed eventuali errori riscontrati.

## Visualizzazione dei dettagli di un'esecuzione di un processo

È possibile scegliere un processo nell'elenco Esecuzioni dei processi nella pagina Monitoraggio, quindi scegliere Visualizza dettagli dell'esecuzione per visualizzare informazioni dettagliate sull'esecuzione del processo.

Le informazioni visualizzate nella scheda dei dettagli dell'esecuzione del processo includono:

| Proprietà | Description |
| --- | --- |
| Nome del lavoro | Il nome del processo . |
| Stato di esecuzione | Lo stato attuale del processo eseguito. I valori possono essere:<br>• `STARTING`<br>• `RUNNING`<br>• `STOPPING`<br>• `STOPPED`<br>• `SUCCEEDED`<br>• `FAILED`<br>• `TIMEOUT` |
| Versione Glue | La versione di AWS Glue utilizzata dall'esec uzione del processo. |
| Tentativo recente | Il numero di tentativi automatici per l'esecuzi one di questo processo. |
| Ora di inizio | La data e ora in cui questa esecuzione di processo è stata avviata. |
| Ora di fine | La data e ora in cui questa elaborazione di processo è stata completata. |
| Ora di inizio | La quantità di tempo impiegato per la preparazi one dell'esecuzione del processo. |
| Ora di esecuzione | La quantità di tempo impiegato per l'esecuzione dello script del processo. |
| Nome trigger | Il nome del trigger associato al processo. |
| Ora ultima modifica | La data dell'ultima modifica apportata al processo. |

| Proprietà | Description |
| --- | --- |
| Configurazione di sicurezza | La configurazione di sicurezza per il processo, che include le impostazioni di crittografia, CloudWatch crittografia e crittografia dei segnalibri di lavoro di Amazon S3. |
| Timeout | Il valore della soglia di timeout per l'esecuzione del processo. |
| Capacità allocata | Il numero di unità di elaborazione AWS Glue dati (DPUs) allocate per questa esecuzion e del processo. Per ulteriori informazioni sulla pianificazione della capacità, consulta Monitoraggio per la pianificazione della capacità DPU nella Guida per gli sviluppatori di AWS Glue. |
| Capacità massima | La capacità massima disponibile per l'esecuzi one del processo. |
| Numero di worker | Il numero di worker utilizzati per l'esecuzione del processo. |

| Proprietà | Description |
| --- | --- |
| Tipo di lavoratore | Il tipo di worker predefiniti allocati per l'esecuzione del processo. I valori possono essere `G.1X` o `G.2X`.<br>• **`G.1X`**: quando si sceglie questo tipo, si fornisce anche un valore per Numero di worker. Ogni worker esegue il mapping su 1 DPU (4 vCPUs, 16 GB di memoria, disco da 64 GB) e fornisce 1 esecutore per lavoratore. Consigliamo questo tipo di lavoratore per lavori che richiedono molta memoria. Questa è l'impostazione predefinita per Tipo di worker per la versione AWS Glue 2.0 o successive.<br>• **`G.2X`**: quando si sceglie questo tipo, si fornisce anche un valore per Numero di worker. Ogni worker esegue il mapping su 2 DPUs (8 vCPUs, 32 GB di memoria, 128 GB di disco) e fornisce 1 esecutore per lavoratore. Suggeriamo questo tipo di dipendente per i processi ad alto consumo di memoria e per i processi che eseguono trasformazioni machine learning. |
| Log | Un collegamento ai log del processo per la registrazione continua (`/aws-glue/jobs/logs-v2` ). |
| Log di output | Un collegamento ai file di log di output del processo (`/aws-glue/jobs/output` ). |
| Log di errore | Un collegamento ai file di log degli errori del processo (`/aws-glue/jobs/error` ). |

È inoltre possibile visualizzare i seguenti elementi aggiuntivi, disponibili anche quando si visualizzano le informazioni relative alle esecuzioni recenti dei processi. Per ulteriori informazioni, consulta the section called "Visualizzare le informazioni sulle esecuzioni dei processi recenti".

- Inserimento di argomenti
- Log continui
- Parametri: puoi visualizzare le visualizzazioni dei parametri di base. Per ulteriori informazioni sui parametri inclusi, consulta the section called "Visualizzazione delle Amazon CloudWatch metriche relative all'esecuzione di un job Spark".
- Interfaccia utente Spark: puoi visualizzare i log di Spark relativi al processo nell'interfaccia utente di Spark. Per ulteriori informazioni sull'utilizzo dell'interfaccia utente di Spark, consulta the section called "Monitoraggio con l'interfaccia utente di Spark". Abilita questa funzionalità seguendo la procedura in the section called "Abilitazione dell'interfaccia utente di Spark per processi".

## Visualizzazione delle Amazon CloudWatch metriche relative all'esecuzione di un job Spark

Nella pagina dei dettagli dell'esecuzione di un lavoro, sotto la sezione Dettagli dell'esecuzione, puoi visualizzare le metriche del processo. AWS Glue Studioinvia le metriche dei job a Amazon CloudWatch per ogni job eseguito.

AWS Glue riporta le metriche Amazon CloudWatch ogni 30 secondi. I parametri AWS Glue rappresentano i valori delta rispetto ai valori segnalati in precedenza. Se appropriato, i pannelli di controllo dei parametri aggregano (sommano) i valori inviati ogni 30 secondi per ottenere un valore per l'intero ultimo minuto. Tuttavia, le metriche di Apache Spark AWS Glue trasmesse a Amazon CloudWatch sono generalmente valori assoluti che rappresentano lo stato corrente nel momento in cui vengono segnalate.

> **Note**
>
> Devi configurare il tuo account per accedere a,. Amazon CloudWatch

I parametri forniscono informazioni sull'esecuzione del processo, ad esempio:

- Spostamento di dati ETL: il numero di byte letti da o scritti in Amazon S3.

- Profilo di memoria: heap utilizzata: il numero di byte di memoria utilizzati dall'heap Java Virtual Machine (JVM).
- Profilo di memoria: utilizzo heap: la frazione di memoria (scala: 0-1), mostrata come percentuale, utilizzata dall'heap JVM.
- Carico CPU: la frazione del carico di sistema della CPU utilizzata (dimensione: 0-1), indicata come percentuale.

## Visualizzazione delle Amazon CloudWatch metriche per l'esecuzione di un job Ray

Nella pagina dei dettagli dell'esecuzione di un processo, sotto la sezione Dettagli dell'esecuzione, puoi visualizzare le metriche del processo. AWS Glue Studioinvia le metriche dei job a Amazon CloudWatch per ogni job eseguito.

AWS Glue riporta le metriche Amazon CloudWatch ogni 30 secondi. I parametri AWS Glue rappresentano i valori delta rispetto ai valori segnalati in precedenza. Se appropriato, i pannelli di controllo dei parametri aggregano (sommano) i valori inviati ogni 30 secondi per ottenere un valore per l'intero ultimo minuto. Tuttavia, le metriche di Apache Spark AWS Glue trasmesse a Amazon CloudWatch sono generalmente valori assoluti che rappresentano lo stato corrente nel momento in cui vengono segnalate.

> **Note**
>
> È necessario configurare l'account per accedere Amazon CloudWatch, come descritto in.

Nei processi Ray, è possibile visualizzare i seguenti grafici di parametri aggregati. Con questi, è possibile creare un profilo del cluster e delle attività, nonché accedere a informazioni dettagliate su ciascun nodo. I dati delle serie temporali che supportano questi grafici sono disponibili CloudWatch per ulteriori analisi.

Profilo dell'attività: stato dell'attività

Mostra il numero di attività Ray nel sistema. A ogni ciclo di vita delle attività viene assegnata una serie temporale.

Profilo dell'attività: nome dell'attività

Mostra il numero di attività Ray nel sistema. Vengono mostrate solo le attività in sospeso e quelle attive. A ogni tipo di attività (in base al nome) viene assegnata una serie temporale distinta.

Profilo del cluster: CPUs in uso

Mostra il numero di core della CPU utilizzati. A ogni nodo viene assegnata una serie temporale. I nodi sono identificati da indirizzi IP, che sono effimeri e vengono utilizzati solo per l'identificazione.

Profilo del cluster: utilizzo della memoria dell'archivio di oggetti

Mostra l'utilizzo della memoria da parte della cache degli oggetti Ray. A ogni posizione di memoria (memoria fisica, memorizzata nella cache su disco e riversata in Amazon S3) viene assegnata una serie temporale distinta. L'archivio oggetti gestisce l'archiviazione di dati su tutti i nodi del cluster. Per ulteriori informazioni, consultare la pagina Objects nella documentazione di Ray.

Profilo del cluster: conteggio dei nodi

Mostra il numero di nodi forniti per il cluster.

Dettaglio del nodo: utilizzo della CPU

Mostra l'utilizzo della CPU su ciascun nodo in percentuale. Ogni serie mostra una percentuale aggregata di utilizzo della CPU su tutti i core del nodo.

Dettaglio del nodo: utilizzo della memoria

Mostra l'utilizzo della memoria su ogni nodo in GB. Ogni serie mostra la memoria aggregata tra tutti i processi sul nodo, incluse le attività Ray e il processo di archiviazione di Plasma. Ciò non rifletterà gli oggetti archiviati su disco o riversati su Amazon S3.

Dettaglio del nodo: utilizzo del disco

Mostra l'utilizzo del disco su ogni nodo in GB.

Dettaglio del nodo: I/O velocità del disco

Mostra il disco I/O su ogni nodo in KB/s.

Dettaglio del nodo: velocità effettiva di rete I/O

Mostra la rete I/O su ogni nodo in KB/s.

Dettaglio del nodo: utilizzo della CPU da parte del componente Ray

Mostra l'utilizzo della CPU in parte dei core. A ogni componente Ray su ogni nodo viene assegnata una serie temporale.

Dettaglio del nodo: utilizzo della memoria da parte del componente Ray

Mostra l'utilizzo della memoria in GiB. A ogni componente Ray su ogni nodo viene assegnata una serie temporale.

# Rileva ed elabora dati sensibili

La trasformazione Detect PII identifica le informazioni personali di identificazione (PII) nell'origine dati. È possibile scegliere l'entità PII da identificare, come si desidera che i dati vengano scansionati e cosa fare con l'entità PII identificata dalla trasformazione Detect PII.

La trasformazione Detect PII fornisce la possibilità di rilevare, mascherare o rimuovere le entità definite o che sono predefinite da AWS. Ciò consente di aumentare la conformità e ridurre la responsabilità. Ad esempio, potresti voler assicurarti che nei tuoi dati non esistano informazioni di identificazione personale che possano essere lette e mascherare i numeri di previdenza sociale con una stringa fissa (ad esempio xxx-xx-xxxx), numeri di telefono o indirizzi.

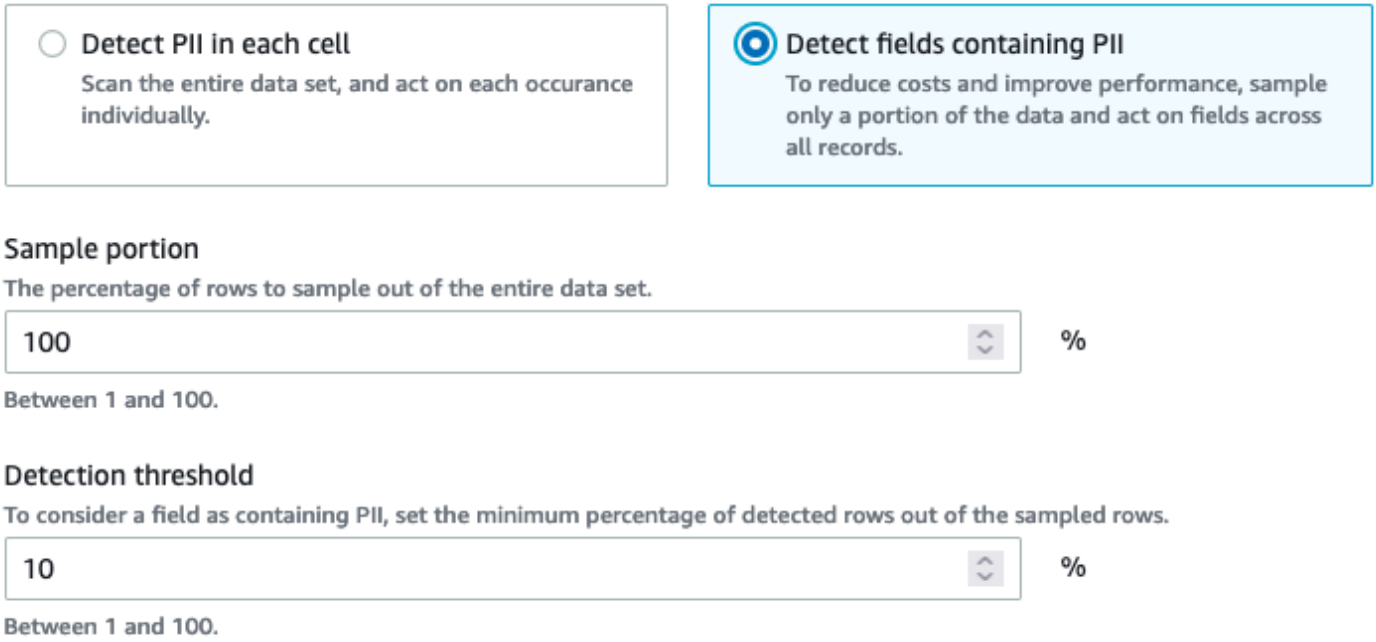
Per lavorare con dati sensibili al di fuori di AWS Glue Studio, consulta Utilizzo del rilevamento di dati sensibili all'esterno di AWS Glue Studio

Argomenti

- Come scegliere il modo in cui desideri che vengano scansionati i dati
- Scelta delle entità PII da rilevare
- Specificazione del livello di distinzione di rilevamento
- Come scegliere cosa fare con i dati PII identificati
- Aggiungere sostituzioni di operazioni granulari

## Come scegliere il modo in cui desideri che vengano scansionati i dati

Quando esegui la scansione del set di dati per i dati sensibili, come le informazioni di identificazione personale (PII), puoi scegliere di rilevare le PII in ogni riga o rilevare le colonne che contengono dati PII.




Quando scegli Detect PII in each cell (Rileva PII in ogni cella), stai scegliendo di scansionare tutte le righe nell'origine dati. Si tratta di una scansione completa per garantire che le entità PII siano identificate.

Quando scegli Detect fields containing PII (Rileva campi contenenti PII), stai scegliendo di scansionare un campione di righe per le entità PII. Questo è un modo per mantenere bassi costi e risorse, identificando al contempo i campi in cui si trovano le entità PII.

Quando si sceglie di rilevare i campi che contengono PII, è possibile ridurre i costi e migliorare le prestazioni campionando una parte di righe. La scelta di questa opzione ti permetterà di specificare opzioni aggiuntive:

- Sample portion (Porzione campione): consente di specificare la percentuale di righe da campionare. Ad esempio, se si immette "50", si specifica che si desidera il 50% delle righe scansionate per l'entità PII.
- Detection threshold (Soglia di rilevamento): consente di specificare la percentuale di righe che contengono l'entità PII in modo che l'intera colonna venga identificata come avente l'entità PII. Ad esempio, se inserisci "10", stai specificando che il numero dell'entità PII, US Phone, nelle righe scansionate deve essere pari o superiore al 10% per poter identificare il campo come entità PII, Numero di telefono degli Stati Uniti. Se la percentuale di righe che contengono l'entità PII è inferiore al 10%, tale campo non verrà etichettato come contenente l'entità PII, Numero di telefono degli Stati Uniti, al suo interno.

# Scelta delle entità PII da rilevare

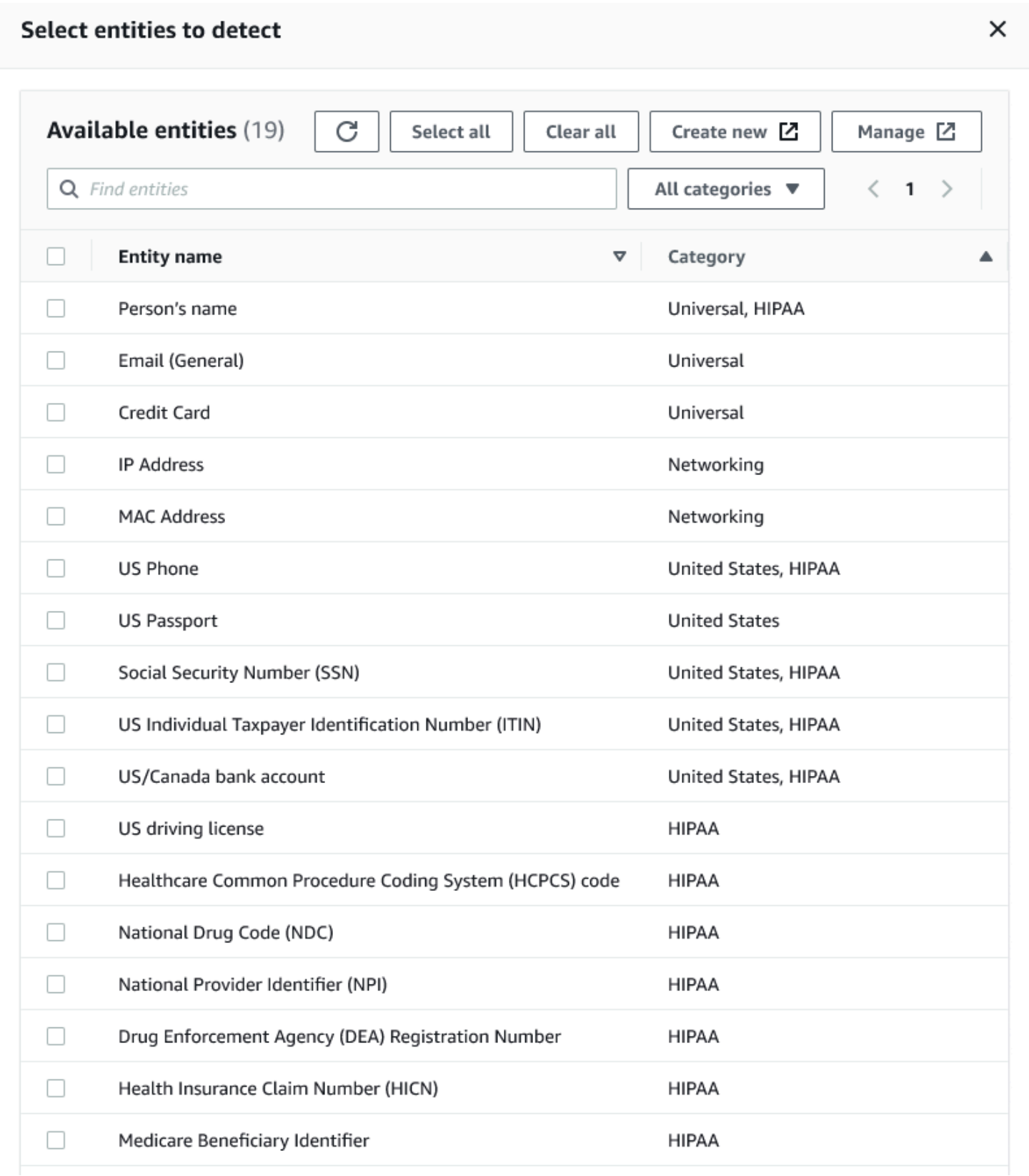
Se hai scelto Rileva PII in ogni cella puoi scegliere tra una delle tre opzioni:

- Tutti i modelli PII disponibili, incluse AWS le entità.
- Seleziona categorie: quando selezioni le categorie, i modelli PII includeranno automaticamente i modelli nelle categorie selezionate.
- Seleziona modelli specifici: verranno rilevati solo i modelli selezionati.

Per un elenco completo dei tipi di dati sensibili gestiti, consulta la pagina Managed Sensitive Data Types.

## Scegli tra tutti i modelli PII disponibili

Se scegli Tutti i modelli PII disponibili, seleziona le entità predefinite da. AWSÈ possibile selezionare una, più di una o tutte le entità.


Select entities to detect
Available entities (19)
Select all
Clear all
Create new
Manage
Find entities
All categories
1
Entity name
Category
Person's name
Universal, HIPAA
Email (General)
Universal
Credit Card
Universal
IP Address
Networking
MAC Address
Networking
US Phone
United States, HIPAA
US Passport
United States
Social Security Number (SSN)
United States, HIPAA
US Individual Taxpayer Identification Number (ITIN)
United States, HIPAA
US/Canada bank account
United States, HIPAA
US driving license
HIPAA
Healthcare Common Procedure Coding System (HCPCS) code
HIPAA
National Drug Code (NDC)
HIPAA
National Provider Identifier (NPI)
HIPAA
Drug Enforcement Agency (DEA) Registration Number
HIPAA
Health Insurance Claim Number (HICN)
HIPAA
Medicare Beneficiary Identifier
HIPAA

## Categorie di selezione

Se hai scelto Categorie di selezione come i modelli PII da rilevare, è possibile selezionare tra le opzioni del menu a discesa. Alcune entità possono appartenere a più di una categoria. Ad esempio: Nome della persona è un'entità che appartiene alle categorie Universale e HIPAA.

- Universale (esempi: e-mail, carta di credito)
- HIPAA (esempi: patente di guida statunitense, codice HCPCS [Healthcare Common Procedure Coding System])
- Rete (esempi: indirizzo IP, indirizzo MAC)
- Argentina
- Australia
- Austria
- Belgio
- Bosnia
- Bulgaria
- Canada
- Cile
- Colombia
- Croazia
- Cipro
- Cechia
- Danimarca
- Estonia
- Finlandia
- Francia
- Germania
- Grecia
- Ungheria
- Irlanda
- Corea
- Giappone

- Messico
- Paesi Bassi
- Nuova Zelanda
- Norvegia
- Portogallo
- Romania
- Singapore
- Slovacchia
- Slovenia
- Spagna
- Svezia
- Svizzera
- Turchia
- Ucraina
- Stati Uniti
- Regno Unito
- Venezuela

## Seleziona modelli specifici

Se scegli Seleziona modelli specifici come modelli PII da rilevare, è possibile cercare o sfogliare da un elenco di modelli già creati o creare un nuovo modello di entità di rilevamento.

I passaggi riportati di seguito descrivono come creare un nuovo modello personalizzato per il rilevamento di dati sensibili. Creerai il modello personalizzato inserendo un nome per il modello, aggiungerai un'espressione regolare e, facoltativamente, definirai le parole contestuali.

1. Per creare un nuovo motivo, fare clic sul pulsante Creare nuovo.

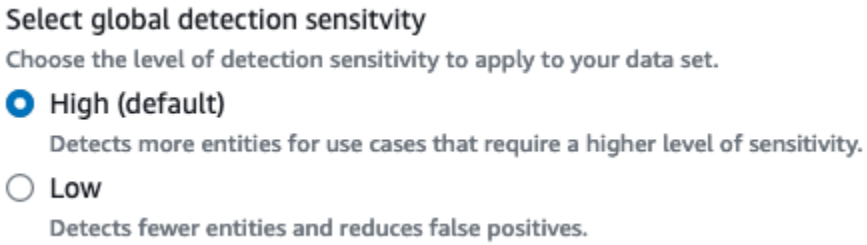
2. Nella pagina Crea entità di rilevamento, immettere il nome dell'entità e un'espressione regolare. L'espressione regolare (Regex) è quella che AWS Glue utilizzerà per abbinare le entità.
3. Fare clic su Convalida. Se la convalida ha esito positivo, verrà visualizzato un messaggio di conferma che indica che la stringa è un'espressione regolare valida. Se la convalida non ha esito positivo, verrà visualizzato un messaggio che indica che la stringa non è conforme alla formattazione corretta e ai valori letterali, operatori o costrutti dei caratteri accettati.
4. È possibile scegliere di aggiungere parole di contesto oltre all'espressione regolare. Le parole contestuali possono aumentare la probabilità di una corrispondenza. Questi possono essere utili nei casi in cui i nomi dei campi non sono descrittivi dell'entità. Ad esempio, i numeri di previdenza sociale possono essere denominati "SSN" o "SS". L'aggiunta di queste parole contestuali può aiutare a far corrispondere l'entità.
5. Fare clic su Crea per creare l'entità di rilevamento. Tutte le entità create sono visibili nella console AWS Glue Studio. Fai clic su Entità di rilevamento nel menu di navigazione a sinistra.

   È possibile modificare, eliminare o creare entità di rilevamento dalla pagina Entità di rilevamento. È inoltre possibile ricercare un modello utilizzando il campo di ricerca.

## Specificazione del livello di distinzione di rilevamento

È possibile impostare il livello di distinzione quando si utilizza il rilevamento di dati sensibili.

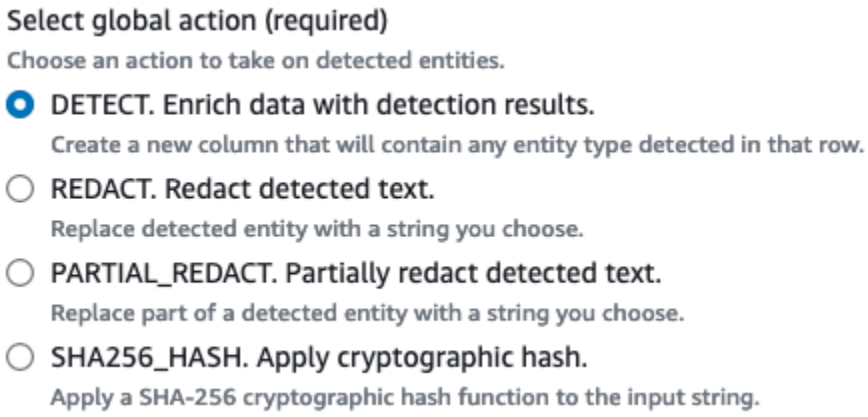
- Alto: (impostazione predefinita) rileva più entità per i casi d'uso che richiedono un livello di distinzione più elevato. Tutti i processi AWS Glue creati dopo novembre 2023 vengono automaticamente attivati per questa impostazione.
- Bassa: rileva un minor numero di entità e riduce i falsi positivi.



# Come scegliere cosa fare con i dati PII identificati

Se hai deciso di rilevare le PII nell'intera origine dati, puoi selezionare l'applicazione di un'azione globale:

- Enrich data with detection results (Arricchisci i dati con i risultati di rilevamento): se scegli Detect PII in ogni cella, potrai archiviare le entità rilevate in una nuova colonna.
- Redact detected text (Rivedi il testo rilevato): è possibile sostituire il valore PII rilevato con una stringa specificata nel campo opzionale Sostituzione del testo. Se non viene specificata alcuna stringa, l'entità PII rilevata viene sostituita con "********".
- Rivedi parzialmente il testo rilevato: è possibile sostituire il valore PII rilevato con una stringa scelta. Esistono due opzioni possibili: lasciare le estremità smascherate o mascherarle fornendo un modello regex esplicito. Questa caratteristica non è disponibile in AWS Glue 2.0.
- Applica hash di crittografia: puoi passare il valore PII rilevato a una funzione hash di crittografia SHA-256 e sostituire il valore con l'output della funzione.




## Differenze tra le versioni di AWS Glue 2.0 e 3.0+

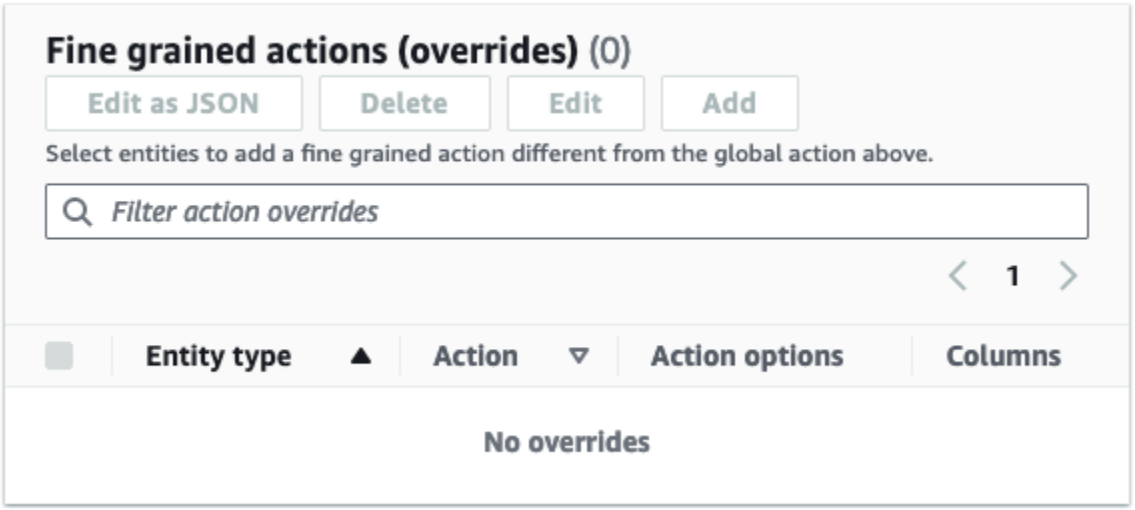
AWS Glue2.0 jobs ne restituirà una nuova DataFrame con le informazioni PII rilevate per ogni colonna in una colonna supplementare. Qualsiasi processo di redazione o hash è visibile all'interno dello script AWS Glue nella scheda visiva.

AWS GlueI lavori 3.0 e 4.0 ne restituiranno uno nuovo DataFrame con la stessa colonna supplementare. È presente una nuova chiave per "actionUsed" che può essere una tra `DETECT`, `REDACT`, `PARTIAL_REDACT` o `SHA256_HASH`. Se viene selezionata un'azione di mascheramento, DataFrame restituirà dati con dati sensibili mascherati.

## Aggiungere sostituzioni di operazioni granulari

È possibile aggiungere ulteriori impostazioni di rilevamento e azione alla tabella dettagliata delle sostituzioni di azioni. Ciò consente di:

- Includere o escludere determinate colonne dal rilevamento: uno schema dedotto sull'origine dati popolerà la tabella con le colonne disponibili.
- Definire le impostazioni specifiche più granulari rispetto all'utilizzo di azioni globali: ad esempio, è possibile specificare diverse impostazioni del testo di redazione per diversi tipi di entità.
- Specificare un'azione diversa da quella globale: se si desidera applicare un'azione diversa a un tipo di dati sensibili diverso, è possibile farlo qui. Tieni presente che non è possibile utilizzare due edit-in-place azioni diverse (redazione e hashing) sulla stessa colonna, ma è sempre possibile utilizzare detect.




# Gestione dei processi ETL con AWS Glue Studio

Per gestire i processi ETL, puoi utilizzare l'interfaccia visiva semplice di AWS Glue Studio. Nel pannello di navigazione, scegli Jobs (Processi) per visualizzare la pagina Jobs (Processi) In questa pagina puoi visualizzare tutti i processi creati con AWS Glue Studio o con la console AWS Glue. In questa pagina puoi visualizzare, gestire ed eseguire i processi.

In questa pagina puoi anche eseguire le seguenti operazioni:

- Avviare un'esecuzione del processo
- Pianificazione delle esecuzioni dei processi

- Gestione delle pianificazioni dei processi
- Interruzione dei processi
- Visualizzazione dei processi
- Visualizzare le informazioni sulle esecuzioni dei processi recenti
- Visualizzare lo script del processo
- Modificare le proprietà del processo
- Salvare il lavoro
- Clonazione di un processo
- Eliminazione dei processi

## Avviare un'esecuzione del processo

In AWS Glue Studio, puoi eseguire i processi on demand. Un processo può essere eseguito più volte e ogni volta AWS Glue raccoglie informazioni sulle attività del processo e sulle prestazioni. Queste informazioni sono indicate come esecuzione del processo e sono identificate da un ID di esecuzione del processo.

Puoi avviare l'esecuzione di un processo in AWS Glue Studio nei seguenti modi:

- Nella pagina Jobs (Processi), scegli il processo che vuoi avviare, quindi scegli il pulsante Run job (Esecuzione del processo).
- Se stai visualizzando un processo nell'editor visivo e questo è stato salvato, puoi scegliere il pulsante Run (Esegui) per avviare l'esecuzione di un processo.

Per ulteriori informazioni sulle esecuzioni dei processi, consulta Uso di processi nella console AWS Glue nella Guida per gli sviluppatori di AWS Glue .

## Pianificazione delle esecuzioni dei processi

In AWS Glue Studio, è possibile creare una pianificazione per eseguire i processi in orari specifici. Puoi specificare i vincoli, ad esempio il numero di volte in cui vengono eseguiti i processi, i giorni della settimana in cui vengono eseguiti e a che ora. Questi vincoli si basano sul comando `cron` e hanno le stesse limitazioni di `cron`. Ad esempio, se vuoi eseguire il processo il giorno 31 di ogni mese, devi ricordare che alcuni mesi non sono di 31 giorni. Per ulteriori informazioni su `cron`, vedi le espressioni Cron nella Guida per gli sviluppatori di AWS Glue .

Per eseguire i processi in base a una pianificazione

1. Crea una pianificazione del processo utilizzando uno dei seguenti metodi:

   - Nella pagina Jobs (Processi), scegli il processo per il quale creare una pianificazione, scegli Actions (Operazioni), quindi Schedule job (Pianifica processo).
   - Se stai visualizzando un processo nell'editor visivo e questo è stato salvato, puoi scegliere la scheda Schedules (pianificazioni). Quindi scegli Create Schedule (Crea pianificazione).

2. Nella pagina Schedule job run (Pianifica esecuzione del processo), inserisci le seguenti informazioni:

   - Name (Nome): inserisci un nome per il processo.
   - Frequency (Frequenza): inseriscila frequenza per la programmazione del processo. Puoi scegliere le seguenti opzioni:
     - Hourly (Orario): il processo verrà eseguito ogni ora, a partire da un minuto specifico. È possibile specificare gli attributi Minute (Minuto) dell'ora in cui il processo deve essere eseguito. Per impostazione predefinita, quando si sceglie la programmazione oraria, il processo viene eseguito all'inizio dell'ora (minuto 0).
     - Daily (Giornaliero): il processo verrà eseguito ogni giorno, a partire da un momento. È possibile specificare gli attributi Minute (Minuto) dell'ora in cui il processo deve essere eseguito e Start hour (Ora di avvio). Le ore sono specificate utilizzando un orologio di 23 ore, in cui si utilizzano i numeri da 13 a 23 per le ore pomeridiane. Il valore predefinito per i minuti e le ore è 0, il che significa che se si seleziona Daily (Giornaliero), il processo verrà eseguito per impostazione predefinita a mezzanotte.
     - Weekly (Settimanale): il processo verrà eseguito uno o più giorni della ogni settimana. Oltre alle stesse impostazioni descritte in precedenza per Daily (Giornaliero), è possibile scegliere i giorni della settimana in cui verrà eseguito il processo. È possibile scegliere uno o più giorni.
     - Monthly (Mensile): il processo verrà eseguito ogni mese in un giorno specifico. Oltre alle stesse impostazioni descritte in precedenza per Daily (Giornaliero), è possibile scegliere i giorni del mese in cui verrà eseguito il processo. Specifica il giorno come un valore numerico compreso tra 1 e 31. Se si seleziona un giorno che in un mese non esiste, ad esempio il 30 febbraio, il processo in quel mese non viene eseguito.
     - Custom (Personalizzato): inserisci un'espressione per la pianificazione del processo utilizzando la sintassi `cron`. Le espressioni cron permettono di creare pianificazioni più

complicate, ad esempio l'ultimo giorno del mese (invece di un giorno specifico del mese) o ogni terzo mese dai giorni 7 al 21.

Consulta le espressioni cron nella Guida per gli sviluppatori di AWS Glue

- Description (Descrizione): è possibile inserire una descrizione per la programmazione dei processi. Se prevedi di utilizzare la stessa pianificazione per più processi, una descrizione può rendere più facile determinare il relativo funzionamento.

3. Scegli Create schedule (Crea pianificazione) per salvare la pianificazione del processo.
4. Dopo aver creato la pianificazione, nella parte superiore della pagina della console viene visualizzato un messaggio di operazione riuscita. Puoi selezionare Job details (Dettagli del processo) in questo banner per visualizzare i dettagli. Si apre la pagina dell'editor visivo dei processi, con la scheda Schedules (Piani) selezionata.

## Gestione delle pianificazioni dei processi

Dopo aver creato le pianificazioni per un processo, puoi aprirlo nell'editor visivo e scegliere la casella di controllo Schedules (Piani) per gestire le pianificazioni.

Nella scheda Schedules (Pianificazioni) dell'editor visivo, puoi eseguire le seguenti attività:

- Creare una nuova pianificazione.

  Scegli Create schedule (Crea pianificazione), quindi inserisci le informazioni per la pianificazione come descritto in the section called "Pianificazione delle esecuzioni dei processi".

- Modificare una pianificazione esistente.

  Seleziona la pianificazione da modificare, quindi Action (Operazioni) e poi Edit schedule (Modifica pianificazione). Quando scegli di modificare una pianificazione esistente, la frequenza è personalizzata e la pianificazione viene visualizzata come espressione `cron`. Puoi modificare l'espressione `cron` oppure specificare una nuova pianificazione utilizzando l'opzione Frequency (Frequenza). Una volta terminate le modifiche, seleziona Update schedule (Aggiorna pianificazione).

- Sospendere una pianificazione attiva.

  Seleziona una pianificazione attiva, quindi Action (Operazioni) e Pause schedule (Sospendi pianificazione). La pianificazione viene disattivata immediatamente. Seleziona il pulsante di aggiornamento (ricarica) per visualizzare lo stato aggiornato della pianificazione.

- Riprendere una pianificazione sospesa.

  Seleziona una pianificazione attiva, quindi Action (Operazioni) e Resume schedule (Riprendi pianificazione). La pianificazione viene attivata immediatamente. Seleziona il pulsante di aggiornamento (ricarica) per visualizzare lo stato aggiornato della pianificazione.

- Eliminare una pianificazione.

  Seleziona la pianificazione da rimuovere, quindi Action (Operazioni) e poi Delete schedule (Elimina pianificazione). La pianificazione viene eliminata immediatamente. Seleziona il pulsante di aggiornamento (ricarica) per visualizzare l'elenco delle pianificazioni aggiornato. La pianificazione mostrerà lo stato Deleting (Eliminazione in corso) fino a quando non è completamente rimossa.

## Interruzione dei processi

Puoi interrompere un processo prima che abbia completato l'esecuzione. Puoi scegliere questa opzione se sai che il processo non è configurato correttamente o se richiede troppo tempo per essere completato.

Nella pagina Monitoring (Monitoraggio), nell'elenco Job runs (Esecuzioni di processo), scegli il processo da interrompere, quindi seleziona Actions (Operazioni) e poi Stop run (Interrompi esecuzione).

## Visualizzazione dei processi

Puoi visualizzare tutti i processi nella pagina Jobs (Processi). Per accedere a questa pagina, seleziona Jobs (Processi) nel pannello di navigazione.

Nella pagina Jobs (Processi) puoi visualizzare tutti i processi creati nell'account. L'elenco Your jobs (I tuoi processi) mostra il nome del processo, il tipo, lo stato dell'ultima esecuzione del processo e le date in cui è stato creato e modificato per l'ultima volta. Puoi selezionare il nome di un processo per visualizzare le relative informazioni dettagliate.

Puoi anche utilizzare il pannello di controllo Your jobs (Monitoraggio) per visualizzare tutti i processi. Puoi accedere al pannello di controllo selezionando Monitoring (Monitoraggio) nel pannello di navigazione.

### Personalizzazione della visualizzazione del processo

Puoi personalizzare la modalità di visualizzazione dei processi nella sezione Your jobs (I tuoi processi) della pagina Jobs (Processi). Puoi inoltre inserire del testo nel campo di ricerca per visualizzare solo i lavori con un nome che contiene tale testo.

Selezionando l'icona delle impostazioni

nella sezione Your jobs (I tuoi processi), puoi personalizzare il modo in cui AWS Glue Studio mostra le informazioni nella tabella. Puoi scegliere di inserire a capo le righe di testo nella visualizzazione, modificare il numero di processi visualizzati nella pagina e specificare le colonne da visualizzare.

## Visualizzare le informazioni sulle esecuzioni dei processi recenti

Un processo può essere eseguito più volte man mano che nuovi dati vengono aggiunti nella posizione di origine. Ogni volta che un processo viene eseguito, all'esecuzione viene assegnato un ID univoco e vengono raccolte informazioni su tale esecuzione. Puoi visualizzare queste informazioni utilizzando i seguenti metodi.

- Seleziona la scheda Runs (Esecuzioni) dell'editor visivo per visualizzare le informazioni sull'esecuzione per il processo attualmente mostrato.

  Nella scheda Runs(Esecuzioni) (la pagina Recent job runs [Esecuzioni dei processi recenti]), è presente una scheda per ogni esecuzione del processo. Le informazioni visualizzate nella scheda Runs (Esecuzioni) includono:
  - ID dell'esecuzione del processo
  - Numero di tentativi di esecuzione del processo
  - Stato dell'esecuzione del processo
  - Ora di inizio e fine dell'esecuzione del processo
  - Il runtime per l'esecuzione del processo
  - Un collegamento ai file di log del processo
  - Un collegamento ai file di log degli errori del processo
  - L'errore restituito per i processi non riusciti
- Puoi selezionare l'esecuzione di un processo per visualizzarne le informazioni aggiuntive, tra cui:
  - Inserimento di argomenti
  - Log continui

- Parametri: puoi visualizzare le visualizzazioni dei parametri di base. Per ulteriori informazioni sui parametri inclusi, consulta the section called "Visualizzazione delle Amazon CloudWatch metriche relative all'esecuzione di un job Spark".
- Interfaccia utente Spark: puoi visualizzare i log di Spark relativi al processo nell'interfaccia utente di Spark. Per ulteriori informazioni sull'utilizzo dell'interfaccia utente di Spark, consulta the section called "Monitoraggio con l'interfaccia utente di Spark". Abilita questa funzionalità seguendo la procedura in the section called "Abilitazione dell'interfaccia utente di Spark per processi".

È possibile selezionare Visualizza dettagli per visualizzare informazioni simili nella pagina dei dettagli dell'esecuzione del processo. In alternativa, è possibile accedere alla pagina dei dettagli dell'esecuzione del processo tramite la pagina Monitoraggio. Nel riquadro di navigazione, scegli Monitoring (Monitoraggio). Scorri in basso fino all'elenco Job runs (Esecuzioni processo). Scegli il processo e poi scegli View run details (Visualizza i dettagli dell'esecuzione). I contenuti sono descritti in Visualizzazione dei dettagli di un'esecuzione di un processo.

Per ulteriori informazioni sui log del processo, consulta Visualizzazione dei log di esecuzione del processo.

## Visualizzare lo script del processo

Dopo aver fornito informazioni per tutti i nodi nel processo, AWS Glue Studio genera uno script utilizzato dal processo per leggere i dati dall'origine, trasformarli e scriverli nella posizione di destinazione. Salvando il processo, puoi visualizzare questo script in qualsiasi momento.

Per visualizzare lo script generato per il processo

1. Nel riquadro di navigazione seleziona Jobs (Processi).
2. Nella pagina Jobs (Processi), nell'elenco Your Jobs (I tuoi processi), scegli il nome del processo da esaminare. In alternativa, puoi selezionare un processo nell'elenco, selezionare il menu Actions (Operazioni), quindi scegliere Edit job (Modifica il processo).
3. Nella pagina dell'editor visivo, scegliere la scheda Script nella parte superiore per visualizzare lo script del processo.

   Se desideri modificare lo script del processo, consulta AWS Glue guida alla programmazione.

# Modificare le proprietà del processo

I nodi nel diagramma processo definiscono le azioni eseguite dal processo, ma sono disponibili anche diverse proprietà che è possibile configurare per il processo. Queste proprietà determinano l'ambiente in cui viene eseguito il processo, le risorse utilizzate, le impostazioni di soglia, le impostazioni di sicurezza e altro ancora.

Per personalizzare l'ambiente di esecuzione dei processi

1. Nel riquadro di navigazione seleziona Jobs (Processi).
2. Nella pagina Jobs (Processi), nell'elenco Your Jobs (I tuoi processi), scegli il nome del processo da esaminare.
3. Nella pagina dell'editor visivo, scegliere la scheda Job details (Dettagli del processo) nella parte superiore del pannello di modifica del processo.
4. Modifica le proprietà del processo secondo le necessità.

   Per ulteriori informazioni sulle proprietà del processo, consulta Definizione delle proprietà del processo nella Guida per gli sviluppatori di AWS Glue .
5. Espandi la sezione Advanced properties (Proprietà avanzate) se devi specificare queste proprietà aggiuntive del processo:

   - Script filename (Nome del file di script): il nome del file che memorizza lo script del processo in Amazon S3.
   - Script path (Percorso dello script): la posizione di Amazon S3 in cui è memorizzato lo script del processo.
   - Job metrics — (non disponibile per i job della shell Python) Attiva la creazione Amazon CloudWatch di metriche durante l'esecuzione di questo processo.
   - Registrazione continua — (non disponibile per i lavori della shell Python) Attiva la registrazione continua CloudWatch a, in modo che i log siano disponibili per la visualizzazione prima del completamento del lavoro.
   - Spark UI (Interfaccia utente di Spark) e Spark UI logs path (Percorso dei log dell'interfaccia utente Spark): (non disponibile per i processi di shell Python) attiva l'uso dell'interfaccia utente Spark per il monitoraggio del processo e specifica la posizione per i log dell'interfaccia utente di Spark.
   - Maximum concurrency (Simultaneità massima): imposta il numero massimo di esecuzioni simultanee consentite per il processo.

- Temporary path (Percorso temporaneo): il percorso di una directory di lavoro in Amazon S3, in cui vengono scritti i risultati intermedi temporanei quando AWS Glue esegue lo script.
- Delay notification threshold (minutes) (Soglia notifica di ritardo [minuti]): specifica una soglia di ritardo per il processo. Se il processo viene eseguito per un periodo più lungo di quello specificato dalla soglia, AWS Glue invia una notifica di ritardo per il lavoro a. CloudWatch
- Configurazione e Server-sidecrittografia della sicurezza: utilizza questi campi per scegliere le opzioni di crittografia per il lavoro.
- Use Glue Data Catalog as the Hive metastore: scegli questa opzione per utilizzare AWS Glue Data Catalog come alternativa ad Apache Hive Metastore.
- Additional network connection (Connessione di rete aggiuntiva): per un'origine dati in un VPC, puoi specificare una connessione di tipo `Network` per assicurarti che il processo acceda ai tuoi dati tramite il VPC.
- Python library path (Percorso libreria Python), Dependent jars path (Percorso file .jar dipendenti) (non disponibili per i processi di shell Python) o Referenced files path (Percorso file di riferimento): utilizza questi campi per specificare la posizione dei file aggiuntivi utilizzati dal processo durante l'esecuzione dello script.
- Job Parameters (Parametri del processo): puoi aggiungere un insieme di coppie chiave-valore che vengono passate come parametri denominati allo script del processo. Nelle chiamate Python a AWS Glue API s, è meglio passare i parametri in modo esplicito per nome. Per ulteriori informazioni sull'utilizzo dei parametri in uno script di processo, consulta Passaggio e accesso ai parametri Python in AWS Glue nella Guida per gli sviluppatori di AWS Glue .
- Tag: puoi aggiungere tag al processo per facilitarne l'organizzazione e l'individuazione.

6. Dopo aver modificato le proprietà del processo, salvalo.

## Memorizza i file Spark shuffle su Amazon S3

Alcuni processi ETL richiedono la lettura e la combinazione di informazioni da più partizioni, ad esempio quando si utilizza una trasformazione di join. Questa operazione è indicata come shuffle. Durante uno shuffle, i dati vengono scritti su disco e trasferiti attraverso la rete. Con AWS Glue versione 3.0, puoi configurare Amazon S3 come posizione di storage per questi file. AWS Glue fornisce un gestore shuffle che scrive e legge file shuffle da e verso Amazon S3. La scrittura e la lettura di file shuffle da Amazon S3 è più lenta (del 5-20%) rispetto al disco locale (o Amazon EBS che è estremamente ottimizzato per Amazon EC2). Tuttavia, Amazon S3 offre una capacità di archiviazione illimitata, pertanto non è necessario preoccuparsi errori "`No space left on device`" durante l'esecuzione del lavoro.

Per configurare il processo per l'utilizzo di Amazon S3 per i file shuffle

1. Nella pagina Jobs (Processi), nell'elenco Your Jobs (I tuoi processi), scegli il nome del processo da modificare.
2. Nella pagina dell'editor visivo, scegliere la scheda Job details (Dettagli del processo) nella parte superiore del pannello di modifica del processo.

   Scorri verso il basso fino alla sezione Job parameters (Parametri del processo).
3. Specifica le seguenti coppie chiave-valore.

   - `--write-shuffle-files-to-s3` — `true`

     Questo è il parametro principale che configura il gestore shuffle in AWS Glue per utilizzare i bucket Amazon S3 per la scrittura e la lettura di dati shuffle. Questo parametro è un flag basato sulla presenza: aggiungendolo a qualsiasi valore abilita lo shuffling di Amazon S3. Per disabilitare Amazon S3 shuffling, rimuovi completamente questo parametro dalla configurazione del processo.
   - (Facoltativo) `--write-shuffle-spills-to-s3` — `true` (Supportato solo nella AWS Glue versione 2.0)

     Questo parametro consente di scaricare i file di riversamento nei bucket Amazon S3, che fornisce ulteriore resilienza al processo Spark in AWS Glue. Questo è necessario solo per carichi di lavoro di grandi dimensioni che riversano molti dati sul disco. Quando il flag non è specificato, non viene scritto alcun file riversato intermedio.
   - (Facoltativo) `--conf spark.shuffle.glue.s3ShuffleBucket` — `S3://<shuffle-bucket>`

     Questo parametro specifica il bucket Amazon S3 da utilizzare durante la scrittura dei file shuffle. Se non viene impostato, la posizione è la cartella `shuffle-data` nella posizione specificata per Temporary path (Percorso temporaneo) (`--TempDir`).

     > **Note**
     >
     > Assicurati che la posizione dello shuffle bucket sia nella stessa Regione AWS in cui viene eseguito il job.
     > Inoltre, il servizio shuffle non pulisce i file al termine dell'esecuzione del processo, pertanto è necessario configurare le policy del ciclo di vita dello storage Amazon S3

nella posizione del bucket shuffle. Per ulteriori informazioni, consulta Gestione del ciclo di vita dello storage nella Guida per l'utente di Amazon S3.

## Salvare il lavoro

Finché non si salva il processo, a sinistra della finestra Save (Salva) viene visualizzato un messaggio in rosso che indica che il processo non è stato salvato.

Per salvare il processo

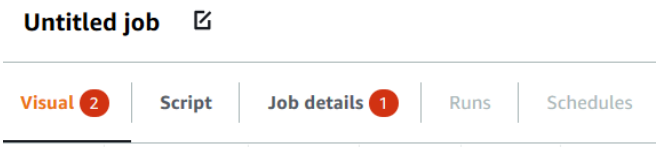
1. Fornisci tutte le informazioni richieste nelle schede Visual (Visivo) e Job details (Dettagli del processo).
2. Seleziona il pulsante Save (Salva).

   Dopo aver salvato il processo, il messaggio 'non salvato' si modifica per mostrare l'ora e la data dell'ultimo salvataggio.

Se esci da AWS Glue Studio prima di salvare il processo, all'accesso successivo ad AWS Glue Studio, verrà visualizzata una notifica. La notifica indica che esiste un processo non salvato e chiede se si desidera ripristinarlo. Se si sceglie di ripristinare il processo, è possibile continuare a modificarlo.

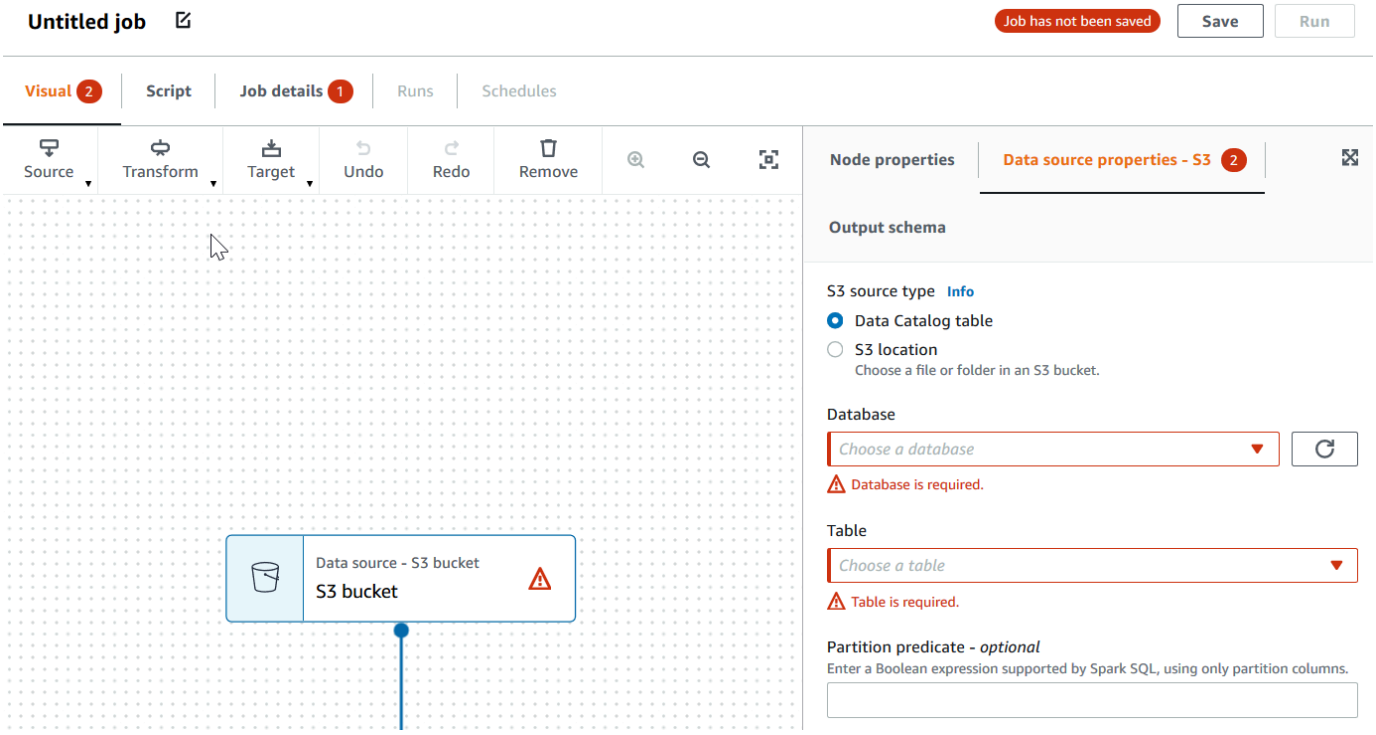
### Risoluzione dei problemi relativi al salvataggio di un processo

Se scegli l'opzione Save (Salva), ma nel tuo lavoro mancano alcune informazioni richieste, nella scheda in cui mancano le informazioni viene visualizzato un messaggio in rosso. Il numero nel messaggio indica quanti campi mancanti sono stati rilevati.

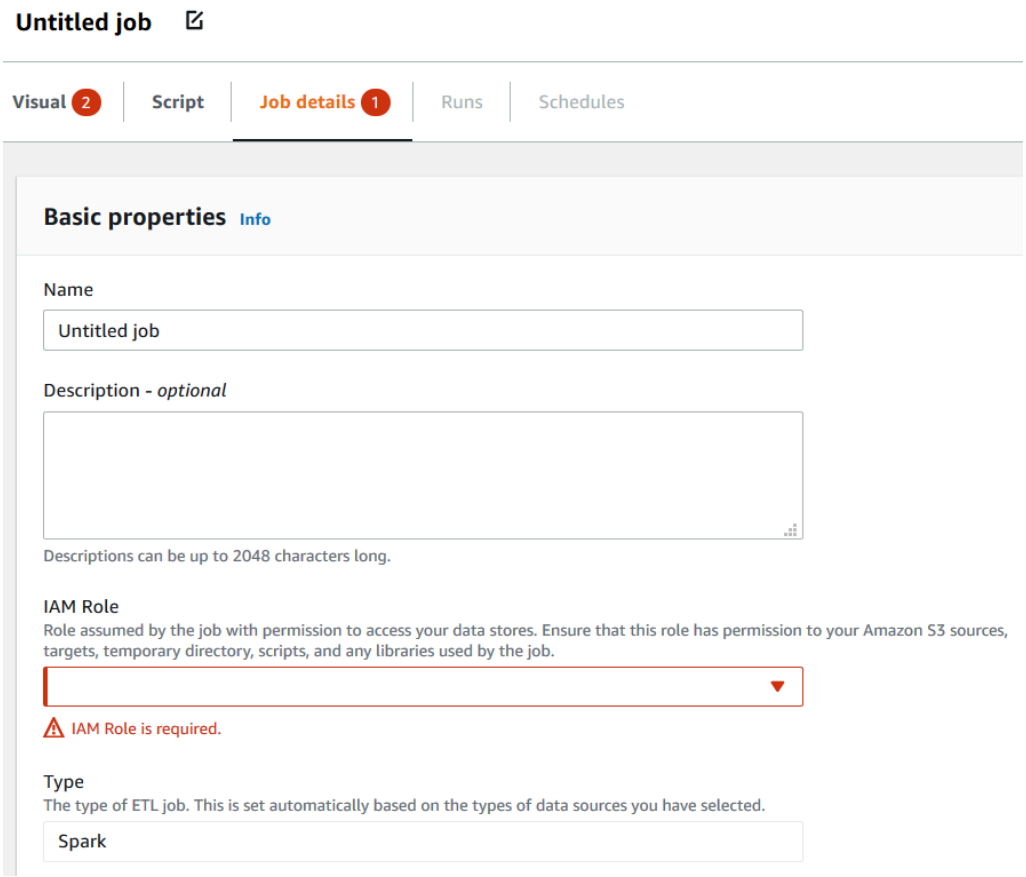



- Se un nodo nell'editor visivo non è configurato correttamente, la scheda Visual (Visivo) mostra un messaggio in rosso e il nodo con l'errore mostra un simbolo di avvertenza ⚠ .

1. Seleziona il nodo. Nel pannello dei dettagli del nodo, nella scheda in cui si trovano le informazioni mancanti o errate viene visualizzato un messaggio in rosso.
2. Scegli la scheda nel pannello dei dettagli del nodo che mostra un messaggio in rosso, quindi individua i campi interessati dal problema, che sono evidenziati. Un messaggio di errore sotto i campi fornisce ulteriori informazioni sul problema.




- Se si verifica un problema con le proprietà del processo, la scheda Job details (Dettagli del processo) mostra un messaggio in rosso. Scegli quella scheda e individua i campi interessati dal problema, che sono evidenziati. Il messaggio di errore sotto i campi fornisce ulteriori informazioni sul problema.




## Clonazione di un processo

Puoi utilizzare l'operazione Clone job (Clona processo) per copiare un processo esistente in un nuovo processo.

Per creare un nuovo processo copiando un processo esistente

1. Nella pagina Jobs (Processi), nell'elenco Your Jobs (I tuoi processi), scegli il processo da duplicare.
2. Nel menu Actions (Operazioni) scegli Clone job (Clona processo).
3. Inserisci un nome per il processo. Puoi quindi salvare o modificare il processo.

## Eliminazione dei processi

È possibile rimuovere i processi che non sono più necessari. È possibile eliminare uno o più processi in un'unica operazione.

Come eliminare i processi da AWS Glue Studio

1. Nella pagina Jobs (Processi), nell'elenco Your Jobs (I tuoi processi), scegli il processo da eliminare.
2. Nel menu Actions (Operazioni) seleziona Delete job (Elimina processo).
3. Conferma di voler eliminare il processo inserendo **`delete`**.

È inoltre possibile eliminare un processo salvato durante la visualizzazione della scheda Job details (Dettagli del processo) per quel lavoro nell'editor visivo.

# Utilizzo dei processi in AWS Glue

Nelle sezioni seguenti vengono fornite informazioni sui processi ETL e Ray in AWS Glue.

Argomenti


- AWS Glue versioni
- Lavorare con Spark jobs in AWS Glue
- Lavorare con Ray Jobs in AWS Glue
- Configurazione delle proprietà del processo per i processi shell di Python in AWS Glue
- Monitoraggio AWS Glue


# AWS Glue versioni

È possibile configurare il parametro della AWS Glue versione quando si aggiunge o si aggiorna un lavoro. La AWS Glue versione determina le versioni di Apache Spark e Python supportate. AWS Glue La versione Python indica la versione supportata per i processi di tipo Spark. La tabella seguente elenca le versioni AWS Glue disponibili, le versioni Spark e Python corrispondenti e altre modifiche di funzionalità.

Puoi utilizzare gli aggiornamenti Generative AI per Apache Spark per aggiornare i tuoi job Glue ETL dalle versioni precedenti di Glue (≥ 2.0) all'ultima versione di Glue.

## AWS Glue versioni

| AWS Glue versione | Versioni dell'ambiente di runtime supportate | Versione di Java supportata | Modifiche della funzionalità |
|---|---|---|---|
| AWS Glue 5.1 | • Spark 3.5.6<br>• Python 3.11<br>• Scala 2.12.18 | Java 17 | AWS Glue 5.1 è la versione predefini ta per i lavori creati senza specificare una versione. AWS Glue<br><br>Oltre all'aggio rnamento del motore Spark, in questa AWS |

| AWS Glue versione | Versioni dell'ambiente di runtime supportate | Versione di Java supportata | Modifiche della funzionalità |
| --- | --- | --- | --- |
| | | | Glue versione sono presenti ottimizza zioni e aggiornamenti, come:<br>• Open Table Formats (OTF) aggiornati a Hudi 1.0.2, Iceberg 1.10.0 e Delta Lake 3.3.2<br>• Formato Iceberg versione 3.0<br>• Hudi Full Table Access (FTA) con letture e scritture<br>• DDL/DML Operazioni di controllo degli accessi a grana fine (FGAC) native di Spark sulle tabelle Hive, Iceberg e Delta Lake registrat e a Lake Formation<br>• Contesto di controllo per le chiamate API Glue and Lake Formation nel CloudTrail registro |

| AWS Glue versione | Versioni dell'ambiente di runtime supportate | Versione di Java supportata | Modifiche della funzionalità |
| --- | --- | --- | --- |
| | | | • Supporto per Iceberg Materiali zed Views |

| AWS Glue versione | Versioni dell'ambiente di runtime supportate | Versione di Java supportata | Modifiche della funzionalità |
| --- | --- | --- | --- |
| AWS Glue 5.0 | • Spark 3.5.4<br>• Python 3.11<br>• Scala 2.12.18 | Java 17 | Oltre agli aggiornamenti del framework, questa versione di AWS Glue presenta ottimizzazioni e aggiornamenti integrati, ad esempio:<br>• Supporto per Amazon SageMaker Unified Studio<br>• Assistenza Amazon SageMaker Lakehouse<br>• Formati a tabella aperta (OTF) aggiornati a Hudi 0.15.0, Iceberg 1.7.1 e Delta Lake 3.3.0<br>• Controllo granulare degli accessi nativo di Spark con Lake Formation.<br>• Supporto per Amazon S3 Access Grants<br>• Supporto `requirements.txt` per l'installazione di |

| AWS Glue versione | Versioni dell'ambiente di runtime supportate | Versione di Java supportata | Modifiche della funzionalità |
| --- | --- | --- | --- |
| | | | librerie Python aggiuntive<br>• Supporto per la derivazione dei dati in Amazon DataZone<br>• Supporto dei bucket di tabelle Amazon S3<br>• AWS Glue Supporto per la visualizzazione multidialettale di Data Catalog<br><br>Limitazioni<br><br>Di seguito sono riportate le limitazio ni della versione 5.0: AWS Glue<br><br>• Il controllo degli accessi a livello di tabella `GlueConte xt` basato su Glue Dynamic Frame/ con AWS Lake Formation autorizza zioni supportate in Glue 4.0 o versioni precedenti non è supportato in |

| AWS Glue versione | Versioni dell'ambiente di runtime supportate | Versione di Java supportata | Modifiche della funzionalità |
| --- | --- | --- | --- |
| | | | Glue 5.0. Usare il nuovo controllo granulare degli accessi (FGAC) nativo di Spark in Glue 5.0.<br><br>Per ulteriori informazioni sulla migrazione a AWS Glue versione 5.0, consultare Migrazione AWS Glue per i job Spark alla versione 5.0 AWS Glue. |

| AWS Glue versione | Versioni dell'ambiente di runtime supportate | Versione di Java supportata | Modifiche della funzionalità |
| --- | --- | --- | --- |
| AWS Glue 4.0 | Versioni dell'ambiente Spark<br>• Spark 3.3.0<br>• Python 3.10 | Java 8 | AWS Glue La versione 4.0 include una serie di ottimizzazioni e aggiornamenti, come: AWS Glue<br>• Numerosi aggiornamenti delle funzionalità Spark da Spark 3.1 a Spark 3.3:<br>• Diversi miglioramenti delle funzionalità se abbinato a Pandas. Per ulteriori informazioni, consulta Novità di Spark 3.3.<br>• Ottimizzazioni aggiuntive sviluppate su Amazon EMR.<br>• Aggiornamento a EMR File System (EMRFS) 2.53.<br>• Migrazione a Log4j 2 da Log4j 1.x<br>• Diversi aggiornamenti del modulo Python da AWS Glue 3.0, come una |

| AWS Glue versione | Versioni dell'ambiente di runtime supportate | Versione di Java supportata | Modifiche della funzionalità |
|---|---|---|---|
| | | | versione aggiornata di Boto.<br>• Aggiornamento di diversi connettori, tra cui il connettor e Amazon Redshift predefinito. Per informazioni, consulta Appendice C: Aggiornamenti dei connettori.<br>• Aggiornamento di diversi driver JDBC. Per informazioni, consulta Appendice B: aggiornamenti dei driver JDBC.<br>• Aggiornato con un nuovo connettore Amazon Redshift e driver JDBC.<br>• Supporto nativo per framework open data lake con Apache Hudi, Delta Lake e Apache Iceberg.<br>• Supporto nativo per il Cloud Shuffle Storage Plugin basato su Amazon S3 (un plug-in Apache Spark) per |

| AWS Glue versione | Versioni dell'ambiente di runtime supportate | Versione di Java supportata | Modifiche della funzionalità |
| --- | --- | --- | --- |
| | | | utilizzare Amazon S3 per lo shuffling e la capacità di archiviazione elastica.<br><br>Limitazioni<br><br>Le limitazioni seguenti sono relative a AWS Glue 4.0:<br><br>• AWS Glue le trasformazioni relative all'apprendimento automatico e alle informazioni di identificazione personale (PII) non sono ancora disponibili nella versione 4.0. AWS Glue<br><br>Per ulteriori informazioni sulla migrazione a AWS Glue versione 4.0, consulta Migrazione AWS Glue per i job Spark alla versione 4.0 AWS Glue. |

| AWS Glue versione | Versioni dell'ambiente di runtime supportate | Versione di Java supportata | Modifiche della funzionalità |
| --- | --- | --- | --- |
| | Versioni dell'ambiente Ray<br>• Ray 2.4.0<br>Python 3.9 | N/D | Crea ed esegui applicazioni Python distribuite con AWS Glue for Ray.<br>• Supporta la distribuzione dei dati Ray-2.4.0 (`ray[data]` ) con Python 3.9. Per ulteriori informazioni su questa versione di Ray, vedete Ray-2.4.0 nel repository Ray. GitHub<br>• Supporta l'installazione di librerie Python aggiuntive nell'ambiente di runtime `Ray2.4`. Per ulteriori informazioni, consulta the section called “Moduli Python aggiuntivi per i processi Ray”.<br>• Integra log e metriche di Ray Jobs con Amazon. CloudWatch Per ulteriori informazioni, consultare |

| AWS Glue versione | Versioni dell'ambiente di runtime supportate | Versione di Java supportata | Modifiche della funzionalità |
|---|---|---|---|
| | | | the section called "Risoluzione degli errori relativi ai processi Ray" e the section called "Parametri dei processi Ray".<br>• Aggrega e visualizza le metriche per i lavori Ray in AWS Glue Studio, su ogni pagina di esecuzione del lavoro.<br>• Supporta la distribuzione di file in ogni directory di lavoro del cluster, il riversamento di oggetti dall'archivio di oggetti Ray ad Amazon S3 e il controllo del numero minimo di nodi worker allocati al processo Ray. Per ulteriori informazioni, consulta the section called "Parametri dei processi Ray". |

| AWS Glue versione | Versioni dell'ambiente di runtime supportate | Versione di Java supportata | Modifiche della funzionalità |
|---|---|---|---|
| | | | Limitazioni sui processi Ray in AWS Glue 4.0<br>• AWS Glue le sessioni interattive per Ray rimangono disponibili in anteprima per questa versione.<br>• AWS Glue l'integrazione di for Ray con Amazon VPC non è attualmente disponibile. Le risorse in un VPC in non AWS saranno accessibili senza un percorso pubblico. Per ulteriori informazioni sull'utilizzo AWS Glue con Amazon VPC, consulta. the section called "Configurazione degli endpoint VPC (AWS PrivateLink) per AWS Glue "<br>• AWS Glue for Ray è disponibile negli Stati Uniti orientali (Virginia settentrionale), Stati Uniti |

| AWS Glue versione | Versioni dell'ambiente di runtime supportate | Versione di Java supportata | Modifiche della funzionalità |
| --- | --- | --- | --- |
| | | | orientali (Ohio), Stati Uniti occidentali (Oregon), Asia Pacifico (Tokyo) ed Europa (Irlanda). |

| AWS Glue versione | Versioni dell'ambiente di runtime supportate | Versione di Java supportata | Modifiche della funzionalità |
| --- | --- | --- | --- |
| AWS Glue 3.0 | • Spark 3.1.1<br>• Python 3.7 | Java 8 | Oltre all'aggiornamento del motore Spark a 3.0, questa versione di AWS Glue presenta ottimizzazioni e aggiornamenti integrati, ad esempio:<br>• Crea la libreria AWS Glue ETL sulla base di Spark 3.0, che è una delle principali release di Spark.<br>• I processi di streaming sono supportati su AWS Glue 3.0.<br>• Include nuove ottimizzazioni del runtime di AWS Glue Spark per prestazioni e affidabilità:<br>  • Elaborazione colonnare in memoria più veloce basata su Apache Arrow per la lettura dei dati CSV.<br>  • Esecuzione basata su |

| AWS Glue versione | Versioni dell'ambiente di runtime supportate | Versione di Java supportata | Modifiche della funzionalità |
| --- | --- | --- | --- |
| | | | SIMD per letture vettorizzate con dati CSV.<br>• L'aggiornamento Spark include anche ulteriori ottimizzazioni sviluppate su Amazon EMR.<br>• EMRFS aggiornato da 2.38 a 2.46, con l'abilitazione di nuove caratteristiche e correzioni di bug per l'accesso ad Amazon S3.<br>• Sono state aggiornate diverse dipendenze necessarie per la nuova versione di Spark.<br>• Driver JDBC aggiornati per le nostre origini dati supportate in modo nativo.<br><br>Limitazioni |

| AWS Glue versione | Versioni dell'ambiente di runtime supportate | Versione di Java supportata | Modifiche della funzionalità |
| --- | --- | --- | --- |
| | | | Le limitazioni seguenti sono relative a AWS Glue 3.0:<br>• AWS Glue le trasformazioni dell'apprendimento automatico non sono ancora disponibili nella versione 3.0. AWS Glue<br>• Alcuni connettori Spark personalizzati non funzionano con AWS Glue 3.0 se dipendono da Spark 2.4 e non sono compatibili con Spark 3.1. |

| AWS Glue versione | Versioni dell'ambiente di runtime supportate | Versione di Java supportata | Modifiche della funzionalità |
| --- | --- | --- | --- |
| AWS Glue 2.0 (fine del ciclo di vita il 1° aprile 2026) | • Spark 2.4.3<br>• Python 3.7 | N/D | Oltre alle funzionalità fornite nella AWS Glue versione 1.0, la AWS Glue versione 2.0 offre anche:<br>• Un'infrastruttura aggiornata per l'esecuzione dei job ETL di Apache Spark AWS Glue con tempi di avvio ridotti.<br>• La registrazione di default è ora in tempo reale, con flussi separati per driver ed esecutori, e contiene output ed errori.<br>• Supporto per la specifica di moduli Python o versioni diverse aggiuntivi a livello di processo.<br>**Note**<br>AWS Glue la versione 2.0 differisce dalla AWS Glue versione |

| AWS Glue versione | Versioni dell'ambiente di runtime supportate | Versione di Java supportata | Modifiche della funzionalità |
| --- | --- | --- | --- |
| | | | 1.0 per alcune dipendenze e versioni a causa delle modifiche architett oniche sottostanti. Convalida i processi AWS Glue prima di eseguire la migrazione tra le versioni principali di AWS Glue . |

| AWS Glue versione | Versioni dell'ambiente di runtime supportate | Versione di Java supportata | Modifiche della funzionalità |
| --- | --- | --- | --- |
| AWS Glue 1.0 (fine del ciclo di vita il 1° aprile 2026) | • Spark 2.4.3<br>• Python 2.7<br>• Python 3.6 | N/D | Puoi mantenere i segnalibri dei processi per i formati Parquet e ORC nei processi AWS Glue ETL (utilizzando AWS Glue versione 1.0). In precedenza, era possibile aggiunger e ai preferiti solo i formati sorgente più comuni di Amazon S3 come JSON, CSV, Apache Avro e XML nei job ETL. AWS Glue<br><br>Quando si impostano le opzioni di formato per gli input e gli output ETL, è possibile specifica re di utilizzare il reader/writer formato Apache Avro 1.8 per supportare la lettura e la scrittura dei tipi logici Avro (utilizzando la versione 1.0). AWS Glue In precedenza, era supportato solo il formato Avro versione 1.7. reader/writer |

| AWS Glue versione | Versioni dell'ambiente di runtime supportate | Versione di Java supportata | Modifiche della funzionalità |
| --- | --- | --- | --- |
| | | | Il tipo di connessione DynamoDB supporta un'opzione di scrittura (utilizzando AWS Glue versione 1.0).<br><br>Limitazioni<br><br>Le limitazioni seguenti sono relative a AWS Glue 1.0:<br><br>• AWS Glue le versioni 0.9 e 1.0 non sono disponibili in Asia Pacifico (Giacarta) (`ap-southeast-3` ), Medio Oriente (Emirati Arabi Uniti) (`me-central-1` ) o in altre nuove regioni in futuro. |

| AWS Glue versione | Versioni dell'ambiente di runtime supportate | Versione di Java supportata | Modifiche della funzionalità |
| --- | --- | --- | --- |
| AWS Glue 0.9 (fine del ciclo di vita il 1° aprile 2026) | • Spark 2.2.1<br>• Python 2.7 | N/D | Limitazioni<br><br>Le limitazioni seguenti sono relative a AWS Glue 0.9:<br><br>• AWS Glue le versioni 0.9 e 1.0 non sono disponibili in Asia Pacifico (Giacarta) (`ap-southeast-3` ), Medio Oriente (Emirati Arabi Uniti) (`me-central-1` ) o in altre nuove regioni in futuro. |

> **Note**
>
> Le seguenti versioni di Glue supportano queste versioni di PythonShell:
>
> - PythonShell la versione 3.6 è supportata nella versione 1.0 di Glue.
> - PythonShell la versione 3.9 è supportata nella versione 3.0 di Glue.
>
> Inoltre, gli endpoint di sviluppo sono supportati solo nelle versioni 1.0 e 0.9 di Glue.

## Policy di supporto versione AWS Glue

AWS Glue è un servizio di integrazione dati serverless che semplifica l'individuazione, la preparazione e la combinazione di dati per analisi dei dati, machine learning e sviluppo di applicazioni. Un processo AWS Glue contiene la logica di business che esegue le attività di

integrazione dei dati in AWS Glue. Esistono tre tipi di processi in AWS Glue: Spark (in batch e in streaming), Ray e shell Python. Quando definisci il processo, specifica la versione di AWS Glue che configura le versioni dell'ambiente di runtime Spark, Ray o Python sottostante. Ad esempio: un job Spark AWS Glue versione 5.0 supporta Spark 3.5.4 e Python 3.11.

## Policy di supporto

AWS Glue le versioni sono basate su una combinazione di sistema operativo, linguaggio di programmazione e librerie software soggette a manutenzione e aggiornamenti di sicurezza. AWS Glue la politica di supporto delle versioni prevede di interrompere il supporto per una versione quando uno dei componenti principali della versione raggiunge la fine del supporto comunitario a lungo termine (LTS) e gli aggiornamenti di sicurezza non sono più disponibili. AWS Glue la politica di supporto delle versioni include i seguenti stati:

End of Support (EOS): quando una AWS Glue versione raggiunge EOS:

- AWS Glue non applicherà più patch di sicurezza o altri aggiornamenti alle versioni EOS.
- AWS Glue le offerte di lavoro sulle versioni EOS non sono idonee al supporto tecnico.
- AWS Glue potrebbe non funzionare SLAs quando i lavori vengono eseguiti su versioni EOS.

Fine del ciclo di vita (EOL) - Quando una AWS Glue versione raggiunge la fine del ciclo di vita (EOL):

- Non è più possibile creare nuovi AWS Glue lavori o sessioni interattive nelle versioni EOL.
- Non è più possibile avviare le esecuzioni di job su queste AWS Glue versioni.
- AWS Glue interromperà le esecuzioni di job e le sessioni interattive esistenti nelle versioni EOL.
- Le versioni EOL verranno rimosse da AWS Glue SDKs e. APIs

Le seguenti versioni AWS Glue hanno raggiunto la fine del supporto e non saranno più disponibili dopo la data di fine del ciclo di vita. Le modifiche allo stato di supporto di una versione iniziano a mezzanotte (fuso orario del Pacifico) della data specificata.

| Tipo | Versione Glue | Fine del supporto | Fine del ciclo di vita |
|---|---|---|---|
| Spark | Glue version 0.9 (Spark 2.2, Scala 2, Python 2) | 01/06/2022 | 01/04/2026 |

| Tipo | Versione Glue | Fine del supporto | Fine del ciclo di vita |
|---|---|---|---|
| Spark | Glue version 1.0 (Spark 2.4, Python 2) | 01/06/2022 | 01/04/2026 |
| Spark | Glue version 1.0 (Spark 2.4, Scala 2, Python 3) | 30/09/2022 | 01/04/2026 |
| Spark | Glue version 2.0 (Spark 2.4, Python 3) | 31/2024 | 1/4/2026 |
| Tipo | Versione di Python | Fine del supporto | Fine del ciclo di vita |
| Shell Python | Python 2 (AWS Glue versione 1.0) | 01/06/2022 | 01/04/2026 |
| Shell Python | PythonShell 3.6 (Glue versione 1.0) | 31/03/2026 | N/A |
| Tipo | Versione notebook | Fine del supporto | Fine del ciclo di vita |
| Endpoint di sviluppo | Notebook Zeppelin | 30/09/2022 | N/A |

> **Note**
>
> La creazione di nuovi lavori in AWS Glue Python Shell 3.6 non sarà consentita una volta raggiunta la fine del supporto il 31 marzo 2026, ma puoi continuare ad aggiornare ed eseguire i lavori esistenti. Tuttavia, i lavori eseguiti su versioni fuori produzione non sono idonei al supporto tecnico. AWS Glue non applicherà patch di sicurezza o altri aggiornamenti alle versioni fuori produzione. AWS Glue inoltre non verrà rispettato SLAs quando i job vengono eseguiti su versioni fuori produzione.

AWS consiglia la migrazione dei processi alle versioni supportate.

Per informazioni sulla migrazione dei job Spark alla AWS Glue versione più recente, consulta Migrazione dei AWS Glue job alla versione 5.1. AWS Glue

Per eseguire la migrazione dei processi di shell Python all'ultima versione di AWS Glue:

- Nella console, scegli `Python 3 (Glue Version 5.1)`.
- Nell'UpdateJobAPI CreateJob/, imposta il `GlueVersion` parametro su e il `5.1` `PythonVersion` to `3` sotto il parametro. `Command` La configurazione `GlueVersion` non influisce sul comportamento dei processi della shell Python, quindi non c'è alcun vantaggio nell'incrementare `GlueVersion`.
- Devi rendere lo script del tuo processo compatibile con Python 3.

## Migrazione AWS Glue per i job Spark alla versione 5.1 AWS Glue

Questo argomento descrive le modifiche tra AWS Glue le versioni 0.9, 1.0, 2.0, 3.0, 4.0 e 5.0 per consentire la migrazione delle applicazioni Spark e dei lavori ETL alla 5.1. AWS Glue Descrive inoltre le funzionalità della AWS Glue versione 5.1 e i vantaggi del suo utilizzo.

Per utilizzare questa funzionalità con i tuoi lavori AWS Glue ETL, scegli `Glue version` quando **`5.1`** crei i tuoi lavori.

Argomenti


- Nuove funzionalità
- Azioni per migrare alla versione 5.1 AWS Glue
- Elenco di controllo della migrazione
- Migrazione da 5.0 a 5.1 AWS Glue AWS Glue
- Migrazione dalle AWS Glue versioni precedenti alla 5.1 AWS Glue
- Migrazione di connettori e driver JDBC per 5.1 AWS Glue


### Nuove funzionalità

Questa sezione descrive le nuove funzionalità e i vantaggi della AWS Glue versione 5.1.

- Aggiornamento di Apache Spark dalla versione 3.5.4 della AWS Glue versione 5.0 alla versione 3.5.6 della versione 5.1. AWS Glue
- Open Table Formats (OTF) aggiornati a Hudi 1.0.2, Iceberg 1.10.0 e Delta Lake 3.3.2
- Iceberg Materialized Views - Crea e gestisci Iceberg Materialized Views (MV). Per ulteriori informazioni, consulta il post del blog

- Formato Iceberg versione 3.0: estende i tipi di dati e le strutture di metadati esistenti per aggiungere nuove funzionalità. Per ulteriori informazioni, consulta Iceberg Table Spec.
- Hudi Full Table Access - Controllo completo dell'accesso alla tabella (FTA) per Apache Hudi in Apache Spark in base alle politiche definite in. AWS Lake Formation Questa funzionalità consente le operazioni di lettura e scrittura dai lavori AWS Glue ETL su tabelle AWS Lake Formation registrate quando il ruolo lavorativo ha accesso completo alla tabella.
- Supporto nativo di Spark per il controllo granulare degli accessi (FGAC) tramite AWS Lake Formation DDL/DML operazioni (come CREATE, ALTER, DELETE, DROP) con controllo granulare degli accessi per le tabelle Apache Hive, Apache Iceberg e Delta Lake registrate in. AWS Lake Formation
- Contesto di controllo per i lavori Spark: il contesto di controllo per i lavori AWS Glue ETL sarà disponibile per le chiamate API nei log. AWS Glue AWS Lake Formation AWS CloudTrail

Problemi noti e limitazioni

Tieni presente i seguenti problemi e limitazioni noti:

- Supporto limitato per la clausola View SQL per la creazione di viste materializzate, la riscrittura delle query e l'aggiornamento incrementale. Maggiori dettagli sono disponibili nella pagina di documentazione della funzionalità Iceberg Materialized Views
- Le scritture Hudi FTA richiedono l'utilizzo HoodieCredentialedHadoopStorage per la vendita di credenziali durante l'esecuzione del lavoro. Imposta la seguente configurazione durante l'esecuzione dei job Hudi:

  ```
  hoodie.storage.class=org.apache.spark.sql.hudi.storage.HoodieCredentialedHadoop
  ```
- Il supporto di scrittura Hudi FTA funziona solo con le configurazioni Hudi predefinite. Le impostazioni Hudi personalizzate o non predefinite potrebbero non essere completamente supportate e potrebbero causare un comportamento imprevisto. Inoltre, il clustering for Hudi Merge-On-Read (MOR) non è supportato nella modalità di scrittura FTA.

Modifiche importanti

Notare le seguenti modifiche speciali:

- Il file system S3A ha sostituito EMRFS come connettore S3 predefinito. Per informazioni su come effettuare la migrazione, consulta. Migrazione da 5.0 a 5.1 AWS Glue AWS Glue

## Azioni per migrare alla versione 5.1 AWS Glue

Per i processi esistenti, modifica la `Glue version` dalla versione precedente a `Glue 5.1` nella configurazione del processo.

- In AWS Glue Studio, scegli `Glue 5.1 - Supports Spark 3.5.6, Scala 2, Python 3` in`Glue version`.
- Nell'API, scegli **`5.1`** nel parametro `GlueVersion` nell'operazione API `UpdateJob`.

Per i nuovi processi, scegli `Glue 5.1` al momento della creazione.

- Nella console, scegli `Spark 3.5.6, Python 3 (Glue Version 5.1) or Spark 3.5.6, Scala 2 (Glue Version 5.1)` in `Glue version`.
- In AWS Glue Studio, scegli `Glue 5.1 - Supports Spark 3.5.6, Scala 2, Python 3` in`Glue version`.
- Nell'API, scegli **`5.1`** nel parametro `GlueVersion` nell'operazione API `CreateJob`.

Per visualizzare i registri degli eventi di Spark della AWS Glue versione 5.1 della versione AWS Glue 2.0 o precedente, avvia un server di cronologia Spark aggiornato per AWS Glue 5.1 utilizzando o Docker. CloudFormation

## Elenco di controllo della migrazione

Rivedi questo elenco di controllo per la migrazione:

- [Python] Aggiorna i riferimenti di avvio da 1.34 a 1.40.

## Migrazione da 5.0 a 5.1 AWS Glue AWS Glue

Tutti i parametri di lavoro e le funzionalità principali esistenti in AWS Glue 5.0 esisteranno nella versione AWS Glue 5.1. Nota le seguenti modifiche durante la migrazione:

- Nella AWS Glue versione 5.1, il filesystem S3A ha sostituito EMRFS come connettore S3 predefinito. Se entrambi `spark.hadoop.fs.s3a.endpoint` e 3 non `spark.hadoop.fs.s3a.endpoint.region` sono impostati, la regione predefinita utilizzata da S3A è. `us-east-2` Ciò può causare problemi, come errori di timeout di caricamento di S3, in particolare per i processi VPC. Per mitigare i problemi causati da questa modifica, imposta la

configurazione `spark.hadoop.fs.s3a.endpoint.region` Spark quando usi il file system S3A nella versione 5.1. AWS Glue

- Per continuare a utilizzare EMRFS anziché S3A, imposta le seguenti configurazioni spark:

```
    --conf spark.hadoop.fs.s3.impl=com.amazon.ws.emr.hadoop.fs.EmrFileSystem
    --conf spark.hadoop.fs.s3n.impl=com.amazon.ws.emr.hadoop.fs.EmrFileSystem
    --conf
spark.hadoop.fs.AbstractFileSystem.s3.impl=org.apache.hadoop.fs.s3.EMRFSDelegate
```

Consulta la documentazione relativa alla migrazione di Spark:

- Guida alla migrazione: Spark Core
- Guida alla migrazione: SQL, set di dati e DataFrame
- Guida alla migrazione: streaming strutturato
- Aggiornamento PySpark

## Migrazione dalle AWS Glue versioni precedenti alla 5.1 AWS Glue

- Per i passaggi di migrazione relativi alla versione AWS Glue 4.0 alla AWS Glue 5.0, consulta Migrazione da AWS Glue 4.0 a AWS Glue 5.0.
- Per le fasi di migrazione relative alla AWS Glue versione 3.0 alla AWS Glue 5.0, consulta Migrazione da AWS Glue 3.0 a AWS Glue 5.0.
- Per le fasi di migrazione relative alla versione AWS Glue 2.0 alla AWS Glue 5.0 e un elenco delle differenze di migrazione tra la AWS Glue versione 2.0 e 4.0, consulta Migrazione dalla AWS Glue 2.0 alla AWS Glue 5.0.

## Migrazione di connettori e driver JDBC per 5.1 AWS Glue

Per le versioni dei connettori JDBC e data lake che sono state aggiornate, consulta:

- Appendice B: aggiornamenti dei driver JDBC
- Appendice C: Aggiornamenti dei connettori
- Appendice D: Aggiornamenti di formato a tabella aperta

Le seguenti modifiche si applicano agli aggiornamenti della versione OTF identificati nella Appendice D: Aggiornamenti di formato a tabella aperta versione 5.1. AWS Glue

Apache Hudi

Nota le seguenti modifiche:

- Supporta l'accesso in lettura e scrittura FTA sulle tabelle registrate di Lake Formation.

Apache Iceberg

Nota le seguenti modifiche:

- Supporta la versione 3 del formato Iceberg. Sono supportate le seguenti funzionalità:
  - Trasformazioni multiargomento per il partizionamento e l'ordinamento.
  - Tracciamento del lignaggio delle righe.
  - Vettori di cancellazione. Scopri di più nel post del blog
  - Chiavi di crittografia delle tabelle.
  - Supporto dei valori predefiniti per le colonne.
- Supporta le scritture FGAC native di Spark su tabelle registrate. AWS Lake Formation
- Compatibilità con Athena SQL - Impossibile leggere le tabelle Iceberg V3 create da EMR Spark a causa di un errore: `GENERIC_INTERNAL_ERROR: Cannot read unsupported version 3`

Delta Lake

Nota le seguenti modifiche:

- Supporta l'accesso in lettura e scrittura FTA sulle tabelle registrate di Lake Formation.

Appendice A: Aggiornamenti importanti delle dipendenze

Di seguito sono riportati gli aggiornamenti delle dipendenze:

| Dipendenza | Versione in 5.1 AWS Glue | Versione in AWS Glue 5.0 | Versione in AWS Glue 4.0 | Versione in AWS Glue 3.0 | Versione in AWS Glue 2.0 | Versione in AWS Glue 1.0 |
|---|---|---|---|---|---|---|
| Java | 17 | 17 | 8 | 8 | 8 | 8 |
| Spark | 3.5.6 | 3,5,4 | 3.3.0-amzn-1 | 3.1.1-amzn-0 | 2.4.3 | 2.4.3 |
| Hadoop | 3.4.1 | 3.4.1 | 3.3.3-amzn-0 | 3.2.1-amzn-3 | 2.8.5-amzn-5 | 2.8.5-amzn-1 |
| Scala | 2,12,18 | 2,12,18 | 2,12 | 2,12 | 2.11 | 2.11 |
| Jackson | 2,15,2 | 2,15,2 | 2,12 | 2,12 | 2.11 | 2.11 |
| Hive | 2.3.9-amzn-4 | 2.3.9-amzn-4 | 2.3.9-amzn-2 | 2.3.7-amzn-4 | 1.2 | 1.2 |
| EMRFS | 2,73,0 | 2,69,0 | 2,54,0 | 2,46,0 | 2.38.0 | 2.30.0 |
| Json4s | 3.7.0-M11 | 3.7.0-M11 | 3.7.0-M11 | 36.6 | 3.5.x | 3.5.x |
| Arrow | 120,1 | 12,0.1 | 7,0,0 | 2.0.0 | 0.10.0 | 0.10.0 |
| AWS Glue Client Data Catalog | 4.9.0 | 4.5.0 | 3.7.0 | 3.0.0 | 1.10.0 | N/D |
| AWS SDK per Java | 2.35.5 | 2,29,52 | 1.12 | 1.12 | | |
| Python | 3,11 | 3,11 | 3,10 | 3.7 | 2.7 e 3.6 | 2.7 e 3.6 |
| Boto | 1,40,61 | 1,34,131 | 1,26 | 1,18 | 1.12 | N/D |
| Connettore EMR DynamoDB | 5.7.0 | 5.6.0 | 4,16,0 | | | |

## Appendice B: aggiornamenti dei driver JDBC

Di seguito sono riportati gli aggiornamenti dei driver JDBC:

| Driver | Versione del driver JDBC in 5.1 AWS Glue | Versione del driver JDBC nella versione 5.0 AWS Glue | Versione del driver JDBC nella versione 4.0 AWS Glue | Versione del driver JDBC nella versione 3.0 AWS Glue | Versione del driver JDBC nelle versioni precedenti AWS Glue |
|---|---|---|---|---|---|
| MySQL | 8.0.33 | 8,0,33 | 8.0.23 | 8.0.23 | 5.1 |
| Microsoft SQL Server | 102,0 | 10.2,0 | 9,40 | 7,0,0 | 6.1.0 |
| Database Oracle | 23,3,023,09 | 23,3,023,09 | 21,7 | 21,1 | 11.2 |
| PostgreSQL | 42,7,3 | 42,7,3 | 42,36 | 4,2,18 | 42,1,0 |
| Amazon Redshift | redshift-jdbc42-2.1.0.29 | redshift-jdbc42-2.1.0.29 | redshift-jdbc42-2.1.0.16 | redshift-jdbc41-1.2.12.1017 | redshift-jdbc41-1.2.12.1017 |
| SAP Hana | 2,20,17 | 2,20,17 | 2,17,12 | | |
| Teradata | 20,00,00,33 | 20,00,00,33 | 20,00,00,06 | | |

## Appendice C: Aggiornamenti dei connettori

Di seguito sono riportati gli aggiornamenti dei connettori:

| Driver | Versione del connettore in 5.1 AWS Glue | Versione del connettore in AWS Glue 5.0 | Versione del connettore in AWS Glue 4.0 | Versione del connettore in AWS Glue 3.0 |
|---|---|---|---|---|
| Connettore EMR DynamoDB | 5.7.0 | 5.6.0 | 4.16.0 | |

| Driver | Versione del connettore in 5.1 AWS Glue | Versione del connettore in AWS Glue 5.0 | Versione del connettore in AWS Glue 4.0 | Versione del connettore in AWS Glue 3.0 |
|---|---|---|---|---|
| Amazon Redshift | 6.4.2 | 64,0 | 6.1.3 | |
| OpenSearch | 1.2.0 | 1.2.0 | 1.0.1 | |
| MongoDB | 10.3.0 | 10.3.0 | 10.0.4 | 3.0.0 |
| Snowflake | 3.1.1 | 3.0.0 | 2.12.0 | |
| Google BigQuery | 0.32.2 | 0,32,2 | 0,32,2 | |
| AzureCosmos | 4,33,0 | 4,33,0 | 42,0 | |
| AzureSQL | 1.3.0 | 1.3.0 | 1.3.0 | |
| Vertica | 33,5 | 3.3.5 | 3.3.5 | |

Appendice D: Aggiornamenti di formato a tabella aperta

Di seguito sono riportati gli aggiornamenti di formato a tabella aperta:

| OTF | Versione del connettore in 5.1 AWS Glue | Versione del connettore in AWS Glue 5.0 | Versione del connettore in AWS Glue 4.0 | Versione del connettore in AWS Glue 3.0 |
|---|---|---|---|---|
| Hudi | 1.0.2 | 0.15.0 | 0.12.1 | 0,10,1 |
| Delta Lake | 3.3.2 | 3.3.0 | 2.1.0 | 1.0.0 |
| Iceberg | 1.10.0 | 1.7.1 | 1.0.0 | 0.13.1 |

## Migrazione AWS Glue per i job Spark alla versione 5.0 AWS Glue

Questo argomento descrive le modifiche tra AWS Glue le versioni 0.9, 1.0, 2.0, 3.0 e 4.0 per consentire la migrazione delle applicazioni Spark e dei lavori ETL alla 5.0. AWS Glue Descrive inoltre le funzionalità della AWS Glue versione 5.0 e i vantaggi del suo utilizzo.

Per utilizzare questa funzionalità con i tuoi lavori AWS Glue ETL, scegli `Glue version` quando **`5.0`** crei i tuoi lavori.

Argomenti


- Nuove funzionalità
- Azioni per migrare alla versione 5.0 AWS Glue
- Elenco di controllo della migrazione
- AWS Glue Funzionalità 5.0
- Migrazione da 4.0 a 5.0 AWS Glue AWS Glue
- Migrazione da AWS Glue 3.0 a 5.0 AWS Glue
- Migrazione da AWS Glue 2.0 a 5.0 AWS Glue
- Modifiche al comportamento di registrazione nella versione 5.0 AWS Glue
- Migrazione di connettori e driver JDBC per 5.0 AWS Glue


## Nuove funzionalità

Questa sezione descrive le nuove funzionalità e i vantaggi della AWS Glue versione 5.0.

- Aggiornamento di Apache Spark dalla versione 3.3.0 della versione AWS Glue 4.0 alla versione 3.5.4 della versione 5.0. AWS Glue Per informazioni, consulta Miglioramenti principali da Spark 3.3.0 a Spark 3.5.4.
- Spark-native controllo degli accessi a grana fine (FGAC) utilizzando Lake Formation. Ciò include FGAC per le tabelle Iceberg, Delta e Hudi. Per ulteriori informazioni, consulta Using AWS Glue with AWS Lake Formation per un controllo granulare degli accessi.

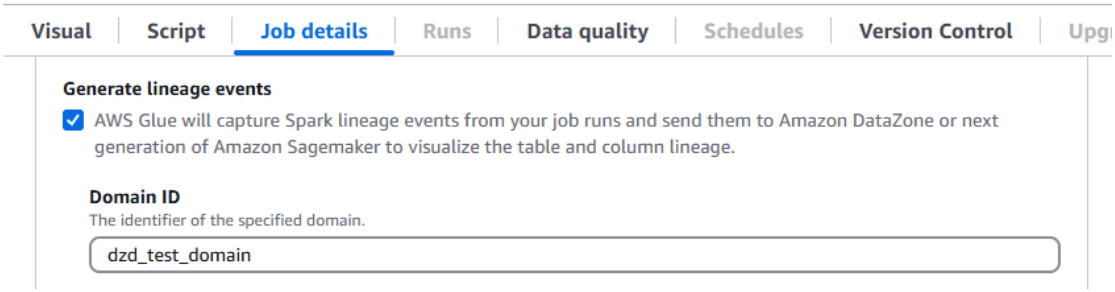
  Tenete presente le seguenti considerazioni o limitazioni per FGAC: Spark-native

  - Attualmente la scrittura dei dati non è supportata
  - Scrivere in Iceberg `GlueContext` tramite Lake Formation richiede invece l'uso del controllo degli accessi IAM

  Per un elenco completo delle limitazioni e delle considerazioni relative all'uso Spark-native di FGAC, vedere. the section called "Considerazioni"
- Supporto per Amazon S3 Access Grants come soluzione scalabile di controllo degli accessi ai tuoi dati Amazon S3 da. AWS Glue Per ulteriori informazioni, consulta Utilizzo di Amazon S3 Access Grants con AWS Glue.

- Formati a tabella aperta (OTF) aggiornati a Hudi 0.15.0, Iceberg 1.7.1 e Delta Lake 3.3.0
- Supporto per Amazon SageMaker Unified Studio.
- Amazon SageMaker Lakehouse e l'integrazione dell'astrazione dei dati. Per ulteriori informazioni, consulta Interrogazione dei cataloghi di dati dei metastore da ETL AWS Glue.
- Supporto per l'installazione di librerie Python aggiuntive utilizzando `requirements.txt`. Per ulteriori informazioni, consulta Installazione di librerie Python aggiuntive in AWS Glue 5.0 o versioni successive utilizzando requirements.txt.
- AWS Glue 5.0 supporta la derivazione dei dati in Amazon DataZone. Puoi AWS Glue configurare la raccolta automatica delle informazioni sulla derivazione durante le esecuzioni dei job Spark e inviare gli eventi di derivazione da visualizzare in Amazon. DataZone Per ulteriori informazioni, consulta Data lineage in Amazon DataZone.

  Per configurarlo sulla AWS Glue console, attiva Generate lineage events e inserisci il tuo ID di DataZone dominio Amazon nella scheda Job details.




  In alternativa, puoi fornire il seguente parametro di lavoro (inserisci il tuo ID di DataZone dominio):

  - Chiave: `--conf`
  - Valore:

```
extraListeners=io.openlineage.spark.agent.OpenLineageSparkListener
—conf spark.openlineage.transport.type=amazon_datazone_api
-conf spark.openlineage.transport.domainId=<your-domain-ID>
```

- Aggiornamenti del connettore e dei driver JDBC. Per ulteriori informazioni, consultare Appendice B: aggiornamenti dei driver JDBC e Appendice C: Aggiornamenti dei connettori.
- Aggiornamento Java da 8 a 17.
- Maggiore spazio di archiviazione AWS Glue `G.1X` e `G.2X` lavoro per i lavoratori con spazio su disco che è aumentato rispettivamente a 94 GB e 138 GB. Inoltre, `R.8X` sono disponibili nuovi tipi di `G.12X` worker e versioni ottimizzate per `R.1X` la `R.2X` memoria nella versione AWS Glue 4.0 e successive. `G.16X` `R.4X` Per ulteriori informazioni, consulta Jobs

- Support for AWS SDK for Java, versione 2 AWS Glue - 5.0, i job possono utilizzare le versioni Java 1.12.569 o 2.28.8 se il job supporta la versione 2. L' AWS SDK for Java 2.x è un'importante riscrittura del codice base della versione 1.x. È stata sviluppata su base Java 8+ e aggiunge diverse caratteristiche richieste frequentemente. Queste includono il supporto per la funzionalità non bloccante I/O e la possibilità di inserire un'implementazione HTTP diversa in fase di esecuzione. Per ulteriori informazioni, inclusa una Guida alla migrazione da SDK per Java v1 a v2, consultare la guida AWS SDK per Java, versione 2.

Modifiche importanti

Notare le seguenti modifiche speciali:

- Nella AWS Glue versione 5.0, quando si utilizza il file system S3A e se sia `fs.s3a.endpoint` che `fs.s3a.endpoint.region` non sono impostati, la regione predefinita utilizzata da S3A è `us-east-2`. Ciò può causare problemi, come errori di timeout di caricamento di S3, in particolare per i processi VPC. Per mitigare i problemi causati da AWS Glue questa modifica, imposta la configurazione Spark `fs.s3a.endpoint.region` quando usi il file system S3A nella versione 5.0.
- Lake Formation Fine-grained Access Control (FGAC)
  - AWS Glue 5.0 supporta solo il nuovo Spark-native FGAC che utilizza Spark. DataFrames Non supporta l'utilizzo di FGAC. AWS Glue DynamicFrames
    - L'uso di FGAC nella versione 5.0 richiede la migrazione da Spark AWS Glue DynamicFrames DataFrames
    - Se non hai bisogno di FGAC, non è necessario migrare a Spark DataFrame e le GlueContext funzionalità, come i segnalibri di lavoro e i predicati push down, continueranno a funzionare.
  - I lavori con Spark-native FGAC richiedono un minimo di 4 dipendenti: un driver utente, un driver di sistema, un esecutore di sistema e un esecutore utente in standby.
  - Per ulteriori informazioni, vedere Using with per il controllo granulare degli AWS Glue accessi AWS Lake Formation.
- Accesso completo alla tabella (FTA) con Lake Formation
  - AWS Glue 5.0 supporta FTA con Spark-native DataFrames (nuove) e GlueContext DynamicFrames (versioni precedenti, con limitazioni)
  - Spark-native FTA
    - Se viene utilizzato lo script 4.0 GlueContext, esegui la migrazione all'utilizzo di Spark nativo.
    - Questa funzionalità è limitata alle tabelle hive e iceberg

    - Per ulteriori informazioni sulla configurazione di un job 5.0 per utilizzare l'FTA nativo di Spark, consulta. Native Spark FTA nella versione 5.0 AWS Glue
  - GlueContext DynamicFrame FTA
    - Non è necessaria alcuna modifica del codice
    - Questa funzionalità è limitata alle tabelle non OTF: non funzionerà con Iceberg, Delta Lake e Hudi.
- Il lettore SIMD CSV vettorializzato non è supportato.
- La registrazione continua nel gruppo di log di output non è supportata. Utilizzare invece il gruppo di log `error`.
- Il AWS Glue job run insights `job-insights-rule-driver` è stato obsoleto. Il flusso di log `job-insights-rca-driver` si trova ora nel gruppo di log di errore.
- Athena-based custom/marketplace i connettori non sono supportati.
- I connettori Adobe Marketo Engage, Facebook Ads, Google Ads, Google Analytics 4, Google Sheets, Hubspot, Instagram Ads, Intercom, Jira Cloud, Oracle, Salesforce, Salesforce Marketing Cloud NetSuite, Salesforce Marketing Cloud Account Engagement, SAP OData, Slack, Snapchat Ads, Stripe, Zendesk e Zoho CRM non sono supportati. ServiceNow
- Le AWS Glue proprietà log4j personalizzate non sono supportate nella versione 5.0.

Miglioramenti principali da Spark 3.3.0 a Spark 3.5.4

Nota i seguenti miglioramenti:

- Client Python per Spark SPARK-39375Connect ().
- Implementa il supporto per i valori DEFAULT per le colonne nelle tabelle () SPARK-38334.
- Supporta «Riferimenti agli alias delle colonne laterali» () SPARK-27561.
- Rafforza l'utilizzo di SQLSTATE per le classi di errore (). SPARK-41994
- Abilita il filtro Bloom Joins per impostazione predefinita (). SPARK-38841
- Migliore scalabilità dell'interfaccia utente Spark e stabilità dei driver per applicazioni di grandi dimensioni (). SPARK-41053
- Monitoraggio asincrono dei progressi in streaming strutturato (). SPARK-39591
- Elaborazione statica arbitraria in Python in streaming strutturato (). SPARK-40434
- Miglioramenti della copertura dell'API Pandas (SPARK-42882) e NumPy supporto di input in (). PySpark SPARK-39405

- Fornisci un profiler di memoria per le funzioni PySpark definite dall'utente (). SPARK-40281
- Implementa PyTorch distributor (). SPARK-41589
- Pubblica artefatti SBOM (). SPARK-41893
- IPv6-only Ambiente di supporto (SPARK-39457).
- Scheduler K8s personalizzato (Apache YuniKorn e Volcano) GA (). SPARK-42802
- Supporto client Scala e Go in Spark Connect (SPARK-42554) e (SPARK-43351).
- PyTorch-based supporto ML distribuito per Spark Connect (SPARK-42471).
- Supporto di streaming strutturato per Spark Connect in Python e Scala SPARK-42938().
- Supporto dell'API Pandas per Python Spark Connect Client (). SPARK-42497
- Introduci le UDF di Arrow Python (). SPARK-40307
- Supporta le funzioni di tabella definite dall'utente in Python (). SPARK-43798
- Migra PySpark gli errori nelle classi di errore (). SPARK-42986
- PySpark framework di test (SPARK-44042).
- Aggiungi il supporto per Datasketches HllSketch () SPARK-16484.
- Built-in Miglioramento della funzione SQL () SPARK-41231.
- CLAUSOLA IDENTIFIER () SPARK-43205.
- Aggiungi funzioni SQL in Scala, Python e R API () SPARK-43907.
- Aggiungi il supporto per argomenti denominati per le funzioni SQL (SPARK-43922).
- Evita la riesecuzione di attività non necessarie su un esecutore dismesso perso se i dati shuffle vengono migrati (). SPARK-41469
- ML distribuito <> spark connect (). SPARK-42471
- DeepSpeed distributore (). SPARK-44264
- Implementa il checkpoint del changelog per RockSDB state store (). SPARK-43421
- Introduci la propagazione della filigrana tra gli operatori (). SPARK-42376
- Introduci drop DuplicatesWithinWatermark () SPARK-42931.
- Miglioramenti alla gestione della memoria del provider di archivi di stato RockSDB (). SPARK-43311

## Azioni per migrare alla versione 5.0 AWS Glue

Per i processi esistenti, modifica la `Glue version` dalla versione precedente a `Glue 5.0` nella configurazione del processo.

- In AWS Glue Studio, scegli `Glue 5.0 - Supports Spark 3.5.4, Scala 2, Python 3` in`Glue version`.
- Nell'API, scegli **`5.0`** nel parametro `GlueVersion` nell'operazione API `UpdateJob`.

Per i nuovi processi, scegli `Glue 5.0` al momento della creazione.

- Nella console, scegli `Spark 3.5.4, Python 3 (Glue Version 5.0) or Spark 3.5.4, Scala 2 (Glue Version 5.0)` in `Glue version`.
- In AWS Glue Studio, scegli `Glue 5.0 - Supports Spark 3.5.4, Scala 2, Python 3` in`Glue version`.
- Nell'API, scegli **`5.0`** nel parametro `GlueVersion` nell'operazione API `CreateJob`.

Per visualizzare i log degli eventi di Spark della AWS Glue versione 5.0 della versione AWS Glue 2.0 o precedente, avvia un server di cronologia Spark aggiornato per la AWS Glue versione 5.0 utilizzando o Docker. CloudFormation

## Elenco di controllo della migrazione

Rivedi questo elenco di controllo per la migrazione:

- Aggiornamenti di Java 17
- [Scala] Aggiorna le chiamate AWS SDK dalla v1 alla v2
- Migrazione da Python 3.10 a 3.11
- [Python] Aggiornare i riferimenti di avvio da 1.26 a 1.34

## AWS Glue Funzionalità 5.0

Questa sezione descrive AWS Glue le funzionalità in modo più dettagliato.

Interrogazione dei cataloghi di dati dei metastore da ETL AWS Glue

Puoi registrare il tuo AWS Glue lavoro per accedere a AWS Glue Data Catalog, il che rende disponibili tabelle e altre risorse di metastore per diversi consumatori. Il Data Catalog supporta una gerarchia multicatalogo, che unifica tutti i dati nei data lake Amazon S3. Fornisce inoltre sia un'API del metastore Hive che un'API Apache Iceberg open source per l'accesso ai dati. Queste funzionalità sono disponibili per altri servizi orientati ai dati come Amazon EMR, Amazon Athena AWS Glue e Amazon Redshift.

Quando crei risorse nel Data Catalog, puoi accedervi da qualsiasi motore SQL che supporti l'API REST di Apache Iceberg. AWS Lake Formation gestisce le autorizzazioni. Dopo la configurazione, è possibile sfruttare le funzionalità AWS Glue di interrogazione di dati diversi interrogando queste risorse di metastore con applicazioni familiari. Queste includono Apache Spark e Trino.

Come sono organizzate le risorse dei metadati

I dati sono organizzati in una gerarchia logica di cataloghi, database e tabelle, utilizzando: AWS Glue Data Catalog

- Catalogo: un contenitore logico che contiene oggetti provenienti da un archivio dati, come schemi o tabelle.
- Database: organizza oggetti di dati come tabelle e viste in un catalogo.
- Tabelle e viste: oggetti di dati in un database che forniscono un livello di astrazione con uno schema comprensibile. Semplificano l'accesso ai dati sottostanti, che possono essere in vari formati e in varie posizioni.

## Migrazione da 4.0 a 5.0 AWS Glue AWS Glue

Tutti i parametri di lavoro e le funzionalità principali esistenti in AWS Glue 4.0 esisteranno nella AWS Glue versione 5.0, ad eccezione delle trasformazioni di machine learning.

Sono stati aggiunti i nuovi parametri seguenti:

- `--enable-lakeformation-fine-grained-access`: abilita la funzionalità di controllo degli accessi a grana fine (FGAC) nelle tabelle di Lake Formation. AWS

Consulta la documentazione relativa alla migrazione di Spark:

- Guida alla migrazione: Spark Core
- Guida alla migrazione: SQL, set di dati e DataFrame
- Guida alla migrazione: streaming strutturato
- Aggiornamento PySpark

## Migrazione da AWS Glue 3.0 a 5.0 AWS Glue

> **Note**
>
> Per le fasi di migrazione relative alla AWS Glue versione 4.0, consultaMigrazione dalla 3.0 alla 4.0 AWS Glue AWS Glue.

Tutti i parametri di lavoro e le funzionalità principali esistenti nella AWS Glue versione 3.0 esisteranno nella AWS Glue versione 5.0, ad eccezione delle trasformazioni di apprendimento automatico.

## Migrazione da AWS Glue 2.0 a 5.0 AWS Glue

> **Note**
>
> Per i passaggi di migrazione relativi alla AWS Glue 4.0 e un elenco delle differenze di migrazione tra la AWS Glue versione 3.0 e 4.0, consultaMigrazione dalla 3.0 alla 4.0 AWS Glue AWS Glue.

Tieni inoltre presente le seguenti differenze di migrazione tra AWS Glue le versioni 3.0 e 2.0:

- Tutti i parametri di lavoro e le funzionalità principali esistenti nella AWS Glue versione 2.0 esisteranno nella AWS Glue versione 5.0, ad eccezione delle trasformazioni di apprendimento automatico.
- Diverse modifiche di Spark da sole potrebbero richiedere la revisione degli script per garantire che non si faccia riferimento alle caratteristiche rimosse. Ad esempio, Spark 3.1.1 e versioni successive non abilitano le Scala-untyped UDF, ma Spark 2.4 le consente.
- Python 2.7 non è supportato.
- Eventuali jar aggiuntivi forniti nei job AWS Glue 2.0 esistenti possono creare dipendenze in conflitto poiché sono stati effettuati aggiornamenti in diverse dipendenze. È possibile evitare conflitti di dipendenze con il parametro `--user-jars-first` del processo.
- Modifiche al comportamento dei file in parquet con timestamp. loading/saving from/to Per ulteriori dettagli, consultare Aggiornamento da Spark SQL 3.0 a 3.1.
- Diverso parallelismo delle attività Spark per la configurazione. driver/executor È possibile regolare il parallelismo delle attività passando l'argomento del processo `--executor-cores`.

## Modifiche al comportamento di registrazione nella versione 5.0 AWS Glue

Di seguito sono riportate le modifiche al comportamento di registrazione nella AWS Glue versione 5.0. Per ulteriori informazioni, vedere Registrazione dei lavori. AWS Glue

- Tutti i log (log di sistema, log dei daemon Spark, log degli utenti e log di Glue Logger) vengono ora scritti nel gruppo di log `/aws-glue/jobs/error` per impostazione predefinita.
- Il gruppo di log `/aws-glue/jobs/logs-v2` utilizzato per la registrazione continua nelle versioni precedenti non viene più utilizzato.
- Non è più possibile rinominare o personalizzare i nomi dei gruppi di log o dei flussi di log utilizzando gli argomenti di registrazione continua rimossi. Consultate invece i nuovi argomenti relativi al lavoro nella AWS Glue versione 5.0.

Nella AWS Glue versione 5.0 vengono introdotti due nuovi argomenti relativi al lavoro

- `--custom-logGroup-prefix`: consente di specificare un prefisso personalizzato per i gruppi di log `/aws-glue/jobs/error` e `/aws-glue/jobs/output`.
- `--custom-logStream-prefix`: consente di specificare un prefisso personalizzato per i nomi dei flussi di log all'interno dei gruppi di log.

  Le regole e le limitazioni di convalida per i prefissi personalizzati includono:
  - L'intero nome del flusso di log deve contenere da 1 a 512 caratteri.
  - Il prefisso personalizzato per i nomi dei flussi di log è limitato a 400 caratteri.
  - I caratteri consentiti nei prefissi includono caratteri alfanumerici, caratteri di sottolineatura (`_`), trattini (`-`) e barre (`/`).

Argomenti di registrazione continua obsoleti nella versione 5.0 AWS Glue

I seguenti argomenti di lavoro per la registrazione continua sono obsoleti nella versione 5.0 AWS Glue

- `--enable-continuous-cloudwatch-log`
- `--continuous-log-logGroup`
- `--continuous-log-logStreamPrefix`
- `--continuous-log-conversionPattern`
- `--enable-continuous-log-filter`

## Migrazione di connettori e driver JDBC per 5.0 AWS Glue

Per le versioni dei connettori JDBC e data lake che sono state aggiornate, consulta:

- Appendice B: aggiornamenti dei driver JDBC
- Appendice C: Aggiornamenti dei connettori
- Appendice D: Aggiornamenti di formato a tabella aperta

Le seguenti modifiche si applicano alle versioni dei connettori o dei driver identificate nelle appendici di Glue 5.0.

Amazon Redshift

Nota le seguenti modifiche:

- Aggiunge il supporto per i nomi di tabella in tre parti per consentire al connettore di interrogare le tabelle di condivisione dei dati di Redshift.
- Corregge la mappatura di Spark `ShortType` per utilizzare Redshift `SMALLINT` anziché `INTEGER` per adattarla meglio alla dimensione prevista dei dati.
- È stato aggiunto il supporto per Custom Cluster Names (CNAME) per Amazon Redshift serverless.

Apache Hudi

Nota le seguenti modifiche:

- Supporta l'indice di livello record.
- Supporta la generazione automatica di chiavi di registrazione. Ora non è necessario specificare il campo della chiave di registrazione.

Apache Iceberg

Nota le seguenti modifiche:

- Supporta il controllo granulare degli accessi con. AWS Lake Formation
- Supporta la ramificazione e il tagging, che sono riferimenti denominati a snapshot con cicli di vita indipendenti.
- È stata aggiunta una procedura di visualizzazione del registro delle modifiche che genera una vista che contiene le modifiche apportate a una tabella in un periodo specificato o tra snapshot specifici.

Delta Lake

Nota le seguenti modifiche:

- Supporta Delta Universal Format (UniForm) che consente un accesso senza interruzioni tramite Apache Iceberg e Apache Hudi.
- Supporta i vettori di cancellazione che implementano un Merge-on-Read paradigma.

AzureCosmos

Nota le seguenti modifiche:

- È stato aggiunto il supporto per chiavi di partizione gerarchiche.
- È stata aggiunta l'opzione per utilizzare lo schema personalizzato con StringType (raw json) per una proprietà annidata.
- È stata aggiunta l'opzione di configurazione `spark.cosmos.auth.aad.clientCertPemBase64` per consentire l'utilizzo dell'autenticazione SPN (ServicePrincipal nome) con certificato anziché il segreto del client.

Per ulteriori informazioni, consulta il registro delle modifiche del connettore Azure Cosmos DB Spark.

Microsoft SQL Server

Nota le seguenti modifiche:

- La crittografia TLS è abilita per impostazione predefinita.
- Quando encrypt = false ma il server richiede la crittografia, il certificato viene convalidato in base all'impostazione della connessione `trustServerCertificate`.
- `aadSecurePrincipalId` e `aadSecurePrincipalSecret` resi obsoleti.
- `getAADSecretPrincipalId` API rimossa.
- È stata aggiunta la risoluzione CNAME quando viene specificato il realm.

MongoDB

Nota le seguenti modifiche:

- Supporto per la modalità micro-batch con Spark Structured Streaming.

- Supporto per i tipi di dati BSON.
- È stato aggiunto il supporto per la lettura di più raccolte quando si utilizzano modalità di streaming micro-batch o continuo.
  - Se il nome di una raccolta utilizzata nell'opzione di configurazione `collection` contiene una virgola, Spark Connector la considera come due raccolte diverse. Per risolvere questo problema, è necessario evitare la virgola facendola precedere da una barra rovesciata (\).
  - Se il nome di una raccolta utilizzata nell'opzione di `collection` configurazione è "*", Spark Connector la interpreta come una specifica per la scansione di tutte le raccolte. Per risolvere questo problema, è necessario evitare l'asterisco facendolo precedere da una barra rovesciata (\).
  - Se il nome di una raccolta utilizzata nell'opzione di configurazione `collection` contiene una barra rovesciata (\), Spark Connector considera la barra rovesciata come un carattere di escape, il che potrebbe cambiare il modo in cui interpreta il valore. Per risolvere questo problema, è necessario evitare la barra rovesciata facendola precedere da un'altra barra rovesciata.

Per ulteriori informazioni, consultare il connettore MongoDB per le note di rilascio di Spark.

Snowflake

Nota le seguenti modifiche:

- È stato introdotto un nuovo parametro `trim_space` che è possibile utilizzare per rifinire automaticamente i valori delle colonne `StringType` durante il salvataggio in una tabella Snowflake. Default: `false`.
- Per impostazione predefinita, il parametro `abort_detached_query` è stato disabilitato a livello di sessione.
- È stato rimosso il requisito del parametro `SFUSER` quando si utilizza OAUTH.
- È stata rimossa la funzionalità Advanced Query Pushdown. Sono disponibili alternative alla funzionalità. Ad esempio, anziché caricare i dati dalle tabelle Snowflake, gli utenti possono caricare direttamente i dati dalle query SQL di Snowflake.

Per ulteriori informazioni, consultare il connettore Snowflake per le note di rilascio di Spark.

Appendice A: Aggiornamenti importanti delle dipendenze

Di seguito sono riportati gli aggiornamenti delle dipendenze:

| Dipendenza | Versione 5.0 AWS Glue | Versione in AWS Glue 4.0 | Versione in AWS Glue 3.0 | Versione in AWS Glue 2.0 | Versione in AWS Glue 1.0 |
|---|---|---|---|---|---|
| Java | 17 | 8 | 8 | 8 | 8 |
| Spark | 3.5.4 | 3.3.0-amzn-1 | 3.1.1-amzn-0 | 2.4.3 | 2.4.3 |
| Hadoop | 3.4.1 | 3.3.3-amzn-0 | 3.2.1-amzn-3 | 2.8.5-amzn-5 | 2.8.5-amzn-1 |
| Scala | 2,12,18 | 2,12 | 2,12 | 2.11 | 2.11 |
| Jackson | 2,15,2 | 2,12 | 2,12 | 2.11 | 2.11 |
| Hive | 2.3.9-amzn-4 | 2.3.9-amzn-2 | 2.3.7-amzn-4 | 1.2 | 1.2 |
| EMRFS | 2,69,0 | 2,54,0 | 2,46,0 | 2.38.0 | 2.30.0 |
| Json4s | 3.7.0-M11 | 3.7.0-M11 | 36.6 | 3.5.x | 3.5.x |
| Arrow | 12,0,1 | 7,0,0 | 2.0.0 | 0.10.0 | 0.10.0 |
| AWS Glue Client Data Catalog | 4.5.0 | 3.7.0 | 3.0.0 | 1.10.0 | N/A |
| AWS SDK per Java | 2.29.52 | 1.12 | 1.12 | | |
| Python | 3,11 | 3,10 | 3.7 | 2.7 e 3.6 | 2.7 e 3.6 |
| Boto | 1,34,131 | 1,26 | 1,18 | 1.12 | N/A |
| Connettore EMR DynamoDB | 5.6.0 | 4,16,0 | | | |

## Appendice B: aggiornamenti dei driver JDBC

Di seguito sono riportati gli aggiornamenti dei driver JDBC:

| Driver | Versione del driver JDBC nella versione 5.0 AWS Glue | Versione del driver JDBC nella versione 4.0 AWS Glue | Versione del driver JDBC nella versione 3.0 AWS Glue | Versione del driver JDBC nelle versioni precedenti AWS Glue |
|---|---|---|---|---|
| MySQL | 8.0.33 | 8.0.23 | 8.0.23 | 5.1 |
| Microsoft SQL Server | 10,2,0 | 9,40 | 7,0,0 | 6.1.0 |
| Database Oracle | 23,3,023,09 | 21,7 | 21,1 | 11.2 |
| PostgreSQL | 42,7,3 | 42,36 | 4,2,18 | 42,1,0 |
| Amazon Redshift | redshift-jdbc42-2.1.0.29 | redshift-jdbc42-2.1.0.16 | redshift-jdbc41-1.2.12.1017 | redshift-jdbc41-1.2.12.1017 |
| SAP Hana | 2,20,17 | 2,17,12 | | |
| Teradata | 20,00,00,33 | 20,00,00,06 | | |

Appendice C: Aggiornamenti dei connettori

Di seguito sono riportati gli aggiornamenti dei connettori:

| Driver | Versione del connettore in 5.0 AWS Glue | Versione del connettore in AWS Glue 4.0 | Versione del connettore in AWS Glue 3.0 |
|---|---|---|---|
| Connettore EMR DynamoDB | 5.6.0 | 4.16.0 | |
| Amazon Redshift | 64,0 | 6.1.3 | |
| OpenSearch | 1.2.0 | 1.0.1 | |
| MongoDB | 10.3,0 | 10.0.4 | 3.0.0 |

| Driver | Versione del connettore in 5.0 AWS Glue | Versione del connettore in AWS Glue 4.0 | Versione del connettore in AWS Glue 3.0 |
|---|---|---|---|
| Snowflake | 3.0.0 | 2.12.0 | |
| Google BigQuery | 0.32.2 | 0,32,2 | |
| AzureCosmos | 4,33,0 | 42,0 | |
| AzureSQL | 1.3.0 | 1.3.0 | |
| Vertica | 33,5 | 3.3.5 | |

Appendice D: Aggiornamenti di formato a tabella aperta

Di seguito sono riportati gli aggiornamenti di formato a tabella aperta:

| OTF | Versione del connettore in 5.0 AWS Glue | Versione del connettore in AWS Glue 4.0 | Versione del connettore in AWS Glue 3.0 |
|---|---|---|---|
| Hudi | 0.15.0 | 0.12.1 | 0,10,1 |
| Delta Lake | 3.3.0 | 2.1.0 | 1.0.0 |
| Iceberg | 1.7.1 | 1.0.0 | 0.13.1 |

## Migrazione AWS Glue per i job Spark alla versione 4.0 AWS Glue

Questo argomento descrive le modifiche tra AWS Glue le versioni 0.9, 1.0, 2.0 e 3.0 per consentire la migrazione delle applicazioni Spark e dei lavori ETL alla 4.0. AWS Glue Descrive inoltre le funzionalità della AWS Glue versione 4.0 e i vantaggi del suo utilizzo.

Per utilizzare questa funzionalità con i tuoi lavori AWS Glue ETL, scegli `Glue version` quando **`4.0`** crei i tuoi lavori.

Argomenti

- Nuove caratteristiche supportate
- Azioni AWS Glue per migrare alla versione 4.0
- Elenco di controllo della migrazione
- Migrazione dalla 3.0 alla 4.0 AWS Glue AWS Glue
- Migrazione da 2.0 a 4.0 AWS Glue AWS Glue
- Migrazione da AWS Glue 1.0 a 4.0 AWS Glue
- Migrazione da 0.9 a 4.0 AWS Glue AWS Glue
- Migrazione di connettori e driver JDBC per 4.0 AWS Glue
- Appendice A: Aggiornamenti importanti delle dipendenze
- Appendice B: aggiornamenti dei driver JDBC
- Appendice C: Aggiornamenti dei connettori


## Nuove caratteristiche supportate

Questa sezione descrive le nuove funzionalità e i vantaggi della AWS Glue versione 4.0.

- Si basa su Apache Spark 3.3.0, ma presenta ottimizzazioni in AWS Glue e Amazon EMR, come esecuzioni adattive delle query, lettori vettorizzati e shuffle e coalescenza delle partizioni ottimizzati.
- Driver JDBC aggiornati per tutte le fonti AWS Glue native tra cui MySQL, Microsoft SQL Server, Oracle, PostgreSQL, MongoDB e le librerie e dipendenze Spark aggiornate introdotte da Spark 3.3.0.
- Aggiornato con un nuovo connettore Amazon Redshift e driver JDBC.
- Accesso Amazon S3 ottimizzato con EMRFS aggiornato e committer di output ottimizzati per Amazon S3 abilitati per impostazione predefinita.
- Accesso ottimizzato al catalogo dati con indici delle partizioni, predicati pushdown, elenco delle partizioni e un client metastore Hive aggiornato.
- Integrazione con Lake Formation per tabelle di catalogo governate con filtraggio a livello di cella e transazioni data lake.
- Ridotta la latenza di avvio per migliorare i tempi complessivi di completamento dei processi e dell'interattività.
- I processi Spark vengono fatturati in incrementi di 1 secondo con una durata minima di fatturazione 10 volte inferiore, da un minimo di 10 minuti a un minimo di 1 minuto.

- Supporto nativo per framework open data lake con Apache Hudi, Delta Lake e Apache Iceberg.
- Supporto nativo per il Cloud Shuffle Storage Plugin basato su Amazon S3 (un plug-in Apache Spark) per utilizzare Amazon S3 per lo shuffling e la capacità di archiviazione elastica.

Miglioramenti principali da Spark 3.1.1 a Spark 3.3.0

Nota i seguenti miglioramenti:

- Filtraggio di runtime a livello di riga (SPARK-32268).
- Miglioramenti ANSI (SPARK-38860).
- Miglioramenti ai messaggi di errore (SPARK-38781).
- Supporto dei tipi complessi per il lettore vettorializzato Parquet (SPARK-34863).
- Supporto di metadati di file nascosti per Spark SQL (SPARK-37273).
- Fornisci un profiler per Python/Pandas UDFs (SPARK-37443).
- Presenta Trigger. AvailableNow per eseguire query di streaming come Trigger.Once in più batch (SPARK-36533).
- Funzionalità di pushdown Datasource V2 più complete (SPARK-38788).
- Migrazione da log4j 1 a log4j 2 (SPARK-37814).

Altre modifiche importanti

Nota le seguenti modifiche:

- Modifiche importanti
  - Elimina i riferimenti al supporto per Python 3.6 nei documenti e Python/docs (SPARK-36977).
  - Rimuove l'hack di tuple specificate sostituendo il pickle integrato con cloudpickle (SPARK-32079).
  - Porta la versione minima di Pandas a 1.0.5 (SPARK-37465).

## Azioni AWS Glue per migrare alla versione 4.0

Per i processi esistenti, modifica la `Glue version` dalla versione precedente a `Glue 4.0` nella configurazione del processo.

- In AWS Glue Studio, scegli `Glue 4.0 - Supports Spark 3.3, Scala 2, Python 3` in`Glue version`.
- Nell'API, scegli **`4.0`** nel parametro `GlueVersion` nell'operazione API `UpdateJob`.

Per i nuovi processi, scegli `Glue 4.0` al momento della creazione.

- Nella console, scegli `Spark 3.3, Python 3 (Glue Version 4.0) or Spark 3.3, Scala 2 (Glue Version 3.0)` in `Glue version`.
- In AWS Glue Studio, scegli `Glue 4.0 - Supports Spark 3.3, Scala 2, Python 3` in`Glue version`.
- Nell'API, scegli **`4.0`** nel parametro `GlueVersion` nell'operazione API `CreateJob`.

Per visualizzare i registri degli eventi di Spark della AWS Glue versione 4.0 della versione AWS Glue 2.0 o precedente, avvia un server di cronologia Spark aggiornato per AWS Glue 4.0 utilizzando o Docker. CloudFormation

## Elenco di controllo della migrazione

- Le librerie Python esterne del processo dipendono da Python 2.7/3.6?
  - Aggiorna le librerie dipendenti da Python 2.7/3.6 a Python 3.10, poiché Spark 3.3.0 ha rimosso il supporto per Python 2.7 e 3.6.

## Migrazione dalla 3.0 alla 4.0 AWS Glue AWS Glue

Nota le seguenti modifiche durante la migrazione:

- Tutti i parametri di lavoro e le funzionalità principali esistenti nella AWS Glue versione 3.0 esisteranno nella AWS Glue versione 4.0.
- AWS Glue 3.0 utilizza Spark 3.1.1 ottimizzato per Amazon EMR e 4.0 AWS Glue utilizza Spark 3.3.0 ottimizzato per Amazon EMR.

  Diverse modifiche di Spark da sole potrebbero richiedere la revisione degli script per garantire che non si faccia riferimento alle funzionalità rimosse.
- AWS Glue 4.0 include anche un aggiornamento a EMRFS e Hadoop. Per la versione specifica, consulta Appendice A: Aggiornamenti importanti delle dipendenze.
- L' AWS SDK fornito nei job ETL è ora aggiornato da 1.11 a 1.12.

- Tutti i processi Python utilizzeranno la versione 3.10 di Python. In precedenza, Python 3.7 veniva utilizzato nella 3.0. AWS Glue

  Di conseguenza, alcuni pymodules forniti out-of-the-box da AWS Glue vengono aggiornati.
- Log4j è stato aggiornato a Log4j2.
  - Per informazioni sul percorso di migrazione di Log4j2, consulta la documentazione di Log4j.
  - È invece necessario rinominare qualsiasi file log4j.properties personalizzato come file log4j2.properties, con le proprietà log4j2 appropriate.
- Per la migrazione di connettori specifici, consulta Migrazione di connettori e driver JDBC per 4.0 AWS Glue.
- L' AWS Encryption SDK viene aggiornato da 1.x a 2.x. AWS Glue sono interessati i lavori che utilizzano configurazioni AWS Glue di sicurezza e i lavori dipendenti dalla dipendenza di AWS Encryption SDK fornita in fase di esecuzione. Consulta le istruzioni per AWS Glue la migrazione dei job.

  È possibile aggiornare in sicurezza un lavoro AWS Glue 2.0/3.0 a un lavoro AWS Glue 4.0 perché AWS Glue 2.0/3.0 contiene già la versione bridge AWS Encryption SDK.

Consulta la documentazione relativa alla migrazione di Spark:

- Aggiornamento da Spark SQL 3.1 a 3.2
- Aggiornamento da Spark SQL 3.2 a 3.3

## Migrazione da 2.0 a 4.0 AWS Glue AWS Glue

Nota le seguenti modifiche durante la migrazione:

> **Note**
>
> Per i passaggi di migrazione relativi alla AWS Glue versione 3.0, consultaMigrazione dalla 3.0 alla 4.0 AWS Glue AWS Glue.

- Tutti i parametri di lavoro e le funzionalità principali esistenti nella AWS Glue versione 2.0 esisteranno nella AWS Glue versione 4.0.

- Il committer ottimizzato per EMRFS S3 per la scrittura di dati Parquet in Amazon S3 è abilitato per impostazione predefinita dalla versione 3.0. AWS Glue Tuttavia, puoi disabilitarlo impostando `--enable-s3-parquet-optimized-committer` a `false`.
- AWS Glue 2.0 utilizza Spark 2.4 open source e AWS Glue 4.0 utilizza Spark 3.3.0 ottimizzato per Amazon EMR.
  - Diverse modifiche di Spark da sole potrebbero richiedere la revisione degli script per garantire che non si faccia riferimento alle funzionalità rimosse.
  - Ad esempio, Spark 3.3.0 non abilita UDFs Scala-untyped, ma Spark 2.4 li consente.
- L' AWS SDK fornito nei job ETL è ora aggiornato da 1.11 a 1.12.
- AWS Glue 4.0 include anche un aggiornamento a EMRFS, driver JDBC aggiornati e inclusioni di ottimizzazioni aggiuntive su Spark stesso fornite da. AWS Glue
- Scala è stato aggiornato da 2.11 a 2.12 e Scala 2.12 non è compatibile con Scala 2.11.
- Python 3.10 è la versione predefinita utilizzata per gli script Python mentre AWS Glue 2.0 utilizzava solo Python 3.7 e 2.7.
  - Python 2.7 non è supportato con Spark 3.3.0. Qualsiasi lavoro che richieda Python 2 nella configurazione del lavoro avrà esito negativo con un. IllegalArgumentException
  - Un nuovo meccanismo di installazione di moduli Python aggiuntivi è disponibile a partire dalla AWS Glue versione 2.0.
- Diversi aggiornamenti delle dipendenze, evidenziati in Appendice A: Aggiornamenti importanti delle dipendenze.
- Qualsiasi file JAR aggiuntivo fornito nei job AWS Glue 2.0 esistenti potrebbe creare dipendenze in conflitto perché sono stati effettuati aggiornamenti in diverse dipendenze dalla versione 4.0 alla 2.0. È possibile evitare conflitti tra percorsi di classe nella AWS Glue versione 4.0 con il parametro job. `--user-jars-first` AWS Glue
- AWS Glue 4.0 utilizza Spark 3.3. A partire da Spark 3.1, c'è stato un cambiamento nel comportamento dei file loading/saving di parquet timestamps from/to . Per ulteriori dettagli, consulta Aggiornamento da Spark SQL 3.0 a 3.1.

  Si consiglia di impostare i seguenti parametri quando si tratta di dati che contengono reading/writing colonne con timestamp. L'impostazione di questi parametri può risolvere il problema di incompatibilità del calendario che si verifica durante l'aggiornamento da Spark 2 a Spark 3, sia per AWS Glue Dynamic Frame che per Spark Data Frame. Utilizzare l'opzione CORRECTED per leggere il valore datetime così com'è e l'opzione LEGACY per cambiare la base dei valori datetime in relazione alla differenza di calendario durante la lettura.

```
- Key: --conf
- Value: spark.sql.legacy.parquet.int96RebaseModeInRead=[CORRECTED|LEGACY] --
conf spark.sql.legacy.parquet.int96RebaseModeInWrite=[CORRECTED|LEGACY] --conf
 spark.sql.legacy.parquet.datetimeRebaseModeInRead=[CORRECTED|LEGACY]
```

- Per la migrazione di connettori specifici, consulta Migrazione di connettori e driver JDBC per 4.0 AWS Glue.
- L' AWS Encryption SDK viene aggiornato da 1.x a 2.x. AWS Glue sono interessati i lavori che utilizzano configurazioni AWS Glue di sicurezza e i lavori dipendenti dalla dipendenza di AWS Encryption SDK fornita in fase di esecuzione. Consulta queste istruzioni per AWS Glue la migrazione dei job:
  - È possibile aggiornare in sicurezza un lavoro AWS Glue 2.0 a un lavoro AWS Glue 4.0 perché la versione AWS Glue 2.0 contiene già la versione bridge AWS Encryption SDK.

Consulta la documentazione relativa alla migrazione di Spark:

- Aggiornamento da Spark SQL 2.4 a 3.0
- Aggiornamento da Spark SQL 3.1 a 3.2
- Aggiornamento da Spark SQL 3.2 a 3.3
- Cambiamenti nel comportamento di Datetime previsti da Spark 3.0.

## Migrazione da AWS Glue 1.0 a 4.0 AWS Glue

Nota le seguenti modifiche durante la migrazione:

- AWS Glue 1.0 utilizza Spark 2.4 e AWS Glue 4.0 open source utilizza Spark 3.3.0 ottimizzato per Amazon EMR.
  - Diverse modifiche di Spark da sole potrebbero richiedere la revisione degli script per garantire che non si faccia riferimento alle funzionalità rimosse.
  - Ad esempio, Spark 3.3.0 non abilita UDFs Scala-untyped, ma Spark 2.4 li consente.
- Tutti i job della AWS Glue versione 4.0 verranno eseguiti con tempi di avvio notevolmente migliorati. I processi Spark verranno fatturati in incrementi di 1 secondo con una durata minima di fatturazione 10 volte inferiore poiché la latenza di avvio passerà da un massimo di 10 minuti a un massimo di 1 minuto.

- Il comportamento di registrazione è cambiato in modo significativo nella AWS Glue versione 4.0, Spark 3.3.0 ha un requisito minimo di Log4j2.
- Diversi aggiornamenti delle dipendenze, descritti nell'appendice.
- Scala è stato inoltre aggiornato da 2.11 a 2.12 e Scala 2.12 non è compatibile con Scala 2.11.
- Python 3.10 è anche la versione di predefinita utilizzata per gli script Python, mentre AWS Glue 0.9 utilizzava solo Python 2.

  Python 2.7 non è supportato con Spark 3.3.0. Qualsiasi lavoro che richieda Python 2 nella configurazione del lavoro avrà esito negativo con un. IllegalArgumentException
- Un nuovo meccanismo di installazione di moduli Python aggiuntivi tramite pip è disponibile dalla versione 2.0. AWS Glue Per ulteriori informazioni, consulta Installazione di moduli Python aggiuntivi con pip in AWS Glue 2.0+.
- AWS Glue 4.0 non funziona su Apache YARN, quindi le impostazioni YARN non si applicano.
- AWS Glue 4.0 non dispone di un Hadoop Distributed File System (HDFS).
- Qualsiasi file JAR aggiuntivo fornito nei job AWS Glue 1.0 esistenti potrebbe creare dipendenze in conflitto perché nella versione 4.0 sono stati effettuati aggiornamenti in diverse dipendenze dalla versione 1.0. Per evitare questo problema, abilitiamo la AWS Glue versione 4.0 con il parametro `--user-jars-first` AWS Glue job di default.
- AWS Glue 4.0 supporta il ridimensionamento automatico. Pertanto, la ExecutorAllocationManager metrica sarà disponibile quando la scalabilità automatica è abilitata.
- Nei lavori della AWS Glue versione 4.0, si specifica il numero di lavoratori e il tipo di lavoratore, ma non si specifica a. `maxCapacity`
- AWS Glue La versione 4.0 non supporta ancora le trasformazioni dell'apprendimento automatico.
- Per la migrazione di connettori specifici, consulta Migrazione di connettori e driver JDBC per 4.0 AWS Glue.
- L' AWS Encryption SDK viene aggiornato da 1.x a 2.x. AWS Glue sono interessati i lavori che utilizzano configurazioni AWS Glue di sicurezza e i lavori dipendenti dalla dipendenza di AWS Encryption SDK fornita in fase di esecuzione. Consulta queste istruzioni per AWS Glue la migrazione dei job.
  - Non è possibile migrare direttamente un lavoro da AWS Glue 0,9/1,0 a un lavoro AWS Glue 4.0. Questo perché quando si esegue l'aggiornamento diretto alla versione 2.x o successiva e si abilitano immediatamente tutte le nuove funzionalità, AWS Encryption SDK non sarà in grado di decrittografare il testo cifrato crittografato con le versioni precedenti di Encryption SDK. AWS

- Per un aggiornamento sicuro, consigliamo innanzitutto di migrare a un AWS Glue job 2.0/3.0 che contenga la versione bridge di Encryption SDK. AWS Esegui il processo una volta per utilizzare la versione bridge Encryption SDK AWS .
- Al termine, è possibile migrare in sicurezza il job AWS Glue 2.0/3.0 a 4.0. AWS Glue

Consulta la documentazione relativa alla migrazione di Spark:

- Aggiornamento da Spark SQL 2.4 a 3.0
- Aggiornamento da Spark SQL 3.0 a 3.1
- Aggiornamento da Spark SQL 3.1 a 3.2
- Aggiornamento da Spark SQL 3.2 a 3.3
- Cambiamenti nel comportamento di Datetime previsti da Spark 3.0.

## Migrazione da 0.9 a 4.0 AWS Glue AWS Glue

Nota le seguenti modifiche durante la migrazione:

- AWS Glue 0.9 utilizza Spark 2.2.1 e 4.0 open source AWS Glue utilizza Spark 3.3.0 ottimizzato per Amazon EMR.
  - Diverse modifiche di Spark da sole potrebbero richiedere la revisione degli script per garantire che non si faccia riferimento alle funzionalità rimosse.
  - Ad esempio, Spark 3.3.0 non abilita Scala-untyped, ma Spark 2.2 li consente. UDFs
- Tutti i job della AWS Glue versione 4.0 verranno eseguiti con tempi di avvio notevolmente migliorati. I processi Spark verranno fatturati in incrementi di 1 secondo con una durata minima di fatturazione 10 volte inferiore poiché la latenza di avvio passerà da un massimo di 10 minuti a un massimo di 1 minuto.
- Il comportamento di registrazione è cambiato in modo significativo rispetto alla AWS Glue versione 4.0, Spark 3.3.0 ha un requisito minimo di Log4j2 come indicato qui (# -32-to-33). https://spark.apache.org/docs/latest/core-migration-guide.html upgrading-from-core
- Diversi aggiornamenti delle dipendenze, descritti nell'appendice.
- Scala è stato inoltre aggiornato da 2.11 a 2.12 e Scala 2.12 non è compatibile con Scala 2.11.
- Python 3.10 è anche la versione di predefinita utilizzata per gli script Python, mentre AWS Glue 0.9 utilizzava solo Python 2.

  - Python 2.7 non è supportato con Spark 3.3.0. Qualsiasi lavoro che richieda Python 2 nella configurazione del lavoro avrà esito negativo con un. IllegalArgumentException
  - È disponibile un nuovo meccanismo di installazione di moduli Python aggiuntivi tramite pip.
- AWS Glue 4.0 non funziona su Apache YARN, quindi le impostazioni YARN non si applicano.
- AWS Glue 4.0 non dispone di un Hadoop Distributed File System (HDFS).
- Qualsiasi file JAR aggiuntivo fornito nei job AWS Glue 0.9 esistenti potrebbe creare dipendenze in conflitto perché nella versione 3.0 sono stati effettuati aggiornamenti in diverse dipendenze dalla versione 0.9. È possibile evitare conflitti tra percorsi di classe nella AWS Glue versione 3.0 con il parametro job. `--user-jars-first` AWS Glue
- AWS Glue 4.0 supporta il ridimensionamento automatico. Pertanto, la ExecutorAllocationManager metrica sarà disponibile quando la scalabilità automatica è abilitata.
- Nei lavori della AWS Glue versione 4.0, si specifica il numero di lavoratori e il tipo di lavoratore, ma non si specifica a. `maxCapacity`
- AWS Glue La versione 4.0 non supporta ancora le trasformazioni dell'apprendimento automatico.
- Per la migrazione di connettori specifici, consulta Migrazione di connettori e driver JDBC per 4.0 AWS Glue.
- L' AWS Encryption SDK viene aggiornato da 1.x a 2.x. AWS Glue sono interessati i lavori che utilizzano configurazioni AWS Glue di sicurezza e i lavori dipendenti dalla dipendenza di AWS Encryption SDK fornita in fase di esecuzione. Consulta queste istruzioni per AWS Glue la migrazione dei job.
  - Non è possibile migrare direttamente un lavoro da AWS Glue 0,9/1,0 a un lavoro AWS Glue 4.0. Questo perché quando si esegue l'aggiornamento diretto alla versione 2.x o successiva e si abilitano immediatamente tutte le nuove funzionalità, AWS Encryption SDK non sarà in grado di decrittografare il testo cifrato crittografato con le versioni precedenti di Encryption SDK. AWS
  - Per un aggiornamento sicuro, consigliamo innanzitutto di migrare a un AWS Glue job 2.0/3.0 che contenga la versione bridge di Encryption SDK. AWS Esegui il processo una volta per utilizzare la versione bridge Encryption SDK AWS .
  - Al termine, è possibile migrare in sicurezza il job AWS Glue 2.0/3.0 a 4.0. AWS Glue

Consulta la documentazione relativa alla migrazione di Spark:

- Aggiornamento da Spark SQL 2.2 a 2.3
- Aggiornamento da Spark SQL 2.3 a 2.4

- Aggiornamento da Spark SQL 2.4 a 3.0
- Aggiornamento da Spark SQL 3.0 a 3.1
- Aggiornamento da Spark SQL 3.1 a 3.2
- Aggiornamento da Spark SQL 3.2 a 3.3
- Cambiamenti nel comportamento di Datetime previsti da Spark 3.0.

## Migrazione di connettori e driver JDBC per 4.0 AWS Glue

Per le versioni dei connettori JDBC e data lake che sono state aggiornate, consulta:

- Appendice B: aggiornamenti dei driver JDBC
- Appendice C: Aggiornamenti dei connettori

### Hudi

- Miglioramenti al supporto Spark SQL:
  - Tramite il comando `Call Procedure`, viene aggiunto il supporto per l'aggiornamento, il downgrade, il bootstrap, la pulizia e la riparazione. In Spark SQL è possibile utilizzare la sintassi `Create/Drop/Show/Refresh Index`.
  - È stato colmato un divario di prestazioni tra l'utilizzo tramite Spark DataSource e Spark SQL. Le scritture di Datasource in passato erano più veloci di SQL.
  - Tutti i generatori di chiavi integrati implementano operazioni API specifiche di Spark più performanti.
  - Ha sostituito la trasformazione UDF nelle `insert` operazioni di massa con le trasformazioni RDD per ridurre i costi di utilizzo. SerDe
  - Spark SQL con Hudi richiede la specifica di una `primaryKey` da parte di `tblproperites` o più opzioni nell'istruzione SQL. Per le operazioni di aggiornamento ed eliminazione, è necessario anche `preCombineField`.
- Qualsiasi tabella Hudi creata prima della versione 0.10.0 senza una `primaryKey` deve essere creata nuovamente con un campo `primaryKey` a partire dalla versione 0.10.0.

### PostgreSQL

- Sono state risolte diverse vulnerabilità (). CVEs

- Java 8 è supportato in modo nativo.
- Se il processo utilizza array di array, ad eccezione degli array di byte, questo scenario può essere trattato come array multidimensionali.

MongoDB

- Il connettore MongoDB corrente supporta Spark versione 3.1 o versione successiva e MongoDB versione 4.0 o successiva.
- A causa dell'aggiornamento del connettore, alcuni nomi di proprietà sono cambiati. Ad esempio, il nome della proprietà URI è stato modificato in `connection.uri`. Per ulteriori informazioni sulle opzioni correnti, consulta il blog di MongoDB Spark Connector.
- L'utilizzo di MongoDB 4.0 ospitato da Amazon DocumentDB presenta alcune differenze funzionali. Per ulteriori informazioni, consulta i seguenti argomenti:
  - Differenze funzionali: Amazon DocumentDB e MongoDB
  - APIsMongoDB, operazioni e tipi di dati supportati.
- L'opzione "partitioner" è limitata a `ShardedPartitioner`, `PaginateIntoPartitionsPartitioner` e `SinglePartitionPartitioner`. Non può utilizzare `SamplePartitioner` e `PaginateBySizePartitioner` predefiniti per Amazon DocumentDB perché l'operatore stage non supporta l'API MongoDB. Per ulteriori informazioni, consulta APIsMongoDB, operazioni e tipi di dati supportati.

Delta Lake

- Delta Lake ora supporta i viaggi nel tempo in SQL per interrogare facilmente i dati più vecchi. Con questo aggiornamento, il viaggio nel tempo è ora disponibile sia in Spark SQL che tramite l'API. DataFrame È stato aggiunto il supporto per la versione corrente di TIMESTAMP in SQL.
- Spark 3.3 introduce Trigger. AvailableNowper eseguire query in streaming come equivalente a quelle in batch`Trigger.Once`. Questo supporto è disponibile anche quando si utilizzano le tabelle Delta come fonte di streaming.
- Supporto per SHOW COLUMNS per restituire l'elenco delle colonne in una tabella.
- Support per DESCRIVE DETAIL nelle API Scala e Python DeltaTable . Recupera informazioni dettagliate su una tabella Delta utilizzando l' DeltaTable API o Spark SQL.

- Supporto per la restituzione di parametri operativi dai comandi SQL Delete, Merge e Update. In precedenza questi comandi SQL restituivano un valore vuoto DataFrame, ora restituiscono un valore DataFrame con metriche utili sull'operazione eseguita.
- Ottimizza i miglioramenti in termini di prestazioni:
  - Imposta l'opzione di configurazione `spark.databricks.delta.optimize.repartition.enabled=true` in modo da utilizzare `repartition(1)` anziché `coalesce(1)` nel comando Optimize per migliorare le prestazioni durante la compattazione di numerosi file di piccole dimensioni.
  - Prestazioni migliorate grazie a un approccio basato su code per parallelizzare i lavori di compattazione.
- Altre modifiche importanti:
  - Supporto per l'utilizzo di variabili nei comandi VACUUM e OPTIMIZE SQL.
  - Miglioramenti per CONVERT TO DELTA con tabelle di catalogo che includono:
    - Completamento automatico dello schema delle partizioni dal catalogo quando non è fornito.
    - Uso delle informazioni sulle partizioni dal catalogo per trovare i file di dati da salvare invece di eseguire una scansione completa della directory. Invece di salvare tutti i file di dati nella directory delle tabelle, verranno salvati solo i file di dati nelle directory delle partizioni attive.
  - Supporto per le letture batch di Change Data Feed (CDF) sulle tabelle abilitate alla mappatura delle colonne quando DROP COLUMN e RENAME COLUMN non sono stati utilizzati. Per ulteriori informazioni, consulta la documentazione di Delta Lake.
  - Miglioramento delle prestazioni dei comandi di aggiornamento abilitando l'eliminazione dello schema nel primo passaggio.

Apache Iceberg

- Sono stati aggiunti diversi miglioramenti delle prestazioni per la pianificazione delle scansioni e le query Spark.
- È stato aggiunto un client di catalogo REST comune che utilizza i commit basati sulle modifiche per risolvere i conflitti di commit lato del servizio.
- La sintassi `AS 0F` per le query SQL relative ai viaggi temporali è supportata.
- È stato aggiunto merge-on-read il supporto per le query MERGE e UPDATE.
- È stato aggiunto il supporto per riscrivere le partizioni utilizzando l'ordine Z.

- Sono state aggiunte una specifica e un'implementazione per Puffin, un formato per statistiche di grandi dimensioni e blob di indici, come schizzi Theta o filtri bloom.
- Sono state aggiunte nuove interfacce per il consumo incrementale dei dati (scansioni di aggiunta e log delle modifiche).
- È stato aggiunto il supporto per operazioni di massa e letture a intervalli alle interfacce FileIO.
- Sono state aggiunte altre tabelle di metadati per mostrare i file di eliminazione nella struttura dei metadati.
- Il comportamento della tabella di eliminazione è cambiato. In Iceberg 0.13.1, l'esecuzione di `DROP TABLE` rimuove la tabella dal catalogo e ne elimina anche il contenuto. In Iceberg 1.0.0, `DROP TABLE` rimuove solo la tabella dal catalogo. Per eliminare il contenuto della tabella, utilizza `DROP TABLE PURGE`.
- Le letture vettorializzate in Parquet sono abilitate per impostazione predefinita in Iceberg 1.0.0. Se desideri disabilitare le letture vettorializzate, imposta `read.parquet.vectorization.enabled` su `false`.

Oracle

Le modifiche sono di lieve entità.

MySQL

Le modifiche sono di lieve entità.

Amazon Redshift

AWS Glue 4.0 presenta un nuovo connettore Amazon Redshift con un nuovo driver JDBC. Per informazioni sui miglioramenti e su come migrare dalle versioni precedenti, consulta. AWS Glue the section called "Connessioni Redshift"

## Appendice A: Aggiornamenti importanti delle dipendenze

Di seguito sono riportati gli aggiornamenti delle dipendenze:

| Dipendenza | Versione in 4.0 AWS Glue | Versione in AWS Glue 3.0 | Versione in AWS Glue 2.0 | Versione in AWS Glue 1.0 |
|---|---|---|---|---|
| Spark | 3.3.0-amzn-1 | 3.1.1-amzn-0 | 2.4.3 | 2.4.3 |

| Dipendenza | Versione in 4.0 AWS Glue | Versione in AWS Glue 3.0 | Versione in AWS Glue 2.0 | Versione in AWS Glue 1.0 |
|---|---|---|---|---|
| Hadoop | 3.3.3-amzn-0 | 3.2.1-amzn-3 | 2.8.5-amzn-5 | 2.8.5-amzn-1 |
| Scala | 2.12 | 2,12 | 2.11 | 2.11 |
| Jackson | 2,13,3 | 2.10.x | 2.7.x | 2.7.x |
| Hive | 2.3.9-amzn-2 | 2.3.7-amzn-4 | 1.2 | 1.2 |
| EMRFS | 2,54,0 | 2,46,0 | 2.38.0 | 2.30.0 |
| Json4s | 3.7.0-M11 | 36.6 | 3.5.x | 3.5.x |
| Arrow | 7,0,0 | 2.0.0 | 0.10.0 | 0.10.0 |
| AWS Glue Client Data Catalog | 3.7.0 | 3.0.0 | 1.10.0 | N/D |
| Python | 3.10 | 3.7 | 2.7 e 3.6 | 2.7 e 3.6 |
| Boto | 1,26 | 1,18 | 1.12 | N/D |

## Appendice B: aggiornamenti dei driver JDBC

Di seguito sono riportati gli aggiornamenti dei driver JDBC:

| Driver | Versione del driver JDBC nelle versioni precedenti AWS Glue | Versione del driver JDBC nella versione 3.0 AWS Glue | Versione del driver JDBC nella versione 4.0 AWS Glue |
|---|---|---|---|
| MySQL | 5.1 | 8.0.23 | 8.0.23 |
| Microsoft SQL Server | 6.1.0 | 7,0,0 | 9,40 |
| Database Oracle | 11.2 | 21,1 | 21,7 |
| PostgreSQL | 42,1,0 | 4,2,18 | 42,36 |

| Driver | Versione del driver JDBC nelle versioni precedenti AWS Glue | Versione del driver JDBC nella versione 3.0 AWS Glue | Versione del driver JDBC nella versione 4.0 AWS Glue |
|---|---|---|---|
| MongoDB | 2.0.0 | 4.0.0 | 4,7,2 |
| Amazon Redshift | redshift-jdbc41-1.2.12.1017 | redshift-jdbc41-1.2.12.1017 | redshift-jdbc42-2.1.0.16 |

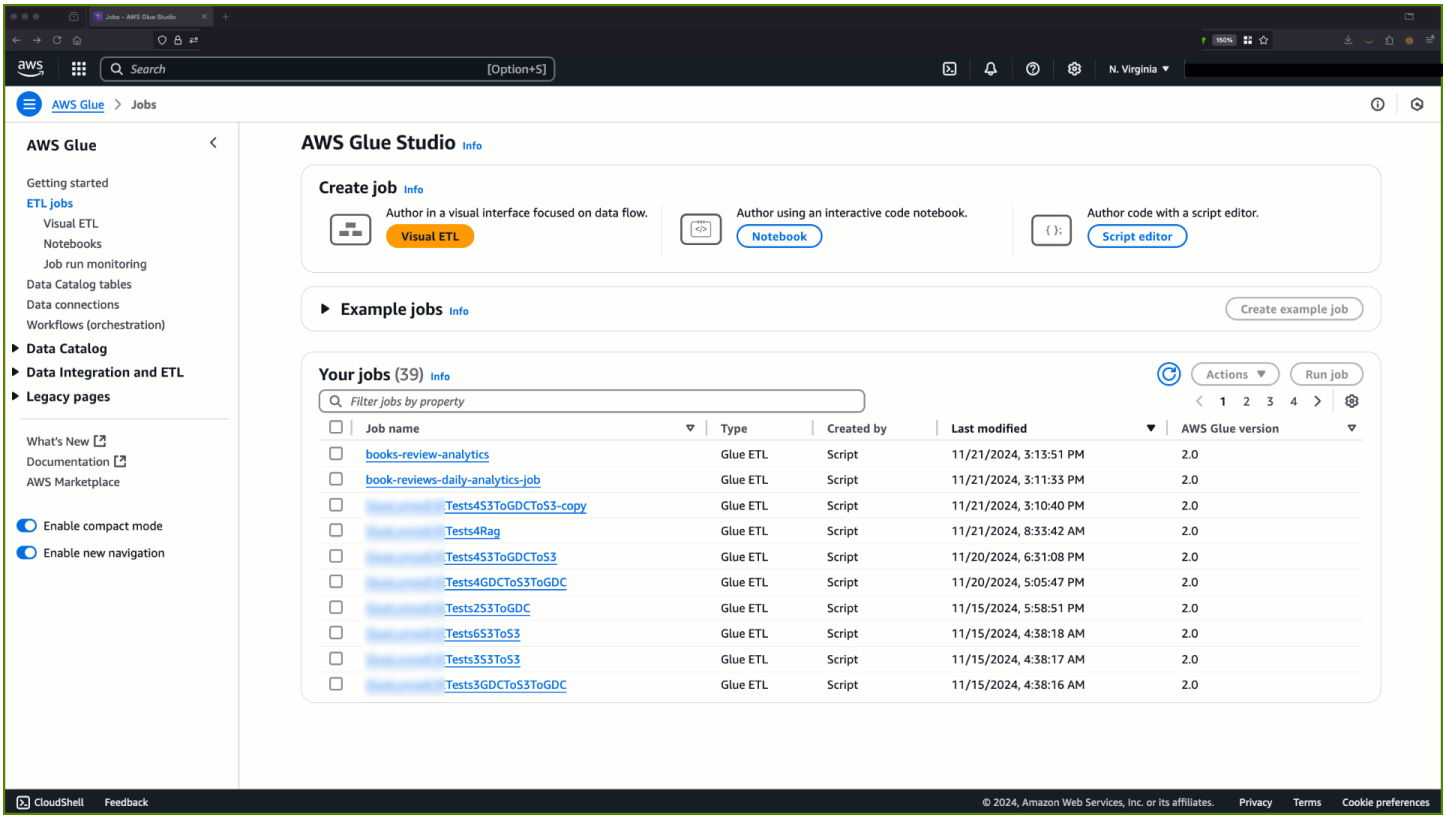
### Appendice C: Aggiornamenti dei connettori

Di seguito sono riportati gli aggiornamenti dei connettori:

| Driver | Versione del connettore in 3.0 AWS Glue | Versione del connettore in AWS Glue 4.0 |
|---|---|---|
| MongoDB | 3.0.0 | 10.0.4 |
| Hudi | 010.1 | 0.12.1 |
| Delta Lake | 1.0.0 | 2.1.0 |
| Iceberg | 0.13.1 | 1.0.0 |
| DynamoDB | 1.11 | 1.12 |

## Aggiornamenti generativi dell'intelligenza artificiale per Apache Spark in Glue AWS

Spark Upgrades in AWS Glue consente ai data engineer e agli sviluppatori di aggiornare e migrare i job AWS Glue Spark esistenti alle ultime release di Spark utilizzando l'intelligenza artificiale generativa. I data engineer possono utilizzarlo per AWS scansionare i lavori di Glue Spark, generare piani di aggiornamento, eseguire piani e convalidare gli output. Riduce i tempi e i costi degli aggiornamenti di Spark automatizzando il lavoro indifferenziato di identificazione e aggiornamento degli script, delle configurazioni, delle dipendenze, dei metodi e delle funzionalità di Spark.




## Come funziona

Quando utilizzi l'analisi degli aggiornamenti, AWS Glue identifica le differenze tra le versioni e le configurazioni nel codice del lavoro per generare un piano di aggiornamento. Il piano di aggiornamento descrive tutte le modifiche al codice e le fasi di migrazione richieste. Successivamente, AWS Glue crea ed esegue l'applicazione aggiornata in un ambiente per convalidare le modifiche e genera un elenco di modifiche al codice per la migrazione del lavoro. È possibile visualizzare lo script aggiornato insieme al riepilogo che descrive in dettaglio le modifiche proposte. Dopo aver eseguito i tuoi test, accetta le modifiche e il lavoro AWS Glue verrà aggiornato automaticamente alla versione più recente con il nuovo script.

Il processo di analisi degli aggiornamenti può richiedere un po' di tempo per il completamento, in funzione della complessità del processo e del carico di lavoro. I risultati dell'analisi dell'aggiornamento verranno archiviati nel percorso Amazon S3 specificato, che può essere esaminato per comprendere l'aggiornamento ed eventuali problemi di compatibilità. Dopo aver esaminato i risultati dell'analisi dell'aggiornamento, si può decidere se procedere con l'aggiornamento effettivo o apportare le modifiche necessarie al processo prima dell'aggiornamento.

## Prerequisiti

I seguenti prerequisiti sono necessari per utilizzare l'intelligenza artificiale generativa per aggiornare i lavori in AWS Glue:

- AWS Glue 2 PySpark jobs: solo i lavori AWS Glue 2 possono essere aggiornati a AWS Glue 5.
- Le autorizzazioni IAM sono necessarie per avviare l'analisi, esaminare i risultati e aggiornare il processo. Per ulteriori informazioni, consultare gli esempi nella sezione Permissions seguente.
- Se si utilizza AWS KMS per crittografare gli artefatti di analisi, sono necessarie autorizzazioni aggiuntive AWS AWS KMS . Per ulteriori informazioni, consultare gli esempi nella sezione AWS KMS politica seguente.

### Permissions

Per avviare una nuova analisi degli aggiornamenti, sono necessarie le seguenti autorizzazioni:

1. Aggiornare la policy IAM del chiamante con la seguente autorizzazione:

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "glue:StartJobUpgradeAnalysis",
                "glue:StartJobRun",
                "glue:GetJobRun",
                "glue:GetJob",
                "glue:BatchStopJobRun"
            ],
            "Resource": [
                "arn:aws:glue:us-east-1:111122223333:job/jobName"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
            ],
```

```
            "Resource": [
              "arn:aws:s3:::amzn-s3-demo-bucket/script-location/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObject"
            ],
            "Resource": [
              "arn:aws:s3:::amzn-s3-demo-bucket/results/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt",
                "kms:GenerateDataKey"
            ],
            "Resource": "arn:aws:kms:us-east-1:111122223333:key/key-id"
        }
    ]
}
```

2. Aggiornare il ruolo di esecuzione del processo in fase di aggiornamento per includere la seguente policy in linea:

```
    {
      "Effect": "Allow",
      "Action": ["s3:GetObject"],
      "Resource": [
        "ARN of the Amazon S3 path provided on API",
        "ARN of the Amazon S3 path provided on API/*"
      ]
    }
```

Ad esempio, se si utilizza il percorso `s3://amzn-s3-demo-bucket/upgraded-result` di Amazon S3, la policy sarà:

```
{
      "Effect": "Allow",
```

```
    "Action": ["s3:GetObject"],
    "Resource": [
      "arn:aws:s3:::amzn-s3-demo-bucket/upgraded-result/",
      "arn:aws:s3:::amzn-s3-demo-bucket/upgraded-result/*"
    ]
  }
```

Per recuperare i dettagli di un'analisi, sono necessarie le seguenti autorizzazioni:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:GetJobUpgradeAnalysis"
      ],
      "Resource": [
        "arn:aws:glue:us-east-1:111122223333:job/jobName"
      ]
    }
  ]
}
```

Per interrompere un'analisi in corso, sono necessarie le seguenti autorizzazioni:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:StopJobUpgradeAnalysis",
        "glue:BatchStopJobRun"
      ],
```

```
      "Resource": [
        "arn:aws:glue:us-east-1:111122223333:job/jobName"
      ]
    }
  ]
}
```

Per elencare tutte le analisi inviate per un processo specifico, sono necessarie le seguenti autorizzazioni:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:ListJobUpgradeAnalyses"
      ],
      "Resource": [
        "arn:aws:glue:us-east-1:111122223333:job/jobName"
      ]
    }
  ]
}
```

Per accettare le modifiche di un'analisi e aggiornare un processo, sono necessarie le seguenti autorizzazioni:

AWS KMS politica

Per passare la tua AWS KMS chiave personalizzata all'avvio di un'analisi, consulta la sezione seguente per configurare le autorizzazioni appropriate sulle AWS KMS chiavi.

Configurazione della crittografia degli artefatti dei risultati utilizzando una chiave: AWS KMS

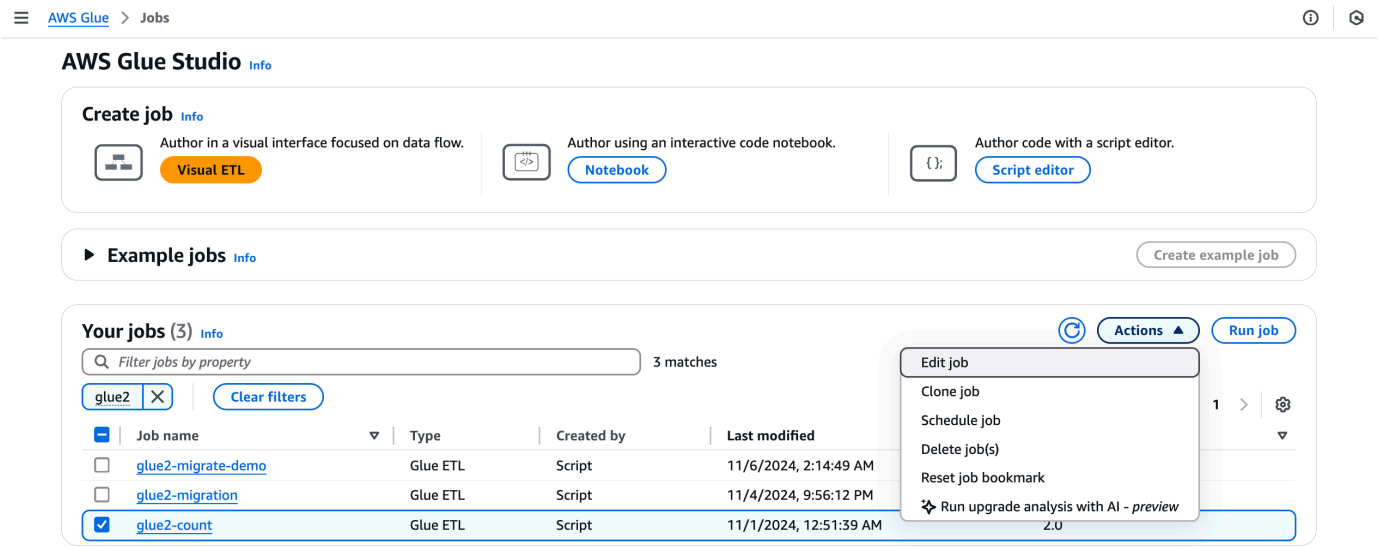
Questa politica garantisce di disporre sia delle autorizzazioni di crittografia che di decrittografia sulla chiave. AWS KMS

```
{
    "Effect": "Allow",
    "Principal":{
        "AWS": "<IAM Customer caller ARN>"
    },
    "Action": [
      "kms:Decrypt",
      "kms:GenerateDataKey",
    ],
    "Resource": "<key-arn-passed-on-start-api>"
}
```

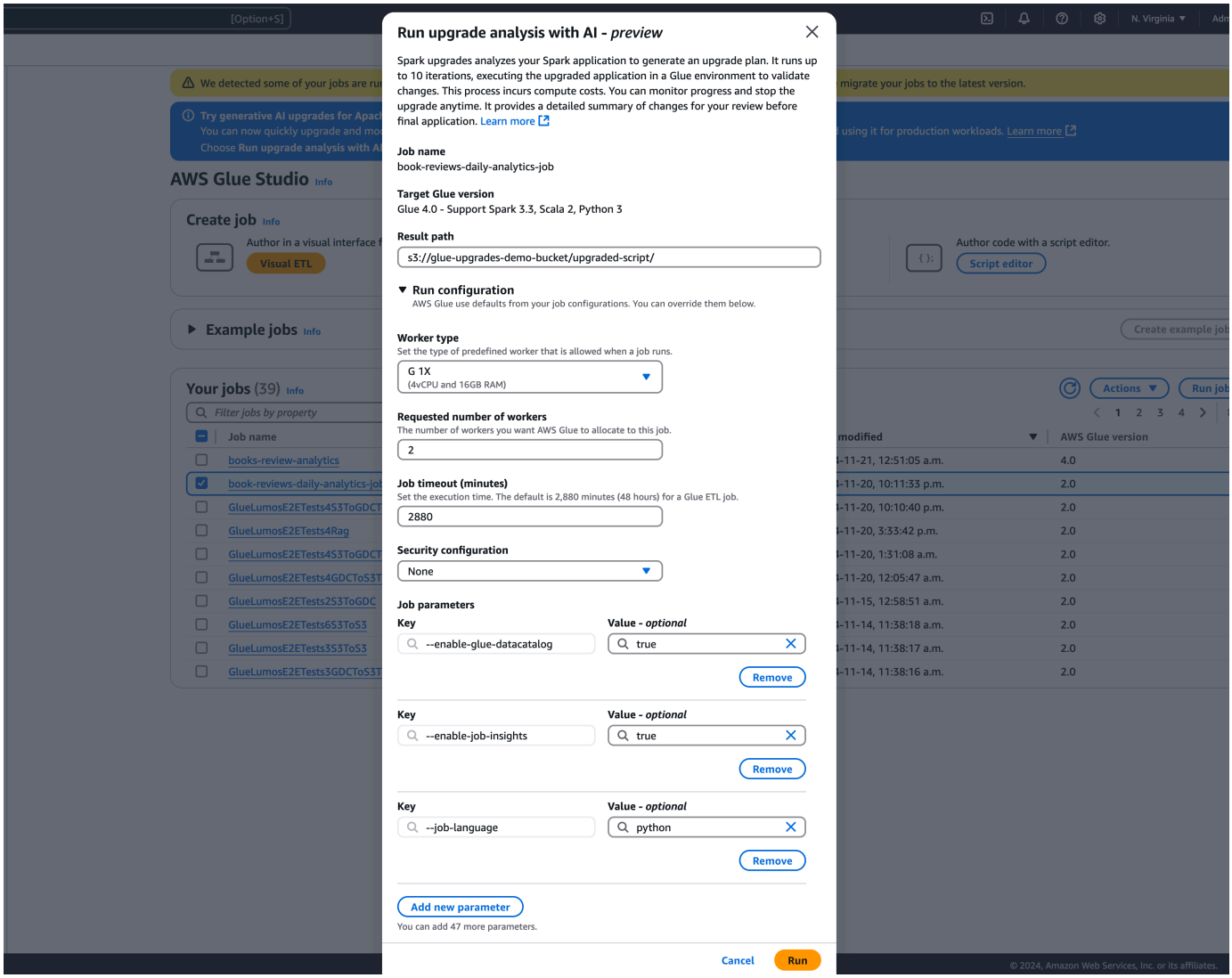
## Esecuzione di un'analisi degli aggiornamenti e applicazione dello script di aggiornamento

È possibile eseguire un'analisi degli aggiornamenti, che genererà un piano di aggiornamento su un processo selezionato dalla vista Processi.

1. Da Jobs, selezionate un job AWS Glue 2.0, quindi scegliete Esegui analisi di aggiornamento dal menu Azioni.




2. Nel modale, selezionare un percorso per memorizzare il piano di aggiornamento generato nel Percorso dei risultati. Questo deve essere il bucket Amazon S3 a cui si può accedere e su cui scrivere.




3. Configurare le opzioni aggiuntive, se necessario:

- Configurazione di esecuzione - opzionale: la configurazione di esecuzione è un'impostazione opzionale che consente di personalizzare vari aspetti delle esecuzioni di convalida eseguite durante l'analisi degli aggiornamenti. Questa configurazione viene utilizzata per eseguire lo script aggiornato e consente di selezionare le proprietà dell'ambiente di calcolo (tipo di worker, numero di worker, ecc.). È necessario utilizzare gli account di sviluppatore non di produzione per eseguire le convalide su set di dati di esempio prima di esaminare, accettare le modifiche e applicarle agli ambienti di produzione. La configurazione di esecuzione include i seguenti parametri personalizzabili:
  - Tipo di worker: è possibile specificare il tipo di worker da utilizzare per le esecuzioni di convalida, in modo da scegliere le risorse di calcolo appropriate in base ai requisiti.

- Numero di worker: è possibile definire il numero di worker da assegnare per le esecuzioni di convalida, in modo da scalare le risorse in base alle esigenze del carico di lavoro.
- Timeout del processo (in minuti): questo parametro consente di impostare un limite di tempo per le esecuzioni di convalida, assicurando che i processi terminino dopo una durata specificata per evitare un consumo eccessivo di risorse.
- Configurazione della sicurezza: è possibile configurare le impostazioni di sicurezza, come la crittografia e il controllo degli accessi, per garantire la protezione dei dati e delle risorse durante le esecuzioni di convalida.
- Parametri di processo aggiuntivi: se necessario, è possibile aggiungere nuovi parametri di processo per personalizzare ulteriormente l'ambiente di esecuzione per le esecuzioni di convalida.

Sfruttando la configurazione di esecuzione, è possibile personalizzare le esecuzioni di convalida in base ai requisiti specifici. Ad esempio, è possibile configurare le esecuzioni di convalida per utilizzare un set di dati più piccolo, che consente di completare l'analisi più rapidamente e di ottimizzare i costi. Questo approccio garantisce che l'analisi degli aggiornamenti venga eseguita in modo efficiente, riducendo al minimo l'utilizzo delle risorse e i costi associati durante la fase di convalida.

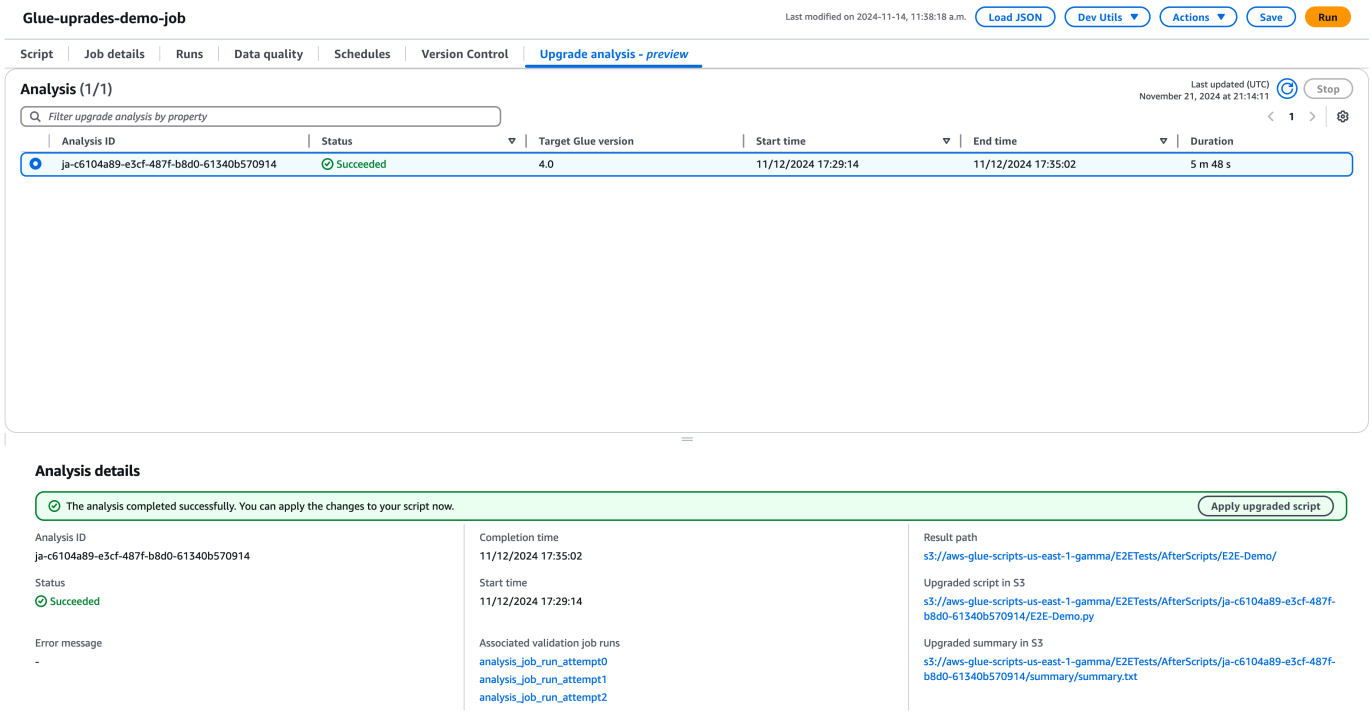
- Configurazione della crittografia - (opzionale):
  - Abilitare la crittografia degli artefatti di aggiornamento: abilitare la crittografia a riposo durante la scrittura dei dati nel percorso dei risultati. Se non si desidera crittografare gli elementi di aggiornamento, lasciare questa opzione deselezionata.

4. Scegliere Esegui per avviare l'analisi degli aggiornamenti. Mentre l'analisi è in esecuzione, è possibile visualizzare i risultati nella scheda Analisi dell'aggiornamento. La finestra dei dettagli dell'analisi mostrerà informazioni sull'analisi e collegamenti agli artefatti di aggiornamento.

   - Percorso dei risultati: è qui che vengono archiviati il riepilogo dei risultati e lo script di aggiornamento.
   - Script aggiornato in Amazon S3: la posizione dello script di aggiornamento in Amazon S3. È possibile visualizzare lo script prima di applicare l'aggiornamento.
   - Riepilogo dell'aggiornamento in Amazon S3: la posizione del riepilogo dell'aggiornamento in Amazon S3. È possibile visualizzare il riepilogo dell'aggiornamento prima di applicarlo.

5. Una volta completata con successo l'analisi dell'aggiornamento, è possibile applicare lo script di aggiornamento per aggiornare automaticamente il processo scegliendo Applica script aggiornato.

Una volta applicata, la versione AWS Glue verrà aggiornata alla 4.0. È possibile visualizzare lo script nella scheda Script.




## Comprendere il riepilogo dell'aggiornamento

Questo esempio dimostra il processo di aggiornamento di un lavoro AWS Glue dalla versione 2.0 alla versione 4.0. Il processo di esempio legge i dati di prodotto da un bucket Amazon S3, applica diverse trasformazioni ai dati utilizzando Spark SQL e quindi salva i risultati trasformati in un bucket Amazon S3.

### Codice originale (AWS Glue 2.0) - prima dell'aggiornamento

```
from awsglue.transforms import *
from pyspark.context import SparkContext
from awsglue.context import GlueContext
from pyspark.sql.types import *
from pyspark.sql.functions import *
from awsglue.job import Job
import json
from pyspark.sql.types import StructType
```

```
sc = SparkContext.getOrCreate()
glueContext = GlueContext(sc)
spark = glueContext.spark_session
job = Job(glueContext)

gdc_database = "s3://aws-glue-scripts-us-east-1-gamma/demo-database/"
schema_location = (
    "s3://aws-glue-scripts-us-east-1-gamma/DataFiles/"
)

products_schema_string = spark.read.text(
    f"{schema_location}schemas/products_schema"
).first()[0]

product_schema = StructType.fromJson(json.loads(products_schema_string))

products_source_df = (
    spark.read.option("header", "true")
    .schema(product_schema)
    .option(
        "path",
        f"{gdc_database}products/",
    )
    .csv(f"{gdc_database}products/")
)

products_source_df.show()
products_temp_view_name = "spark_upgrade_demo_product_view"
products_source_df.createOrReplaceTempView(products_temp_view_name)

query = f"select {products_temp_view_name}.*, format_string('%0$s-%0$s', category,
 subcategory) as unique_category from {products_temp_view_name}"
products_with_combination_df = spark.sql(query)
products_with_combination_df.show()

products_with_combination_df.createOrReplaceTempView(products_temp_view_name)
product_df_attribution = spark.sql(
    f"""
SELECT *,
unbase64(split(product_name, ' ')[0]) as product_name_decoded,
unbase64(split(unique_category, '-')[1]) as subcategory_decoded
FROM {products_temp_view_name}
"""
)
```

```
product_df_attribution.show()


product_df_attribution.write.mode("overwrite").option("header", "true").option(
    "path", f"{gdc_database}spark_upgrade_demo_product_agg/"
).saveAsTable("spark_upgrade_demo_product_agg", external=True)

spark_upgrade_demo_product_agg_table_df = spark.sql(
    f"SHOW TABLE EXTENDED in default like 'spark_upgrade_demo_product_agg'"
)
spark_upgrade_demo_product_agg_table_df.show()
job.commit()
```

Nuovo codice (Glue 4.0), dopo l'aggiornamento

```
from awsglue.transforms import *
from pyspark.context import SparkContext
from awsglue.context import GlueContext
from pyspark.sql.types import *
from pyspark.sql.functions import *
from awsglue.job import Job
import json
from pyspark.sql.types import StructType

sc = SparkContext.getOrCreate()
glueContext = GlueContext(sc)
spark = glueContext.spark_session
# change 1
spark.conf.set("spark.sql.adaptive.enabled", "false")
# change 2
spark.conf.set("spark.sql.legacy.pathOptionBehavior.enabled", "true")
job = Job(glueContext)

gdc_database = "s3://aws-glue-scripts-us-east-1-gamma/demo-database/"
schema_location = (
    "s3://aws-glue-scripts-us-east-1-gamma/DataFiles/"
)

products_schema_string = spark.read.text(
    f"{schema_location}schemas/products_schema"
).first()[0]

product_schema = StructType.fromJson(json.loads(products_schema_string))
```

```
products_source_df = (
    spark.read.option("header", "true")
    .schema(product_schema)
    .option(
        "path",
        f"{gdc_database}products/",
    )
    .csv(f"{gdc_database}products/")
)

products_source_df.show()
products_temp_view_name = "spark_upgrade_demo_product_view"
products_source_df.createOrReplaceTempView(products_temp_view_name)

# change 3
query = f"select {products_temp_view_name}.*, format_string('%1$s-%1$s', category,
 subcategory) as unique_category from {products_temp_view_name}"
products_with_combination_df = spark.sql(query)
products_with_combination_df.show()

products_with_combination_df.createOrReplaceTempView(products_temp_view_name)
# change 4
product_df_attribution = spark.sql(
    f"""
SELECT *,
try_to_binary(split(product_name, ' ')[0], 'base64') as product_name_decoded,
try_to_binary(split(unique_category, '-')[1], 'base64') as subcategory_decoded
FROM {products_temp_view_name}
"""
)
product_df_attribution.show()


product_df_attribution.write.mode("overwrite").option("header", "true").option(
    "path", f"{gdc_database}spark_upgrade_demo_product_agg/"
).saveAsTable("spark_upgrade_demo_product_agg", external=True)

spark_upgrade_demo_product_agg_table_df = spark.sql(
    f"SHOW TABLE EXTENDED in default like 'spark_upgrade_demo_product_agg'"
)
spark_upgrade_demo_product_agg_table_df.show()
job.commit()
```

Spiegazione del riepilogo dell'analisi

```
In Spark 3.2, spark.sql.adaptive.enabled is enabled by default. To restore the behavior before
Spark 3.2,
you can set spark.sql.adaptive.enabled to false.


In Spark 3.1, path option cannot coexist when the following methods are called with path
parameter(s): DataFrameReader.load(), DataFrameWriter.save(), DataStreamReader.load(), or
DataStreamWriter.start(). In addition, paths option cannot coexist for DataFrameReader.load().
For example, spark.read.format(csv).option(path, /tmp).load(/tmp2) or spark.read.option(path, /
tmp).csv(/tmp2) will throw org.apache.spark.sql.AnalysisException. In Spark version 3.0 and
below, path option is overwritten if one path parameter is passed to above methods; path option
is added to the overall paths if multiple path parameters are passed to DataFrameReader.load().
To restore the behavior before Spark 3.1, you can set
spark.sql.legacy.pathOptionBehavior.enabled to true.


Since Spark 3.3, the `strfmt` in `format_string(strfmt, obj, ...)` and `printf(strfmt,
obj, ...)` will no longer support to use 0$ to specify the first argument, the first argument
should always reference by 1$ when use argument index to indicating the position of the argument
in the argument list.


Since Spark 3.3, the unbase64 function throws error for a malformed str input. Use
try_to_binary(<str>, 'base64') to tolerate malformed input and return NULL instead. In Spark 3.2
and earlier, the unbase64 function returns a best-efforts result for a malformed str input.
```

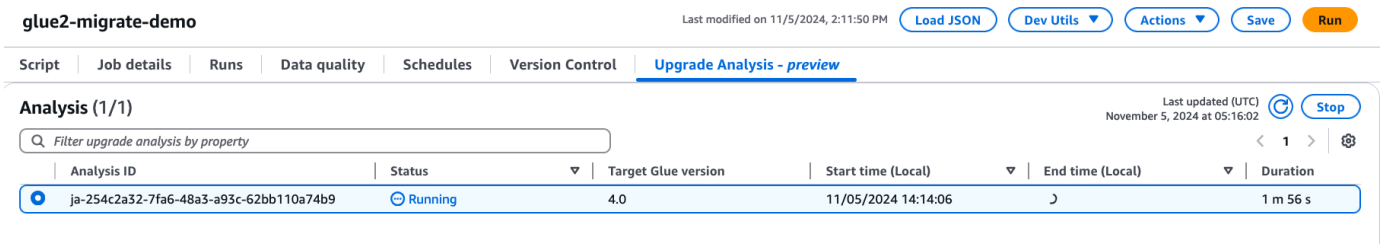
In base al riepilogo, ci sono quattro modifiche proposte da AWS Glue per aggiornare correttamente lo script da AWS Glue 2.0 a AWS Glue 4.0:

1. Configurazione SQL di Spark (spark.sql.adaptive.enabled): questa modifica serve a ripristinare il comportamento dell'applicazione quando una nuova funzionalità per l'esecuzione adattiva delle query Spark SQL viene introdotta a partire da Spark 3.2. Si può ispezionare questa modifica alla configurazione e abilitarla o disabilitarla ulteriormente secondo le preferenze.
2. DataFrame Modifica dell'API: l'opzione path non può coesistere con altre DataFrameReader operazioni come`load()`. Per mantenere il comportamento precedente, AWS Glue ha aggiornato lo script per aggiungere una nuova configurazione SQL (spark.sql.legacy). pathOptionBehavior.abilitato).
3. Modifica dell'API SQL di Spark: il comportamento di `strfmt` in `format_string(strfmt, obj, ...)` è stato aggiornato in modo da non consentire `0$` come primo argomento. Per garantire la compatibilità, AWS Glue ha modificato lo script da utilizzare invece 1$ come primo argomento.
4. Modifica dell'API SQL di Spark: la funzione `unbase64` non consente input di stringhe in formato errato. Per mantenere il comportamento precedente, AWS Glue ha aggiornato lo script per utilizzare la `try_to_binary` funzione.

## Interruzione di un'analisi di aggiornamento in corso

È possibile annullare un'analisi dell'aggiornamento in corso o semplicemente interromperla.

1. Scegliere la scheda Analisi dell'aggiornamento.
2. Selezionare il processo in esecuzione, poi scegliere Interrompi. Questo interromperà l'analisi. È quindi possibile eseguire un'altra analisi degli aggiornamenti sullo stesso processo.




## Considerazioni

Quando inizi a utilizzare Spark Upgrades, ci sono diversi aspetti importanti da considerare per un utilizzo ottimale del servizio.

- Ambito e limitazioni del servizio: la versione corrente si concentra sugli aggiornamenti PySpark del codice dalle versioni 2.0 alla versione 5.0 di AWS Glue. Al momento, il servizio gestisce PySpark codice che non si basa su dipendenze di libreria aggiuntive. È possibile eseguire aggiornamenti automatici per un massimo di 10 processi contemporaneamente in un AWS account, in modo da aggiornare in modo efficiente più processi mantenendo al contempo la stabilità del sistema.
  - Sono supportati solo i PySpark lavori.
  - L'analisi degli aggiornamenti scadrà dopo 24 ore.
  - È possibile eseguire solo un'analisi di aggiornamento attiva alla volta per un singolo processo. A livello di account, possono essere eseguite fino a 10 analisi degli aggiornamenti attive nello stesso momento.
- Ottimizzazione dei costi durante il processo di aggiornamento: poiché Spark Upgrades utilizza l'intelligenza artificiale generativa per convalidare il piano di aggiornamento attraverso più iterazioni, con ogni iterazione eseguita come processo AWS Glue nel tuo account, è essenziale ottimizzare le configurazioni di esecuzione del processo di convalida per ridurre i costi. A tal fine, consigliamo di specificare una configurazione di esecuzione all'avvio di un'analisi degli aggiornamenti nel modo seguente:

  - Usare account di sviluppo non di produzione e selezionare esempi di set di dati fittizi che rappresentino i dati di produzione ma di dimensioni più piccole per la convalida con Spark Upgrades.
  - Utilizzare risorse di calcolo della giusta dimensione, come i worker G.1X, e selezionare un numero appropriato di worker per l'elaborazione dei dati di esempio.
  - Attivazione dell'auto-scaling dei job di AWS Glue, se applicabile, per regolare automaticamente le risorse in base al carico di lavoro.

  Ad esempio, se il processo di produzione elabora terabyte di dati con 20 worker G.2X, è possibile configurare il processo di aggiornamento per elaborare alcuni gigabyte di dati rappresentativi con 2 worker G.2X e il dimensionamento automatico abilitato per la convalida.
- Migliori pratiche: consigliamo vivamente di iniziare il percorso di aggiornamento con lavori non di produzione. Questo approccio consente di acquisire familiarità con il flusso di lavoro di aggiornamento e di comprendere come il servizio gestisce i diversi tipi di modelli di codice Spark.
- Allarmi e notifiche: quando utilizzi la funzionalità di aggiornamento dell'intelligenza artificiale generativa su un lavoro, assicurati che alarms/notifications le esecuzioni dei job non riusciti siano disattivate. Durante il processo di aggiornamento, potrebbero verificarsi fino a 10 esecuzioni di processi non riusciti nell'account prima che vengano forniti gli artefatti aggiornati.
- Regole di rilevamento delle anomalie: disattivate le regole di rilevamento delle anomalie sul processo in fase di aggiornamento, poiché i dati scritti nelle cartelle di output durante le esecuzioni intermedie dei processi potrebbero non essere nel formato previsto durante la convalida dell'aggiornamento.
- Utilizzare l'analisi degli aggiornamenti con processi di idempotenza: utilizzare l'analisi degli aggiornamenti con processi idempotenti per garantire che ogni successivo tentativo di esecuzione del processo di convalida sia simile a quello precedente e non comporti problemi. I processi idempotenti sono processi che possono essere eseguiti più volte con gli stessi dati di input e produrranno ogni volta lo stesso output. Quando si utilizzano gli aggiornamenti di intelligenza artificiale generativa per Apache Spark in AWS Glue, il servizio eseguirà più iterazioni del job come parte del processo di convalida. Durante ogni iterazione, apporterà modifiche al codice e alle configurazioni Spark per convalidare il piano di aggiornamento. Se il processo Spark non è idempotente, eseguirlo più volte con gli stessi dati di input potrebbe causare problemi.

### Regioni supportate

Gli aggiornamenti generativi dell'intelligenza artificiale per Apache Spark sono disponibili nelle seguenti regioni:

- Asia Pacifico: Tokyo (ap-northeast-1), Seul (ap-northeast-2), Mumbai (ap-south-1), Singapore (ap-southeast-1) e Sydney (ap-southeast-2)
- Nord America: Canada (ca-central-1)
- Europa: Francoforte (eu-central-1), Stoccolma (eu-north-1), Irlanda (eu-west-1), Londra (eu-west-2) e Parigi (eu-west-3)
- Sud America: San Paolo (sa-east-1)
- Stati Uniti: Virginia del Nord (us-east-1), Ohio (us-east-2) e Oregon (us-west-2)

### Inferenza tra regioni in Spark Upgrades

Spark Upgrades è basato e sfrutta l'inferenza interregionale (CRIS). Amazon Bedrock Con CRIS, Spark Upgrades selezionerà automaticamente la regione ottimale all'interno dell'area geografica (come descritto più dettagliatamente qui) per elaborare la richiesta di inferenza, massimizzare le risorse di calcolo disponibili e la disponibilità dei modelli e fornire la migliore esperienza al cliente. L'utilizzo dell'inferenza tra regioni non prevede costi aggiuntivi.

Le richieste di inferenza tra regioni vengono conservate all'interno delle AWS regioni che fanno parte della geografia in cui risiedono originariamente i dati. Ad esempio, una richiesta effettuata negli Stati Uniti viene conservata nelle AWS regioni degli Stati Uniti. Sebbene i dati rimangano archiviati solo nella regione primaria, quando si utilizza l'inferenza tra regioni, i prompt di input e i risultati di output potrebbero spostarsi al di fuori della regione primaria. Tutti i dati verranno trasmessi crittografati attraverso la rete sicura di Amazon.

## Lavorare con Spark jobs in AWS Glue

Fornisce informazioni sui lavori AWS Glue Spark ETL.

Argomenti

- Utilizzo dei parametri del lavoro nei lavori AWS Glue
- AWS Glue Spark e lavori PySpark
- AWS Glue tipi di lavoratori

- Aggiunta di processi di streaming ETL in AWS Glue
- Record di abbinamento con AWS Lake Formation FindMatches
- Esegui la migrazione dei programmi Apache Spark a AWS Glue

## Utilizzo dei parametri del lavoro nei lavori AWS Glue

Quando si crea un lavoro AWS Glue, si impostano alcuni campi standard, come `Role` e`WorkerType`. È possibile fornire informazioni di configurazione aggiuntive tramite i campi `Argument` (Parametri del processo nella console). In questi campi, è possibile fornire ai job AWS Glue gli argomenti (parametri) elencati in questo argomento.

Per ulteriori informazioni sull'API AWS Glue Job, consultathe section called "Jobs".

> **Note**
>
> Gli argomenti del processo possono avere una dimensione massima di 260 KB. Se la dimensione dell'argomento è superiore a 260 KB, un controllo di convalida genererà un errore.

### Impostazione dei parametri del processo

È possibile configurare un processo tramite la console nella scheda Job details (Dettagli del processo), sotto l'intestazione Job parameters (Parametri del processo). Puoi anche configurare un lavoro tramite AWS CLI by setting `DefaultArguments` o `NonOverridableArguments` on a job oppure impostando `Arguments` on a job run. Gli argomenti impostati nel processo verranno trasmessi ogni volta che il processo viene eseguito, mentre gli argomenti impostati durante l'esecuzione del processo verranno trasmessi solo per quella singola esecuzione.

Ad esempio, di seguito è riportata la sintassi per l'esecuzione di un processo utilizzando `--arguments` per impostare un parametro di processo.

```
$ aws glue start-job-run --job-name "CSV to CSV" --arguments='--scriptLocation="s3://
my_glue/libraries/test_lib.py"'
```

## Accesso ai parametri di processo

Quando scrivi script AWS Glue, potresti voler accedere ai valori dei parametri di lavoro per modificare il comportamento del tuo codice. Forniamo metodi di supporto per eseguire questa operazione tramite le nostre librerie. Questi metodi riescono a risolvere i valori dei parametri di esecuzione del processo che sostituiscono i valori dei parametri del processo. Quando si risolvono i parametri impostati in più posizioni, il processo `NonOverridableArguments` sostituisce l'esecuzione del processo `Arguments`, che sostituisce il processo `DefaultArguments`.

In Python:

Nei processi Python, forniamo una funzione denominata `getResolvedParameters`. Per ulteriori informazioni, consulta the section called "getResolvedOptions". I parametri del processo sono disponibili nella variabile `sys.argv`.

In Scala:

Nei processi Scala, forniamo un oggetto denominato `GlueArgParser`. Per ulteriori informazioni, consulta the section called "GlueArgParser". I parametri del processo sono disponibili nella variabile `sysArgs`.

## Riferimento ai parametri di processo

AWS Glue riconosce i seguenti nomi di argomenti che è possibile utilizzare per configurare l'ambiente di script per i job e le esecuzioni dei job:

**`--additional-python-modules`**

Un elenco delimitato da virgole che rappresenta un insieme di pacchetti Python da installare. Puoi installare pacchetti da PyPI o fornire una distribuzione personalizzata. Una voce del pacchetto PyPI sarà nel formato *`package`*`==`*`version`*, con il nome e la versione del pacchetto di destinazione. Una voce della distribuzione personalizzata è rappresentata dal percorso S3 della distribuzione.

Le voci utilizzano la versione di Python corrispondente al pacchetto e alla versione, in modo che l'utente non debba utilizzare due segni uguali, come ==. Per ulteriori informazioni su altri operatori che corrispondono alle versioni, consulta la pagina PEP 440.

Per inviare le opzioni di installazione del modulo a `pip3`, utilizza il parametro --python-modules-installer-option.

**`--auto-scale-within-microbatch`**

Il valore di default è true. Questo parametro può essere utilizzato solo per i lavori di streaming AWS Glue, che elaborano i dati di streaming in una serie di micro batch, e la scalabilità automatica deve essere abilitata. Quando si imposta questo valore su "false", viene calcolata la media mobile esponenziale della durata del batch per i microbatch completati e questo valore viene confrontato con la dimensione della finestra per determinare se aumentare o ridurre verticalmente il numero di esecutori. Il dimensionamento avviene solo quando viene completato un microbatch. Quando si imposta questo valore su "true", durante un microbatch, l'aumento avviene quando il numero di attività Spark rimane invariato per 30 secondi o se l'elaborazione del batch corrente è maggiore della dimensione della finestra. Il numero di esecutori diminuisce se un esecutore è rimasto inattivo per più di 60 secondi o se la media mobile esponenziale della durata del batch è bassa.

**`--class`**

La classe Scala che funge da punto di accesso per lo script Scala. Questo vale solo se il tuo `--job-language` è impostato su `scala`.

**`--continuous-log-conversionPattern`**

Specifica un modello di log di conversione personalizzato per un processo abilitato per la registrazione continua. Il modello di conversione si applica solo ai log dei driver e ai log delle esecuzioni. Non influisce sulla barra di avanzamento di AWS Glue.

**`--continuous-log-logGroup`**

Specifica un nome di gruppo di CloudWatch log Amazon personalizzato per un job abilitato alla registrazione continua.

**`--continuous-log-logStreamPrefix`**

Specifica un prefisso di CloudWatch log stream personalizzato per un job abilitato alla registrazione continua.

**`--customer-driver-env-vars`** e **`--customer-executor-env-vars`**

Questi parametri impostano le variabili di ambiente nel sistema operativo rispettivamente per ogni worker (driver o executor). Puoi utilizzare questi parametri quando crei piattaforme e framework personalizzati su AWS Glue, per consentire ai tuoi utenti di scrivere lavori su di esso. L'attivazione di questi due flag consentirà di impostare diverse variabili di ambiente rispettivamente sul driver e sull'executor senza dover inserire la stessa logica nello script del processo.

Esempio di utilizzo

Di seguito è riportato un esempio di utilizzo di questi parametri:

```
"–customer-driver-env-vars", "CUSTOMER_KEY1=VAL1,CUSTOMER_KEY2=\"val2,val2 val2\"",
"–customer-executor-env-vars", "CUSTOMER_KEY3=VAL3,KEY4=VAL4"
```

La loro impostazione nell'argomento dell'esecuzione del processo equivale all'esecuzione dei seguenti comandi:

Nel driver:

- export CUSTOMER_ = KEY1 VAL1
- export CUSTOMER_ KEY2 ="val2, val2 val2"

Nell'executor:

- esporta CUSTOMER_ KEY3 = VAL3

Quindi, nello script del processo, è possibile recuperare le variabili di ambiente utilizzando `os.environ.get("CUSTOMER_KEY1")` o `System.getenv("CUSTOMER_KEY1")`.

Sintassi applicata

Osservare i seguenti standard quando si definiscono le variabili di ambiente:

- Ogni chiave deve avere il `CUSTOMER_ prefix`.

  Ad esempio: per `"CUSTOMER_KEY3=VAL3,KEY4=VAL4"`, `KEY4=VAL4` verrà ignorato e non impostato.
- Ogni coppia di chiavi e valori deve essere delineata con una sola virgola.

  Ad esempio: `"CUSTOMER_KEY3=VAL3,CUSTOMER_KEY4=VAL4"`
- Se "value" contiene spazi o virgole, deve essere definito tra virgolette.

  Ad esempio: `CUSTOMER_KEY2=\"val2,val2 val2\"`

Questa sintassi modella da vicino gli standard di impostazione delle variabili di ambiente bash.

**`--datalake-formats`**

Supportato in AWS Glue 3.0 e versioni successive.

Specifica il framework del data lake da utilizzare. AWS Glue aggiunge i file JAR richiesti per i framework specificati in. `classpath` Per ulteriori informazioni, consulta Utilizzo di framework di data lake con AWS Glue processi ETL.

Puoi specificare uno o più dei seguenti valori, separati da una virgola:

- `hudi`
- `delta`
- `iceberg`

Ad esempio, invia il seguente argomento per specificare tutti e tre i framework.

```
'--datalake-formats': 'hudi,delta,iceberg'
```

**`--disable-proxy-v2`**

Disattiva il proxy del servizio per consentire le chiamate di AWS servizio ad Amazon S3 e AWS Glue provenienti dallo script tramite il tuo VPC. CloudWatch Per ulteriori informazioni, consulta Configurazione di chiamate AWS affinché passino attraverso il tuo VPC. Per disabilitare il proxy del servizio, imposta il valore di questo parametro su `true`.

**`--enable-auto-scaling`**

Attiva la scalabilità automatica e la fatturazione per operatore quando imposti il valore su `true`.

**`--enable-continuous-cloudwatch-log`**

Consente la registrazione continua in tempo reale per i lavori AWS Glue. Puoi visualizzare i log dei processi Apache Spark in tempo reale in CloudWatch.

**`--enable-continuous-log-filter`**

Specifica un filtro standard (`true`) o nessun filtro (`false`) durante la creazione o la modifica di un processo abilitato per la registrazione continua. La scelta del filtro standard elimina i messaggi di registro del battito cardiaco non utili di Apache Spark driver/executor e Apache Hadoop YARN. Non scegliendo alcun filtro si ottengono tutti i messaggi di log.

**`--enable-glue-datacatalog`**

Consente di utilizzare AWS Glue Data Catalog come metastore Apache Spark Hive. Imposta questo valore su `true` per abilitare questa funzionalità.

**`--enable-job-insights`**

Consente un monitoraggio aggiuntivo dell'analisi degli errori con AWS Glue job run insights. Per informazioni dettagliate, vedi the section called “Monitoraggio con AWS Glue Job Run

Insights". Per impostazione predefinita, il valore è impostato su `true` e le informazioni dettagliate sull'esecuzione dei processi sono abilitate.

Questa opzione è disponibile per AWS Glue versione 2.0 e 3.0.

**`--enable-lakeformation-fine-grained-access`**

Consente un controllo granulare degli accessi per i lavori AWS Glue. Per ulteriori informazioni, consulta the section called "Lake Formation per FGAC".

**`--enable-metrics`**

Abilita la raccolta di parametri per la profilatura del processo per questa esecuzione. Queste metriche sono disponibili sulla console AWS Glue e sulla CloudWatch console Amazon. Il valore di questo parametro non è rilevante. Per abilitare questa funzionalità, è possibile fornire a questo parametro qualsiasi valore, ma `true` è consigliabile per motivi di chiarezza. Per disabilitare la funzionalità, rimuovi questo parametro dalla configurazione del processo.

**`--enable-observability-metrics`**

Abilita una serie di metriche di osservabilità per generare informazioni su ciò che accade all'interno di ogni lavoro eseguito nella pagina Job Runs Monitoring nella console AWS Glue e nella Amazon CloudWatch console. Per abilitare questa funzionalità, imposta il valore del parametro su "true". Per disabilitare questa funzionalità, impostalo su `false` o rimuovi questo parametro dalla configurazione del processo.

**`--enable-rename-algorithm-v2`**

Imposta la versione dell'algoritmo di ridenominazione EMRFS alla versione 2. Quando un processo Spark utilizza la modalità di sovrascrittura della partizione dinamica, è possibile che venga creata una partizione duplicata. Ad esempio, si può ottenere una partizione duplicata come `s3://bucket/table/location/p1=1/p1=1`. Qui, P1 è la partizione che viene sovrascritta. La versione 2 dell'algoritmo di ridenominazione risolve questo problema.

Questa opzione è disponibile solo nella versione 1.0 di AWS Glue.

**`--enable-s3-parquet-optimized-committer`**

Abilita il committer ottimizzato EMRFS S3 per la scrittura dei dati Parquet in Amazon S3. È possibile fornire la parameter/value coppia tramite la console AWS Glue durante la creazione o l'aggiornamento di un lavoro AWS Glue. L'impostazione del valore su **`true`** abilita il committer. Per impostazione predefinita, il flag è attivato in AWS Glue 3.0 e disattivato in AWS Glue 2.0.

Per ulteriori informazioni, consulta Utilizzo del committer ottimizzato EMRFS S3.

**--enable-spark-ui**

Se impostato su`true`, attiva la funzionalità per utilizzare l'interfaccia utente Spark per monitorare ed eseguire il debug dei lavori AWS Glue ETL.

**--executor-cores**

Numero di attività spark che possono essere eseguite in parallelo. Questa opzione è supportata su AWS Glue 3.0+. Il valore non deve superare il doppio del numero di v CPUs del tipo di lavoratore, ovvero 8 on`G.1X`, 16 on`G.2X`, 32 on`G.4X`, 64 on`G.8X`, 96 on`G.12X`, 128 on `G.16X` e 8 on`R.1X`, 16 on, 32 on `R.2X`R`.4X`, 64 on. `R.8X` È necessario prestare attenzione durante l'aggiornamento di questa configurazione in quanto potrebbe influire sulle prestazioni del processo; l'incremento del parallelismo, infatti, esercita pressione sulla memoria e sul disco, oltre a limitare i sistemi di origine e destinazione (ad esempio, potrebbe causare più connessioni simultanee su Amazon RDS).

**--extra-files**

I percorsi Amazon S3 dei file aggiuntivi, come file di configurazione, che AWS Glue copia nella directory di lavoro del tuo script sul nodo del driver prima di eseguirlo. I valori multipli devono essere percorsi completi separati dalla virgola (`,`). Il valore può essere rappresentato da singoli file o posizioni di directory. Questa opzione non è supportata per i tipi di processo shell Python.

**--extra-jars**

Amazon S3 conduce a file aggiuntivi che AWS Glue copia nel driver e negli executor. AWS Glue aggiunge anche questi file al classpath Java prima di eseguire lo script. I valori multipli devono essere percorsi completi separati dalla virgola (`,`). L'estensione non deve essere `.jar`

**--extra-py-files**

I percorsi di Amazon S3 verso moduli Python aggiuntivi che AWS Glue aggiunge al percorso Python sul nodo driver prima di eseguire lo script. I valori multipli devono essere percorsi completi separati dalla virgola (`,`). Solo i singoli file sono supportati, non il percorso di una directory.

**--job-bookmark-option**

Controlla il comportamento di un segnalibro del processo. É possibile impostare i seguenti valori opzione.

| --job-bookmark-option Valore | Description |
| --- | --- |
| `job-bookmark-enable` | Tieni traccia dei dati elaborati in precedenza. Quando si esegue un processo, elabora i nuovi dati a partire dall'ultimo punto di controllo. |
| `job-bookmark-disable` | Elabora sempre l'intero set di dati. Sei responsab ile della gestione dell'output delle esecuzioni di job precedenti. |
| `job-bookmark-pause` | Elabora i dati incrementali dall'ultima esecuzione riuscita o i dati nell'intervallo identificato dalle seguenti opzioni secondarie, senza aggiornare lo stato dell'ulti mo segnalibro. L'utente è responsabile della gestione dell'output delle esecuzioni di job precedenti. Le due opzioni secondarie sono le seguenti:<br>• **`job-bookmark-from`** `<from-value>` è l'ID di esecuzione che rappresenta tutto l'input che è stato elaborato fino all'ultima esecuzione riuscita prima, incluso l'ID di esecuzione specificato. L'input corrispondente viene ignorato.<br>• **`job-bookmark-to`** `<to-value>` è l'ID di esecuzione che rappresenta tutto l'input che è stato elaborato fino all'ultima esecuzione riuscita prima, incluso l'ID di esecuzione specificato. L'input corrispondente escluso l'input identificato da `<from-value>` viene elaborato dal processo. Qualsiasi input successivo a questo è escluso anche dall'elab orazione.<br><br>Lo stato dei segnalibri di processo non viene aggiornato quando viene specificato questo set di opzioni.<br><br>Le opzioni secondarie sono facoltative. Tuttavia, se vengono utilizzate, devono essere fornite entrambe le opzioni secondarie. |

Ad esempio, per abilitare un segnalibro di processo, passa l'argomento seguente.

```
'--job-bookmark-option': 'job-bookmark-enable'
```

**`--job-language`**

Il linguaggio di programmazione script. Questo valore deve essere `scala` o `python`. Se questo parametro non è presente, il valore predefinito è `python`.

**`--python-modules-installer-option`**

Una stringa di testo semplice che definisce le opzioni da inviare a `pip3` quando si installano i moduli con --additional-python-modules. Fornisci le opzioni come faresti nella riga di comando, separate da spazi e precedute da trattini. Per ulteriori informazioni sull'utilizzo, consulta the section called "Installazione di moduli Python aggiuntivi con pip in AWS Glue 2.0 o versioni successive".

> **Note**
>
> Questa opzione non è supportata per i lavori AWS Glue quando si utilizza Python 3.9.

**`--scriptLocation`**

La posizione di Amazon Simple Storage Service (Amazon S3) in cui si trova lo script ETL (nel formato `s3://path/to/my/script.py`). Questo parametro sovrascrive un percorso script impostato nell'oggetto `JobCommand`.

**`--spark-event-logs-path`**

Specifica un percorso Amazon S3. Quando si utilizza la funzionalità di monitoraggio dell'interfaccia utente di Spark, AWS Glue scarica i log degli eventi Spark su questo percorso Amazon S3 ogni 30 secondi in un bucket utilizzabile come directory temporanea per la memorizzazione di eventi dell'interfaccia utente Spark.

**`--TempDir`**

Specifica un percorso Amazon S3 a un bucket utilizzabile come directory temporanea per il processo.

Ad esempio, per impostare una directory temporanea, passa l'argomento seguente.

```
'--TempDir': 's3-path-to-directory'
```

> **Note**
>
> AWS Glue crea un bucket temporaneo per i lavori se un bucket non esiste già in una regione. Questo bucket potrebbe consentire l'accesso pubblico. Puoi modificare il bucket in Amazon S3 per impostare il blocco dell'accesso pubblico oppure eliminare il bucket in un secondo momento dopo che tutti i processi in quella regione sono stati completati.

**`--use-postgres-driver`**

Impostando questo valore su `true`, assegna la priorità al driver JDBC Postgres nella variabile classpath per evitare un conflitto con il driver JDBC Amazon Redshift. Questa opzione è disponibile solo nella versione 2.0 di AWS Glue.

**`--user-jars-first`**

Impostando questo valore su `true`, dà la priorità ai file JAR aggiuntivi del cliente nella variabile classpath. Questa opzione è disponibile solo nella versione 2.0 o successiva di AWS Glue.

**`--conf`**

Controlla i parametri di configurazione di Spark. È per casi d'uso avanzati.

**`--encryption-type`**

Parametro legacy. Il comportamento corrispondente deve essere configurato utilizzando le configurazioni di sicurezza. Per ulteriori informazioni sulle configurazioni di sicurezza, consulta la pagina the section called "Crittografia dei dati scritti da AWS Glue".

AWS Glue utilizza internamente i seguenti argomenti e non dovresti mai usarli:

- `--debug`— Interno a AWS Glue. Non impostare.
- `--mode`— Interno a AWS Glue. Non impostare.
- `--JOB_NAME`— Interno a AWS Glue. Non impostare.
- `--endpoint`— Interno a AWS Glue. Non impostare.

AWS Glue supporta il bootstrap di un ambiente con il `site` modulo Python utilizzato `sitecustomize` per eseguire personalizzazioni specifiche del sito. L'avvio delle proprie funzioni di inizializzazione è consigliato solo per casi d'uso avanzati ed è supportato al massimo su Glue 4.0. AWS

Il prefisso della variabile di ambiente, `GLUE_CUSTOMER`, è riservato all'uso da parte dei clienti.

## AWS Glue Spark e lavori PySpark

AWS Glue supporta Spark e jobs. PySpark Un job Spark viene eseguito in un ambiente Apache Spark gestito da. AWS Glue Elabora i dati in batch. Un processo ETL di streaming è simile a un processo Spark, ad eccezione del fatto che esegue ETL sui flussi di dati. Esso utilizza il framework Apache Spark Structured Streaming. Alcune caratteristiche dei processi Spark non sono disponibili per i processi ETL in streaming.

Le seguenti sezioni forniscono informazioni su AWS Glue Spark e sui job. PySpark

Argomenti

- Configurazione delle proprietà dei job per i job Spark in AWS Glue
- Modifica degli script Spark nella console AWS Glue
- Processi (legacy)
- Monitoraggio dei dati elaborati mediante segnalibri di processo
- Memorizzazione dei dati shuffle di Spark
- Monitoraggio dei processi Spark AWS Glue
- Risoluzione dei problemi di intelligenza artificiale generativa per Apache Spark in Glue AWS
- Utilizzo di viste materializzate con AWS Glue

### Configurazione delle proprietà dei job per i job Spark in AWS Glue

Quando definisci il tuo job sulla AWS Glue console, fornisci i valori delle proprietà per controllare l'ambiente AWS Glue di runtime.

Definire le proprietà di processo per i processi Spark

L'elenco seguente descrive le proprietà di un processo Spark. Per le proprietà di un processo shell di Python, consulta Definire le proprietà del processo per i processi shell di Python. Per le proprietà di

un processo ETL di streaming, vedere the section called "Definizione delle proprietà di processo per un processo di streaming ETL".

Le proprietà sono elencate nell'ordine in cui appaiono nella procedura guidata Aggiungi lavoro sulla AWS Glue console.

Nome

Stringa UTF-8 con un massimo di 255 caratteri.

Description

Fornisci una descrizione facoltativa di un massimo di 2.048 caratteri.

Ruolo IAM

Specifica il ruolo IAM; utilizzato per definire l'autorizzazione alle risorse utilizzate per eseguire il processo e accedere agli archivi dati. Per ulteriori informazioni sulle autorizzazioni per l'esecuzione di lavori in AWS Glue, vedere. Gestione delle identità e degli accessi per AWS Glue

Tipo

Il tipo di processo ETL. Viene impostato automaticamente in base al tipo di origini dati selezionate.

- Spark esegue uno script ETL di Apache Spark con il comando del processo denominato `glueetl`.
- Spark Streaming esegue uno script ETL di streaming Apache Spark con il comando del processo denominato `gluestreaming`. Per ulteriori informazioni, consulta the section called "Aggiunta di processi di streaming ETL".
- Shell di Python esegue uno script Python con il comando di processo denominato `pythonshell`. Per ulteriori informazioni, consulta Configurazione delle proprietà del processo per i processi shell di Python in AWS Glue.

Versione AWS Glue

AWS Glue version determina le versioni di Apache Spark e Python disponibili per il job, come specificato nella tabella seguente.

| AWS Glue versione | Versioni Spark e Python supportate |
| --- | --- |
| 5.1 | • Spark 3.5.6 |

| AWS Glue versione | Versioni Spark e Python supportate |
| --- | --- |
| | • Python 3.11 |
| 5.0 | • Spark 3.5.4<br>• Python 3.11 |
| 4.0 | • Spark 3.3.0<br>• Python 3.10 |
| 3.0 | • Spark 3.1.1<br>• Python 3.7 |

Lavori creati senza specificare una AWS Glue versione predefinita 5.1. AWS Glue

Lingua

Il codice nello script ETL definisce la logica del processo. Lo script può essere codificato in Python o Scala. È possibile scegliere se lo script eseguito dal job viene generato AWS Glue o fornito dall'utente. Puoi fornire il nome e la posizione dello script in Amazon Simple Storage Service (Amazon S3). Conferma che non esiste un file con lo stesso nome della directory di script nel percorso. Per ulteriori informazioni sull'uso degli script, consulta AWS Glue guida alla programmazione.

Tipo di lavoratore

Sono disponibili i seguenti tipi di worker:

Le risorse disponibili per i AWS Glue lavoratori vengono misurate in DPUs. Una DPU è una misura relativa della potenza di elaborazione costituita da 4 V CPUs di capacità di elaborazione e 16 GB di memoria.

- **`G.025X`**: quando si sceglie questo tipo, si fornisce anche un valore per Numero di worker. Ogni worker esegue il mapping a 0,25 DPU (2 vCPUs, 4 GB di memoria) con disco da 84 GB (circa 34 GB liberi). Consigliamo questo tipo di worker per i processi di streaming a basso volume. Questo tipo di worker è disponibile solo per i lavori di streaming della AWS Glue versione 3.0 o successiva.
- **`G.1X`**: quando si sceglie questo tipo, si fornisce anche un valore per Numero di worker. Ogni lavoratore esegue il mapping su 1 DPU (4 vCPUs, 16 GB di memoria) con disco da 94 GB (circa 44 GB gratuiti). Questi tipi di worker sono raccomandati per carichi di lavoro come

trasformazioni di dati, join e query, in quanto offrono un modo scalabile ed economico per eseguire la maggior parte dei processi.

- **`G.2X`**: quando si sceglie questo tipo, si fornisce anche un valore per Numero di worker. Ogni worker esegue il mapping su 2 DPU (8 vCPUs, 32 GB di memoria) con disco da 138 GB (circa 78 GB gratuiti). Questi tipi di worker sono raccomandati per carichi di lavoro come trasformazioni di dati, join e query, in quanto offrono un modo scalabile ed economico per eseguire la maggior parte dei processi.
- **`G.4X`**: quando si sceglie questo tipo, si fornisce anche un valore per Numero di worker. Ogni worker esegue il mapping a 4 DPU (16 vCPUs, 64 GB di memoria) con disco da 256 GB (circa 230 GB gratuiti). Questi tipi di worker sono raccomandati per i processi i cui carichi di lavoro contengono trasformazioni, aggregazioni, join e query con i requisiti più elevati.
- **`G.8X`**: quando si sceglie questo tipo, si fornisce anche un valore per Numero di worker. Ogni worker esegue il mapping a 8 DPU (32 vCPUs, 128 GB di memoria) con disco da 512 GB (circa 485 GB gratuiti). Questi tipi di worker sono raccomandati per i processi i cui carichi di lavoro contengono trasformazioni, aggregazioni, join e query con i requisiti più elevati.
- **`G.12X`**: quando si sceglie questo tipo, si fornisce anche un valore per Numero di worker. Ogni worker esegue il mapping a 12 DPU (48 vCPUs, 192 GB di memoria) con disco da 768 GB (circa 741 GB liberi). Consigliamo questo tipo di worker per processi con carichi di lavoro molto grandi e che richiedono molte risorse con una capacità di calcolo significativa.
- **`G.16X`**: quando si sceglie questo tipo, si fornisce anche un valore per Numero di worker. Ogni worker esegue il mapping a 16 DPU (64 vCPUs, 256 GB di memoria) con disco da 1024 GB (circa 996 GB gratuiti). Consigliamo questo tipo di worker per processi con i carichi di lavoro più grandi e che richiedono più risorse con una capacità di calcolo massima.
- **`R.1X`**: quando si sceglie questo tipo, si fornisce anche un valore per Numero di worker. Ogni worker esegue la mappatura su 1 DPU con configurazione ottimizzata per la memoria. Consigliamo questo tipo di worker per carichi di lavoro che richiedono molta memoria e che presentano spesso errori o richiedono rapporti elevati. out-of-memory memory-to-CPU
- **`R.2X`**: quando si sceglie questo tipo, si fornisce anche un valore per Numero di worker. Ogni worker esegue la mappatura su 2 DPU con configurazione ottimizzata per la memoria. Consigliamo questo tipo di worker per carichi di lavoro che richiedono molta memoria e che presentano spesso errori o richiedono rapporti elevati. out-of-memory memory-to-CPU
- **`R.4X`**: quando si sceglie questo tipo, si fornisce anche un valore per Numero di worker. Ogni worker esegue la mappatura su 4 DPU con configurazione ottimizzata per la memoria. Consigliamo questo tipo di worker per carichi di lavoro che richiedono grandi quantità di

memoria e che presentano spesso errori o richiedono rapporti elevati. out-of-memory memory-to-CPU

- **`R.8X`**: quando si sceglie questo tipo, si fornisce anche un valore per Numero di worker. Ogni worker esegue la mappatura su 8 DPU con configurazione ottimizzata per la memoria. Consigliamo questo tipo di worker per carichi di lavoro di grandi dimensioni che richiedono un uso intensivo della memoria e che presentano spesso errori o richiedono rapporti elevati. out-of-memory memory-to-CPU

Specifiche del tipo di worker

La tabella seguente fornisce le specifiche dettagliate per tutti i tipi di worker G disponibili:

Specifiche del tipo worker G

| Tipo di worker | DPU per nodo | VPCU | Memoria (GB) | Disco (GB) | Spazio libero su disco (GB) | Esecutori Spark per nodo |
|---|---|---|---|---|---|---|
| G.025X | 0.25 | 2 | 4 | 84 | 34 | 1 |
| G.1X | 1 | 4 | 16 | 94 | 44 | 1 |
| G.2X | 2 | 8 | 32 | 138 | 78 | 1 |
| G.4X | 4 | 16 | 64 | 256 | 230 | 1 |
| G.8X | 8 | 32 | 128 | 512 | 485 | 1 |
| G.12X | 12 | 48 | 192 | 768 | 741 | 1 |
| G.16X | 16 | 64 | 256 | 1024 | 996 | 1 |

Importante: i tipi di worker G.12X e G.16X, così come tutti i tipi di worker R (da R.1X a R.8X), hanno una latenza di avvio più elevata.

Ti viene addebitata una tariffa oraria basata sul numero di job ETL DPUs utilizzati per eseguire i tuoi job ETL. Per ulteriori informazioni, consultare la pagina dei prezzi di AWS Glue.

Per la AWS Glue versione 1.0 o precedente, quando si configura un lavoro utilizzando la console e si specifica un tipo di lavoratore su Standard, viene impostata la capacità massima e il numero

di lavoratori diventa il valore di Capacità massima - 1. Se si utilizza AWS Command Line Interface (AWS CLI) o AWS SDK, è possibile specificare il parametro Capacità massima oppure specificare sia il tipo di lavoratore che il numero di lavoratori.

Per i lavori della AWS Glue versione 2.0 o successiva, non è possibile specificare una capacità massima. È invece necessario specificare un Worker type (Tipo di worker) e il Number of workers (Numero di worker).

**`G.4X`**e i tipi di **`G.8X`** lavoratori sono disponibili solo per i lavori Spark ETL AWS Glue versione 3.0 o successiva AWS nelle seguenti regioni: Stati Uniti orientali (Ohio), Stati Uniti orientali (Virginia settentrionale), Stati Uniti occidentali (California settentrionale), Stati Uniti occidentali (Oregon), Asia Pacifico (Mumbai), Asia Pacifico (Seoul), Asia Pacifico (Singapore), Asia Pacifico (Sydney), Asia Pacifico (Tokyo), Canada (Centrale), Europa (Francoforte), Europa (Irlanda), Europa (Londra), Europa (Spagna), Europa (Stoccolma) e Sud America (San Paolo).

**`G.12X`****`G.16X`**, e **`R.1X`** tramite i tipi di **`R.8X`** worker sono disponibili solo per i job Spark ETL AWS Glue versione 4.0 o successiva AWS nelle seguenti regioni: Stati Uniti orientali (Virginia settentrionale), Stati Uniti occidentali (Oregon), Stati Uniti orientali (Ohio), Europa (Irlanda) ed Europa (Francoforte). In futuro verranno supportate regioni aggiuntive.

Numero di worker richiesto

Per la maggior parte dei tipi di worker è necessario specificare il numero di worker allocati quando il processo viene eseguito.

Segnalibro Job

Specificate in che modo AWS Glue i processi definiscono le informazioni durante l'esecuzione del processo. Puoi ricordare di aver già elaborato i dati, aggiornato le informazioni sullo stato o ignorato le informazioni sullo stato. Per ulteriori informazioni, consulta the section called "Monitoraggio dei dati elaborati mediante segnalibri di processo".

Coda per l'esecuzione dei processi

Specifica se le esecuzioni dei processi vengono messe in coda per essere eseguite in un secondo momento quando le quote di servizio non sono sufficienti.

Se selezionato, la coda per l'esecuzione dei processi viene abilitata. Se non è compilato, le esecuzioni del processo non saranno prese in considerazione per la coda.

Se questa impostazione non corrisponde al set di valori impostato nell'esecuzione del processo, verrà utilizzato il valore presente nel campo di esecuzione del processo.

Esecuzione flessibile

Quando configuri un lavoro utilizzando AWS Studio o l'API, puoi specificare una classe di esecuzione del lavoro standard o flessibile. I tuoi processo possono avere diversi gradi di priorità e sensibilità temporale. La classe di esecuzione standard è ideale per carichi di lavoro sensibili al tempo che richiedono un avvio rapido dei processi e risorse dedicate.

La classe di esecuzione flessibile è adatta per processi non urgenti come i processi di pre-produzione, test e caricamenti di dati una tantum. Le esecuzioni di job flessibili sono supportate per i lavori che utilizzano la AWS Glue versione 3.0 o successiva `G.1X` e/o per i tipi di `G.2X` worker. I nuovi tipi di worker (`G.12X`, `G.16X` e da `R.1X` a `R.8X`) non supportano l'esecuzione flessibile.

Le esecuzioni dei processi flessibili vengono fatturate in base al numero di worker che vengono eseguiti alla volta. Il numero di worker può essere aggiunto o rimosso per un'esecuzione di lavoro flessibile in esecuzione. Invece di fatturare come semplice calcolo di `Max Capacity * Execution Time`, ogni worker contribuirà per il tempo che è stato eseguito durante l'esecuzione del processo. La fattura è la somma di (`Number of DPUs per worker * time each worker ran`).

Per ulteriori informazioni, consulta il pannello di aiuto in AWS Studio oppure Jobs eEsecuzioni di processi.

Numero di tentativi

Specificate il numero di volte, da 0 a 10, che AWS Glue devono riavviare automaticamente il processo in caso di errore. I processi che raggiungono il limite di timeout non vengono riavviati.

Timeout dei processi

Imposta il tempo di esecuzione massimo in minuti. Il valore massimo è 7 giorni o 10.080 minuti. In caso contrario, i processi genereranno un'eccezione.

Quando il valore viene lasciato vuoto, il timeout è predefinito a 2.880 minuti per Glue versione 4.0 e precedenti o 480 minuti per Glue versione 5.0 e successive.

Tutti i AWS Glue lavori esistenti con un valore di timeout superiore a 7 giorni verranno impostati come impostazione predefinita su 7 giorni. Ad esempio, se un processo in batch ha impostato un timeout di 20 giorni, sarà interrotto al settimo giorno.

> **Best practice per i timeout dei processi**
>
> I processi vengono fatturati in base al tempo di esecuzione. Per evitare addebiti imprevisti, configura i valori di timeout appropriati per il tempo di esecuzione previsto del processo.

Proprietà avanzate

Nome del file dello script

Un nome univoco dello script per il processo. Non può essere denominato Processo senza titolo.

Percorso dello script

La posizione Amazon S3 dello script. Il percorso deve essere nel formato `s3://bucket/prefix/path/`. Deve terminare con una barra (/) e non includere alcun file.

Parametri del processo

Attiva o disattiva la creazione di CloudWatch metriche Amazon durante l'esecuzione di questo processo. Per visualizzare i dati di profiling, è necessario abilitare questa opzione. Per ulteriori informazioni su come attivare e visualizzare i parametri, consulta Monitoraggio e debug dei processi.

Metriche di osservabilità dei processi

Attiva la creazione di CloudWatch parametri di osservabilità aggiuntivi durante l'esecuzione di questo processo. Per ulteriori informazioni, consulta the section called "Monitoraggio con parametri AWS Glue di osservabilità".

Registrazione continua

Attiva la registrazione continua su Amazon CloudWatch. Se questa opzione non è abilitata, i registri sono disponibili solo dopo il completamento del processo. Per ulteriori informazioni, consulta the section called "Registrazione dei lavori AWS Glue".

Interfaccia utente Spark

Attiva l'uso dell'interfaccia utente di Spark per monitorare questo processo. Per ulteriori informazioni, consulta Abilitazione dell'interfaccia utente Web di Apache Spark per processi AWS Glue.

Percorso dei log dell'interfaccia utente di Spark

Il percorso per scrivere i log quando l'interfaccia utente di Spark è abilitata.

Configurazione della registrazione e del monitoraggio dell'interfaccia utente di Spark

Scegli una delle seguenti opzioni:

- Standard: scrivi i log usando l'ID di esecuzione del AWS Glue processo come nome del file. Attiva il monitoraggio dell'interfaccia utente Spark nella console. AWS Glue
- Legacy: scrivi i log usando "spark-application- {timestamp}" come nome del file. Non attivare il monitoraggio dell'interfaccia utente di Spark.
- Standard e legacy: scrivi i log sia nelle posizioni standard che in quelle legacy. Attiva il monitoraggio dell'interfaccia utente Spark nella AWS Glue console.

Simultaneità massima

Imposta il numero massimo di esecuzioni simultanee consentite per il processo. Il valore di default è 1. Viene restituito un errore al raggiungimento della soglia. Il valore massimo che è possibile specificare è controllato da un limite di servizio. Ad esempio, se un'esecuzione di un processo precedente non è ancora terminata quando una nuova istanza viene avviata, è possibile restituire un errore per evitare che due istanze dello stesso processo vengano eseguite simultaneamente.

Percorso temporaneo

Fornisci la posizione di una directory di lavoro in Amazon S3 in cui vengono scritti i risultati intermedi temporanei durante l' AWS Glue esecuzione dello script. Conferma che non esiste un file con lo stesso nome della directory temporanea nel percorso. Questa directory viene utilizzata durante la AWS Glue lettura e la scrittura su Amazon Redshift e per determinate AWS Glue trasformazioni.

> **Note**
>
> AWS Glue crea un bucket temporaneo per i lavori se un bucket non esiste già in una regione. Questo bucket potrebbe consentire l'accesso pubblico. Puoi modificare il bucket in Amazon S3 per impostare il blocco dell'accesso pubblico oppure eliminare il bucket in un secondo momento dopo che tutti i processi in quella regione sono stati completati.

Soglia notifica di ritardo (minuti)

Imposta la soglia (in minuti) prima di inviare una notifica di ritardo. Puoi impostare questa soglia per inviare notifiche quando l'esecuzione di un processo `RUNNING`, `STARTING` o `STOPPING` impiega di più rispetto alla quantità di minuti attesa.

Configurazione di sicurezza

Scegliere una configurazione di sicurezza dall'elenco. Una configurazione di sicurezza specifica come vengono crittografati i dati del target Amazon S3: nessuna crittografia, crittografia lato server con chiavi gestite da AWS KMS(SSE-KMS) o chiavi di crittografia gestite da Amazon S3 (SSE-S3).

Crittografia lato server

Se selezioni questa opzione, quando il processo ETL scrive in Amazon S3, i dati vengono crittografati quando sono inattivi tramite crittografia SSE-S3. Vengono crittografati sia i dati di destinazione Amazon S3 sia tutti gli altri dati scritti in una directory temporanea Amazon S3. Questa opzione viene passata come parametro del processo. Per ulteriori informazioni, consulta Protezione dei dati mediante la crittografia lato server con chiavi di crittografia gestite da Amazon S3 (SSE-S3) nella Guida per l'utente di Amazon Simple Storage Service.

> **Important**
>
> Questa opzione viene ignorata se viene specificata una configurazione di protezione.

Opzione per l'uso del catalogo dati di Glue come metastore Hive

Seleziona di utilizzare AWS Glue Data Catalog come metastore Hive. Il ruolo IAM utilizzato per il processo deve disporre dell'autorizzazione `glue:CreateDatabase`. Viene creato un database chiamato "default" nel catalogo dati, nel caso non fosse già presente.

Connessioni

Scegli una configurazione VPC per accedere alle origini dati di Amazon S3 situate nel cloud privato virtuale (VPC). Puoi creare e gestire una connessione di rete in AWS Glue. Per ulteriori informazioni, consulta Connessione ai dati.

Libraries (Librerie)

Percorso della libreria Python, percorso dipendente e JARs percorso dei file di riferimento

Specificare queste opzioni se lo script le richiede. Puoi definire percorsi separati da virgole Amazon S3 per queste opzioni quando definisci il processo. Puoi sostituire tali percorsi quando esegui il processo. Per ulteriori informazioni, consulta Fornire i propri script personalizzati.

Parametri Job

Un insieme di coppie chiave-valore che vengono passate come parametri denominati allo script. Si tratta di valori predefiniti che vengono utilizzati quando lo script viene eseguito, ma è possibile ignorarli nei trigger o quando si esegue il processo. È necessario prefissare il nome della chiave con `--`; ad esempio: `--myKey`. I parametri del lavoro vengono passati come mappa quando si utilizza AWS Command Line Interface.

Per ulteriori esempi, vedere i parametri Python in Passaggio di parametri Python in AWS Glue e accesso ai parametri.

Tag

Il tag si applica al processo tramite una Tag key (Chiave tag) e un Tag value (Valore tag) facoltativo. Una volta create, le chiavi di tag sono di sola lettura. Usa i tag su alcune risorse per facilitarne l'organizzazione e l'individuazione. Per ulteriori informazioni, consulta AWS tag in AWS Glue.

Restrizioni per i processi che accedono alle tabelle gestite da Lake Formation

Tieni presente le seguenti note e restrizioni quando crei lavori che leggono o scrivono su tabelle gestite da AWS Lake Formation:

- Le seguenti caratteristiche non sono supportate nei processi che accedono alle tabelle con filtri a livello di cella:
  - Segnalibri di processo ed esecuzione limitata
  - Predicati pushdown
  - Predicati delle partizioni dei cataloghi lato server
  - enableUpdateCatalog

## Modifica degli script Spark nella console AWS Glue

Uno script contiene il codice che estrae i dati dalle fonti, li trasforma e li carica in obiettivi. AWS Glue esegue uno script quando avvia un processo.

Gli script ETL AWS Glue possono essere codificati in Python o Scala. Gli script Python utilizzano un linguaggio che è un'estensione del dialetto PySpark Python per i lavori di estrazione, trasformazione e caricamento (ETL). Lo script contiene costrutti estesi per gestire le trasformazioni ETL. Quando si genera automaticamente la logica del codice sorgente per un processo, viene creato lo script. Puoi modificare questo script oppure puoi fornire il tuo script per elaborare il lavoro ETL.

Per informazioni sulla definizione e sulla modifica di script in AWS Glue, consulta la pagina AWS Glue guida alla programmazione.

Librerie o file aggiuntivi

Se lo script richiede librerie o file aggiuntivi, puoi specificarli come segue:

Percorso della libreria Python

Percorsi Amazon Simple Storage Service (Amazon S3) separati da virgole per le librerie Python richieste dallo script.

> **Note**
>
> È possibile utilizzare solo librerie Python pure. Le librerie che si basano sulle estensioni C, come la libreria di analisi dati Python pandas, non sono ancora supportate.

Percorso dei vasi dipendenti

Percorsi Amazon S3 separati da virgole dei file JAR richiesti dallo script.

> **Note**
>
> Al momento possono essere utilizzate solo le librerie pure Java o Scala (2.11).

Percorso dei file di riferimento

Percorsi Amazon S3 separati da virgole di file aggiuntivi (ad esempio i file di configurazione) richiesti dallo script.

## Processi (legacy)

Uno script contiene il codice che estrae, trasforma e carica il lavoro (ETL). Puoi fornire uno script personalizzato oppure AWS Glue può generare uno script seguendo le tue indicazioni. Per informazioni su come creare gli script, consulta Fornire i propri script personalizzati.

Puoi modificare uno script nella console AWS Glue. Quando modifichi uno script, puoi aggiungere origini, destinazioni e trasformazioni.

Per modificare uno script

1. Accedi a Console di gestione AWS e apri la AWS Glue console all'indirizzo https://console.aws.amazon.com/glue/. Selezionare Processi scheda.
2. Scegli un processo nell'elenco, quindi Action (Operazione), Edit script (Modifica script) per aprire l'editor di script.

   Puoi inoltre accedere all'editor di script dalla pagina dei dettagli del processo. Scegli la scheda Script, quindi scegli Edit script (Modifica script).

Editor di script

L'editor di script AWS Glue permette di inserire, modificare ed eliminare origini, destinazioni e trasformazioni nello script. L'editor di script visualizza sia lo script sia un diagramma per aiutarti a visualizzare il flusso di dati.

Per creare un diagramma per lo script, scegli Generate diagram (Genera diagramma). AWS Glue usa righe di annotazione nello script che iniziano con # # per visualizzare il diagramma. Per rappresentare correttamente lo script nel diagramma, è necessario mantenere sincronizzati i parametri nelle annotazioni e i parametri nel codice Apache Spark.

L'editor di script ti consente di aggiungere modelli di codice ovunque il cursore sia posizionato nello script. Nella parte superiore dell'editor, sceglie tra le seguenti opzioni:

- Per aggiungere una tabella di origine allo script, scegli Source (Origine).
- Per aggiungere una tabella di destinazione allo script, scegli Target (Destinazione).
- Per aggiungere una posizione di destinazione allo script, scegli Target location (Posizione di destinazione).

- Per aggiungere una trasformazione allo script, scegli Transform (Trasformazione). Per informazioni sulle funzioni richiamate nel tuo script, consulta Programma gli script ETL di AWS Glue in PySpark.
- Per aggiungere una trasformazione Spigot allo script, scegli Spigot.

Nel codice inserito, modifica i `parameters` nelle annotazioni e nel codice Apache Spark. Ad esempio, se aggiungi una trasformazione Spigot, verifica che `path` sia sostituito sia nella riga di annotazione `@args` sia nella riga di codice `output`.

La scheda Logs (Log) mostra i log che sono associati al tuo processo durante l'esecuzione. Vengono visualizzate le 1.000 righe più recenti.

La scheda Schema mostra lo schema delle origini e delle destinazioni selezionate, quando disponibili nel catalogo dati.

## Monitoraggio dei dati elaborati mediante segnalibri di processo

AWS Glue monitora i dati già elaborati durante una precedente esecuzione di un processo ETL conservando le informazioni sullo stato dall'esecuzione del processo. Questa informazione sullo stato persistente è chiamato segnalibro di processo. I segnalibri del processo aiutano AWS Glue a mantenere le informazioni sullo stato e prevenire la rielaborazione dei dati precedenti. Con i segnalibri del processo, è possibile elaborare nuovi dati quando vengono rieseguiti su un intervallo pianificato. Un segnalibro di un processo è costituito dagli stati dei vari elementi dei processi, come le origini, le trasformazioni e le destinazioni. Ad esempio, il processo ETL potrebbe leggere nuove partizioni in un file Amazon S3. AWS Glue tiene traccia delle partizioni che sono state elaborate correttamente dal processo al fine di evitare elaborazioni ripetute o dati duplicati nel datastore di destinazione del processo.

I segnalibri di processo sono implementati per le origini dati JDBC, la trasformazione Relationalize e alcune origini Amazon Simple Storage Service (Amazon S3). La tabella seguente elenca i formati di origine Amazon S3 supportati da AWS Glue per i segnalibri del processo.

| Versione AWS Glue | Formati Amazon S3 di origine |
|---|---|
| Versione 0.9 | JSON, CSV, Apache Avro, XML |
| Versione 1.0 e successive. | JSON, CSV, Apache Avro, XML, Parquet, ORC |

Per ulteriori informazioni sulle versioni AWS Glue, consulta Definire le proprietà di processo per i processi Spark.

La funzionalità dei segnalibri di lavoro ha funzionalità aggiuntive quando si accede tramite AWS Glue script. Quando si sfoglia lo script generato, è possibile visualizzare contesti di trasformazione correlati a questa funzionalità. Per ulteriori informazioni, consulta the section called "Utilizzo di segnalibri di processo".

Argomenti

- Utilizzo di segnalibri di processo in AWS Glue
- Dettagli operativi della funzione dei segnalibri di processo

## Utilizzo di segnalibri di processo in AWS Glue

L'opzione di segnalibro di processo viene passata come parametro all'avvio del processo. La tabella seguente descrive le opzioni per impostare i segnalibri di processo nella console AWS Glue.

| Segnalibro Job | Description |
| --- | --- |
| Attiva | Provoca l'aggiornamento dello stato del processo dopo un'esecuzione per tenere traccia dei dati elaborati in precedenza. Se il processo ha un'origine con un supporto segnalibro del processo, questo tiene traccia dei dati elaborati e, quando un processo viene eseguito, elabora i nuovi dati a partire dall'ultimo punto di controllo. |
| Disabilita | I segnalibri del processo non vengono utilizzati e il processo elabora sempre l'intero set di dati. L'utente è responsabile della gestione dell'outp ut delle esecuzioni di job precedenti. Questa è l'impostazione predefinita. |
| Metti in pausa | Elabora i dati incrementali dall'ultima esecuzione riuscita o i dati nell'inte rvallo identificato dalle seguenti opzioni secondarie, senza aggiornar e lo stato dell'ultimo segnalibro. L'utente è responsabile della gestione dell'output delle esecuzioni di job precedenti. Le due opzioni secondarie sono:<br>• job-bookmark-from<from-value>è l'ID di esecuzione che rappresen ta tutto l'input elaborato fino all'ultima esecuzione riuscita precedente |

| Segnalibro Job | Description |
| --- | --- |
| | e include l'ID di esecuzione specificato. L'input corrispondente viene ignorato.<br>• job-bookmark-to<to-value>è l'ID di esecuzione che rappresenta tutto l'input elaborato fino all'ultima esecuzione riuscita precedente e include l'ID di esecuzione specificato. L'input corrispondente escluso l'input identificato da <from-value> viene elaborato dal processo. Qualsiasi input successivo a questo è escluso anche dall'elaborazione.<br><br>Lo stato dei segnalibri di processo non viene aggiornato quando viene specificato questo set di opzioni.<br><br>Le opzioni secondarie sono facoltative, tuttavia se utilizzate, entrambe le opzioni secondarie devono essere specificate. |

Per informazioni dettagliate sui parametri passati a un processo nella riga di comando e, in particolare, sui segnalibri di lavoro, consulta Utilizzo dei parametri del lavoro nei lavori AWS Glue.

Per le origini di input Amazon S3, i segnalibri del processo AWS Glue controllano l'ora dell'ultima modifica degli oggetti per individuare gli oggetti da rielaborare. Se i dati dell'origine di input sono stati modificati dall'ultima esecuzione del processo, i file vengono rielaborati alla nuova esecuzione del processo.

Per le origini JDBC, si applicano le seguenti regole:

- Per ogni tabella, AWS Glue utilizza una o più colonne come chiavi di segnalibro per determinare i dati nuovi ed elaborati. Le chiavi dei segnalibri si combinano per formare un'unica chiave composta.
- Per impostazione predefinita, AWS Glue utilizza la chiave primaria come chiave di segnalibro, a condizione che sia in ordine sequenziale (senza spazi vuoti) crescente o decrescente.
- È possibile specificare le colonne da utilizzare come chiavi di segnalibro nello script AWS Glue . Per ulteriori informazioni sull'utilizzo dei segnalibri Job negli AWS Glue script, vedere. the section called "Utilizzo di segnalibri di processo"
- AWS Glue non supporta l'utilizzo di colonne che fanno distinzione tra maiuscole e minuscole come chiavi di segnalibro del processo.

È possibile ripristinare i segnalibri di processo per i processi ETL di AWS Glue Spark a qualsiasi esecuzione precedente del processo. È possibile supportare meglio gli scenari di recupero dei dati ripristinando i segnalibri di lavoro a qualsiasi esecuzione di lavoro precedente, con conseguente esecuzione del lavoro di rielaborazione dei dati solo relativi all'esecuzione del lavoro con segnalibro.

Se hai intenzione di rielaborare tutti i dati utilizzando lo stesso processo, reimposta il segnalibro del processo. Per reimpostare lo stato del segnalibro del processo, utilizza la console AWS Glue, l'operazione API ResetJobBookmark azione (Python: reset_job_bookmark) o la AWS CLI. Ad esempio, inserisci il seguente comando utilizzando AWS CLI:

```
aws glue reset-job-bookmark --job-name my-job-name
```

Quando si riavvolge o si reimposta un segnalibro, AWS Glue non pulisce i file di destinazione, perché potrebbero esserci più destinazioni e le destinazioni non sono monitorate con i segnalibri di processo. Solo i file di origine vengono tracciati con i segnalibri di processo. È possibile creare diverse destinazioni dell'output durante il riavvolgimento e la rielaborazione dei file di origine per evitare la duplicazione dei dati nell'output.

AWS Glue tiene traccia dei segnalibri dei processi per ogni processo. All'eliminazione del processo, seguirà l'eliminazione del segnalibro di processo.

In alcuni casi, potresti aver abilitato i segnalibri del processo AWS Glue, ma il processo ETL rielabora i dati già elaborati in una esecuzione precedente. Per ulteriori informazioni sulla risoluzione delle cause comuni di questo errore, consulta Risoluzione degli errori di configurazione comuni di Glue.

Dettagli operativi della funzione dei segnalibri di processo

Questa sezione descrive ulteriori dettagli operativi utilizzando i segnalibri del processo.

I segnalibri del processo archiviano gli stati per un processo. Ogni istanza dello stato viene contrassegnata da un nome processo e da un numero di versione. Quando uno script richiama `job.init`, ne recupera lo stato e ottiene sempre la versione più recente. Uno stato contiene più elementi dello stato, specifici per ogni origine, trasformazione e istanza sink nello script. Gli elementi dello stato sono identificati da un contesto di trasformazione collegato all'elemento corrispondente (origine, trasformazione o sink) nello script. Gli elementi dello stato vengono salvati in modo atomico quando `job.commit` viene richiamato dallo script dell'utente. Lo script ottiene dagli argomenti il nome del processo e l'opzione di controllo per i segnalibri del processo.

Gli elementi dello stato nel segnalibro del processo sono origine, trasformazione oppure dati specifici del sink. Ad esempio, supponiamo che si desideri leggere i dati incrementali da una posizione Amazon S3 costantemente scritta da un processo upstream o da un processo. In questo caso, lo script deve determinare quanto è stato elaborato fino a ora. L'implementazione del segnalibro del processo per l'origine Amazon S3 salva le informazioni in modo che, quando il processo viene eseguito nuovamente, è possibile filtrare solo i nuovi oggetti utilizzando le informazioni salvate e ricalcolare lo stato per l'esecuzione successiva del processo. Un timestamp viene utilizzato per filtrare i nuovi file.

In aggiunta agli elementi dello stato, i segnalibri del processo dispongono di un numero di esecuzione, un numero di tentativo e un numero di versione. Il numero di esecuzione monitora l'esecuzione del processo e il numero di tentativo registra i tentativi di esecuzione di un processo. Il numero di esecuzione di un processo è un numero che aumenta in maniera monotona e che subisce incrementi a ogni esecuzione riuscita. Il numero di tentativi monitora i tentativi per ogni esecuzione e viene incrementato solo in caso di un'esecuzione dopo un tentativo non riuscito. Il numero di versione aumenta in maniera monotona e monitora gli aggiornamenti a un segnalibro del processo.

Nel database dei AWS Glue servizi, gli stati dei segnalibri per tutte le trasformazioni vengono memorizzati insieme come coppie chiave-valore:

```
{
  "job_name" : ...,
  "run_id": ...,
  "run_number": ..,
  "attempt_number": ...
  "states": {
    "transformation_ctx1" : {
      bookmark_state1
    },
    "transformation_ctx2" : {
      bookmark_state2
    }
  }
}
```

Best practice

Di seguito sono indicate le best practice per l'utilizzo dei segnalibri di processo.

- Non modificare la proprietà dell'origine dati con il segnalibro abilitato. Ad esempio, esiste un datasource0 che punta a un percorso di input di Amazon S3 A e il lavoro è stato letto da un'origine

che è in esecuzione per diversi round con il segnalibro abilitato. Se modifichi il percorso di input di datasource0 nel percorso Amazon S3 B senza modificare il`transformation_ctx`, il AWS Glue job utilizzerà il vecchio stato del segnalibro memorizzato. Ciò comporterà la mancanza o il salto di file nel percorso di input B, come si suppone AWS Glue che tali file siano stati elaborati in esecuzioni precedenti.

- Utilizzare una tabella di catalogo con segnalibri per una migliore gestione delle partizioni. I segnalibri funzionano sia per le origini dati del Catalogo dati che per le opzioni. Tuttavia, è difficile creare remove/add nuove partizioni con l'approccio from options. L'utilizzo di una tabella di catalogo con crawler può fornire una migliore automazione per tracciare le partizioni appena aggiunte e ti offre la flessibilità necessaria per selezionare partizioni particolari con Predicato pushdown.
- Utilizzo dell'elenco di file AWS Glue Amazon S3 per set di dati di grandi dimensioni. Un segnalibro elenca tutti i file sotto ogni partizione di input e esegue il filering, quindi se ci sono troppi file sotto una singola partizione il segnalibro può essere eseguito nel driver OOM. Usa il lister di file AWS Glue Amazon S3 per evitare di elencare tutti i file in memoria contemporaneamente.

## Memorizzazione dei dati shuffle di Spark

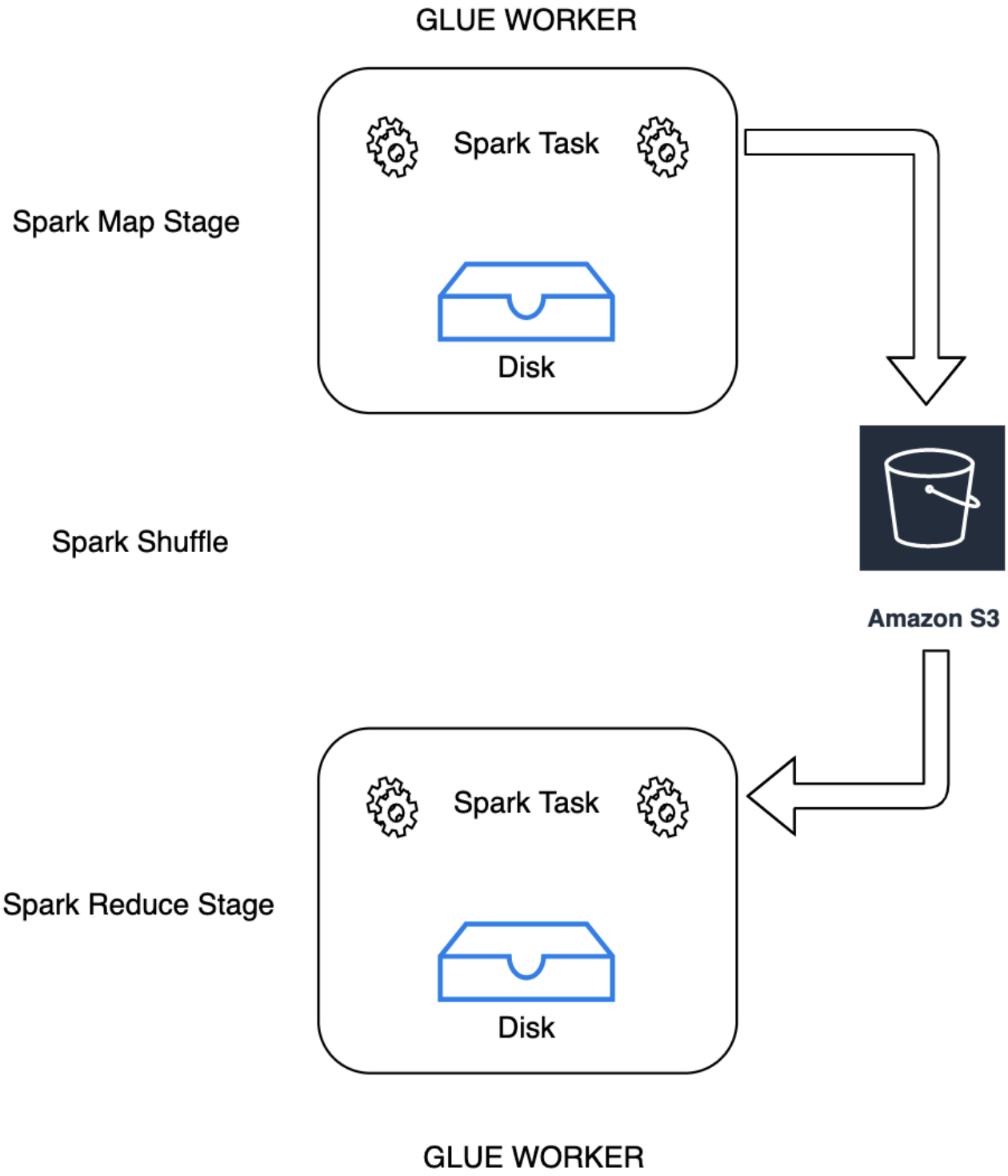
Lo shuffle rappresenta un passaggio importante in un processo Spark quando i dati vengono riorganizzati tra le partizioni. È necessario perché trasformazioni estese, come `join`, `groupByKey`, `reduceByKey` e `repartition`, hanno bisogno di informazioni da altre partizioni per completare l'elaborazione. Spark raccoglie i dati richiesti da ciascuna partizione e li combina in una nuova partizione. Durante uno shuffle, i dati vengono scritti su disco e trasferiti attraverso la rete. Di conseguenza, l'operazione di shuffle è legata alla capacità del disco locale. Spark genera un errore `No space left on device` o `MetadataFetchFailedException` quando sull'executor non è rimasto sufficiente spazio su disco e non vi è un ripristino.

> **Note**
>
> AWS Glue Il plug-in Spark shuffle con Amazon S3 è supportato solo per i lavori ETL. AWS Glue

### Soluzione

ConAWS Glue, ora puoi usare Amazon S3 per archiviare i dati di Spark shuffle utilizzando il Cloud Shuffle Storage Plugin. Amazon S3 è un servizio di archiviazione di oggetti che offre scalabilità,

disponibilità dei dati, sicurezza e prestazioni tra le migliori del settore. Questa soluzione disaggrega calcolo e storage per i processi Spark e offre elasticità completa e storage per lo shuffle a basso costo, consentendo di eseguire in modo affidabile i carichi di lavoro con shuffle intensivo.



Puoi attivare lo shuffle Amazon S3 per eseguire i processi AWS Glue in modo affidabile senza errori se sono legati alla capacità del disco locale per operazioni di shuffle di grandi dimensioni. In alcuni casi, lo shuffle su Amazon S3 è leggermente più lento del disco locale (o EBS) se si dispone di un numero elevato di piccole partizioni o file shuffle scritti su Amazon S3.

Prerequisiti per l'utilizzo del plug-in Cloud Shuffle Storage

Per utilizzare il Cloud Shuffle Storage Plugin con i job ETL, è necessario quanto segue: AWS Glue

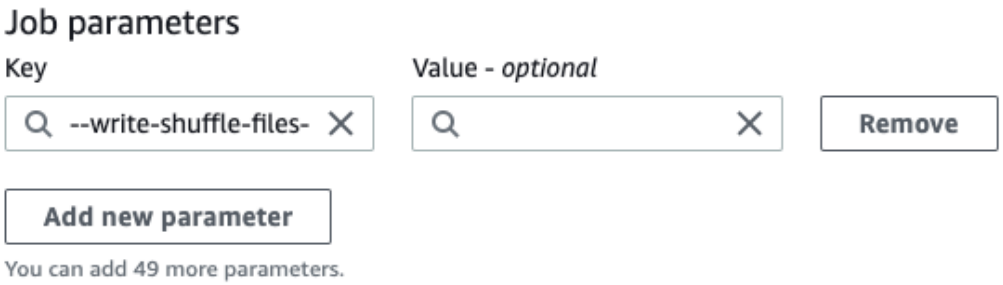
- Un bucket Amazon S3 situato nella stessa regione in cui viene eseguito il processo, per archiviare i dati intermedi e i dati riversati. Il prefisso Amazon S3 dell'archiviazione con shuffle può essere specificato con `--conf spark.shuffle.glue.s3ShuffleBucket=s3://shuffle-bucket/prefix/`, come nell'esempio seguente:

```
--conf spark.shuffle.glue.s3ShuffleBucket=s3://glue-shuffle-123456789-us-east-1/glue-shuffle-data/
```

- Imposta le policy del ciclo di vita dell'archiviazione di Amazon S3 sul prefisso (come `glue-shuffle-data`), poiché lo shuffle manager non pulisce i file al termine del processo. Lo shuffle intermedio e i dati riversati devono essere eliminati al termine di un processo. Gli utenti possono impostare policy del ciclo di vita breve sul prefisso. Le istruzioni per configurare il ciclo di vita per Amazon S3 sono disponibili nella sezione Setting lifecycle configuration on a bucket nella Guida per l'utente di Amazon Simple Storage Service.

Utilizzo di AWS Glue Spark shuffle manager dalla console AWS

Per impostare lo shuffle manager di AWS Glue Spark utilizzando la console AWS Glue o AWS Glue Studio durante la configurazione di un processo: scegli il parametro --write-shuffle-files-to-s3 per attivare lo shuffle su Amazon S3 per il processo.



Utilizzo del plug-in shuffle di AWS Glue Spark

I seguenti parametri di processo attivano e regolano AWS Glue shuffle manager. Questi parametri sono flag, quindi i valori forniti non vengono considerati.

> **Important**
>
> Per disabilitare Amazon S3 shuffling, devi rimuovere completamente il `--write-shuffle-files-to-s3` parametro dalla configurazione del processo. L'impostazione del valore su `false` non disabilita lo shuffling di Amazon S3: il parametro funge da flag basato sulla presenza, il che significa che qualsiasi valore (incluso) abilita lo shuffling di `false` Amazon S3 quando il parametro è presente.

- `--write-shuffle-files-to-s3`: il flag principale, che abilita lo shuffle manager di AWS Glue Spark all'utilizzo dei bucket Amazon S3 per scrivere e leggere i dati shuffle. Quando il flag non è specificato, lo shuffle manager non viene utilizzato.
- `--write-shuffle-spills-to-s3`: (supportato solo in AWS Glue versione 2.0). Un flag facoltativo che consente di scaricare i file riversati nei bucket Amazon S3, fornendo ulteriore resilienza al processo Spark. Questo è necessario solo per carichi di lavoro di grandi dimensioni che riversano molti dati sul disco. Quando il flag non è specificato, non viene scritto alcun file riversato intermedio.
- `--conf spark.shuffle.glue.s3ShuffleBucket=s3://<shuffle-bucket>` — Un altro flag opzionale che specifica il bucket Amazon S3 in cui vengono scritti i file shuffle. `--TempDir`Per impostazione predefinita, /shuffle-data. AWS Glue 3.0+ supporta la scrittura di file shuffle su più bucket specificando i bucket con un delimitatore di virgola, come in. `--conf spark.shuffle.glue.s3ShuffleBucket=s3://`*`shuffle-bucket-1`*`/`*`prefix`*`,s3://`*`shuffle-bucket-2`*`/`*`prefix`*`/` L'utilizzo di più bucket migliora le prestazioni.

Per abilitare la crittografia a riposo per i dati shuffle, fornisci le impostazioni di configurazione della sicurezza. Per ulteriori informazioni sulle configurazioni di sicurezza, consulta the section called "Configurazione della crittografia". AWS Glue supporta tutte le altre configurazioni relative ai dati shuffle fornite da Spark.

File binari software per il plug-in di archiviazione cloud shuffle

Puoi anche scaricare i file binari software del plug-in di archiviazione cloud shuffle per Apache Spark con la licenza Apache 2.0 ed eseguirli in qualsiasi ambiente Spark. Il nuovo plug-in include il supporto predefinito per Amazon S3 e può essere facilmente configurato per utilizzare anche altre forme di archiviazione cloud come Google Cloud Storage e Microsoft Azure Blob Storage. Per ulteriori informazioni, consulta Plug-in di archiviazione cloud shuffle per Apache Spark.

Note e limitazioni

Di seguito sono riportate note o limitazioni per AWS Glue shuffle manager:

- AWS Glue shuffle manager non elimina automaticamente i file di dati shuffle (temporanei) archiviati nel bucket Amazon S3 dopo il completamento di un processo. Per garantire la protezione dei dati, segui le istruzioni riportate nella sezione Prerequisiti per l'utilizzo del plug-in Cloud Shuffle Storage prima di abilitare il plug-in Cloud Shuffle Storage.
- È possibile usare questa funzionalità se i dati sono asimmetrici.

Plug-in di archiviazione cloud shuffle per Apache Spark

Il plug-in di archiviazione cloud shuffle è un plug-in Apache Spark compatibile con l'API `ShuffleDataIO` che consente di archiviare dati shuffle su sistemi di archiviazione cloud (come Amazon S3). Ti aiuta a integrare o sostituire la capacità di archiviazione su disco locale per operazioni shuffle di grandi dimensioni, generalmente innescate da trasformazioni come `groupByKey` e `repartition` nelle applicazioni Spark `joinreduceByKey`, riducendo così gli errori comuni o la price/performance dislocazione dei processi e delle pipeline di analisi dei dati senza server.

AWS Glue

Le versioni 3.0 e 4.0 di AWS Glue includono il plug-in preinstallato e pronto per abilitare lo shuffling su Amazon S3 senza passaggi aggiuntivi. Per ulteriori informazioni, consulta Plug-in shuffle di AWS Glue Spark con Amazon S3 per abilitare la funzionalità per le tue applicazioni Spark.

Altri ambienti Spark

Il plug-in richiede che in altri ambienti Spark siano impostate le seguenti configurazioni Spark:

- `--conf spark.shuffle.sort.io.plugin.class=com.amazonaws.spark.shuffle.io.cloud.Choppe` indica a Spark di utilizzare questo plug-in per Shuffle IO.

- `--conf spark.shuffle.storage.path=s3://bucket-name/shuffle-file-dir`: il percorso in cui verranno archiviati i file shuffle.

> **Note**
>
> Il plug-in sovrascrive una classe principale di Spark. Di conseguenza, il jar del plug-in deve essere caricato prima dei jar di Spark. Ciò è possibile utilizzando `userClassPathFirst` in ambienti YARN on-premise se il plug-in viene utilizzato all'esterno di AWS Glue.

Creazione di bundle per il plug-in con le applicazioni Spark

È possibile raggruppare il plug-in con le applicazioni Spark e le distribuzioni Spark (versioni 3.1 e successive) aggiungendo la dipendenza del plug-in nel file Mavenpom.xml mentre si sviluppano le applicazioni Spark in locale. Per ulteriori informazioni sulle versioni del plug-in e di Spark, consulta Versioni del plug-in.

```
<repositories>
   ...
    <repository>
        <id>aws-glue-etl-artifacts</id>
        <url>https://aws-glue-etl-artifacts.s3.amazonaws.com/release/ </url>
    </repository>
</repositories>
...
<dependency>
    <groupId>com.amazonaws</groupId>
    <artifactId>chopper-plugin</artifactId>
    <version>3.1-amzn-LATEST</version>
</dependency>
```

In alternativa, puoi scaricare i file binari direttamente dagli artefatti di AWS Glue Maven e includerli nell'applicazione Spark come segue.

```
#!/bin/bash
sudo wget -v https://aws-glue-etl-artifacts.s3.amazonaws.com/release/com/amazonaws/
chopper-plugin/3.1-amzn-LATEST/chopper-plugin-3.1-amzn-LATEST.jar -P /usr/lib/spark/
jars/
```

Esempio spark-submit

```
spark-submit --deploy-mode cluster \
--conf spark.shuffle.storage.s3.path=s3://<ShuffleBucket>/<shuffle-dir> \
--conf spark.driver.extraClassPath=<Path to plugin jar> \
--conf spark.executor.extraClassPath=<Path to plugin jar> \
--class <your test class name> s3://<ShuffleBucket>/<Your application jar> \
```

Configurazioni facoltative

Questi sono i valori delle configurazioni facoltative che controllano il comportamento dello shuffle di Amazon S3.

- `spark.shuffle.storage.s3.enableServerSideEncryption`: Enable/disable S3 SSE per shuffle e spill file. Il valore predefinito è `true`.
- `spark.shuffle.storage.s3.serverSideEncryption.algorithm`: l'algoritmo SSE da utilizzare. Il valore predefinito è `AES256`.
- `spark.shuffle.storage.s3.serverSideEncryption.kms.key`: l'ARN della chiave KMS quando è abilitato SSE aws:kms.

Oltre a queste configurazioni, potrebbe essere necessario impostarne altre come `spark.hadoop.fs.s3.enableServerSideEncryption` e configurazioni aggiuntive specifiche dell'ambiente per garantire l'applicazione della crittografia appropriata per il caso d'uso.

Versioni del plug-in

Questo plugin è supportato per le versioni Spark associate a ciascuna versione. AWS Glue La tabella seguente mostra la AWS Glue versione, la versione Spark e la versione del plug-in associata con la posizione Amazon S3 per il file binario del software del plug-in.

| Versione AWS Glue | Versione di Spark | Versione del plug-in | Posizione di Amazon S3. |
|---|---|---|---|
| 3.0 | 3.1 | 3.1-amzn-LATEST | s3:///-Plugin-3.1-amzn-latest.jar aws-glue-etl-artifacts release/com/amazon aws/chopper-plugin /3.1-amzn-0/chopper |

| Versione AWS Glue | Versione di Spark | Versione del plug-in | Posizione di Amazon S3. |
|---|---|---|---|
| 4.0 | 3.3 | 3.3-amzn-LATEST | s3://aws-glue-etl-artifacts/release/com/amazonaws/chopper-plugin/3.3-amzn-0/chopper-Plugin-3.3-AMZN-latest.jar |

Licenza

Il software del plug-in è concesso in licenza ai sensi della licenza Apache-2.0.

## Monitoraggio dei processi Spark AWS Glue

Argomenti

- Spark Metrics disponibile in AWS Glue Studio
- Monitoraggio dei processi tramite l'interfaccia utente Web di Apache Spark
- Monitoraggio con AWS Glue Job Run Insights
- Monitoraggio con Amazon CloudWatch
- Monitoraggio e debug dei processi

### Spark Metrics disponibile in AWS Glue Studio

La scheda Metrics (Parametri) mostra i parametri raccolti quando un processo viene eseguito ed è attivata la profilatura. Nei processi Spark vengono visualizzati i grafici seguenti:

- Spostamento di dati ETL
- Profilo di memoria: driver ed executor

Scegli View additional metrics (Visualizza parametri aggiuntivi) per visualizzare i grafici relativi agli elementi seguenti:

- Spostamento di dati ETL
- Profilo di memoria: driver ed executor

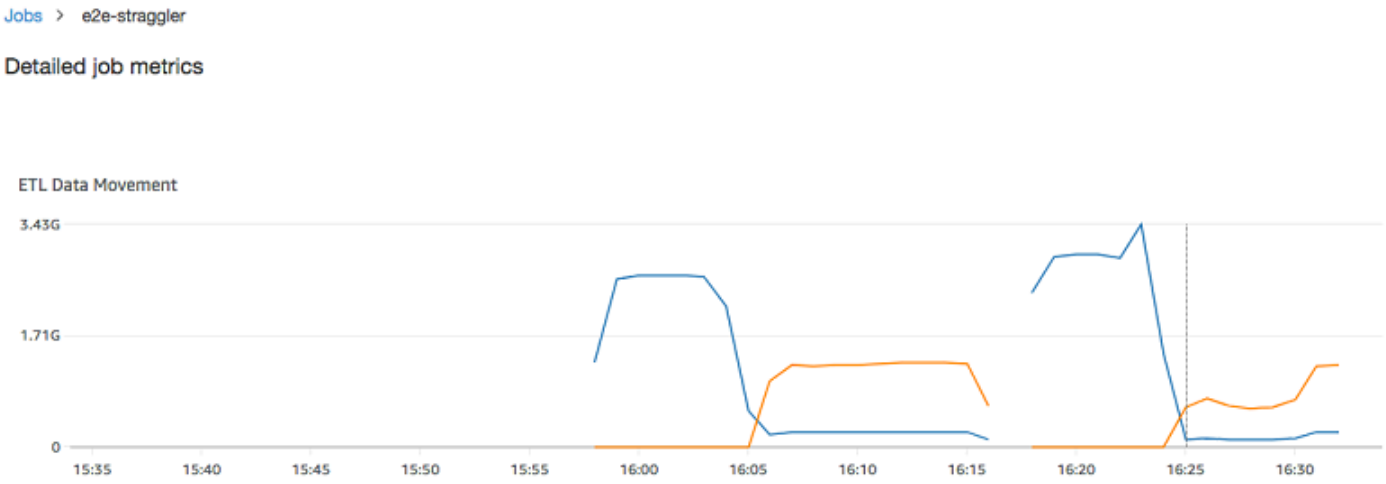
- Distribuzione casuale dei dati tra executor
- Carico CPU: driver ed executor
- Esecuzione del processo: executor attivi, fasi completate e numero massimo di executor necessari

I dati di questi grafici vengono inseriti nelle CloudWatch metriche se il job è configurato per raccogliere metriche. Per ulteriori informazioni su come abilitare i parametri e interpretare i grafici, consulta Monitoraggio e debug dei processi.

Example Grafico relativo allo spostamento di dati ETL

Il grafico relativo allo spostamento di dati ETL mostra i parametri seguenti:

- Numero di byte letti da Amazon S3 da tutti gli executor: `glue.ALL.s3.filesystem.read_bytes`
- Numero di byte scritti in Amazon S3 da tutti gli executor: `glue.ALL.s3.filesystem.write_bytes`

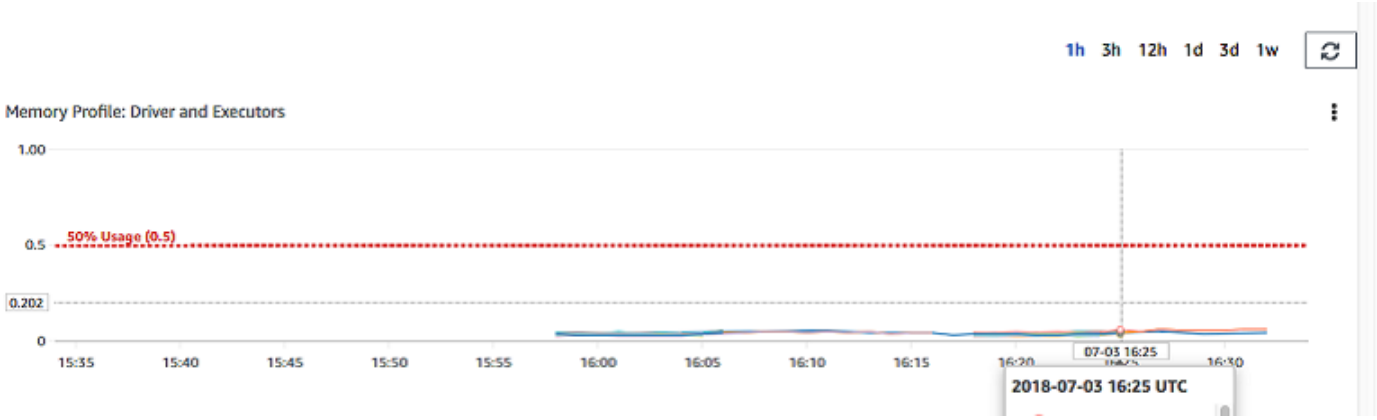



Example Grafico relativo al profilo di memoria

Il grafico relativo al profilo di memoria mostra i parametri seguenti:

- Frazione di memoria usata dall'heap JVM per questo driver (dimensione: 0-1) dal driver, da un executor identificato da executorId o da tutti gli executor—
  - `glue.driver.jvm.heap.usage`
  - `glue.executorId.jvm.heap.usage`

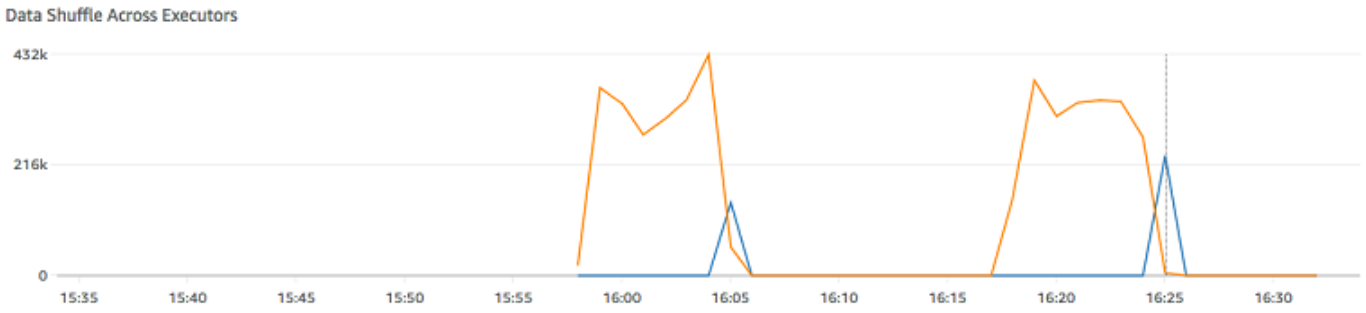
- glue.ALL.jvm.heap.usage




Example Grafico relativo alla distribuzione casuale dei dati tra executor

Il grafico relativo alla distribuzione casuale dei dati tra executor mostra i parametri seguenti:

- Numero di byte letti da tutti gli executor per distribuire i dati in modo casuale: glue.driver.aggregate.shuffleLocalBytesRead
- Numero di byte scritti da tutti gli executor per distribuire i dati in modo casuale: glue.driver.aggregate.shuffleBytesWritten

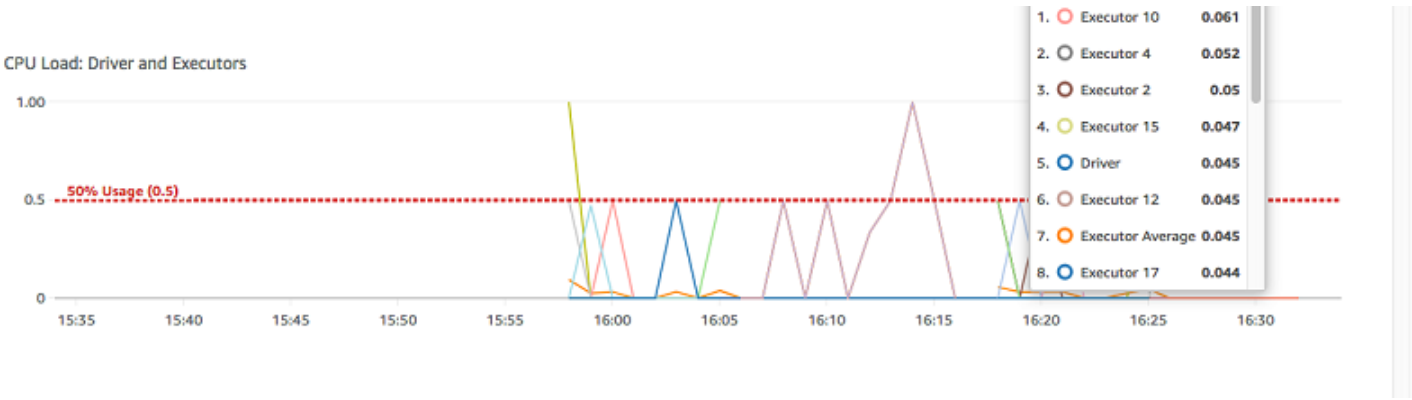



Example Grafico relativo al carico CPU

Il grafico relativo al carico CPU mostra i parametri seguenti:

- Frazione del carico di sistema della CPU usata (dimensione: 0-1) dal driver, da un executor identificato da executorId o da tutti gli executor:
  - glue.driver.system.cpuSystemLoad

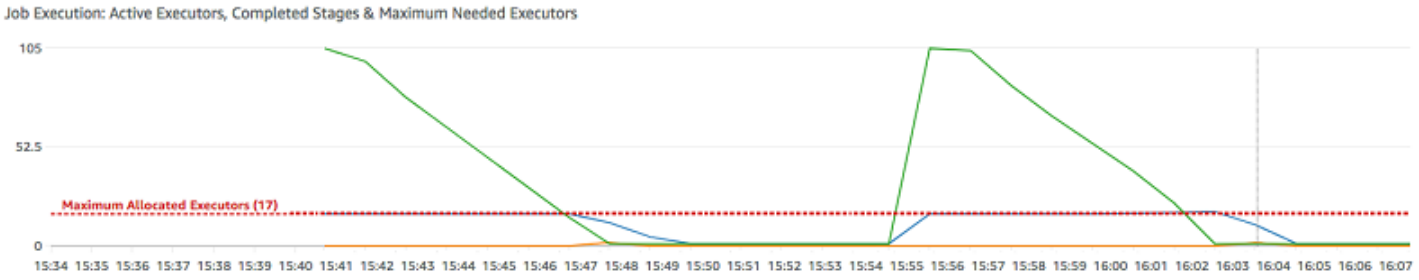
- glue.*executorId*.system.cpuSystemLoad
- glue.ALL.system.cpuSystemLoad




Example Grafico relativo all'esecuzione del processo

Il grafico relativo all'esecuzione del processo mostra i parametri seguenti:

- Numero di executor attivamente in esecuzione: glue.driver.ExecutorAllocationManager.executors.numberAllExecutors
- Numero di fasi completate: glue.aggregate.numCompletedStages
- Numero massimo di executor necessari: glue.driver.ExecutorAllocationManager.executors.numberMaxNeededExecutors




Monitoraggio dei processi tramite l'interfaccia utente Web di Apache Spark

Puoi utilizzare l'interfaccia utente Web di Apache Spark per monitorare ed eseguire il debug dei processi ETL AWS Glue in esecuzione sul sistema di processi AWS Glue e anche delle applicazioni Spark in esecuzione sugli endpoint di sviluppo AWS Glue. L'interfaccia utente di Spark consente di controllare quanto segue per ogni processo:

- Tempistica eventi di ogni fase Spark
- Un grafo aciclico orientato (DAG) del processo
- Piani fisici e logici per le query SparkSQL
- Le variabili ambientali Spark sottostanti per ogni processo

Per ulteriori informazioni sull'utilizzo dell'interfaccia utente Web di Spark, consulta l'interfaccia utente Web nella documentazione di Spark. Per indicazioni su come interpretare i risultati dell'interfaccia utente di Spark per migliorare le prestazioni del tuo lavoro, consulta le migliori pratiche per l'ottimizzazione delle prestazioni per i lavori di Apache Spark in AWS Glue Prescriptive Guidance. AWS

Puoi vedere l'interfaccia utente di Spark nella console. AWS Glue È disponibile quando un AWS Glue job viene eseguito su versioni AWS Glue 3.0 o successive con registri generati nel formato Standard (anziché legacy), che è l'impostazione predefinita per i lavori più recenti. Se disponi di file di registro di dimensioni superiori a 0,5 GB, puoi abilitare il supporto roll-log per i job run su versioni AWS Glue 4.0 o successive per semplificare l'archiviazione, l'analisi e la risoluzione dei problemi dei log.

Puoi abilitare l'interfaccia utente di Spark utilizzando la AWS Glue console o AWS Command Line Interface ()AWS CLI. Quando abiliti l'interfaccia utente di Spark, i processi ETL AWS Glue e le applicazioni Spark su endpoint di sviluppo AWS Glue possono eseguire il backup dei log degli eventi Spark in un percorso specificato in Amazon Simple Storage Service (Amazon S3). Puoi utilizzare i log degli eventi sottoposti a backup in Amazon S3 con l'interfaccia utente di Spark sia in tempo reale, ovvero durante l'esecuzione del processo, sia al termine dello stesso. Sebbene i log rimangano in Amazon S3, l'interfaccia utente Spark nella console può AWS Glue visualizzarli.

Permissions

Per utilizzare l'interfaccia utente Spark nella AWS Glue console, puoi utilizzare `UseGlueStudio` o aggiungere tutti i singoli servizi. APIs Tutti APIs sono necessari per utilizzare completamente l'interfaccia utente di Spark, tuttavia gli utenti possono accedere alle funzionalità di SparkUI aggiungendo il relativo servizio APIs nell'autorizzazione IAM per un accesso granulare.

`RequestLogParsing` è la più importante in quanto esegue l'analisi dei log. I restanti servono per leggere APIs i rispettivi dati analizzati. Ad esempio, `GetStages` fornisce l'accesso ai dati relativi a tutte le fasi di un processo Spark.

L'elenco dei servizi dell'interfaccia utente Spark APIs mappati è riportato di `UseGlueStudio` seguito nella policy di esempio. La policy riportata di seguito fornisce accesso per utilizzare solo le

funzionalità dell'interfaccia utente di Spark. Per aggiungere altre autorizzazioni come Amazon S3 e IAM, consulta Creazione di politiche IAM personalizzate per. AWS Glue Studio

L'elenco dei servizi di interfaccia utente Spark APIs mappati `UseGlueStudio` è riportato di seguito nella policy di esempio. Quando si utilizza un'API del servizio Spark UI, usare il seguente namespace: `glue:<ServiceAPI>`.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "AllowGlueStudioSparkUI",
      "Effect": "Allow",
      "Action": [
        "glue:RequestLogParsing",
        "glue:GetLogParsingStatus",
        "glue:GetEnvironment",
        "glue:GetJobs",
        "glue:GetJob",
        "glue:GetStage",
        "glue:GetStages",
        "glue:GetStageFiles",
        "glue:BatchGetStageFiles",
        "glue:GetStageAttempt",
        "glue:GetStageAttemptTaskList",
        "glue:GetStageAttemptTaskSummary",
        "glue:GetExecutors",
        "glue:GetExecutorsThreads",
        "glue:GetStorage",
        "glue:GetStorageUnit",
        "glue:GetQueries",
        "glue:GetQuery"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

Limitazioni

- L'interfaccia utente di Spark nella AWS Glue console non è disponibile per le esecuzioni di job avvenute prima del 20 novembre 2023 perché sono nel formato di registro precedente.
- L'interfaccia utente di Spark nella AWS Glue console supporta i rolling log per la AWS Glue versione 4.0, come quelli generati di default nei job di streaming. La somma massima di tutti i file di eventi dei log in sequenza è di 2 GB. Per i AWS Glue lavori senza supporto per i rolllog, la dimensione massima del file degli eventi di registro supportata per SparkUI è 0,5 GB.
- L'interfaccia utente di Spark serverless non è disponibile per i log degli eventi Spark archiviati in un bucket Amazon S3 a cui è possibile accedere solo dal proprio VPC.

Esempio: interfaccia utente Web di Apache Spark

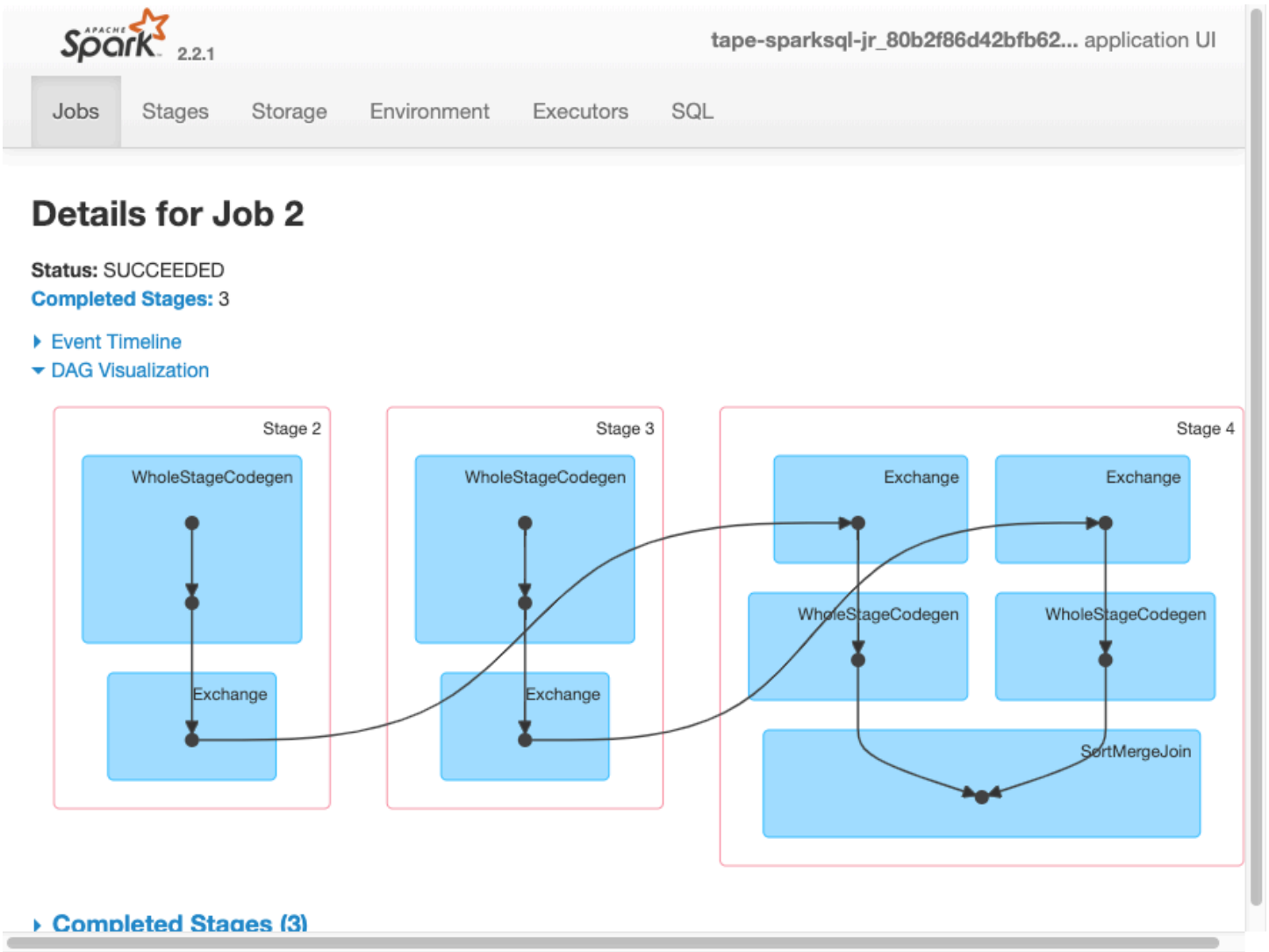
Questo esempio illustra come utilizzare l'interfaccia utente di Spark per comprendere le prestazioni del processo. Gli screenshot mostrano l'interfaccia utente Web di Spark fornita da un server della cronologia Spark autogestito. L'interfaccia utente Spark nella AWS Glue console offre visualizzazioni simili. Per ulteriori informazioni sull'utilizzo dell'interfaccia utente Web di Spark, consulta l'interfaccia utente Web nella documentazione di Spark.

Di seguito è riportato un esempio di un'applicazione Spark che legge da due origini dati, esegue una trasformazione join e la scrive in Amazon S3 nel formato Parquet.

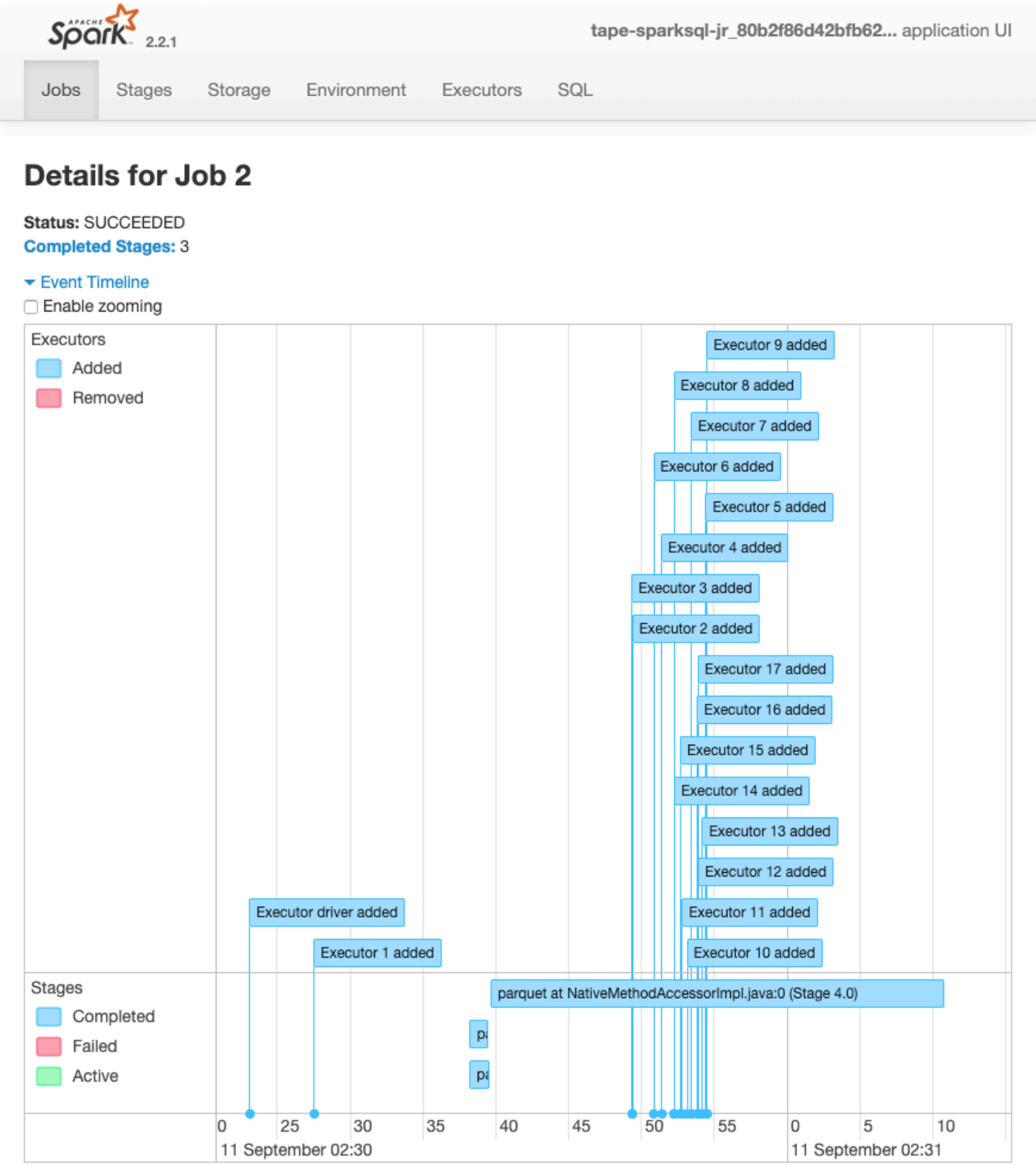
```
import sys
from awsglue.transforms import *
from awsglue.utils import getResolvedOptions
from pyspark.context import SparkContext
from awsglue.context import GlueContext
from awsglue.job import Job
from pyspark.sql.functions import count, when, expr, col, sum, isnull
from pyspark.sql.functions import countDistinct
from awsglue.dynamicframe import DynamicFrame

args = getResolvedOptions(sys.argv, ['JOB_NAME'])

sc = SparkContext()
glueContext = GlueContext(sc)
spark = glueContext.spark_session

job = Job(glueContext)
job.init(args['JOB_NAME'])
```

```
df_persons = spark.read.json("s3://awsglue-datasets/examples/us-legislators/all/
persons.json")
df_memberships = spark.read.json("s3://awsglue-datasets/examples/us-legislators/all/
memberships.json")

df_joined = df_persons.join(df_memberships, df_persons.id == df_memberships.person_id,
 'fullouter')
df_joined.write.parquet("s3://aws-glue-demo-sparkui/output/")

job.commit()
```

La seguente visualizzazione DAG mostra le diverse fasi in questo processo Spark.




La seguente tempistica eventi per un processo mostra l'avvio, l'esecuzione e l'arresto di diversi executor Spark.


Spark 2.2.1
tape-sparksql-jr_80b2f86d42bfb62... application UI
Jobs
Stages
Storage
Environment
Executors
SQL
Details for Job 2
Status: SUCCEEDED
Completed Stages: 3
Event Timeline
Enable zooming
Executors
Added
Removed
Executor 9 added
Executor 8 added
Executor 7 added
Executor 6 added
Executor 5 added
Executor 4 added
Executor 3 added
Executor 2 added
Executor 17 added
Executor 16 added
Executor 15 added
Executor 14 added
Executor 13 added
Executor 12 added
Executor 11 added
Executor 10 added
Executor driver added
Executor 1 added
Stages
Completed
Failed
Active
parquet at NativeMethodAccessorImpl.java:0 (Stage 4.0)
0
25
30
35
40
45
50
55
0
5
10
11 September 02:30
11 September 02:31
DAG Visualization
Completed Stages (3)

La schermata seguente mostra i dettagli dei piani di query SparkSQL:

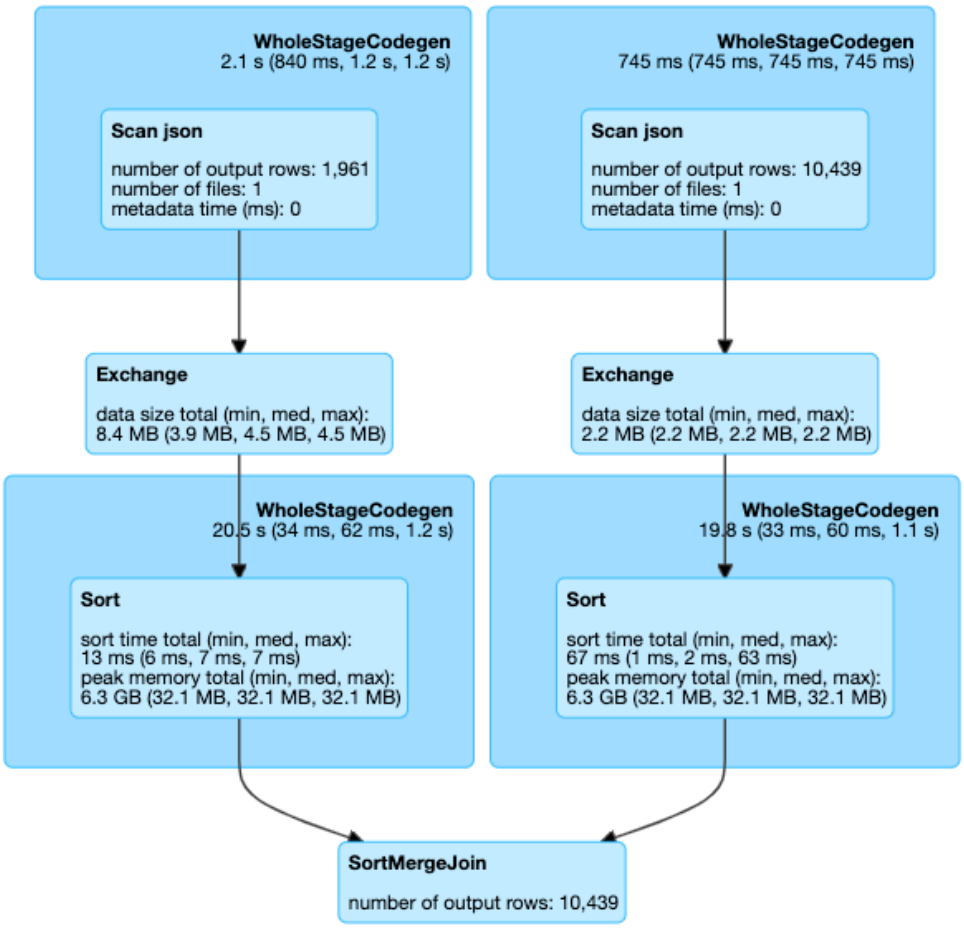
- Piano logico esaminato
- Piano logico analizzato
- Piano logico ottimizzato
- Piano fisico per l'esecuzione

Spark 2.2.1 tape-sparksql-jr_80b2f86d42bfb62... application UI

Jobs Stages Storage Environment Executors SQL

# Details for Query 0

**Submitted Time:** 2019/09/11 02:30:37
**Duration:** 34 s
**Succeeded Jobs:** 2




▾ Details

```
== Parsed Logical Plan ==
Join FullOuter, (id#14 = person_id#50)
:-
Relation[birth_date#8,contact_details#9,death_date#10,family_name#11,gender#12,given_name#13,id#14,identifiers#15
,image#16,images#17,links#18,name#19,other_names#20,sort_name#21] json
+-
Relation[area_id#45,end_date#46,legislative_period_id#47,on_behalf_of_id#48,organization_id#49,person_id#50,role#
51,start_date#52] json

== Analyzed Logical Plan ==
birth_date: string, contact_details: array<struct<type:string,value:string>>, death_date: string, family_name:
string, gender: string, given_name: string, id: string, identifiers:
array<struct<identifier:string,scheme:string>>, image: string, images: array<struct<url:string>>, links:
array<struct<note:string,url:string>>, name: string, other_names:
array<struct<lang:string,name:string,note:string>>, sort_name: string, area_id: string, end_date: string,
legislative_period_id: string, on_behalf_of_id: string, organization_id: string, person_id: string, role: string,
start_date: string
Join FullOuter, (id#14 = person_id#50)
```

Argomenti


- Abilitazione dell'interfaccia utente Web di Apache Spark per processi AWS Glue
- Avvio del server della cronologia di Spark


## Abilitazione dell'interfaccia utente Web di Apache Spark per processi AWS Glue

Puoi utilizzare l'interfaccia utente Web di Apache Spark per monitorare ed eseguire il debug dei processi ETL AWS Glue in esecuzione sul sistema di processi AWS Glue. Puoi configurare l'interfaccia utente di Spark tramite la console AWS Glue o AWS Command Line Interface (AWS CLI).

Ogni 30 secondi, AWS Glue esegue il backup dei log degli eventi Spark nel percorso Amazon S3 specificato.

Argomenti


- Configurazione dell'interfaccia utente di Spark (console)
- Configurazione dell'interfaccia utente di Spark (AWS CLI)
- Configurazione dell'interfaccia utente di Spark per sessioni che utilizzano notebook
- Abilitare log in sequenza


### Configurazione dell'interfaccia utente di Spark (console)

Segui queste fasi per configurare l'interfaccia utente di Spark mediante la Console di gestione AWS. Quando si crea un AWS Glue lavoro, l'interfaccia utente di Spark è abilitata per impostazione predefinita.

Per attivare l'interfaccia utente di Spark durante la creazione o la modifica di un processo

1. Accedi Console di gestione AWS e apri la AWS Glue console all'indirizzo https://console.aws.amazon.com/glue/.
2. Nel riquadro di navigazione scegliere Jobs (Processi).
3. Scegli Aggiungi processo o selezionane uno esistente.
4. In Dettagli processo, apri le Proprietà avanzate.
5. Nella scheda Interfaccia utente Spark, scegli Scrivi i log dell'interfaccia utente di Spark su Amazon S3.
6. Specifica un percorso Amazon S3 per archiviare i log di eventi Spark per il processo. Tieni presente che, se utilizzi una configurazione di sicurezza nel processo, la crittografia verrà

applicata anche al file di log dell'interfaccia utente di Spark. Per ulteriori informazioni, consulta Crittografia dei dati scritti da AWS Glue.

7. Nella sezione Configurazione della registrazione e del monitoraggio dell'interfaccia utente di Spark:

   - Seleziona Standard se stai generando log da visualizzare nella AWS Glue console.
   - Seleziona Legacy se stai generando i log da visualizzare su un server della cronologia di Spark.
   - Puoi anche decidere di generarli entrambi.

Configurazione dell'interfaccia utente di Spark (AWS CLI)

Per generare log da visualizzare con l'interfaccia utente Spark, nella AWS Glue console, usa AWS CLI per passare i seguenti parametri di lavoro ai AWS Glue job. Per ulteriori informazioni, consulta the section called "Parametri Job".

```
'--enable-spark-ui': 'true',
'--spark-event-logs-path': 's3://s3-event-log-path'
```

Per distribuire i log nelle rispettive posizioni precedenti, imposta il parametro `--enable-spark-ui-legacy-path` su `"true"`. Se non desideri generare log in entrambi i formati, rimuovi il parametro `--enable-spark-ui`.

Configurazione dell'interfaccia utente di Spark per sessioni che utilizzano notebook

> **Warning**
>
> AWS Glue le sessioni interattive attualmente non supportano l'interfaccia utente Spark nella console. Configura un server della cronologia di Spark.

Se usi AWS Glue notebook, configura la configurazione di SparkUI prima di iniziare la sessione. A tale scopo, utilizza il magic per celle `%%configure`:

```
%%configure { "--enable-spark-ui": "true", "--spark-event-logs-path": "s3://path" }
```

Abilitare log in sequenza

L'abilitazione dei file di eventi SparkUI e Rolling Log per i job offre diversi vantaggi: AWS Glue

- Rolling Log Event Files: con i file di eventi Rolling Log abilitati, AWS Glue genera file di log separati per ogni fase dell'esecuzione del lavoro, semplificando l'identificazione e la risoluzione dei problemi specifici di una particolare fase o trasformazione.
- Migliore gestione dei log: i file di eventi dei log in sequenza aiutano a gestire i file di log in modo più efficiente. Invece di avere un unico file di log potenzialmente di grandi dimensioni, i log vengono suddivisi in file più piccoli e più gestibili in base alle fasi di esecuzione del processo. Questo può semplificare l'archiviazione, l'analisi e la risoluzione dei problemi dei log.
- Migliore tolleranza agli errori: se un AWS Glue job fallisce o viene interrotto, i file degli eventi del rolling log possono fornire informazioni preziose sull'ultima fase riuscita, facilitando la ripresa del lavoro da quel punto piuttosto che ricominciare da zero.
- Ottimizzazione dei costi: abilitando i file di eventi dei log in sequenza, è possibile risparmiare sui costi di archiviazione associati ai file di log. Invece di archiviare un singolo file di log potenzialmente di grandi dimensioni, vengono archiviati file di log più piccoli e più gestibili, il che può essere più conveniente, soprattutto per processi complessi o di lunga durata.

In un nuovo ambiente, gli utenti possono abilitare esplicitamente i log in sequenza tramite:

```
'–conf': 'spark.eventLog.rolling.enabled=true'
```

or

```
'–conf': 'spark.eventLog.rolling.enabled=true –conf
spark.eventLog.rolling.maxFileSize=128m'
```

Quando i log in sequenza sono attivati, `spark.eventLog.rolling.maxFileSize` specifica la dimensione massima del file di log degli eventi prima che venga eseguito il rollover. Se non specificato, il valore predefinito di questo parametro facoltativo è 128 MB. Il minimo è 10 MB.

La somma massima di tutti i file di eventi dei log in sequenza è di 2 GB. Per i AWS Glue lavori senza supporto per i rolllog, la dimensione massima del file degli eventi di registro supportata per SparkUI è 0,5 GB.

Puoi disattivare i log in sequenza per un processo di streaming inserendo una configurazione aggiuntiva. Tieni presente che la manutenzione di file di log molto grandi può essere costosa.

Per disattivare i log in sequenza, fornisci la seguente configurazione:

```
'--spark-ui-event-logs-path': 'true',
```

```
'--conf': 'spark.eventLog.rolling.enabled=false'
```

Avvio del server della cronologia di Spark

Puoi utilizzare un server della cronologia Spark per visualizzare i log di Spark sull'infrastruttura. Puoi vedere le stesse visualizzazioni nella AWS Glue console per i AWS Glue job run su versioni AWS Glue 4.0 o successive con i log generati nel formato Standard (anziché legacy). Per ulteriori informazioni, consulta the section called “Monitoraggio con l'interfaccia utente di Spark”.

Puoi avviare il server di cronologia Spark utilizzando un AWS CloudFormation modello che ospita il server su un'istanza EC2 o avviarlo localmente utilizzando Docker.

Argomenti


- Avvio del server di cronologia Spark e visualizzazione dell'interfaccia utente Spark utilizzando AWS CloudFormation
- Avvio del server della cronologia Spark e visualizzazione dell'interfaccia utente di Spark mediante Docker


Avvio del server di cronologia Spark e visualizzazione dell'interfaccia utente Spark utilizzando AWS CloudFormation

Puoi utilizzare un AWS CloudFormation modello per avviare il server di cronologia Apache Spark e visualizzare l'interfaccia utente web di Spark. Questi modelli sono esempi che è necessario modificare per soddisfare i requisiti.

Per avviare il server di cronologia Spark e visualizzare l'interfaccia utente Spark utilizzando CloudFormation

1. Scegli uno dei pulsanti Launch Stack (Avvia stack) nella tabella seguente. Questo avvia lo stack sulla console. CloudFormation

| Region | Avvia |
| --- | --- |
| Stati Uniti orientali (Ohio) | Launch Stack |
| Stati Uniti orientali (Virginia settentrionale) | Launch Stack |

| Region | Avvia |
| --- | --- |
| Stati Uniti occidentali (California settentrionale) | Launch Stack |
| Stati Uniti occidentali (Oregon) | Launch Stack |
| Africa (Città del Capo) | Launch Stack |
| Asia Pacific (Hong Kong) | Launch Stack |
| Asia Pacific (Mumbai) | Launch Stack |
| Asia Pacifico (Osaka) | Launch Stack |
| Asia Pacifico (Seoul) | Launch Stack |
| Asia Pacifico (Singapore) | Launch Stack |
| Asia Pacifico (Sydney) | Launch Stack |
| Asia Pacifico (Tokyo) | Launch Stack |
| Canada (Centrale) | Launch Stack |
| Europa (Francoforte) | Launch Stack |
| Europa (Irlanda) | Launch Stack |
| Europa (Londra) | Launch Stack |
| Europa (Milano) | Launch Stack |

| Region | Avvia |
| --- | --- |
| Europa (Parigi) | |
| Europa (Stoccolma) | |
| Medio Oriente (Bahrein) | |
| Sud America (San Paolo) | |

2. Nella pagina Specify template (Specifica modello), scegli Next (Avanti).
3. Nella pagina Specify stack details (Specifica dettagli stack), immetti Stack name (Nome stack). Inserisci informazioni aggiuntive sotto Parameters (Parametri).

   a. Configurazione dell'interfaccia utente di Spark

      Inserisci le informazioni che seguono:

      - Intervallo di indirizzi IP: l'intervallo di indirizzi IP che può essere utilizzato per visualizzare l'interfaccia utente di Spark. Se desideri limitare l'accesso da un intervallo di indirizzi IP specifico, devi utilizzare un valore personalizzato.
      - Porta del server di cronologia: la porta per l'interfaccia utente di Spark. Puoi usare il valore predefinito.
      - Directory del log di eventi: scegli la posizione in cui sono archiviati i log di eventi Spark dagli endpoint di sviluppo o dal processo AWS Glue. È necessario utilizzare **`s3a://`** per lo schema di percorso dei log di eventi.
      - Posizione del pacchetto Spark: puoi usare il valore di default.
      - Keystore path: percorso SSL/TLS keystore per HTTPS. Se desideri utilizzare un file keystore personalizzato, puoi specificare il percorso S3 `s3://path_to_your_keystore_file` qui. Se lasci questo parametro vuoto, viene generato e utilizzato un keystore basato su certificato autofirmato.
      - Keystore password (Password keystore): inserisci una password del keystore SSL/TLS per HTTPS.

b. Configurazione di un'istanza EC2

Inserisci le informazioni che seguono:

- Tipo di istanza: il tipo di istanza Amazon EC2 che ospita il server della cronologia di Spark. Poiché questo modello avvia un'istanza Amazon EC2 nel tuo account, il costo di Amazon EC2 verrà addebitato separatamente nel tuo account.
- ID AMI più recente: l'ID AMI di Amazon Linux 2 per l'istanza del server della cronologia di Spark. Puoi usare il valore predefinito.
- ID VPC: l'ID del cloud privato virtuale (VPC) per l'istanza del server della cronologia di Spark. Puoi utilizzare uno qualsiasi dei file VPCs disponibili nel tuo account L'utilizzo di un VPC predefinito con un ACL di rete predefinito non è consigliato. Per ulteriori informazioni, consulta VPC predefinito e sottoreti predefinite e Creazione di un VPC nella Guida per l'utente di Amazon VPC.
- ID sottorete: l'ID dell'istanza del server della cronologia di Spark. Puoi utilizzare una qualsiasi delle sottoreti nel VPC. Devi essere in grado di raggiungere la rete dal client alla sottorete. Se desideri accedere tramite Internet, devi utilizzare una sottorete pubblica che dispone dell'Internet gateway nella tabella di routing.

c. Scegli Next (Successivo).

4. Nella pagina Configura le opzioni dello stack, per utilizzare le credenziali utente correnti per determinare come CloudFormation creare, modificare o eliminare le risorse nello stack, scegli Avanti. Inoltre, nella sezione Autorizzazioni, è possibile specificare un ruolo da utilizzare al posto delle autorizzazioni utente correnti, dopodiché occorre scegliere Successivo.
5. Nella pagina Review (Revisione), rivedi il modello.

   Seleziona Riconosco che CloudFormation potrebbe creare risorse IAM, quindi scegli Create stack.
6. Attendi la creazione dello stack.
7. Apri la scheda Outputs (Output).

   a. Copia l'URL di SparkUiPublicUrlse stai usando una sottorete pubblica.

   b. Copia l'URL di SparkUiPrivateUrlse stai usando una sottorete privata.
8. Apri un browser Web e incolla l'URL. Ciò consente di accedere al server tramite HTTPS sulla porta specificata. Il tuo browser potrebbe non riconoscere il certificato del server. Se ciò accade, ignora la protezione e procedi comunque.

### Avvio del server della cronologia Spark e visualizzazione dell'interfaccia utente di Spark mediante Docker

Se preferisci l'accesso locale (non mantenere un'istanza EC2 per il server della cronologia Apache Spark), puoi anche utilizzare Docker per avviare il server della cronologia Apache Spark e visualizzare l'interfaccia utente di Spark in locale. Questo Dockerfile è un esempio che devi modificare per soddisfare i tuoi requisiti.

#### Prerequisiti

Per informazioni su come installare Docker sul laptop, vedi la community di Docker Engine.

#### Per avviare il server della cronologia Spark e visualizzare l'interfaccia utente di Spark in locale utilizzando Docker

1. Scarica file da GitHub.

   Scarica il Dockerfile e `pom.xml` dagli esempi di codice di AWS Glue.
2. Stabilisci se desideri utilizzare le credenziali utente o le credenziali dell'utente federato per accedere a AWS.

   - Per utilizzare le credenziali utente correnti per l'accesso AWS, recuperate i valori da utilizzare per `AWS_ACCESS_KEY_ID` e `AWS_SECRET_ACCESS_KEY` nel `docker run` comando. Per ulteriori informazioni, consulta Gestione delle chiavi di accesso per gli utenti IAM nella Guida per l'utente di IAM .
   - Per utilizzare gli utenti federati SAML 2.0 per l'accesso AWS, ottieni i valori per `AWS_ACCESS_KEY_ID`, `AWS_SECRET_ACCESS_KEY` e. `AWS_SESSION_TOKEN` Per ulteriori informazioni, consulta la sezione relativa alla richiesta di credenziali di sicurezza provvisorie
3. Determina la posizione della directory del log di eventi da utilizzare nel comando `docker run`.
4. Crea l'immagine Docker utilizzando i file nella directory locale, utilizzando il nome `glue/sparkui` e il tag `latest`.

   ```
   $ docker build -t glue/sparkui:latest .
   ```
5. Crea e avvia il container docker.

   Nei comandi seguenti, utilizza i valori ottenuti in precedenza nei passaggi 2 e 3.

   a. Per creare il container docker utilizzando le credenziali utente, utilizza un comando simile al seguente

```
docker run -itd -e SPARK_HISTORY_OPTS="$SPARK_HISTORY_OPTS -
Dspark.history.fs.logDirectory=s3a://path_to_eventlog
 -Dspark.hadoop.fs.s3a.access.key=AWS_ACCESS_KEY_ID -
Dspark.hadoop.fs.s3a.secret.key=AWS_SECRET_ACCESS_KEY"
 -p 18080:18080 glue/sparkui:latest "/opt/spark/bin/spark-class
 org.apache.spark.deploy.history.HistoryServer"
```

b. Per creare il container docker utilizzando credenziali temporanee, utilizzare `org.apache.hadoop.fs.s3a.TemporaryAWSCredentialsProvider` come provider e fornisci i valori delle credenziali ottenuti nel passaggio 2. Per ulteriori informazioni, consulta Using Session Credentials with Temporary AWSCredentials Provider nella documentazione Hadoop: Integration with Amazon Web Services.

```
docker run -itd -e SPARK_HISTORY_OPTS="$SPARK_HISTORY_OPTS -
Dspark.history.fs.logDirectory=s3a://path_to_eventlog
 -Dspark.hadoop.fs.s3a.access.key=AWS_ACCESS_KEY_ID -
Dspark.hadoop.fs.s3a.secret.key=AWS_SECRET_ACCESS_KEY
 -Dspark.hadoop.fs.s3a.session.token=AWS_SESSION_TOKEN
 -
Dspark.hadoop.fs.s3a.aws.credentials.provider=org.apache.hadoop.fs.s3a.TemporaryAWSCred
 -p 18080:18080 glue/sparkui:latest "/opt/spark/bin/spark-class
 org.apache.spark.deploy.history.HistoryServer"
```

> **Note**
>
> Questi parametri di configurazione provengono dal Modulo Hadoop-AWS. Potrebbe essere necessario aggiungere una configurazione specifica in base al proprio caso d'uso. Ad esempio: gli utenti in regioni isolate dovranno configurare il `spark.hadoop.fs.s3a.endpoint`.

6. Apri `http://localhost:18080` nel browser per visualizzare l'interfaccia utente di Spark in locale.

### Monitoraggio con AWS Glue Job Run Insights

AWS Glue job run insights è una funzionalità AWS Glue che semplifica il debug e l'ottimizzazione dei job. AWS Glue AWS Glue fornisce l'interfaccia utente di Spark e CloudWatch registri e metriche

per il monitoraggio dei lavori. AWS Glue Con questa funzionalità, ottieni queste informazioni sull'esecuzione del tuo AWS Glue lavoro:

- Numero di riga dello script di AWS Glue lavoro che ha avuto un errore.
- Azione Spark eseguita per l'ultima volta nel piano di query di Spark poco prima dell'errore del processo.
- Eventi di eccezione Spark correlati all'errore riscontrato in un flusso di log in ordine cronologico.
- Analisi della causa principale e azione consigliata (come l'ottimizzazione dello script) per risolvere il problema.
- Eventi Spark comuni (messaggi log relativi a un'azione Spark) con un'azione consigliata che affronta la causa principale.

Tutte queste informazioni sono disponibili utilizzando due nuovi flussi di log nei CloudWatch log dei lavori. AWS Glue

Requisiti

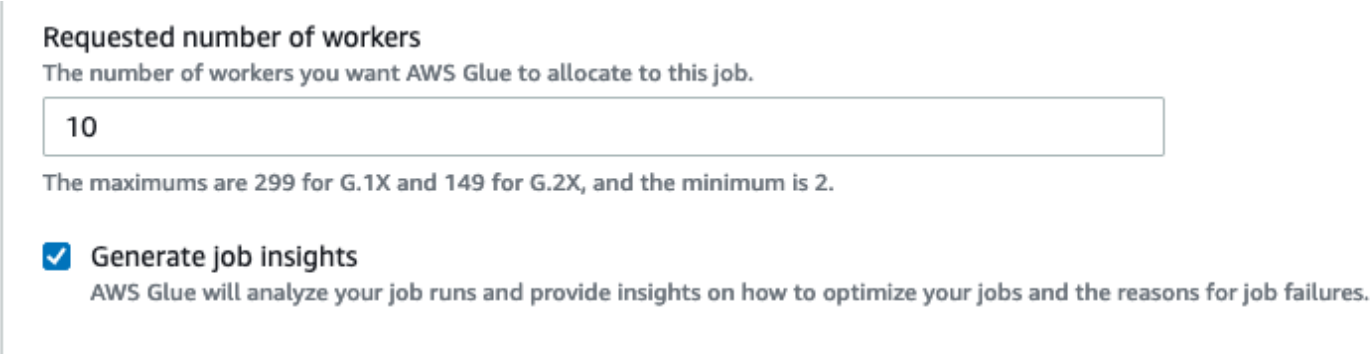
La funzionalità AWS Glue job run insights è disponibile per AWS Glue le versioni 2.0 e successive. Puoi seguire la guida alla migrazione per i lavori esistenti per aggiornarli da AWS Glue versioni precedenti.

Abilitare Job Run Insights per un job AWS Glue ETL

Puoi abilitare Job Run Insights tramite AWS Glue Studio o la CLI.

AWS Glue Studio

Quando si crea un lavoro tramite AWS Glue Studio, è possibile abilitare o disabilitare Job Run Insights nella scheda Job Details. Controlla che la casella Generate job insights (Genera informazioni sul lavoro) sia selezionata.



Riga di comando

Se si crea un processo tramite CLI, è possibile avviare un processo con un singolo nuovo parametro del processo: `--enable-job-insights = true`.

Per impostazione predefinita, i flussi di log di informazioni dell'esecuzione del processo vengono creati nello stesso gruppo di log predefinito utilizzato da Registrazione continua AWS Glue, cioè `/aws-glue/jobs/logs-v2/`. È possibile impostare il nome del gruppo di log personalizzato, i filtri di log e le configurazioni del gruppo di log utilizzando lo stesso set di argomenti per la registrazione continua. Per ulteriori informazioni, consulta Abilitazione della registrazione continua per i AWS Glue lavori.

Accesso al job run insights registra i flussi in CloudWatch

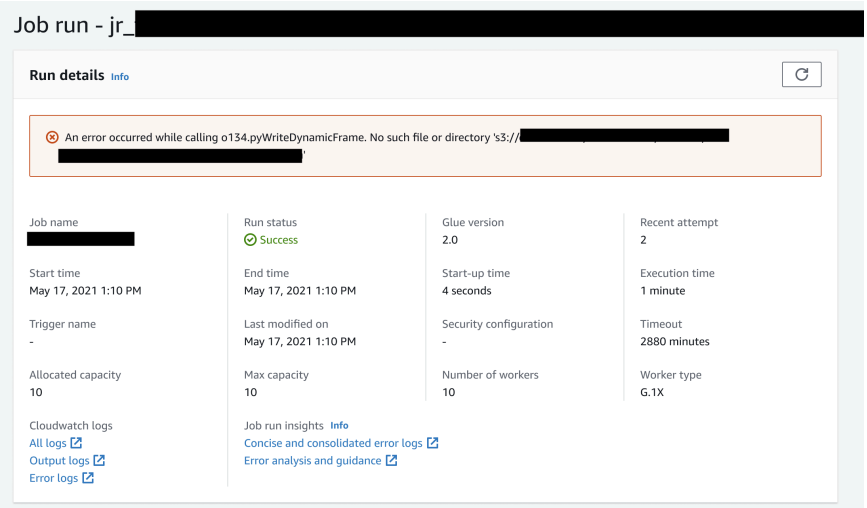
Con la funzione di informazioni dell'esecuzione del processo abilitata, potrebbero esserci due flussi di log creati quando un processo non riesce. Quando un processo termina correttamente, nessuno dei flussi viene generato.

1. Flusso di log analisi delle eccezioni: `<job-run-id>-job-insights-rca-driver`. Questo flusso fornisce quanto segue:
   - Numero di riga dello script di AWS Glue lavoro che ha causato l'errore.
   - Azione Spark eseguita per ultima nel piano di query Spark (DAG).
   - Eventi brevi in ordine cronologico dal driver e dagli esecutori Spark correlati all'eccezione. Se necessario, è possibile trovare dettagli come i messaggi di errore completi, l'attività Spark non riuscita e il relativo ID esecutori che consentono di concentrarsi sul flusso di log dell'esecutore specifico per un'indagine più approfondita.
2. Flusso di informazioni basato su regole:
   - Analisi della causa principale e consigli su come correggere gli errori (ad esempio l'utilizzo di un parametro di lavoro specifico per ottimizzare le prestazioni).
   - Eventi Spark rilevanti come base per l'analisi della causa principale e un'azione consigliata.

> **Note**
>
> Il primo flusso esiste solo se sono disponibili eventi di eccezione Spark per un'esecuzione del processo non riuscita e il secondo stream esisterà solo se sono disponibili informazioni per l'esecuzione del processo non riuscita. Ad esempio, se il processo termina correttamente,

nessuno dei flussi verrà generato. Se il processo fallisce ma non esiste una regola definita dal servizio che può corrispondere allo scenario di errore, verrà generato solo il primo flusso.

Se il job viene creato da AWS Glue Studio, i link agli stream precedenti sono disponibili anche nella scheda dei dettagli del job run (Job run insights) come «Log di errore concisi e consolidati» e «Analisi e guida agli errori».




Esempio di Job Run Insights AWS Glue

In questa sezione, presentiamo un esempio di come la funzione Informazioni dell'esecuzione del processo può essere d'aiuto nella risoluzione di un problema nel processo non riuscito. In questo esempio, un utente ha dimenticato di importare il modulo richiesto (tensorflow) in un AWS Glue job per analizzare e creare un modello di machine learning sui propri dati.

```
import sys
from awsglue.transforms import *
from awsglue.utils import getResolvedOptions
from pyspark.context import SparkContext
from awsglue.context import GlueContext
from awsglue.job import Job
from pyspark.sql.types import *
from pyspark.sql.functions import udf,col

args = getResolvedOptions(sys.argv, ['JOB_NAME'])

sc = SparkContext()
glueContext = GlueContext(sc)
spark = glueContext.spark_session
```

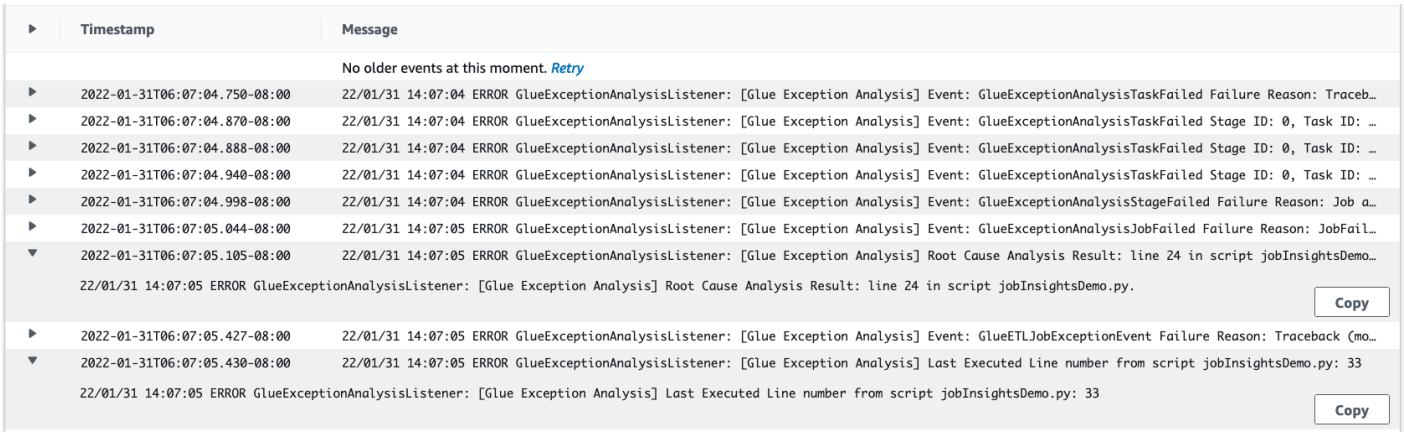
```
job = Job(glueContext)
job.init(args['JOB_NAME'], args)

data_set_1 = [1, 2, 3, 4]
data_set_2 = [5, 6, 7, 8]

scoresDf = spark.createDataFrame(data_set_1, IntegerType())

def data_multiplier_func(factor, data_vector):
    import tensorflow as tf
    with tf.compat.v1.Session() as sess:
        x1 = tf.constant(factor)
        x2 = tf.constant(data_vector)
        result = tf.multiply(x1, x2)
        return sess.run(result).tolist()

data_multiplier_udf = udf(lambda x:data_multiplier_func(x, data_set_2),
 ArrayType(IntegerType(),False))
factoredDf = scoresDf.withColumn("final_value", data_multiplier_udf(col("value")))
print(factoredDf.collect())
```

Senza la funzione di Informazioni dell'esecuzione del processo, data la non riuscita del processo, viene visualizzato solo questo messaggio generato da Spark:

```
An error occurred while calling o111.collectToPython. Traceback (most
recent call last):
```

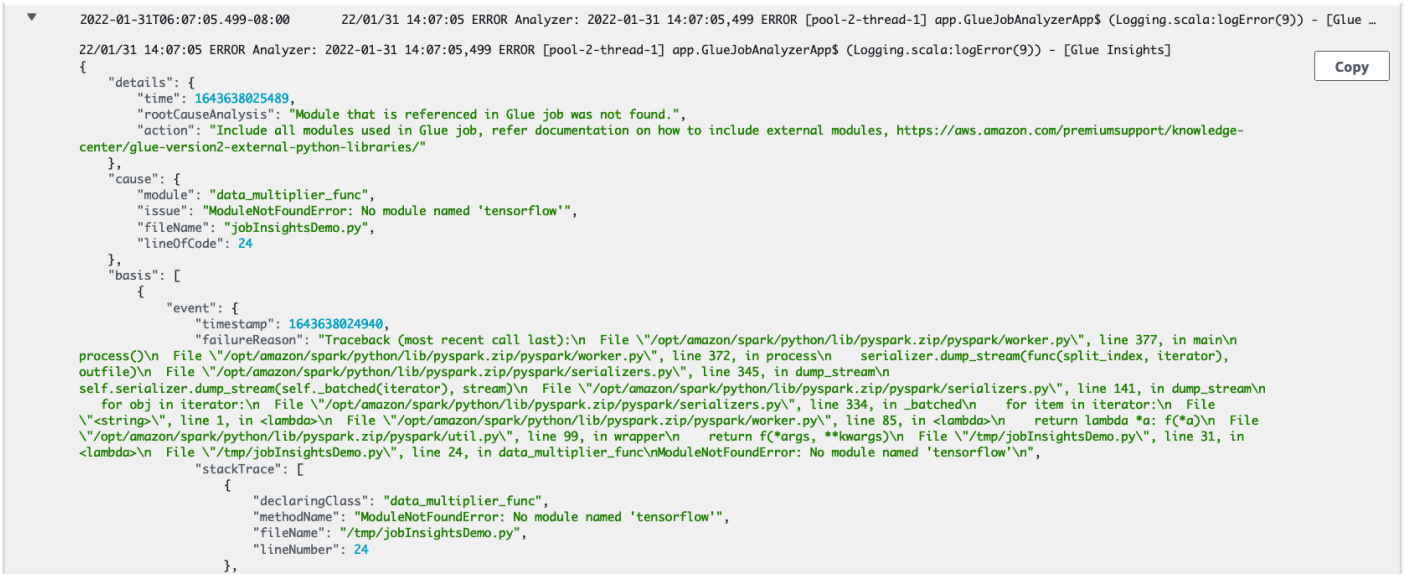
Il messaggio è ambiguo e limita l'esperienza di debug. In questo caso, questa funzionalità fornisce informazioni aggiuntive in due flussi di log: CloudWatch

1. Il flusso di log `job-insights-rca-driver`:
   - Eventi eccezioni: questo flusso di log fornisce gli eventi di eccezione Spark relativi all'errore raccolto dal driver Spark e dai diversi lavoratori distribuiti. Questi eventi ti aiutano a comprendere la propagazione ordinata nel tempo dell'eccezione man mano che il codice difettoso viene eseguito tra le attività, gli esecutori e le fasi di Spark distribuite tra i worker. AWS Glue
   - Numeri riga: questo flusso di log identifica la riga 21, che ha eseguito la chiamata per importare il modulo Python mancante che ha causato l'errore; identifica anche la riga 24, la chiamata all'azione Spark `collect()`, come ultima riga eseguita nello script.




2. Il flusso di log `job-insights-rule-driver`:

   - Causa principale e raccomandazione: oltre al numero di riga e al numero dell'ultima riga eseguita per l'errore nello script, questo flusso di log mostra l'analisi della causa principale e la raccomandazione per seguire il AWS Glue documento e impostare i parametri di lavoro necessari per utilizzare un modulo Python aggiuntivo nel lavoro. AWS Glue
   - Evento base: questo flusso di log mostra anche l'evento di eccezione Spark che è stato valutato con la regola definita dal servizio per dedurre la causa principale e fornire una raccomandazione.




## Monitoraggio con Amazon CloudWatch

Puoi monitorare AWS Glue utilizzando Amazon CloudWatch, che raccoglie ed elabora i dati grezzi AWS Glue trasformandoli in metriche leggibili. near-real-time Queste statistiche vengono registrate per un periodo di due settimane, per permettere l'accesso alle informazioni storiche e offrire

una prospettiva migliore sulle prestazioni del servizio o dell'applicazione Web. Per impostazione predefinita, i dati AWS Glue delle metriche vengono inviati automaticamente a. CloudWatch Per ulteriori informazioni, consulta What Is Amazon CloudWatch? nella Amazon CloudWatch User Guide, eAWS Glue metriche.

Registrazione continua

AWS Glue supporta anche la registrazione continua in tempo reale per i processi AWS Glue. Quando la registrazione continua è abilitata per un lavoro, puoi visualizzare i log in tempo reale sulla AWS Glue console o sulla dashboard della CloudWatch console. Per ulteriori informazioni, consulta Registrazione dei lavori AWS Glue.

Parametri di osservabilità

Quando le metriche di osservabilità del Job sono abilitate, vengono generate Amazon CloudWatch metriche aggiuntive quando il lavoro viene eseguito. Utilizza i parametri AWS Glue di osservabilità per generare approfondimenti su ciò che accade all'interno di AWS Glue e migliorare la classificazione e l'analisi dei problemi.

Argomenti

- Monitoraggio AWS Glue tramite CloudWatch parametri Amazon
- Configurazione degli CloudWatch allarmi Amazon sui profili AWS Glue professionali
- Registrazione dei lavori AWS Glue
- Monitoraggio con parametri AWS Glue di osservabilità

## Monitoraggio AWS Glue tramite CloudWatch parametri Amazon

Puoi creare il profilo delle attività AWS Glue e monitorarle utilizzando il profiler dei processi AWS Glue. Raccoglie ed elabora i dati grezzi dei AWS Glue lavori in metriche leggibili e quasi in tempo reale archiviate in Amazon. CloudWatch Queste statistiche vengono conservate e aggregate in CloudWatch modo da poter accedere alle informazioni storiche per una migliore prospettiva sulle prestazioni dell'applicazione.

> **Note**
>
> È possibile che vengano addebitati costi aggiuntivi quando si abilitano le metriche relative ai lavori e CloudWatch si creano metriche personalizzate. Per ulteriori informazioni, consulta i CloudWatch prezzi di Amazon.

Panoramica dei parametri AWS Glue

Quando interagisci conAWS Glue, invia le metriche a. CloudWatch Puoi visualizzare queste metriche utilizzando la AWS Glue console (il metodo preferito), la dashboard della CloudWatch console o AWS Command Line Interface (AWS CLI).

Per visualizzare i parametri usando il pannello di controllo della console AWS Glue

Puoi visualizzare grafici dettagliati o di riepilogo dei parametri per un processo oppure grafici dettagliati per un'esecuzione di un processo.

1. Accedi a Console di gestione AWS e apri la AWS Glue console all'indirizzo https://console.aws.amazon.com/glue/.
2. Nel riquadro di navigazione, scegli Monitoraggio dell'esecuzione del processo.
3. In Esecuzioni del processo, scegli Operazioni per interrompere un processo attualmente in esecuzione, visualizzare un processo o riavvolgerne il segnalibro.
4. Seleziona un processo, quindi scegli Visualizza dettagli di esecuzione per visualizzare informazioni aggiuntive sull'esecuzione del processo.

Per visualizzare le metriche utilizzando la dashboard della CloudWatch console

I parametri vengono raggruppati prima in base allo spazio dei nomi del servizio e successivamente in base alle diverse combinazioni di dimensioni all'interno di ogni spazio dei nomi.

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione, seleziona Parametri.
3. Selezionare lo spazio dei nomi Glue.

Per visualizzare le metriche utilizzando il AWS CLI

- Al prompt dei comandi utilizza il comando seguente.

```
aws cloudwatch list-metrics --namespace Glue
```

AWS Glueriporta le metriche CloudWatch ogni 30 secondi e i dashboard delle CloudWatch metriche sono configurate per visualizzarle ogni minuto. I parametri AWS Glue rappresentano i valori delta

rispetto ai valori segnalati in precedenza. Se appropriato, i pannelli di controllo dei parametri aggregano (sommano) i valori inviati ogni 30 secondi per ottenere un valore per l'intero ultimo minuto.

AWS Glue comportamento delle metriche per i lavori Spark

I parametri di AWS Glue vengono abilitati al momento dell'inizializzazione di un `GlueContext` in uno script e vengono in genere aggiornati solo al termine di un'attività di Apache Spark. Rappresentano i valori aggregati per tutte le attività di Spark completate fino al momento attuale.

Tuttavia, le metriche Spark che vengono AWS Glue trasmesse a CloudWatch sono generalmente valori assoluti che rappresentano lo stato attuale nel momento in cui vengono segnalate. AWS Glueli riporta CloudWatch ogni 30 secondi e i dashboard delle metriche generalmente mostrano la media dei punti dati ricevuti nell'ultimo minuto.

I nomi dei parametri AWS Glue sono tutti preceduti da uno dei seguenti tipi di prefisso:

- `glue.driver.`: i parametri i cui nomi iniziano con questo prefisso rappresentano parametri AWS Glue aggregati da tutti gli executor nel driver Spark oppure parametri Spark corrispondenti al driver Spark.
- `glue.executorId.`: executorId è il numero di un executor Spark specifico. Corrisponde agli executor elencati nei log.
- `glue.ALL.`: i parametri i cui nomi iniziano con questo prefisso aggregano i valori di tutti gli executor Spark.

AWS Glue metriche

AWS Glue profila e invia le seguenti metriche CloudWatch ogni 30 secondi e la AWS Glue Metrics Dashboard le riporta una volta al minuto:

| Metrica | Description |
| --- | --- |
| `glue.driver.aggregate.bytesRead` | Il numero di byte letti da tutte le origini dati da tutti i processi Spark completati in esecuzione in tutti gli executor.<br><br>Dimensioni valide: `JobName` (il nome del AWS Glue Job), `JobRunId` (l' JobRun ID. or`ALL`) e `Type` (count). |

| Metrica | Description |
| --- | --- |
| | Statistiche valide: SUM (Somma). Questa metrica è un valore delta dell'ultimo valore riportato, quindi nella AWS Glue Metrics Dashboard viene utilizzata una statistica SUM per l'aggregazione.<br><br>Unità: byte<br><br>Può essere utilizzato per monitorare:<br><br>• I byte letti.<br>• L'avanzamento del processo.<br>• L'origine dati JDBC.<br>• Problemi relativi ai segnalibri di processo.<br>• Varianza tra esecuzioni di processo.<br><br>Questo parametro può essere utilizzato come il parametro `glue.ALL.s3.filesystem.read_bytes` , con la differenza che questo viene aggiornato alla fine di un processo Spark e acquisisce anche origini dati non S3. |

| Metrica | Description |
| --- | --- |
| `glue.driver.aggregate.elapsedTime` | Il tempo di ETL trascorso in millisecondi (non include i tempi di bootstrap del processo).<br><br>Dimensioni valide: `JobName` (il nome del AWS Glue Job), `JobRunId` (l' JobRun lD. or`ALL`) e `Type` (count).<br><br>Statistiche valide: SUM (Somma). Questa metrica è un valore delta dell'ultimo valore riportato, quindi nella AWS Glue Metrics Dashboard viene utilizzata una statistica SUM per l'aggregazione.<br><br>Unità: millisecondi<br><br>Può essere utilizzato per determinare il tempo medio di esecuzione di un processo.<br><br>Alcuni modi per utilizzare i dati:<br><br>• Impostare allarmi per i ritardi.<br>• Misurare la varianza tra esecuzioni di processo. |

| Metrica | Description |
| --- | --- |
| `glue.driver.aggregate.numCompletedStages` | Il numero di fasi completate nel processo.<br><br>Dimensioni valide: `JobName` (il nome del AWS Glue Job), `JobRunId` (l' JobRun ID. or`ALL`) e `Type` (count).<br><br>Statistiche valide: SUM (Somma). Questa metrica è un valore delta dell'ultimo valore riportato, quindi nella AWS Glue Metrics Dashboard viene utilizzata una statistica SUM per l'aggregazione.<br><br>Unità: numero<br><br>Può essere utilizzato per monitorare:<br><br>• L'avanzamento del processo.<br>• La sequenza temporale per fase dell'esecuzione del processo, se correlata ad altri parametri.<br><br>Alcuni modi per utilizzare i dati:<br><br>• Identificare fasi impegnative nell'esecuzione di un processo.<br>• Impostare gli allarmi per picchi correlati (fasi impegnative) tra le esecuzioni dei lavori. |

| Metrica | Description |
| --- | --- |
| `glue.driver.aggregate.numCompletedTasks` | Il numero di attività completate nel processo.<br><br>Dimensioni valide: `JobName` (il nome del AWS Glue Job), `JobRunId` (l' JobRun ID. or`ALL`) e `Type` (count).<br><br>Statistiche valide: SUM (Somma). Questa metrica è un valore delta dell'ultimo valore riportato, quindi nella AWS Glue Metrics Dashboard viene utilizzata una statistica SUM per l'aggregazione.<br><br>Unità: numero<br><br>Può essere utilizzato per monitorare:<br><br>• L'avanzamento del processo.<br>• Parallelismo all'interno di una fase. |

| Metrica | Description |
| --- | --- |
| `glue.driver.aggregate.numFailedTasks` | Il numero di processi non riusciti.<br><br>Dimensioni valide: `JobName` (il nome del AWS Glue Job), `JobRunId` (l' JobRun ID. or`ALL`) e `Type` (count).<br><br>Statistiche valide: SUM (Somma). Questa metrica è un valore delta dell'ultimo valore riportato, quindi nella AWS Glue Metrics Dashboard viene utilizzata una statistica SUM per l'aggregazione.<br><br>Unità: numero<br><br>Può essere utilizzato per monitorare:<br><br>• Anomalie dei dati che causano la non riuscita delle attività del processo.<br>• Anomalie del cluster che causano la non riuscita delle attività del processo.<br>• Anomalie dello script che causano la non riuscita delle attività del processo.<br><br>I dati possono essere utilizzati per impostare allarmi per errori maggiori che potrebbero suggerire anomalie nei dati, nel cluster o negli script. |

| Metrica | Description |
| --- | --- |
| `glue.driver.aggregate.numKilledTasks` | Il numero di attività interrotte.<br><br>Dimensioni valide: `JobName` (il nome del AWS Glue Job), `JobRunId` (l' JobRun ID. or`ALL`) e `Type` (count).<br><br>Statistiche valide: SUM (Somma). Questa metrica è un valore delta dell'ultimo valore riportato, quindi nella AWS Glue Metrics Dashboard viene utilizzata una statistica SUM per l'aggregazione.<br><br>Unità: numero<br><br>Può essere utilizzato per monitorare:<br><br>• Anomalie nel Data Skew che provocano eccezioni () che interrompono le attività. OOMs<br>• Anomalie degli script che generano eccezioni () che interrompono le attività. OOMs<br><br>Alcuni modi per utilizzare i dati:<br><br>• Impostare allarmi per errori maggiori che potrebber o suggerire anomalie nei dati.<br>• Impostare allarmi per errori maggiori che potrebber o suggerire anomalie nel cluster.<br>• Impostare allarmi per errori maggiori che potrebber o suggerire anomalie nello script. |

| Metrica | Description |
| --- | --- |
| `glue.driver.aggregate.recordsRead` | Il numero di record letti da tutte le origini dati da tutti i processi Spark completati in esecuzione in tutti gli executor.<br><br>Dimensioni valide: `JobName` (il nome del AWS Glue Job), `JobRunId` (l' JobRun ID. or`ALL`) e `Type` (count).<br><br>Statistiche valide: SUM (Somma). Questa metrica è un valore delta dell'ultimo valore riportato, quindi nella AWS Glue Metrics Dashboard viene utilizzata una statistica SUM per l'aggregazione.<br><br>Unità: numero<br><br>Può essere utilizzato per monitorare:<br><br>• Record letti.<br>• L'avanzamento del processo.<br>• L'origine dati JDBC.<br>• Problemi relativi ai segnalibri di processo.<br>• Differenza nelle esecuzioni dei processi nei giorni.<br><br>Questo parametro può essere utilizzato come il parametro `glue.ALL.s3.filesystem.read_bytes` , con la differenza che questo viene aggiornato alla fine di un processo Spark. |

| Metrica | Description |
| --- | --- |
| `glue.driver.aggregate.shuffleBytesWritten` | Il numero di byte scritti da tutti gli executor per mescolare i dati tra di loro rispetto al report precedente (aggregato dal AWS Glue Metrics Dashboard come numero di byte scritti a questo scopo nel minuto precedente).<br><br>Dimensioni valide: `JobName` (il nome del AWS Glue Job), `JobRunId` (l' JobRun lD. or`ALL`) e `Type` (count).<br><br>Statistiche valide: SUM (Somma). Questa metrica è un valore delta dell'ultimo valore riportato, quindi nella AWS Glue Metrics Dashboard viene utilizzata una statistica SUM per l'aggregazione.<br><br>Unità: byte<br><br>Può essere utilizzato per monitorare: la distribuzione casuale dei dati nei processi (join di grandi dimensioni, groupBy, repartition, coalesce).<br><br>Alcuni modi per utilizzare i dati:<br><br>• Ripartizionare o decomprimere file di input di grandi dimensioni prima di ulteriori elaborazioni.<br>• Ripartizionare i dati in modo più uniforme per evitare i tasti di scelta rapida.<br>• Prefiltrare i dati prima delle operazioni di join o groupBy. |

| Metrica | Description |
| --- | --- |
| `glue.driver.aggregate.shuffleLocalBytesRead` | Il numero di byte letti da tutti gli executor per mescolare i dati tra di loro rispetto al report precedente (aggregato dal AWS Glue Metrics Dashboard come numero di byte letti a tale scopo nel minuto precedente).<br><br>Dimensioni valide: `JobName` (il nome del AWS Glue Job), `JobRunId` (l' JobRun lD. or`ALL`) e `Type` (count).<br><br>Statistiche valide: SUM (Somma). Questa metrica è un valore delta dell'ultimo valore riportato, quindi nella AWS Glue Metrics Dashboard viene utilizzata una statistica SUM per l'aggregazione.<br><br>Unità: byte<br><br>Può essere utilizzato per monitorare: la distribuzione casuale dei dati nei processi (join di grandi dimensioni, groupBy, repartition, coalesce).<br><br>Alcuni modi per utilizzare i dati:<br><br>• Ripartizionare o decomprimere file di input di grandi dimensioni prima di ulteriori elaborazioni.<br>• Ripartizionare i dati in modo più uniforme utilizzando i tasti di scelta rapida.<br>• Prefiltrare i dati prima delle operazioni di join o groupBy. |

| Metrica | Description |
| --- | --- |
| `glue.driver.BlockManager.disk.diskSpaceUsed_MB` | Numero di megabyte di spazio su disco utilizzati in tutti gli executor.<br><br>Dimensioni valide: `JobName` (il nome del AWS Glue Job), `JobRunId` (l' JobRun ID. or`ALL`) e `Type` (gauge).<br><br>Statistiche valide: Average (Media). Si tratta di un parametro Spark, riportato come valore assoluto.<br><br>Unità: megabyte<br><br>Può essere utilizzato per monitorare:<br><br>• Spazio su disco utilizzato per i blocchi che rappresentano partizioni RDD memorizzate nella cache.<br>• Spazio su disco utilizzato per i blocchi che rappresentano uscite shuffle intermedie.<br>• Spazio su disco utilizzato per i blocchi che rappresentano le trasmissioni.<br><br>Alcuni modi per utilizzare i dati:<br><br>• Identificare gli errori dei processi dovuti a un maggiore utilizzo del disco.<br>• Identificare partizioni di grandi dimensioni con conseguente riversamento o distribuzione casuale.<br>• Aumentare la capacità DPU sottoposta a provision ing per risolvere questi problemi. |

| Metrica | Description |
| --- | --- |
| `glue.driver.ExecutorAllocationManager.executors.numberAllExecutors` | Numero di executor di processo attivi.<br><br>Dimensioni valide: `JobName` (il nome del AWS Glue Job), `JobRunId` (l' JobRun ID. or`ALL`) e `Type` (gauge).<br><br>Statistiche valide: Average (Media). Si tratta di un parametro Spark, riportato come valore assoluto.<br><br>Unità: numero<br><br>Può essere utilizzato per monitorare:<br><br>• L'attività dei processi.<br>• Calo degli executor (con solo pochi executor attivi)<br>• Parallelismo attuale a livello di executor.<br><br>Alcuni modi per utilizzare i dati:<br><br>• Ripartizionare o decomprimere i file di input di grandi dimensioni in anticipo se il cluster è sottoutilizzato.<br>• Identificare i ritardi di esecuzione delle fasi o dei processi dovuti a scenari di rallentamento.<br>• • Effettua un confronto con numberMaxNeeded gli Executor per comprendere meglio il backlog e aumentare il provisioning. DPUs |

| Metrica | Description |
| --- | --- |
| `glue.driver.ExecutorAllocationManager.executors.numberMaxNeededExecutors` | Il numero massimo di executor di processo (attivi e in sospeso) necessari per soddisfare il carico corrente.<br><br>Dimensioni valide: `JobName` (il nome del AWS Glue Job), `JobRunId` (l' JobRun ID. or`ALL`) e `Type` (gauge).<br><br>Statistiche valide: Maximum (Massimo). Si tratta di un parametro Spark, riportato come valore assoluto.<br><br>Unità: numero<br><br>Può essere utilizzato per monitorare:<br><br>• L'attività dei processi.<br>• Parallelismo attuale a livello di esecutore e arretrato di attività in sospeso non ancora pianificate a causa dell'indisponibilità degli esecutori a causa della capacità o degli esecutori della DPU. killed/failed<br><br>Alcuni modi per utilizzare i dati:<br><br>• pending/backlog Identificazione della coda di pianificazione.<br>• Identificare i ritardi di esecuzione delle fasi o dei processi dovuti a scenari di rallentamento.<br>• Confrontate con numberAllExecutors per comprendere meglio il backlog e il provisioning. DPUs<br>• Aumentare la capacità DPU sottoposta a provisioning per correggere il backlog dell'executor in sospeso. |

| Metrica | Description |
| --- | --- |
| `glue.driver.jvm.heap.usage`<br><br>`glue.executorId.jvm.heap.usage`<br><br>`glue.ALL.jvm.heap.usage` | Frazione di memoria usata dall'heap JVM per questo driver (dimensione: 0-1) per driver, executor identific ato da executorId o TUTTI gli executor.<br><br>Dimensioni valide: `JobName` (il nome del AWS Glue Job), `JobRunId` (l' JobRun ID. or`ALL`) e `Type` (gauge).<br><br>Statistiche valide: Average (Media). Si tratta di un parametro Spark, riportato come valore assoluto.<br><br>Unità: percentuale<br><br>Può essere utilizzato per monitorare:<br><br>• Utilizzo out-of-memory delle condizioni del driver (OOM). `glue.driver.jvm.heap.usage`<br>• Utilizzo out-of-memory `glue.ALL.jvm.heap.usage` delle condizioni dell'esecutore (OOM).<br><br>Alcuni modi per utilizzare i dati:<br><br>• Identificare gli ID e le fasi dell'executor che consumano memoria.<br>• Identificare gli ID e le fasi dell'executor in calo.<br>• Identifica una out-of-memory condizione del driver (OOM).<br>• Identifica una out-of-memory condizione dell'esec utore (OOM) e ottieni l'ID dell'esecutore corrispon dente in modo da poter ottenere una traccia dello stack dal log dell'esecutore.<br>• Identifica i file o le partizioni che possono presentar e una distorsione dei dati che causa ritardi o condizioni (). out-of-memory OOMs |

| Metrica | Description |
| --- | --- |
| `glue.driver.jvm.heap.used`<br><br>`glue.executorId.jvm.heap.used`<br><br>`glue.ALL.jvm.heap.used` | Il numero di byte di memoria utilizzati dall'heap JVM per il driver, l'executor identificato da executorId o TUTTI gli executor.<br><br>Dimensioni valide: `JobName` (il nome del AWS Glue Job), `JobRunId` (l' JobRun ID. or`ALL`) e `Type` (gauge).<br><br>Statistiche valide: Average (Media). Si tratta di un parametro Spark, riportato come valore assoluto.<br><br>Unità: byte<br><br>Può essere utilizzato per monitorare:<br><br>• out-of-memoryCondizioni del conducente (OOM).<br>• out-of-memoryCondizioni dell'esecutore (OOM).<br><br>Alcuni modi per utilizzare i dati:<br><br>• Identificare gli ID e le fasi dell'executor che consumano memoria.<br>• Identificare gli ID e le fasi dell'executor in calo.<br>• Identifica una out-of-memory condizione del driver (OOM).<br>• Identifica una out-of-memory condizione dell'esec utore (OOM) e ottieni l'ID dell'esecutore corrispon dente in modo da poter ottenere una traccia dello stack dal log dell'esecutore.<br>• Identifica i file o le partizioni che possono presentar e una distorsione dei dati che causa ritardi o condizioni (). out-of-memory OOMs |

| Metrica | Description |
| --- | --- |
| `glue.driver.s3.filesystem.read_bytes`<br><br>`glue.executorId.s3.filesystem.read_bytes`<br><br>`glue.ALL.s3.filesystem.read_bytes` | Il numero di byte letti da Amazon S3 dal driver, da un executor identificato da ExecutorID o da ALL executor rispetto al report precedente (aggregato dal Metrics Dashboard come il numero di byte letti AWS Glue nel minuto precedente).<br><br>Dimensioni valide: `JobName`, `JobRunId` e `Type` (valutazione).<br><br>Statistiche valide: SUM (Somma). Questa metrica è un valore delta dell'ultimo valore riportato, quindi nella Metrics Dashboard viene utilizzata una statistic a SUM per l'aggregazione AWS Glue . L'area sotto la curva nella AWS Glue Metrics Dashboard può essere utilizzata per confrontare visivamente i byte letti da due diverse esecuzioni di lavoro.<br><br>Unità: byte.<br><br>Può essere utilizzato per monitorare:<br><br>• Spostamento di dati ETL.<br>• L'avanzamento del processo.<br>• Problemi relativi ai segnalibri di processo (dati elaborati, rielaborati e saltati).<br>• Confronto tra letture e velocità di importazione da origini dati esterne.<br>• Varianza tra esecuzioni di processo.<br><br>I dati risultanti possono essere utilizzati per:<br><br>• Pianificazione della capacità DPU.<br>• Impostare gli allarmi per picchi o riduzioni di grandi dimensioni nei dati letti per le esecuzioni le fasi dei processi. |

| Metrica | Description |
| --- | --- |
| `glue.driver.s3.filesystem.write_bytes`<br><br>`glue.executorId.s3.filesystem.write_bytes`<br><br>`glue.ALL.s3.filesystem.write_bytes` | Il numero di byte scritti su Amazon S3 dal driver, da un executor identificato da ExecutorID o da ALL executor a partire dal report precedente (aggregato dal Metrics Dashboard come il numero di byte scritti AWS Glue nel minuto precedente).<br><br>Dimensioni valide: `JobName`, `JobRunId` e `Type` (valutazione).<br><br>Statistiche valide: SUM (Somma). Questa metrica è un valore delta dell'ultimo valore riportato, quindi nella Metrics Dashboard viene utilizzata una statistic a SUM per l'aggregazione AWS Glue . L'area sotto la curva nella AWS Glue Metrics Dashboard può essere utilizzata per confrontare visivamente i byte scritti da due diverse esecuzioni di job.<br><br>Unità: byte<br><br>Può essere utilizzato per monitorare:<br><br>• Spostamento di dati ETL.<br>• L'avanzamento del processo.<br>• Problemi relativi ai segnalibri di processo (dati elaborati, rielaborati e saltati).<br>• Confronto tra letture e velocità di importazione da origini dati esterne.<br>• Varianza tra esecuzioni di processo.<br><br>Alcuni modi per utilizzare i dati:<br><br>• Pianificazione della capacità DPU.<br>• Impostare gli allarmi per picchi o riduzioni di grandi dimensioni nei dati letti per le esecuzioni le fasi dei processi. |

| Metrica | Description |
| --- | --- |
| `glue.driver.streaming.numRecords` | Numero di record ricevuti in un micro-batch. Questa metrica è disponibile solo per i lavori di AWS Glue streaming con la AWS Glue versione 2.0 e successive.<br><br>Dimensioni valide: `JobName` (il nome del AWS Glue lavoro), `JobRunId` (l' JobRun ID. or`ALL`) e `Type` (count).<br><br>Valid Statistics: Sum (Somma), Maximum (Massimo), Minimum (Minimo), Average (Media), Percentile (Percentuale)<br><br>Unità: numero<br><br>Può essere utilizzato per monitorare:<br><br>• Record letti.<br>• L'avanzamento del processo. |

| Metrica | Description |
| --- | --- |
| `glue.driver.streaming.batchProcessingTimeInMs` | Il tempo necessario per elaborare i batch in millisecondi. Questa metrica è disponibile solo per i lavori di AWS Glue streaming con la AWS Glue versione 2.0 e successive.<br><br>Dimensioni valide: `JobName` (il nome del AWS Glue lavoro), `JobRunId` (l' JobRun ID. or`ALL`) e `Type` (count).<br><br>Valid Statistics: Sum (Somma), Maximum (Massimo), Minimum (Minimo), Average (Media), Percentile (Percentuale)<br><br>Unità: numero<br><br>Può essere utilizzato per monitorare:<br><br>• L'avanzamento del processo.<br>• Prestazioni dello script. |

| Metrica | Description |
| --- | --- |
| `glue.driver.system.cpuSystemLoad`<br><br>`glue.executorId.system.cpuSystemLoad`<br><br>`glue.ALL.system.cpuSystemLoad` | Frazione del carico di sistema della CPU usata (dimensione: 0-1) dal driver, da un executor identificato da executorId o da tutti gli executor.<br><br>Dimensioni valide: `JobName` (il nome del AWS Glue lavoro), `JobRunId` (l' JobRun ID. or`ALL`) e `Type` (gauge).<br><br>Statistiche valide: Average (Media). Questo parametro è riportato come valore assoluto.<br><br>Unità: percentuale<br><br>Può essere utilizzato per monitorare:<br><br>• Carico della CPU del driver.<br>• Carico della CPU dell'executor.<br>• Rilevamento di executor o fasi associati alla CPU o all'IO in un processo.<br><br>Alcuni modi per utilizzare i dati:<br><br>• Pianificazione della capacità della DPU insieme ai parametri IO (Bytes Read/Shuffle Bytes, Task Parallelism) e alla metrica del numero massimo di esecutori necessari.<br>• Identificare il rapporto CPU/IO. Questo consente il ripartizionamento e l'aumento della capacità di provisioning per i processi a esecuzione prolungata con set di dati suddivisibili con minore utilizzo della CPU. |

Dimensioni per le metriche AWS Glue

AWS Glue le metriche utilizzano lo spazio dei AWS Glue nomi e forniscono metriche per le seguenti dimensioni:

| Dimensione | Description |
| --- | --- |
| `JobName` | Questa dimensione filtra le metriche di tutte le esecuzioni di un processo specifico. AWS Glue |
| `JobRunId` | Questa dimensione filtra le metriche di un AWS Glue lavoro specifico eseguito da un JobRun ID o. `ALL` |
| `Type` | Questa dimensione filtra i parametri in base a `count` (numero aggregato) o `gauge` (valore in un determina to momento). |

Per ulteriori informazioni, consulta la Amazon CloudWatch User Guide.

Configurazione degli CloudWatch allarmi Amazon sui profili AWS Glue professionali

AWS Gluele metriche sono disponibili anche in Amazon CloudWatch. È possibile configurare allarmi per qualsiasi parametro AWS Glue per i processi pianificati.

Di seguito sono illustrati alcuni scenari comuni per l'impostazione degli allarmi:

- Processi che stanno per esaurire la memoria: è possibile impostare un allarme quando l'utilizzo della memoria supera la media normale per il driver o un executor per un processo AWS Glue.
- Calo degli executor: è possibile impostare un allarme quando il numero di executor scende sotto una determinata soglia per un intervallo di tempo lungo in un processo AWS Glue.
- Backlog o rielaborazione dei dati: confronta le metriche dei singoli lavori in un flusso di lavoro utilizzando un'espressione matematica. CloudWatch È quindi possibile attivare un allarme in base al valore dell'espressione risultante (ad esempio il rapporto tra byte scritti da un processo e byte letti dal processo seguente).

Per istruzioni dettagliate sull'impostazione degli allarmi, consulta Creare o modificare un CloudWatch allarme nella Guida per l'utente di Amazon CloudWatch Events.

Per l'utilizzo di scenari di monitoraggio e debug, consulta. CloudWatch Monitoraggio e debug dei processi

Registrazione dei lavori AWS Glue

Nella AWS Glue versione 5.0, tutti i lavori dispongono di funzionalità di registrazione in tempo reale. È inoltre possibile specificare le opzioni di configurazione personalizzate per personalizzare il comportamento di log. Queste opzioni includono l'impostazione del nome del gruppo di Amazon CloudWatch log, del prefisso del flusso di Amazon CloudWatch log (che precederà l'ID e l'ID del AWS Glue job run) e driver/executor il modello di conversione dei log per i messaggi di log. Queste configurazioni consentono di aggregare i log in gruppi di log Amazon CloudWatch personalizzati con politiche di scadenza diverse. Inoltre, è possibile analizzare i log in modo più efficace utilizzando prefissi e modelli di conversione personalizzati per i flussi di log. Questo livello di personalizzazione consente di ottimizzare la gestione e l'analisi dei log in base ai requisiti specifici.

Comportamento di registrazione nella versione 5.0 AWS Glue

Per impostazione predefinita, i log di sistema, i log dei daemon Spark e i log dei AWS Glue logger degli utenti vengono scritti nel gruppo di log in. `/aws-glue/jobs/error` Amazon CloudWatch D'altra parte, i log degli utenti stdout (standard output) e stderr (standard error) vengono scritti nel gruppo di log `/aws-glue/jobs/output` per impostazione predefinita.

Registrazione personalizzata

È possibile personalizzare i prefissi predefiniti del gruppo di log e del flusso di log utilizzando i seguenti argomenti di lavoro:

- `--custom-logGroup-prefix`: consente di specificare un prefisso personalizzato per i gruppi di log `/aws-glue/jobs/error` e `/aws-glue/jobs/output`. Se si fornisce un prefisso personalizzato, i nomi dei gruppi di log avranno il seguente formato:
  - `/aws-glue/jobs/error` sarà *`<customer prefix>`*`/error`
  - `/aws-glue/jobs/output` sarà *`<customer prefix>`*`/output`
- `--custom-logStream-prefix`: consente di specificare un prefisso personalizzato per i nomi dei flussi di log all'interno dei gruppi di log. Se si fornisce un prefisso personalizzato, i nomi dei flussi di log avranno il seguente formato:
  - `jobrunid-driver` sarà *`<customer log stream>`*`-driver`
  - `jobrunid-executorNum` sarà *`<customer log stream>`*`-executorNum`

Regole e limitazioni di convalida per i prefissi personalizzati:

- L'intero nome del flusso di log deve contenere da 1 a 512 caratteri.
- Il prefisso personalizzato stesso è limitato a 400 caratteri.
- Il prefisso personalizzato deve corrispondere al modello di espressione regolare "[^: *] *" (i caratteri speciali consentiti sono "_", "-" e "/").

Registrazione di messaggi specifici di applicazioni tramite logger di script personalizzato

È possibile utilizzare il AWS Glue logger per registrare tutti i messaggi specifici dell'applicazione nello script che vengono inviati in tempo reale al flusso di registro del driver.

Il seguente esempio mostra uno script Python.

```
from awsglue.context import GlueContext
from pyspark.context import SparkContext

sc = SparkContext()
glueContext = GlueContext(sc)
logger = glueContext.get_logger()
logger.info("info message")
logger.warn("warn message")
logger.error("error message")
```

Il seguente esempio mostra uno script Scala.

```
import com.amazonaws.services.glue.log.GlueLogger

object GlueApp {
  def main(sysArgs: Array[String]) {
    val logger = new GlueLogger
    logger.info("info message")
    logger.warn("warn message")
    logger.error("error message")
  }
}
```

Abilitazione della barra di avanzamento per visualizzare l'avanzamento del processo

AWS Glue fornisce una barra di avanzamento in tempo reale sotto il flusso di `JOB_RUN_ID-progress-bar` log per controllare AWS Glue lo stato di esecuzione del lavoro. Al momento,

supporta solo i processi che inizializzano `glueContext`. Se esegui un processo Spark puro senza inizializzarlo`glueContext`, la barra di AWS Glue avanzamento non viene visualizzata.

La barra di avanzamento mostra il seguente aggiornamento dell'avanzamento ogni 5 secondi.

```
Stage Number (Stage Name): > (numCompletedTasks + numActiveTasks) /
 totalNumOfTasksInThisStage]
```

Configurazione di sicurezza con registrazione Amazon CloudWatch

Quando una configurazione di sicurezza è abilitata per Amazon CloudWatch i log, AWS Glue crea gruppi di log con modelli di denominazione specifici che incorporano il nome della configurazione di sicurezza.

Denominazione dei gruppi di log con configurazione di sicurezza

I gruppi di log predefiniti e personalizzati saranno i seguenti:

- Gruppo di log degli errori predefinito: `/aws-glue/jobs/`*`Security-Configuration-Name`*`-role/`*`glue-job-role`*`/error`
- Gruppo di log di output predefinito: `/aws-glue/jobs/`*`Security-Configuration-Name`*`-role/`*`glue-job-role`*`/output`
- Gruppo di log degli errori personalizzato (AWS Glue 5.0): *`custom-log-group-prefix`*`/`*`Security-Configuration-Name`*`-role/`*`glue-job-role`*`/error`
- Gruppo di log di output personalizzato (AWS Glue 5.0): *`custom-log-group-prefix`*`/`*`Security-Configuration-Name`*`-role/`*`glue-job-role`*`/output`

Autorizzazioni IAM richieste

Se si abilita una configurazione di sicurezza con Amazon CloudWatch Logs, è necessario aggiungere l'autrozzazione `logs:AssociateKmsKey` alle autorizzazioni del ruolo IAM. Se tale autorizzazione non è inclusa, la registrazione continua verrà disabilitata.

Inoltre, per configurare la crittografia per Amazon CloudWatch i log, segui le istruzioni in Encrypt Log Data in Amazon CloudWatch Logs Using nella Amazon Amazon CloudWatch Logs AWS Key Management Service User Guide.

Informazioni aggiuntive

Per ulteriori informazioni sulla creazione di configurazioni di sicurezza, consulta Gestione delle configurazioni di sicurezza sulla console. AWS Glue

Argomenti

- Abilitazione della registrazione continua per i AWS Glue lavori 4.0 e precedenti
- Visualizzazione dei registri dei lavori AWS Glue

Abilitazione della registrazione continua per i AWS Glue lavori 4.0 e precedenti

> **Note**
>
> Nella AWS Glue 4.0 e nelle versioni precedenti, la registrazione continua era una funzionalità disponibile. Tuttavia, con l'introduzione della AWS Glue versione 5.0, tutti i lavori dispongono di funzionalità di registrazione in tempo reale. Per ulteriori dettagli sulle funzionalità di registrazione e sulle opzioni di configurazione in AWS Glue 5.0, vedere Logging for jobs. AWS Glue

È possibile abilitare la registrazione continua utilizzando la AWS Glue console o tramite (). AWS Command Line Interface AWS CLI

Puoi abilitare la registrazione continua al momento della creazione di un nuovo processo e della modifica di un processo esistente oppure puoi abilitarla attraverso la AWS CLI.

È inoltre possibile specificare opzioni di configurazione personalizzate come il nome del gruppo di Amazon CloudWatch log, il prefisso del flusso di CloudWatch log prima dell' driver/executor ID del AWS Glue job run e il modello di conversione dei log per i messaggi di log. Queste configurazioni consentono di impostare log aggregati in gruppi di CloudWatch log personalizzati con politiche di scadenza diverse e di analizzarli ulteriormente con prefissi e modelli di conversione personalizzati per i flussi di log.

Argomenti

- Utilizzando il Console di gestione AWS
- Registrazione di messaggi specifici di applicazioni tramite logger di script personalizzato
- Abilitazione della barra di avanzamento per visualizzare l'avanzamento del processo

- Configurazione di sicurezza con la registrazione continua.

Utilizzando il Console di gestione AWS

Segui questi passaggi per utilizzare la console per abilitare la registrazione continua durante la creazione o la modifica di un AWS Glue lavoro.

Per creare un nuovo AWS Glue lavoro con registrazione continua

1. Accedi a Console di gestione AWS e apri la AWS Glue console all'indirizzo https://console.aws.amazon.com/glue/.
2. Nel riquadro di navigazione scegli Processi ETL.
3. Scegli ETL visivo.
4. Nella scheda Dettagli del lavoro, espandi la sezione Proprietà avanzate.
5. In Registrazione continua seleziona Abilita accessi. CloudWatch

Per abilitare la registrazione continua per un lavoro esistente AWS Glue

1. Apri la AWS Glue console all'indirizzo https://console.aws.amazon.com/glue/.
2. Nel riquadro di navigazione scegliere Jobs (Processi).
3. Scegliere un processo esistente dall'elenco Jobs (Processi).
4. Scegliere Action (Operazione), Edit job (Modifica processo).
5. Nella scheda Dettagli del lavoro, espandi la sezione Proprietà avanzate.
6. In Registrazione continua seleziona Abilita accessi. CloudWatch

Usando il AWS CLI

Per abilitare la registrazione continua, si passano i parametri del processo a un AWS Glue lavoro. Passate i seguenti parametri di lavoro speciali in modo simile agli altri parametri di AWS Glue lavoro. Per ulteriori informazioni, consulta Utilizzo dei parametri del lavoro nei lavori AWS Glue.

```
'--enable-continuous-cloudwatch-log': 'true'
```

Puoi specificare un nome di gruppo di CloudWatch log Amazon personalizzato. Se non specificato, il nome predefinito del gruppo di log è `/aws-glue/jobs/logs-v2`.

```
'--continuous-log-logGroup': 'custom_log_group_name'
```

Puoi specificare un prefisso Amazon CloudWatch Log Stream personalizzato. Se non specificato, il prefisso del flusso di log predefinito è l'ID di esecuzione del processo.

```
'--continuous-log-logStreamPrefix': 'custom_log_stream_prefix'
```

È possibile specificare un modello di conversione di registrazione continua personalizzato. Se non specificato, il modello di conversione predefinito è `%d{yy/MM/dd HH:mm:ss} %p %c{1}: %m%n`. Tieni presente che il modello di conversione si applica solo ai log dei driver e ai log delle esecuzioni. Non interessa la barra di avanzamento di AWS Glue .

```
'--continuous-log-conversionPattern': 'custom_log_conversion_pattern'
```

Registrazione di messaggi specifici di applicazioni tramite logger di script personalizzato

È possibile utilizzare il AWS Glue logger per registrare tutti i messaggi specifici dell'applicazione nello script che vengono inviati in tempo reale al flusso di registro del driver.

Il seguente esempio mostra uno script Python.

```
from awsglue.context import GlueContext
from pyspark.context import SparkContext

sc = SparkContext()
glueContext = GlueContext(sc)
logger = glueContext.get_logger()
logger.info("info message")
logger.warn("warn message")
logger.error("error message")
```

Il seguente esempio mostra uno script Scala.

```
import com.amazonaws.services.glue.log.GlueLogger

object GlueApp {
  def main(sysArgs: Array[String]) {
    val logger = new GlueLogger
    logger.info("info message")
    logger.warn("warn message")
```

```
    logger.error("error message")
  }
}
```

Abilitazione della barra di avanzamento per visualizzare l'avanzamento del processo

AWS Glue fornisce una barra di avanzamento in tempo reale sotto il flusso di `JOB_RUN_ID-progress-bar` log per controllare AWS Glue lo stato di esecuzione del lavoro. Al momento, supporta solo i processi che inizializzano `glueContext`. Se esegui un processo Spark puro senza inizializzarlo`glueContext`, la barra di AWS Glue avanzamento non viene visualizzata.

La barra di avanzamento mostra il seguente aggiornamento dell'avanzamento ogni 5 secondi.

```
Stage Number (Stage Name): > (numCompletedTasks + numActiveTasks) /
 totalNumOfTasksInThisStage]
```

Configurazione di sicurezza con la registrazione continua.

Se è abilitata una configurazione di sicurezza per CloudWatch i log, AWS Glue creerà un gruppo di log denominato come segue per i log continui:

```
<Log-Group-Name>-<Security-Configuration-Name>
```

I gruppi di log predefiniti e personalizzati saranno i seguenti:

- Il gruppo di log continuo di default sarà `/aws-glue/jobs/error-`*`<Security-Configuration-Name>`*
- Il gruppo di log continuo di default sarà *`<custom-log-group-name>`*`-`*`<Security-Configuration-Name>`*

È necessario aggiungere le autorizzazioni `logs:AssociateKmsKey` al ruolo IAM, se si abilita una configurazione di sicurezza con Logs. CloudWatch Se tale autorizzazione non è inclusa, la registrazione continua verrà disabilitata. Inoltre, per configurare la crittografia per CloudWatch i log, segui le istruzioni in Encrypt Log Data in CloudWatch Logs Using nella Amazon CloudWatch Logs AWS Key Management Service User Guide.

Per ulteriori informazioni sulla creazione delle configurazioni di sicurezza, consulta Gestire le configurazioni di sicurezza nella console AWS Glue.

> **Note**
>
> È possibile che vengano addebitati costi aggiuntivi quando si abilita la registrazione e vengono creati eventi di registro aggiuntivi. CloudWatch Per ulteriori informazioni, consulta i CloudWatch prezzi di Amazon.

Visualizzazione dei registri dei lavori AWS Glue

Puoi visualizzare i log in tempo reale utilizzando la AWS Glue console o la CloudWatch console Amazon.

Per visualizzare i log in tempo reale utilizzando la dashboard della console AWS Glue

1. Accedi Console di gestione AWS e apri la AWS Glue console all'indirizzo https://console.aws.amazon.com/glue/.
2. Nel riquadro di navigazione scegliere Jobs (Processi).
3. Aggiungere o avviare un processo esistente. Scegliere Action (Operazione), Run job (Esegui processo).

   Quando avvii l'esecuzione di un processo, puoi accedere a una pagina che contiene informazioni sul processo in esecuzione:

   - La scheda Logs (Log) mostra i precedenti log dell'applicazione aggregati.
   - La scheda Log mostra una barra di avanzamento in tempo reale quando il processo è in esecuzione con `glueContext` inizializzato.
   - La scheda Registri contiene anche i registri dei driver, che acquisiscono i log dei driver Apache Spark in tempo reale, e i registri delle applicazioni dallo script registrato utilizzando il logger dell' AWS Glue applicazione quando il processo è in esecuzione.
4. Per processi precedenti, è anche possibile visualizzare i log in tempo reale nella vista Job History (Cronologia processi) scegliendo Logs (Log). Questa azione ti porta alla CloudWatch console che mostra tutti i flussi di log del driver Spark, dell'executor e della barra di avanzamento relativi all'esecuzione del job.

Per visualizzare i log in tempo reale utilizzando la dashboard della console CloudWatch

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.

2. Nel riquadro di navigazione selezionare Log.
3. Scegli il gruppo /aws- glue/jobs/error /log.
4. Nella casella Filter (Filtro), incollare l'ID dell'esecuzione del processo.

   È possibile visualizzare i log di driver, i log di executor e la barra di avanzamento (se si utilizza Standard filter (Filtro standard)).

Monitoraggio con parametri AWS Glue di osservabilità

> **Note**
>
> I parametri di osservabilità AWS Glue sono disponibili in AWS Glue 4.0 e versioni successive.

Utilizza i parametri AWS Glue di osservabilità per generare approfondimenti su ciò che accade all'interno di AWS Glue per i processi di Apache Spark e migliorare la classificazione e l'analisi dei problemi. I parametri di osservabilità vengono visualizzati tramite i pannelli di controllo Amazon CloudWatch e possono essere utilizzati per aiutare a eseguire l'analisi delle cause principali degli errori e diagnosticare i rallentamenti delle prestazioni. È possibile ridurre il tempo impiegato per il debug dei problemi su larga scala così da poterti concentrare sulla risoluzione dei problemi in modo più rapido ed efficace.

AWS GlueL'osservabilità fornisce Amazon CloudWatch metriche classificate nei seguenti quattro gruppi:

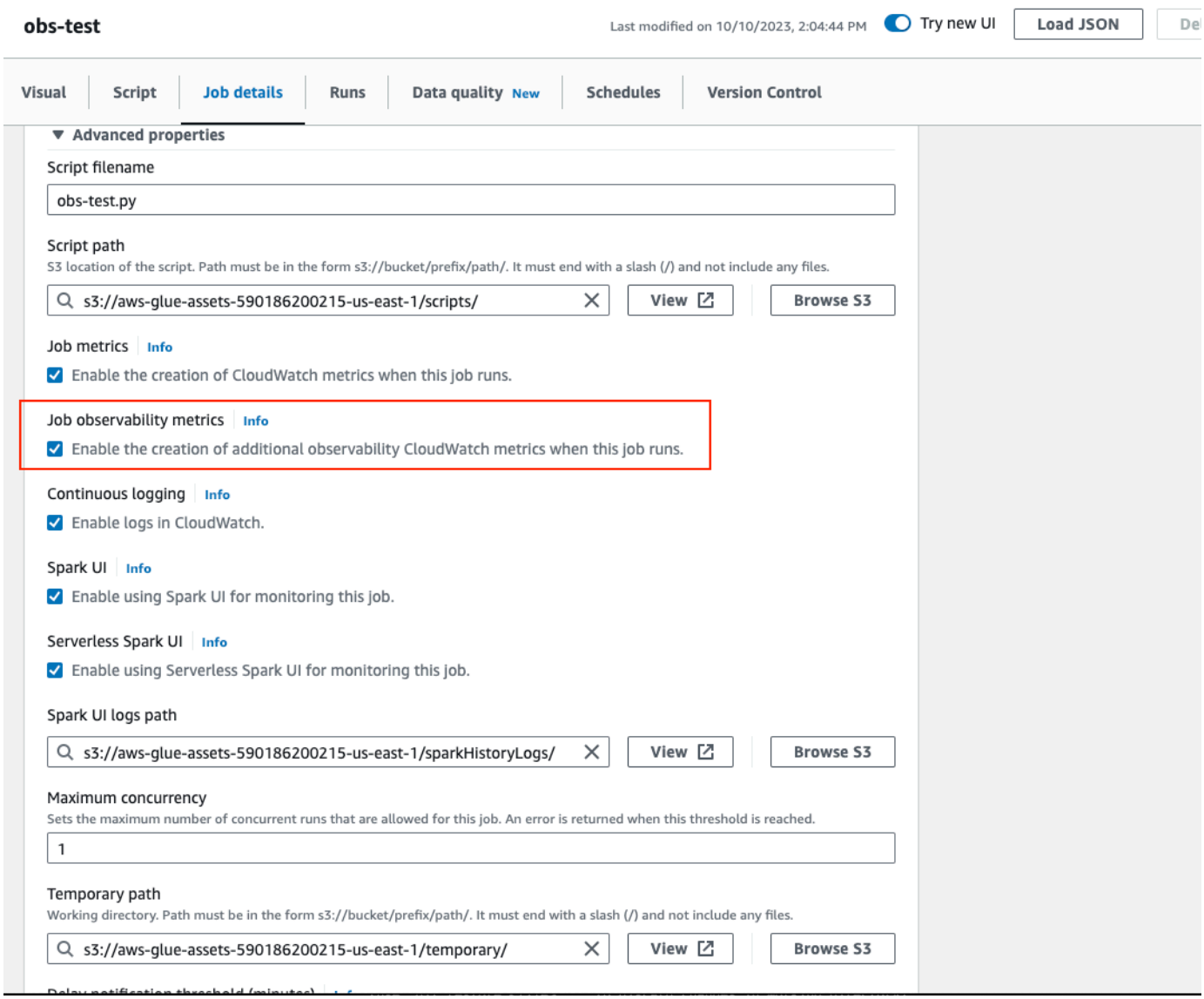
- Affidabilità (ad esempio, classi di errori): identifica facilmente i motivi di errore più comuni in un determinato intervallo di tempo che potresti voler risolvere.
- Prestazioni (ad esempio, asimmetria): individua un ostacolo prestazionale e applica tecniche di ottimizzazione. Ad esempio, quando riscontri un peggioramento delle prestazioni a causa dell'asimmetria del processo, potresti voler abilitare l'esecuzione delle query adattive Spark e ottimizzare la soglia di unione skew.
- Throughput (ovvero, per source/sink throughput): monitora le tendenze delle letture e scritture dei dati. Puoi anche configurare Amazon CloudWatch allarmi per anomalie.
- Utilizzo delle risorse (ad esempio, personale, utilizzo della memoria e del disco): individuazione efficiente dei processi con un basso utilizzo della capacità. Potresti voler abilitare il dimensionamento automatico AWS Glue per questi processi.

Guida introduttiva ai parametri AWS Glue di osservabilità

> **Note**
>
> I nuovi parametri sono abilitati per impostazione predefinita nella console AWS Glue Studio.

Per configurare i parametri di osservabilità in AWS Glue Studio:

1. Accedi alla console AWS Glue e scegli processi ETL dal menu della console.
2. Scegli un processo facendo clic sul suo nome nella sezione I tuoi processi.
3. Seleziona la scheda Job details (Dettagli del processo).
4. Scorri verso il basso e scegli Proprietà avanzate, quindi Parametri di osservabilità del processo.




Per abilitare le metriche di AWS Glue osservabilità utilizzando: AWS CLI

- Aggiungi alla mappa `--default-arguments` il seguente valore-chiave nel file JSON di input:

```
--enable-observability-metrics, true
```

Utilizzo dell'osservabilità AWS Glue

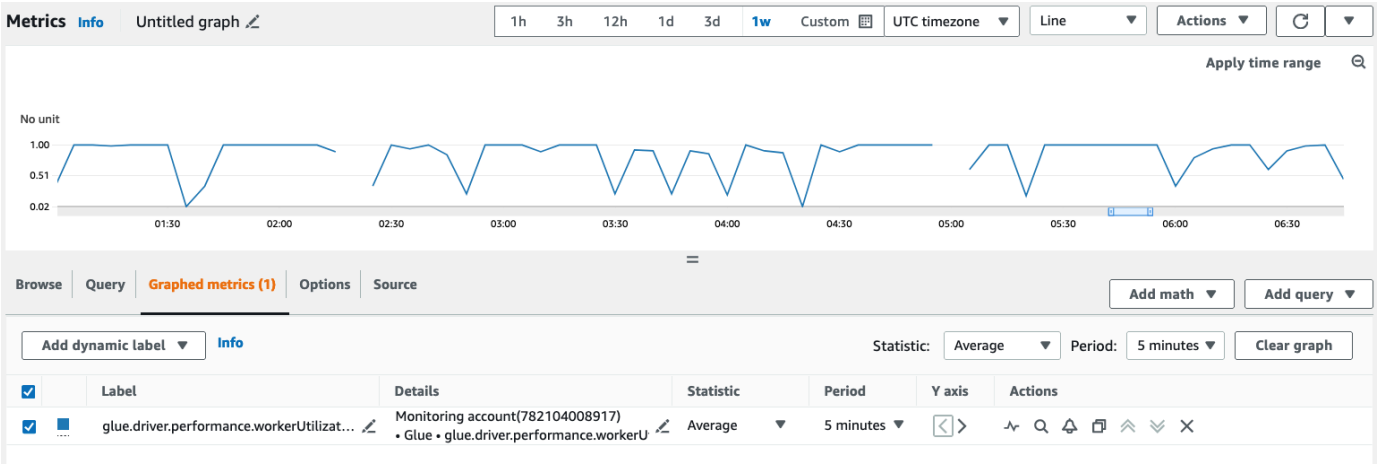
Poiché le metriche AWS Glue di osservabilità vengono fornite tramite Amazon CloudWatch, puoi utilizzare la Amazon CloudWatch console, l'SDK o l'API per interrogare i AWS CLI datapoint delle

metriche di osservabilità. Consulta Utilizzo dell'osservabilità Glue per monitorare l'utilizzo delle risorse per ridurre i costi per un caso d'uso di esempio su quando utilizzare i parametri AWS Glue di osservabilità.

Utilizzo dell'osservabilità nella console AWS Glue Amazon CloudWatch

Per interrogare e visualizzare le metriche nella console: Amazon CloudWatch

1. Apri la Amazon CloudWatch console e scegli Tutte le metriche.
2. In Spazi dei nomi personalizzati, seleziona AWS Glue.
3. Scegli Parametri di osservabilità del processo, Parametri di osservabilità per origine oppure Parametri di osservabilità per Sink.
4. Cerca il nome specifico del parametro, il nome del processo, l'ID di esecuzione del processo e selezionali.
5. Nella scheda Parametri nel grafico, configura la statistica, il periodo e altre opzioni che preferisci.




Per interrogare una metrica di osservabilità utilizzando: AWS CLI

1. Crea un file JSON di definizione dei parametri e sostituisci `your-Glue-job-name` e `your-Glue-job-run-id` con quelli pertinenti.

```
$ cat multiplequeries.json
[
    {
        "Id": "avgWorkerUtil_0",
        "MetricStat": {
            "Metric": {
                "Namespace": "Glue",
```

```
                "MetricName": "glue.driver.workerUtilization",
                "Dimensions": [
                    {
                        "Name": "JobName",
                        "Value": "<your-Glue-job-name-A>"
                    },
                    {
                        "Name": "JobRunId",
                        "Value": "<your-Glue-job-run-id-A>"
                    },
                    {
                        "Name": "Type",
                        "Value": "gauge"
                    },
                    {
                        "Name": "ObservabilityGroup",
                        "Value": "resource_utilization"
                    }
                ]
            },
            "Period": 1800,
            "Stat": "Minimum",
            "Unit": "None"
        }
    },
    {
        "Id": "avgWorkerUtil_1",
        "MetricStat": {
            "Metric": {
                "Namespace": "Glue",
                "MetricName": "glue.driver.workerUtilization",
                "Dimensions": [
                    {
                        "Name": "JobName",
                        "Value": "<your-Glue-job-name-B>"
                    },
                    {
                        "Name": "JobRunId",
                        "Value": "<your-Glue-job-run-id-B>"
                    },
                    {
                        "Name": "Type",
                        "Value": "gauge"
                    },
```

```
                    {
                        "Name": "ObservabilityGroup",
                        "Value": "resource_utilization"
                    }
                ]
            },
            "Period": 1800,
            "Stat": "Minimum",
            "Unit": "None"
        }
    }
]
```

2. Eseguire il comando `get-metric-data`:

```
$ aws cloudwatch get-metric-data --metric-data-queries file: //multiplequeries.json
 \
     --start-time '2023-10-28T18: 20' \
     --end-time '2023-10-28T19: 10'  \
     --region us-east-1
{
    "MetricDataResults": [
        {
            "Id": "avgWorkerUtil_0",
            "Label": "<your-label-for-A>",
            "Timestamps": [
                "2023-10-28T18:20:00+00:00"
            ],
            "Values": [
                0.06718750000000001
            ],
            "StatusCode": "Complete"
        },
        {
            "Id": "avgWorkerUtil_1",
            "Label": "<your-label-for-B>",
            "Timestamps": [
                "2023-10-28T18:50:00+00:00"
            ],
            "Values": [
                0.5959183673469387
            ],
```

```
            "StatusCode": "Complete"
        }
    ],
    "Messages": []
}
```

Parametri di osservabilità

AWS GlueL'osservabilità profila e invia le seguenti metriche Amazon CloudWatch ogni 30 secondi e alcune di queste metriche possono essere visibili nella pagina AWS Glue Studio Job Runs Monitoring.

| Metrica | Description | Categoria |
| --- | --- | --- |
| glue.driver.skewness.stage | Categoria parametro: job_performance<br><br>Indice di asimmetria di esecuzione delle fasi spark: questa metrica è un indicator e della durata massima dell'attività in una determina ta fase rispetto alla durata media dell'attività in questa fase. Rileva l'asimmetria di esecuzione, che potrebbe essere causata dall'asim metria dei dati di input o da una trasformazione (ad es. join asimmetrico). I valori di questo parametro rientrano nell'intervallo [0, infinito], dove 0 indica il rapporto tra il tempo di esecuzione massimo e quello medio delle attività. Tra tutte le attività nella fase, è inferiore a un determinato | job_performance |

| Metrica | Description | Categoria |
| --- | --- | --- |
| | fattore di asimmetria della stessa. Il fattore predefinito di asimmetria della fase è "5" e può essere sovrascri tto tramite la configurazione spark: spark.metrics.conf .driver.source.glue.jobPerf ormance.skewnessFactor<br><br>Un valore di asimmetria della fase pari a 1 significa che il rapporto è il doppio del fattore di asimmetria della fase.<br><br>Il valore dell'asimmetria della fase viene aggiornato ogni 30 secondi per riflettere l'asimmet ria corrente. Il valore alla fine della fase riflette l'asimmetria della fase finale.<br><br>Questa metrica a livello di fase viene utilizzata per calcolare la metrica a livello di lavoro `glue.driver.skewne ss.job` .<br><br>Dimensioni valide: JobName (il nome del AWS Glue Job), JobRunId ( JobRun ID. o ALL), Type (gauge) e Observabi lityGroup (job_performance)<br><br>Statistiche valide: media, massimo, minimo, percentuale | |

| Metrica | Description | Categoria |
| --- | --- | --- |
| | Unità: numero | |

| Metrica | Description | Categoria |
| --- | --- | --- |
| glue.driver.skewness.job | Categoria parametro: job_performance<br><br>L'asimmetria del lavoro è l'asimmetria ponderata massima di tutte le fasi. L'asimmetria della fase (glue.driver.skewness.stage) viene ponderata in base alla durata della fase. In questo modo si evita il caso limite in cui una fase molto asimmetri ca viene eseguita per un periodo molto breve rispetto ad altre fasi (quindi la sua asimmetria non è significativa per le prestazioni complessi ve del processo e non vale la pena cercare di correggerla).<br><br>Questo parametro viene aggiornato al completam ento di ogni fase, perciò l'ultimo valore riflette l'effettiva asimmetria complessiva del processo.<br><br>Dimensioni valide: JobName (il nome del AWS Glue Job), JobRunId ( JobRun ID. o ALL), Type (gauge) e Observabi lityGroup (job_performance)<br><br>Statistiche valide: media, massimo, minimo, percentuale | job_performance |

| Metrica | Description | Categoria |
| --- | --- | --- |
| | Unità: numero | |
| glue.succeed.ALL | Categoria parametro: errore<br><br>Numero totale di processi eseguiti con successo, per completare il quadro delle categorie di errori<br><br>Dimensioni valide: JobName (il nome del AWS Glue Job), JobRunId ( JobRun ID. o ALL), Type (count) e Observabi lityGroup (error)<br><br>Statistiche valide: SOMMA<br><br>Unità: numero | error |
| glue.error.ALL | Categoria parametro: errore<br><br>Numero totale di errori di esecuzione del processo, per completare il quadro delle categorie di errori<br><br>Dimensioni valide: JobName (il nome del AWS Glue Job), JobRunId ( JobRun ID. o ALL), Type (count) e Observabi lityGroup (error)<br><br>Statistiche valide: SOMMA<br><br>Unità: numero | error |

| Metrica | Description | Categoria |
| --- | --- | --- |
| glue.error.[error category] | Categoria parametro: errore<br><br>Questo insieme di parametri viene aggiornato solo se l'esecuzione di un processo fallisce. La categorizzazione degli errori facilita la classific azione e il debug. Quando l'esecuzione di un processo fallisce, la causa dell'erro re viene classificata e il parametro della categoria di errore corrispondente viene impostato su 1. Ciò consente di eseguire l'analisi degli errori nel corso tempo, nonché quella relativa a tutti i processi, per identificare le categorie di errore più comuni e risolverle. AWS Glue include 28 categorie di errore, tra cui OUT_OF_ME MORY (driver ed executor) , AUTORIZZAZIONE, SINTASSI e LIMITAZIONE (DELLA LARGHEZZA DI BANDA DELLA RETE). Le categorie di errore includono anche COMPILAZIONE, AVVIO e TIMEOUT.<br><br>Dimensioni valide: JobName (il nome del AWS Glue Job), JobRunId ( JobRun ID. o ALL), | error |

| Metrica | Description | Categoria |
| --- | --- | --- |
| | Type (count) e Observabi lityGroup (error)<br><br>Statistiche valide: SOMMA<br><br>Unità: numero | |
| glue.driver.workerUtilization | Categoria parametro: resource_utilization<br><br>La percentuale dei worker allocati che vengono effettiva mente utilizzati. Se non va bene, può essere utile il dimensionamento automatico.<br><br>Dimensioni valide: JobName (il nome del AWS Glue Job), JobRunId ( JobRun ID. o ALL), Type (gauge) e Observabi lityGroup (resource_utilization)<br><br>Statistiche valide: media, massimo, minimo, percentuale<br><br>Unità: percentuale | resource_utilization |

| Metrica | Description | Categoria |
| --- | --- | --- |
| glue.driver.memory.heap.[available \| used] | Categoria parametro: resource_utilization<br><br>La memoria heap del driver disponibile/utilizzata durante l'esecuzione del processo. Ciò è utile per comprendere le tendenze di utilizzo della memoria, soprattutto nel tempo, il che può contribuire a evitare potenziali errori e a eseguirne il debug.<br><br>Dimensioni valide: JobName (il nome del AWS Glue Job), JobRunId ( JobRun ID. o ALL), Type (gauge) e ObservabilityGroup (resource_utilization)<br><br>Statistiche valide: media<br><br>Unità: byte | resource_utilization |

| Metrica | Description | Categoria |
| --- | --- | --- |
| glue.driver.memory.heap.used.percentage | Categoria parametro: resource_utilization<br><br>La memoria heap del driver utilizzata (%) durante l'esecuzione del processo. Ciò è utile per comprendere le tendenze di utilizzo della memoria, soprattutto nel tempo, il che può contribuire a evitare potenziali errori e a eseguirne il debug.<br><br>Dimensioni valide: JobName (il nome del AWS Glue Job), JobRunId ( JobRun ID. o ALL), Type (gauge) e ObservabilityGroup (resource_utilization)<br><br>Statistiche valide: media<br><br>Unità: percentuale | resource_utilization |

| Metrica | Description | Categoria |
| --- | --- | --- |
| glue.driver.memory.non-heap.[available \| used] | Categoria parametro: resource_utilization<br><br>La memoria non heap del driver disponibile/utilizzata durante l'esecuzione del processo. Ciò è utile per comprendere le tendenze di utilizzo della memoria, soprattutto nel tempo, il che può contribuire a evitare potenziali errori e a eseguirne il debug.<br><br>Dimensioni valide: JobName (il nome del AWS Glue Job), JobRunId ( JobRun ID. o ALL), Type (gauge) e Observabilità Group (resource_utilization)<br><br>Statistiche valide: media<br><br>Unità: byte | resource_utilization |

| Metrica | Description | Categoria |
| --- | --- | --- |
| glue.driver.memory.non-heap .used.percentage | Categoria parametro: resource_utilization<br><br>La memoria non heap del driver utilizzata (%) durante l'esecuzione del processo. Ciò è utile per comprendere le tendenze di utilizzo della memoria, soprattutto nel tempo, il che può contribuire a evitare potenziali errori e a eseguirne il debug.<br><br>Dimensioni valide: JobName (il nome del AWS Glue Job), JobRunId ( JobRun ID. o ALL), Type (gauge) e Observabi lityGroup (resource_utilization)<br><br>Statistiche valide: media<br><br>Unità: percentuale | resource_utilization |

| Metrica | Description | Categoria |
| --- | --- | --- |
| glue.driver.memory.total.[available \| used] | Categoria parametro: resource_utilization<br><br>La memoria totale del driver disponibile/utilizzata durante l'esecuzione del processo. Ciò è utile per comprendere le tendenze di utilizzo della memoria, soprattutto nel tempo, il che può contribuire a evitare potenziali errori e a eseguirne il debug.<br><br>Dimensioni valide: JobName (il nome del AWS Glue Job), JobRunId ( JobRun ID. o ALL), Type (gauge) e ObservabilityGroup (resource_utilization)<br><br>Statistiche valide: media<br><br>Unità: byte | resource_utilization |

| Metrica | Description | Categoria |
| --- | --- | --- |
| glue.driver.memory.total.used.percentage | Categoria parametro: resource_utilization<br><br>La memoria totale del driver utilizzata (%) durante l'esecuzione del processo. Ciò è utile per comprendere le tendenze di utilizzo della memoria, soprattutto nel tempo, il che può contribuire a evitare potenziali errori e a eseguirne il debug.<br><br>Dimensioni valide: JobName (il nome del AWS Glue Job), JobRunId ( JobRun ID. o ALL), Type (gauge) e ObservabilityGroup (resource_utilization)<br><br>Statistiche valide: media<br><br>Unità: percentuale | resource_utilization |

| Metrica | Description | Categoria |
| --- | --- | --- |
| glue.ALL.memory.heap.[available \| used] | Categoria parametro: resource_utilization<br><br>La memoria heap degli esecutori. available/used ALL significa tutti gli executor.<br><br>Dimensioni valide: JobName (il nome del AWS Glue Job), JobRunId ( JobRun ID. o ALL), Type (gauge) e ObservabilityGroup (resource_utilization)<br><br>Statistiche valide: media<br><br>Unità: byte | resource_utilization |
| glue.ALL.memory.heap.used.percentage | Categoria parametro: resource_utilization<br><br>La memoria heap degli executor utilizzata (%). ALL significa tutti gli executor.<br><br>Dimensioni valide: JobName (il nome del AWS Glue Job), JobRunId ( JobRun ID. o ALL), Type (gauge) e ObservabilityGroup (resource_utilization)<br><br>Statistiche valide: media<br><br>Unità: percentuale | resource_utilization |

| Metrica | Description | Categoria |
| --- | --- | --- |
| glue.ALL.memory.non-heap.[available \| used] | Categoria parametro: resource_utilization<br><br>La memoria non heap degli esecutori. available/used ALL significa tutti gli executor.<br><br>Dimensioni valide: JobName (il nome del AWS Glue Job), JobRunId ( JobRun ID. o ALL), Type (gauge) e ObservabilityGroup (resource_utilization)<br><br>Statistiche valide: media<br><br>Unità: byte | resource_utilization |
| glue.ALL.memory.non-heap.used.percentage | Categoria parametro: resource_utilization<br><br>La memoria non heap degli executor utilizzata (%). ALL significa tutti gli executor.<br><br>Dimensioni valide: JobName (il nome del AWS Glue Job), JobRunId ( JobRun ID. o ALL), Type (gauge) e ObservabilityGroup (resource_utilization)<br><br>Statistiche valide: media<br><br>Unità: percentuale | resource_utilization |

| Metrica | Description | Categoria |
| --- | --- | --- |
| glue.ALL.memory.total.[available \| used] | Categoria parametro: resource_utilization<br><br>La memoria totale degli esecutori. available/used ALL significa tutti gli executor.<br><br>Dimensioni valide: JobName (il nome del AWS Glue Job), JobRunId ( JobRun ID. o ALL), Type (gauge) e ObservabilityGroup (resource_utilization)<br><br>Statistiche valide: media<br><br>Unità: byte | resource_utilization |
| glue.ALL.memory.total.used.percentage | Categoria parametro: resource_utilization<br><br>La memoria totale degli executor utilizzata (%). ALL significa tutti gli executor.<br><br>Dimensioni valide: JobName (il nome del AWS Glue Job), JobRunId ( JobRun ID. o ALL), Type (gauge) e ObservabilityGroup (resource_utilization)<br><br>Statistiche valide: media<br><br>Unità: percentuale | resource_utilization |

| Metrica | Description | Categoria |
| --- | --- | --- |
| glue.driver.disk.[available_GB \| used_GB] | Categoria parametro: resource_utilization<br><br>Lo spazio su disco del driver durante l'esecuzione del processo. available/used Ciò è utile per comprende re le tendenze di utilizzo del disco, soprattutto nel tempo, il che può contribuire a evitare potenziali errori e a eseguire il debug di quelli relativi alla presenza di spazio non sufficiente sul disco.<br><br>Dimensioni valide: JobName (il nome del AWS Glue Job), JobRunId ( JobRun ID. o ALL), Type (gauge) e Observabi lityGroup (resource_utilization)<br><br>Statistiche valide: media<br><br>Unità: gigabyte | resource_utilization |

| Metrica | Description | Categoria |
| --- | --- | --- |
| glue.driver.disk.used.percentage] | Categoria parametro: resource_utilization<br><br>Lo spazio su disco del driver durante l'esecuzione del processo. available/used Ciò è utile per comprende re le tendenze di utilizzo del disco, soprattutto nel tempo, il che può contribuire a evitare potenziali errori e a eseguire il debug di quelli relativi alla presenza di spazio non sufficiente sul disco.<br><br>Dimensioni valide: JobName (il nome del AWS Glue Job), JobRunId ( JobRun ID. o ALL), Type (gauge) e Observabi lityGroup (resource_utilization)<br><br>Statistiche valide: media<br><br>Unità: percentuale | resource_utilization |

| Metrica | Description | Categoria |
| --- | --- | --- |
| glue.ALL.disk.[available_GB \| used_GB] | Categoria parametro: resource_utilization<br><br>Lo spazio su disco degli esecutori. available/used ALL significa tutti gli executor.<br><br>Dimensioni valide: JobName (il nome del AWS Glue Job), JobRunId ( JobRun ID. o ALL), Type (gauge) e Observabi lityGroup (resource_utilization)<br><br>Statistiche valide: media<br><br>Unità: gigabyte | resource_utilization |
| glue.ALL.disk.used.percenta ge | Categoria parametro: resource_utilization<br><br>Lo spazio su disco (%) degli esecutori. available/used/used ALL significa tutti gli executor.<br><br>Dimensioni valide: JobName (il nome del AWS Glue Job), JobRunId ( JobRun ID. o ALL), Type (gauge) e Observabi lityGroup (resource_utilization)<br><br>Statistiche valide: media<br><br>Unità: percentuale | resource_utilization |

| Metrica | Description | Categoria |
| --- | --- | --- |
| glue.driver.bytesRead | Categoria parametro: velocità di trasmissione effettiva<br><br>Il numero di byte letti per ogni origine di input in questa esecuzione del processo e per TUTTE le origini. È possibile così comprendere il volume dei dati e le relative variazion i nel tempo, il che consente di risolvere problemi come l'asimmetria dei dati.<br><br>Dimensioni valide: JobName (il nome del AWS Glue Job), JobRunId ( JobRun ID. o ALL), Type (gauge), (resource_utilization) e Source ObservabilityGroup (posizione dei dati di origine)<br><br>Statistiche valide: media<br><br>Unità: byte | velocità di trasmissione effettiva |

| Metrica | Description | Categoria |
| --- | --- | --- |
| glue.driver.[recordsRead \| filesRead] | Categoria parametro: velocità di trasmissione effettiva<br><br>Il numero di records/files letture per sorgente di input in questo processo e per TUTTE le fonti. È possibile così comprendere il volume dei dati e le relative variazion i nel tempo, il che consente di risolvere problemi come l'asimmetria dei dati.<br><br>Dimensioni valide: JobName (il nome del AWS Glue Job), JobRunId ( JobRun ID. o ALL), Type (gauge), (resource_utilization) e Source ObservabilityGroup (posizione dei dati di origine)<br><br>Statistiche valide: media<br><br>Unità: numero | velocità di trasmissione effettiva |

| Metrica | Description | Categoria |
| --- | --- | --- |
| glue.driver.partitionsRead | Categoria parametro: velocità di trasmissione effettiva<br><br>Il numero di partizioni lette per ogni origine di input di Amazon S3 in questa esecuzione del processo e per TUTTE le origini.<br><br>Dimensioni valide: JobName (il nome del AWS Glue Job), JobRunId ( JobRun ID. o ALL), Type (gauge), (resource_utilization) e Source ObservabilityGroup (posizione dei dati di origine)<br><br>Statistiche valide: media<br><br>Unità: numero | velocità di trasmissione effettiva |

| Metrica | Description | Categoria |
| --- | --- | --- |
| glue.driver.bytesWrittten | Categoria parametro: velocità di trasmissione effettiva<br><br>Il numero di byte scritti per ogni sink di output in questa esecuzione del processo e per TUTTI i sink. È possibile così comprendere il volume dei dati e il modo in cui evolve nel tempo, il che consente di risolvere problemi come l'asimmetria dell'elaborazione.<br><br>Dimensioni valide: JobName (il nome del AWS Glue Job), JobRunId ( JobRun ID. o ALL), Type (gauge), (resource _utilization) e Sink Observabi lityGroup (posizione dei dati del sink)<br><br>Statistiche valide: media<br><br>Unità: byte | velocità di trasmissione effettiva |

| Metrica | Description | Categoria |
| --- | --- | --- |
| glue.driver.[recordsWritten \| filesWritten] | Categoria parametro: velocità di trasmissione effettiva<br><br>Il numero di sink records/files scritti per uscita in questa esecuzione di processo e per TUTTI i sink. È possibile così comprendere il volume dei dati e il modo in cui evolve nel tempo, il che consente di risolvere problemi come l'asimmetria dell'elaborazione.<br><br>Dimensioni valide: JobName (il nome del AWS Glue Job), JobRunId ( JobRun ID. o ALL), Type (gauge), (resource_utilization) e Sink ObservabilityGroup (posizione dei dati del sink)<br><br>Statistiche valide: media<br><br>Unità: numero | velocità di trasmissione effettiva |

Categorie di errore

| Categorie di errore | Description |
| --- | --- |
| COMPILATION_ERROR | Gli errori si verificano durante la compilazione del codice Scala. |
| CONNECTION_ERROR | Si verificano errori durante la connessione a un servizio host/database, ecc. service/remote |

| Categorie di errore | Description |
| --- | --- |
| DISK_NO_SPACE_ERROR | Gli errori si verificano quando non c'è più spazio nel disco sul driver/executor. |
| OUT_OF_MEMORY_ERROR | Gli errori si verificano quando non c'è più spazio nella memoria sul driver/executor. |
| IMPORT_ERROR | Gli errori si verificano durante l'importazione delle dipendenze. |
| INVALID_ARGUMENT_ERROR | Gli errori si verificano quando gli argomenti di input sono non validi o illegali. |
| PERMISSION_ERROR | Gli errori si verificano in mancanza di autorizza zioni per il servizio, per i dati, ecc. |
| RESOURCE_NOT_FOUND_ERROR | Gli errori si verificano quando i dati, la posizione, ecc. non esistono. |
| QUERY_ERROR | Gli errori derivano dall'esecuzione delle query di Spark SQL. |
| SYNTAX_ERROR | Gli errori si verificano quando nello script è presente un errore di sintassi. |
| THROTTLING_ERROR | Gli errori si verificano quando si supera la limitazione della concorrenza del servizio o il limite della quota di servizio. |
| DATA_LAKE_FRAMEWORK_ERROR | Gli errori derivano da framework data lake supportati nativamente da AWS Glue, come Hudi, Iceberg, ecc. |
| UNSUPPORTED_OPERATION_ERROR | Gli errori si verificano quando si eseguono operazioni non supportate. |
| RESOURCES_ALREADY_EXISTS_ERROR | Gli errori si verificano quando una risorsa da creare o aggiungere esiste già. |

| Categorie di errore | Description |
| --- | --- |
| GLUE_INTERNAL_SERVICE_ERROR | Gli errori si verificano quando c'è un problema interno al servizio AWS Glue. |
| GLUE_OPERATION_TIMEOUT_ERROR | Gli errori si verificano quando un'operazione AWS Glue è in timeout. |
| GLUE_VALIDATION_ERROR | Gli errori si verificano quando un valore richiesto non può essere convalidato per un processo AWS Glue. |
| GLUE_JOB_BOOKMARK_VERSION_MISMATCH_ERROR | Gli errori si verificano quando lo stesso processo si trova sullo stesso bucket di origine e scrive contemporaneamente nella same/different destinazione (concorrenza >1) |
| LAUNCH_ERROR | Gli errori si verificano durante la fase di avvio del processo AWS Glue. |
| DYNAMODB_ERROR | Gli errori generici derivano dal servizio. Amazon DynamoDB |
| GLUE_ERROR | Gli errori generici derivano dal servizio AWS Glue. |
| LAKEFORMATION_ERROR | Gli errori generici derivano dal AWS Lake Formation servizio. |
| REDSHIFT_ERROR | Gli errori generici derivano dal Amazon Redshift servizio. |
| S3_ERROR | Gli errori generici derivano dal servizio Amazon S3. |
| SYSTEM_EXIT_ERROR | Errore generico di uscita dal sistema. |
| TIMEOUT_ERROR | Gli errori generici si verificano quando il processo fallisce per timeout dell'operazione. |

| Categorie di errore | Description |
| --- | --- |
| UNCLASSIFIED_SPARK_ERROR | Gli errori generici derivano da Spark. |
| UNCLASSIFIED_ERROR | Categoria di errore predefinita. |

Limitazioni

> **Note**
>
> `glueContext` deve essere inizializzato per poter pubblicare i parametri.

Nella dimensione di origine, il valore corrisponde al percorso o al nome della tabella Amazon S3, a seconda del tipo di origine. Inoltre, se l'origine è JDBC e viene utilizzata l'opzione di query, la stringa di query viene impostata nella dimensione di origine. Se il valore supera i 500 caratteri, viene ridotto per rispettare questo limite. Di seguito sono riportate le limitazioni del valore:

- I caratteri non ASCII verranno rimossi.
- Se il nome dell'origine non contiene alcun carattere ASCII, verrà convertito in <non-ASCII input>.

Limitazioni e considerazioni relative ai parametri della velocità di trasmissione effettiva

- DataFrame e DataFrame based DynamicFrame (ad esempio JDBC, lettura da parquet su Amazon S3) sono supportati, mentre quelli DynamicFrame basati su RDD (ad esempio la lettura di csv, json su Amazon S3, ecc.) non sono supportati. Tecnicamente, tutte le letture e le scritture visibili sull'interfaccia utente di Spark sono supportate.
- Il parametro `recordsRead` viene emesso se l'origine dati è una tabella di catalogo e il formato è JSON, CSV, testo o Iceberg.
- I parametri `glue.driver.throughput.recordsWritten`, `glue.driver.throughput.bytesWritten` e `glue.driver.throughput.filesWritten` non sono disponibili nelle tabelle JDBC e Iceberg.
- I parametri potrebbero subire ritardi. Se il lavoro termina in circa un minuto, è possibile che in Metrics non sia presente alcuna metrica di throughput. Amazon CloudWatch

Monitoraggio e debug dei processi

Puoi raccogliere metriche sui AWS Glue lavori e visualizzarle sulle CloudWatch console AWS Glue Amazon per identificare e risolvere i problemi. Per la profilatura dei processi AWS Glue sono necessarie le fasi seguenti:

1. Abilitare i parametri:

   a. Abilitare l'opzione Job metrics (Parametri processo) nella definizione del processo. È possibile abilitare la profilatura nella console AWS Glue o come parametro per il processo. Per ulteriori informazioni, consultare Definire le proprietà di processo per i processi Spark o Utilizzo dei parametri del lavoro nei lavori AWS Glue.

   b. Abilita l'opzione Parametri AWS Glue di osservabilità nella definizione del processo. È possibile abilitare l'osservabilità nella console AWS Glue o come parametro per il processo. Per ulteriori informazioni, consulta Monitoraggio con parametri AWS Glue di osservabilità.

2. Verificare che lo script del processo inizializzi un oggetto `GlueContext`. Il frammento di script seguente inizializza ad esempio un oggetto `GlueContext` e mostra dove viene inserito il codice profilato nello script. Questo formato generale viene usato negli scenari di debug seguenti.

```
import sys
from awsglue.transforms import *
from awsglue.utils import getResolvedOptions
from pyspark.context import SparkContext
from awsglue.context import GlueContext
from awsglue.job import Job
import time

## @params: [JOB_NAME]
args = getResolvedOptions(sys.argv, ['JOB_NAME'])

sc = SparkContext()
glueContext = GlueContext(sc)
spark = glueContext.spark_session
job = Job(glueContext)
job.init(args['JOB_NAME'], args)

...
...
code-to-profile
...
```

```
...


job.commit()
```

3. Esegui il processo.
4. Visualizzare i parametri:

   a. Visualizza i parametri nella console AWS Glue e identifica quelli anomali per il driver o per un executor.

   b. Controlla le metriche di osservabilità nella pagina di monitoraggio del Job run, nella pagina dei dettagli del job run o su Amazon. CloudWatch Per ulteriori informazioni, consulta Monitoraggio con parametri AWS Glue di osservabilità.

5. Risalire alla causa principale usando il parametro identificato.
6. Facoltativamente, confermare la causa principale usando il flusso di log del driver o dell'executor del processo identificato.

Casi d'uso per i parametri AWS Glue di osservabilità

- Debug di eccezioni di memoria esaurita (OOM) e anomalie dei processi
- Debug di fasi impegnative e attività in ritardo
- Monitoraggio dell'avanzamento di processi multipli
- Monitoraggio per la pianificazione della capacità DPU
- Utilizzo dell'osservabilità AWS Glue per monitorare l'utilizzo delle risorse per ridurre i costi

Debug di eccezioni di memoria esaurita (OOM) e anomalie dei processi

È possibile eseguire il debug di eccezioni out-of-memory (OOM) e anomalie del lavoro in. AWS Glue Le sezioni seguenti descrivono gli scenari per il debug delle out-of-memory eccezioni del driver Apache Spark o di un esecutore Spark.

- Debug dell'eccezione di memoria esaurita (OOM) di un driver
- Debug di un'eccezione di memoria esaurita (OOM) dell'executor

Debug dell'eccezione di memoria esaurita (OOM) di un driver

In questo scenario un processo Spark sta leggendo un numero elevato di piccoli file da Amazon Simple Storage Service (Amazon S3). I file vengono convertiti nel formato Apache Parquet e quindi scritti in Amazon S3. Il driver Spark sta per esaurire la memoria. I dati Amazon S3 di input hanno più di 1 milione di file in partizioni Amazon S3 diverse.

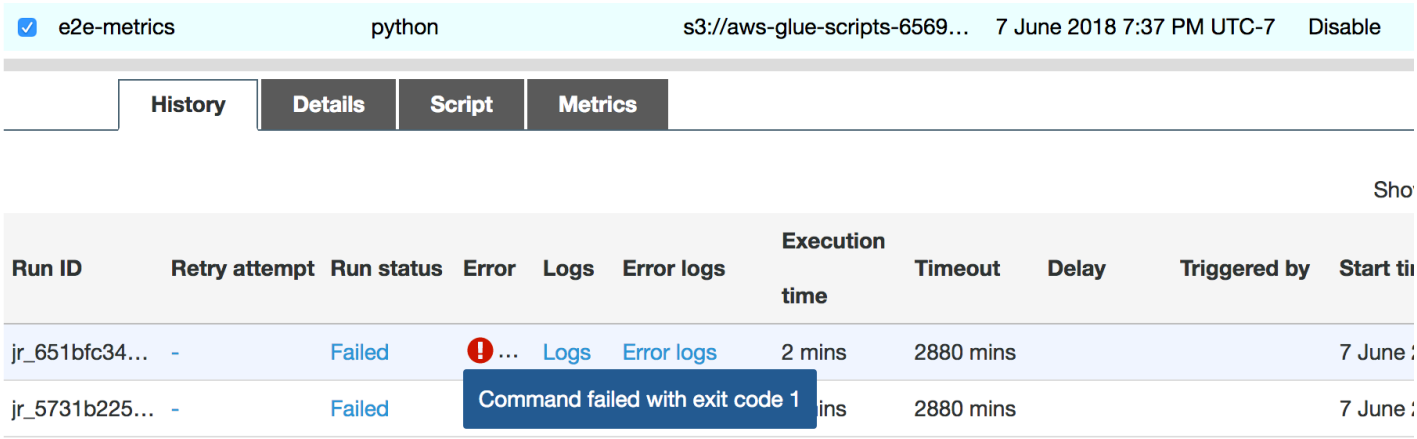
Il codice profilato è il seguente:

```
data = spark.read.format("json").option("inferSchema", False).load("s3://input_path")
data.write.format("parquet").save(output_path)
```

Visualizza AWS Glue le metriche profilate sulla console

Il grafico seguente mostra l'utilizzo di memoria sotto forma di percentuale per il driver e gli executor. Il grafico dell'utilizzo viene tracciato usando punti dati che rappresentano la media dei valori segnalati nell'ultimo minuto. Nel profilo di memoria del processo è possibile vedere che la memoria del driver supera la soglia di sicurezza del 50% di utilizzo in modo rapido. L'utilizzo di memoria medio di tutti gli executor rimane invece inferiore al 4%. Ciò indica chiaramente un'anomalia nell'esecuzione del driver nel processo Spark.

L'esecuzione del processo ha esito negativo in breve tempo e viene visualizzato l'errore seguente nella scheda History (Cronologia) della console AWS Glue: Command Failed with Exit Code 1 (Comando non riuscito con codice di uscita 1). Questa stringa di errore indica che il processo non è riuscito a causa di un errore di sistema, che in questo caso è l'esaurimento di memoria del driver.




Sulla console, scegli il link Registri degli errori nella scheda Cronologia per confermare la scoperta del driver OOM nei registri. CloudWatch Cerca "**`Error`**" nel log di errori del processo per verificare che la mancata riuscita del processo sia effettivamente dovuta a un'eccezione di memoria esaurita:

```
# java.lang.OutOfMemoryError: Java heap space
# -XX:OnOutOfMemoryError="kill -9 %p"
# Executing /bin/sh -c "kill -9 12039"...
```

Nella scheda History (Cronologia) per il processo scegli Logs (Log). È possibile trovare la seguente traccia dell'esecuzione del driver nei CloudWatch registri all'inizio del processo. Il driver Spark cerca di elencare tutti i file in tutte le directory, crea una oggetto `InMemoryFileIndex` e avvia un'attività per ogni file. Di conseguenza, il driver Spark deve gestire una quantità elevata di stato in memoria per tenere traccia di tutte le attività. Viene memorizzato nella cache l'elenco completo di un numero elevato di file per l'indice in memoria, provocando l'esaurimento della memoria del driver.

Correzione dell'elaborazione di più file mediante il raggruppamento

È possibile correggere l'elaborazione di più file usando la caratteristica di raggruppamento in AWS Glue. Il raggruppamento viene abilitato automaticamente quando si usano frame dinamici e quando il set di dati di input include un numero elevato di file (più di 50.000). Il raggruppamento permette di unire più file in un gruppo e permette a un'attività di elaborare l'intero gruppo invece di un singolo file. Di conseguenza, il driver Spark archivia una quantità significativamente inferiore di stato in memoria

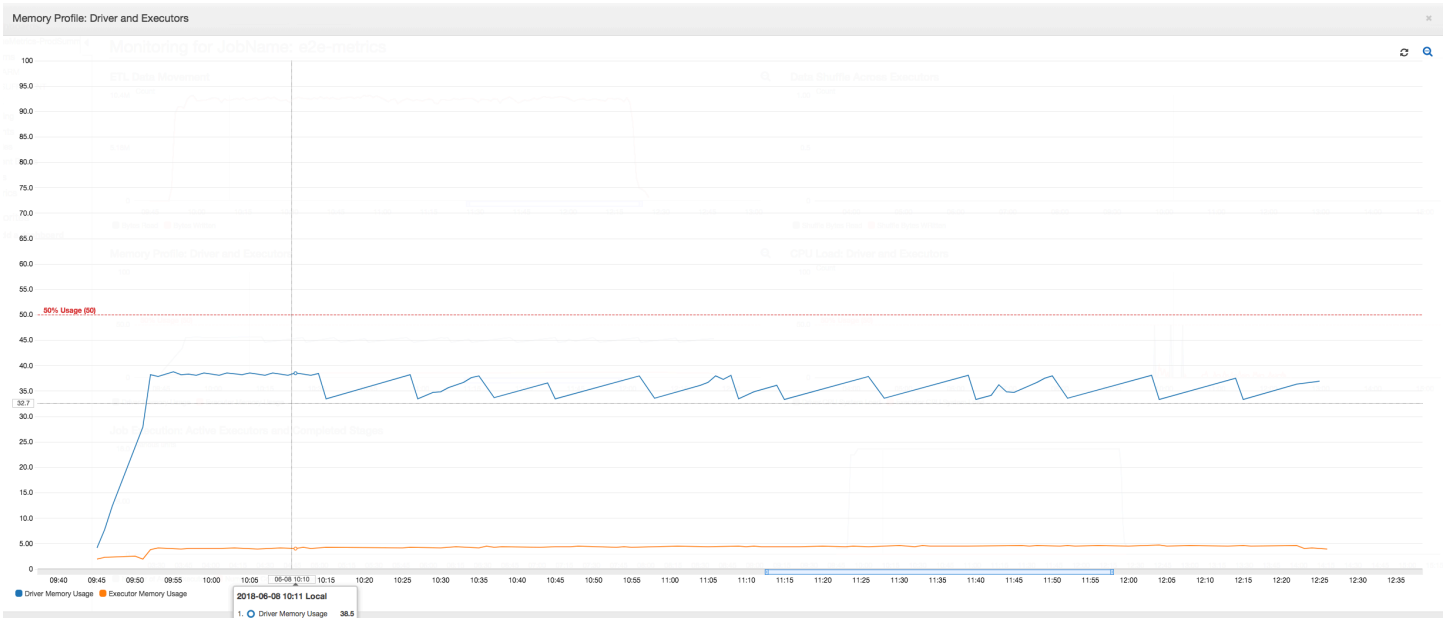
per tenere traccia di un numero minore di attività. Per ulteriori informazioni sull'abilitazione manuale del raggruppamento per un set di dati, consulta Lettura di file di input in gruppi di grandi dimensioni.

Per controllare il profilo di memoria del processo AWS Glue, profila il codice seguente con il raggruppamento abilitato:

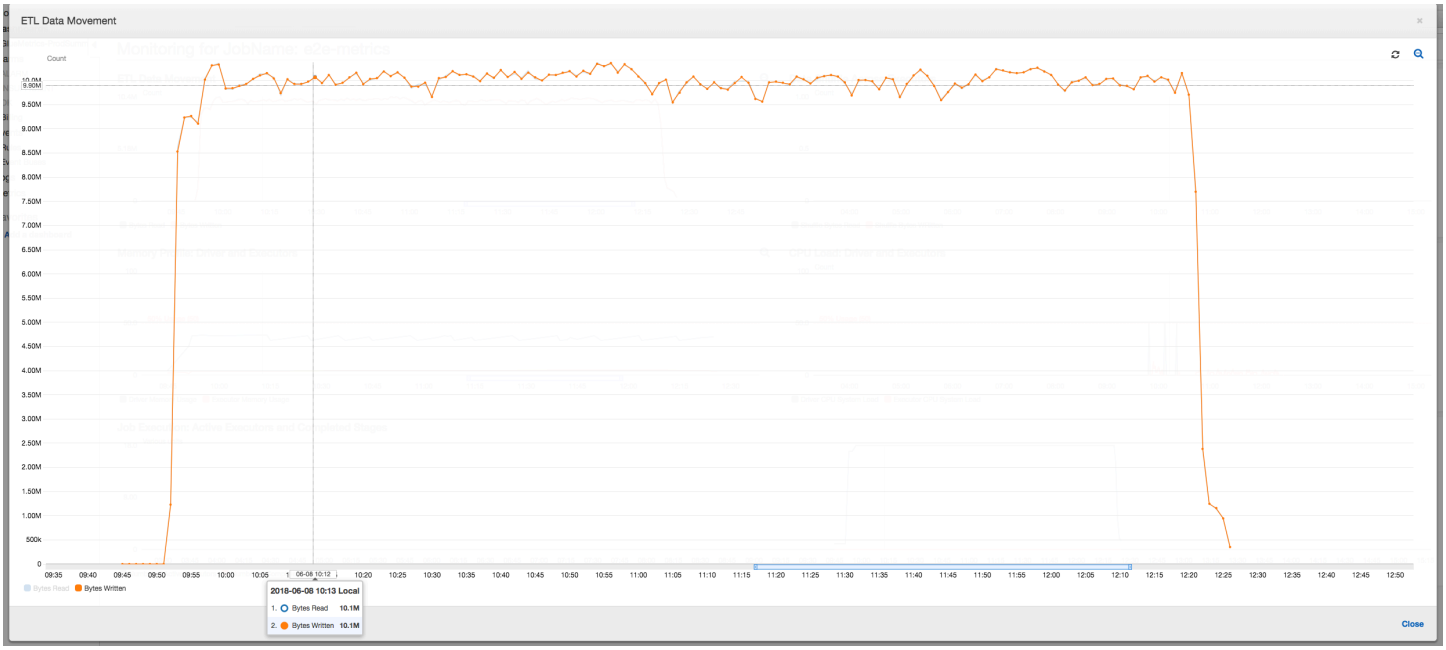
```
df = glueContext.create_dynamic_frame_from_options("s3", {'paths': ["s3://input_path"],
 "recurse":True, 'groupFiles': 'inPartition'}, format="json")
datasink = glueContext.write_dynamic_frame.from_options(frame = df, connection_type
 = "s3", connection_options = {"path": output_path}, format = "parquet",
 transformation_ctx = "datasink")
```

È possibile monitorare il profilo di memoria e lo spostamento di dati ETL nel profilo del processo AWS Glue.

Il driver viene eseguito sotto la soglia del 50% di utilizzo di memoria per l'intera durata del processo AWS Glue. Gli executor trasmettono i dati da Amazon S3, li elaborano e li scrivono in Amazon S3. Di conseguenza, usano meno del 5% di memoria in qualsiasi momento.




Il profilo di spostamento dei dati seguente mostra il numero totale di byte Amazon S3 che vengono letti e scritti nell'ultimo minuto da tutti gli executor man mano che il processo avanza. In entrambi i casi viene seguito un modello simile mentre i dati vengono trasmessi tra tutti gli executor. Il processo completa l'elaborazione di tutto il milione di file in meno di tre ore.



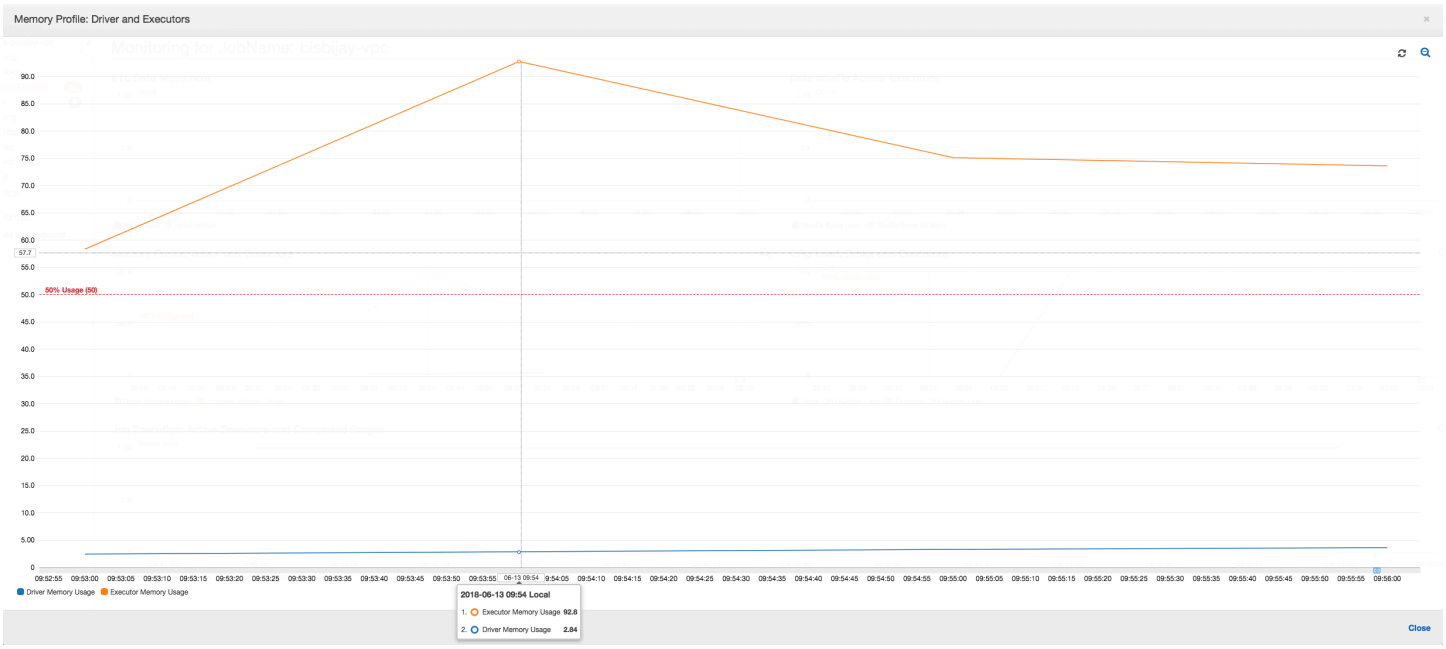

Debug di un'eccezione di memoria esaurita (OOM) dell'executor

In questo scenario è possibile imparare a eseguire il debug delle eccezioni di memoria esaurita che potrebbero verificarsi negli executor Apache Spark. Il codice seguente usa il lettore Spark MySQL per leggere una tabella di grandi dimensioni con circa 34 milioni di righe in un dataframe Spark. Scrive quindi i dati in Amazon S3 in formato Parquet. È possibile fornire le proprietà di connessione e usare le configurazioni Spark predefinite per leggere la tabella.

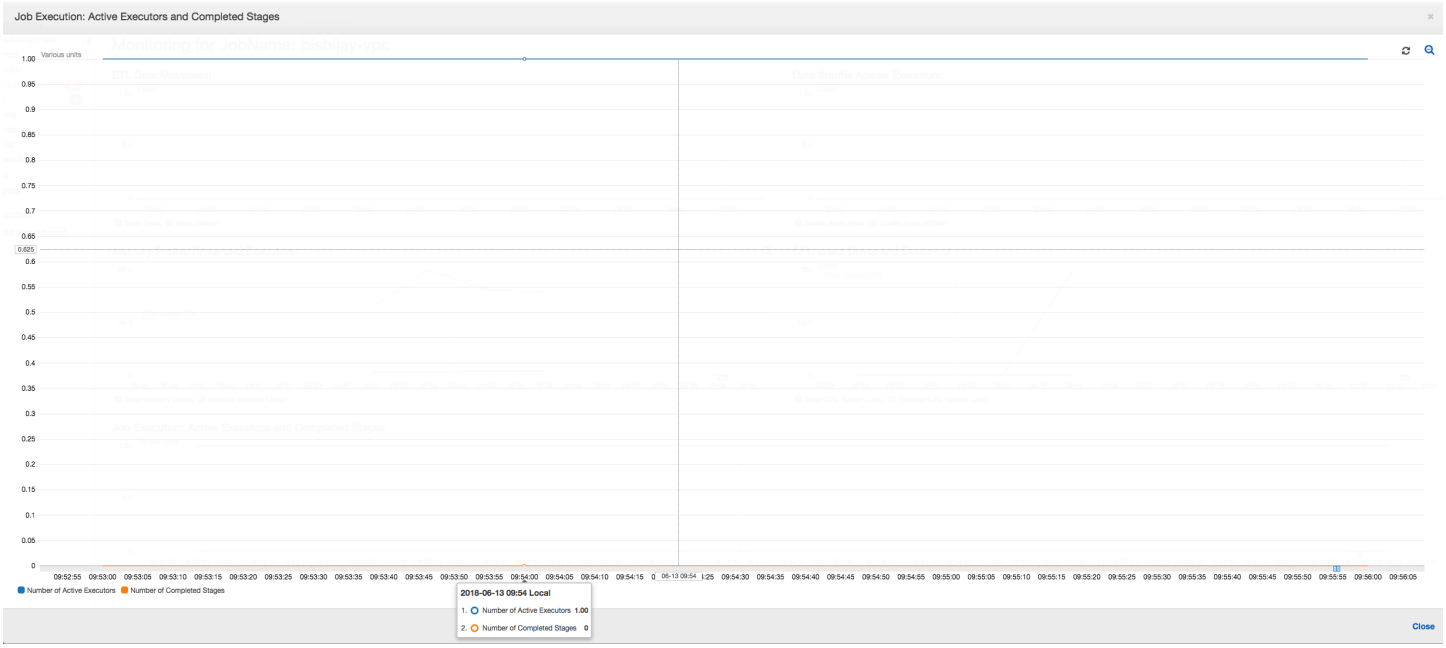
```
val connectionProperties = new Properties()
connectionProperties.put("user", user)
connectionProperties.put("password", password)
connectionProperties.put("Driver", "com.mysql.jdbc.Driver")
val sparkSession = glueContext.sparkSession
val dfSpark = sparkSession.read.jdbc(url, tableName, connectionProperties)
dfSpark.write.format("parquet").save(output_path)
```

Visualizzazione dei parametri profilati nella console AWS Glue

Se la pendenza del grafico di utilizzo della memoria è positiva e supera il 50% e se il processo non riesce prima che venga emesso il parametro successivo, l'esaurimento della memoria può probabilmente essere la causa. Il grafico seguente mostra che entro un minuto di esecuzione, l'utilizzo di memoria medio in tutti gli executor sale rapidamente sopra il 50%. L'utilizzo raggiunge il 92% e il container che esegue l'executor viene interrotto da Apache Hadoop YARN.

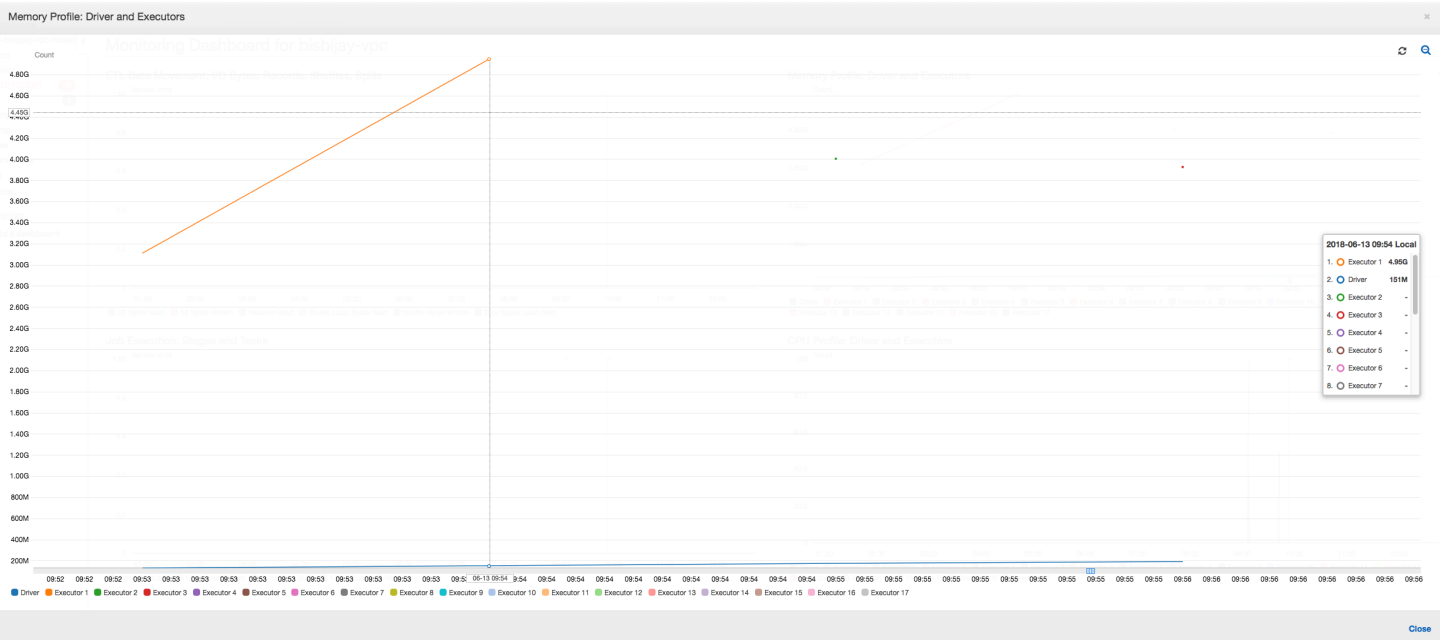



Come mostra il grafico seguente, c'è sempre un singolo executor in esecuzione fino a quando il processo non ha esito negativo. Ciò avviene perché viene avviato un nuovo executor per sostituire quello interrotto. Le operazioni di lettura dell'origine dati JDBC non sono parallelizzate per impostazione predefinita perché ciò richiederebbe il partizionamento della tabella in una colonna e l'apertura di più connessioni. Di conseguenza, un solo executor legge la tabella completa sequenzialmente.

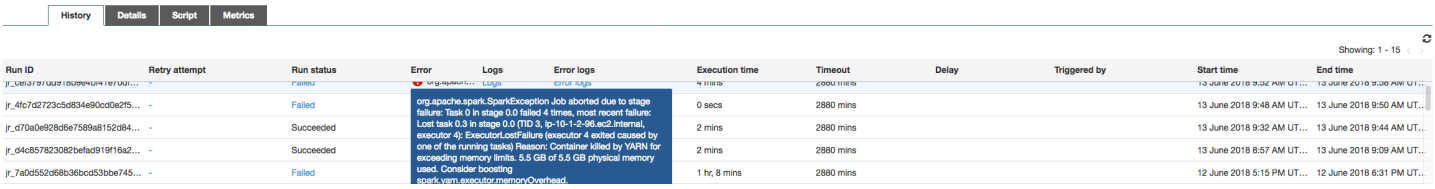



Come mostra il grafico seguente, Spark cerca di avviare una nuova attività quattro volte prima che il processo abbia esito negativo. È possibile visualizzare il profilo di memoria di tre executor.

Ogni executor consuma rapidamente tutta la relativa memoria. Il quarto executor esaurisce la memoria e il processo ha esito negativo. Di conseguenza, il relativo parametro non viene segnalato immediatamente.




È possibile verificare dalla stringa di errore nella console AWS Glue che il processo ha avuto esito negativo a causa di eccezioni di memoria esaurita, come illustrato nell'immagine seguente.




Job output logs: per confermare ulteriormente l'individuazione di un'eccezione OOM dell'executor, guarda i CloudWatch log. Eseguendo la ricerca di **`Error`**, puoi trovare quattro executor interrotti circa nella stessa finestra temporale, come indicato nel pannello di controllo dei parametri. Gli executor sono stati terminati tutti da YARN a causa del superamento dei limiti di memoria.

Executor 1

```
18/06/13 16:54:29 WARN YarnAllocator: Container killed by YARN for exceeding
 memory limits. 5.5 GB of 5.5 GB physical memory used. Consider boosting
 spark.yarn.executor.memoryOverhead.
18/06/13 16:54:29 WARN YarnSchedulerBackend$YarnSchedulerEndpoint: Container killed
 by YARN for exceeding memory limits. 5.5 GB of 5.5 GB physical memory used. Consider
 boosting spark.yarn.executor.memoryOverhead.
```

```
18/06/13 16:54:29 ERROR YarnClusterScheduler: Lost executor 1 on
 ip-10-1-2-175.ec2.internal: Container killed by YARN for exceeding
 memory limits. 5.5 GB of 5.5 GB physical memory used. Consider boosting
 spark.yarn.executor.memoryOverhead.
18/06/13 16:54:29 WARN TaskSetManager: Lost task 0.0 in stage 0.0 (TID 0,
 ip-10-1-2-175.ec2.internal, executor 1): ExecutorLostFailure (executor 1
 exited caused by one of the running tasks) Reason: Container killed by YARN for
 exceeding memory limits. 5.5 GB of 5.5 GB physical memory used. Consider boosting
 spark.yarn.executor.memoryOverhead.
```

### Executor 2

```
18/06/13 16:55:35 WARN YarnAllocator: Container killed by YARN for exceeding
 memory limits. 5.8 GB of 5.5 GB physical memory used. Consider boosting
 spark.yarn.executor.memoryOverhead.
18/06/13 16:55:35 WARN YarnSchedulerBackend$YarnSchedulerEndpoint: Container killed
 by YARN for exceeding memory limits. 5.8 GB of 5.5 GB physical memory used. Consider
 boosting spark.yarn.executor.memoryOverhead.
18/06/13 16:55:35 ERROR YarnClusterScheduler: Lost executor 2 on
 ip-10-1-2-16.ec2.internal: Container killed by YARN for exceeding
 memory limits. 5.8 GB of 5.5 GB physical memory used. Consider boosting
 spark.yarn.executor.memoryOverhead.
18/06/13 16:55:35 WARN TaskSetManager: Lost task 0.1 in stage 0.0 (TID 1,
 ip-10-1-2-16.ec2.internal, executor 2): ExecutorLostFailure (executor 2 exited
 caused by one of the running tasks) Reason: Container killed by YARN for
 exceeding memory limits. 5.8 GB of 5.5 GB physical memory used. Consider boosting
 spark.yarn.executor.memoryOverhead.
```

### Executor 3

```
18/06/13 16:56:37 WARN YarnAllocator: Container killed by YARN for exceeding
 memory limits. 5.8 GB of 5.5 GB physical memory used. Consider boosting
 spark.yarn.executor.memoryOverhead.
18/06/13 16:56:37 WARN YarnSchedulerBackend$YarnSchedulerEndpoint: Container killed
 by YARN for exceeding memory limits. 5.8 GB of 5.5 GB physical memory used. Consider
 boosting spark.yarn.executor.memoryOverhead.
18/06/13 16:56:37 ERROR YarnClusterScheduler: Lost executor 3 on
 ip-10-1-2-189.ec2.internal: Container killed by YARN for exceeding
 memory limits. 5.8 GB of 5.5 GB physical memory used. Consider boosting
 spark.yarn.executor.memoryOverhead.
18/06/13 16:56:37 WARN TaskSetManager: Lost task 0.2 in stage 0.0 (TID 2,
 ip-10-1-2-189.ec2.internal, executor 3): ExecutorLostFailure (executor 3
 exited caused by one of the running tasks) Reason: Container killed by YARN for
```

```
 exceeding memory limits. 5.8 GB of 5.5 GB physical memory used. Consider boosting
 spark.yarn.executor.memoryOverhead.
```

Executor 4

```
18/06/13 16:57:18 WARN YarnAllocator: Container killed by YARN for exceeding
 memory limits. 5.5 GB of 5.5 GB physical memory used. Consider boosting
 spark.yarn.executor.memoryOverhead.
18/06/13 16:57:18 WARN YarnSchedulerBackend$YarnSchedulerEndpoint: Container killed
 by YARN for exceeding memory limits. 5.5 GB of 5.5 GB physical memory used. Consider
 boosting spark.yarn.executor.memoryOverhead.
18/06/13 16:57:18 ERROR YarnClusterScheduler: Lost executor 4 on
 ip-10-1-2-96.ec2.internal: Container killed by YARN for exceeding
 memory limits. 5.5 GB of 5.5 GB physical memory used. Consider boosting
 spark.yarn.executor.memoryOverhead.
18/06/13 16:57:18 WARN TaskSetManager: Lost task 0.3 in stage 0.0 (TID 3,
 ip-10-1-2-96.ec2.internal, executor 4): ExecutorLostFailure (executor 4 exited
 caused by one of the running tasks) Reason: Container killed by YARN for
 exceeding memory limits. 5.5 GB of 5.5 GB physical memory used. Consider boosting
 spark.yarn.executor.memoryOverhead.
```

Correzione dell'impostazione relativa alle dimensioni di recupero tramite frame dinamici AWS Glue

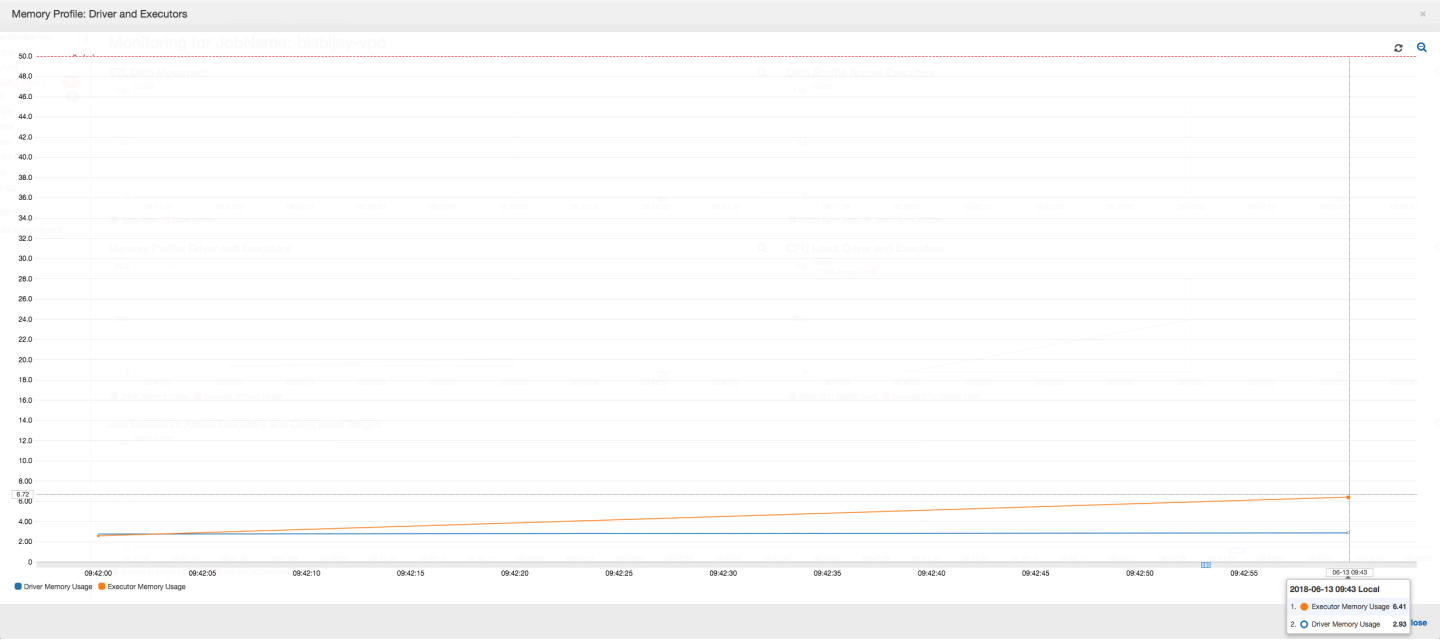
L'executor ha esaurito la memoria durante la lettura della tabella JDBC perché la configurazione predefinita per le dimensioni di recupero JDBC Spark è zero. Ciò significa che il driver JDBC nell'executor Spark cerca di recuperare i 34 milioni di righe del database contemporaneamente e di eseguirne la memorizzazione nella cache, anche se il flusso di Spark avviene una riga per volta. Con Spark, è possibile evitare questo scenario impostando il parametro relativo alle dimensioni di recupero su un valore predefinito diverso da zero.

È inoltre possibile risolvere questo errore usando frame dinamici AWS Glue. Per impostazione predefinita, i frame dinamici utilizzano una dimensione di recupero di 1.000 righe che in genere è un valore sufficiente. Di conseguenza, l'executor non usa più del 7% della memoria totale. Il processo AWS Glue termina in meno di due minuti con solo un singolo executor. Sebbene l'uso dei frame dinamici AWS Glue sia l'approccio consigliato, è anche possibile impostare la dimensione di recupero utilizzando la proprietà `fetchsize` di Apache Spark. Consulta la guida Spark, DataFrames SQL e Datasets.

```
val (url, database, tableName) = {
 ("jdbc_url", "db_name", "table_name")
 }
```

```
val source = glueContext.getSource(format, sourceJson)
val df = source.getDynamicFrame
glueContext.write_dynamic_frame.from_options(frame = df, connection_type = "s3",
 connection_options = {"path": output_path}, format = "parquet", transformation_ctx =
 "datasink")
```

Parametri profilati normali: la memoria dell'executor con frame dinamici AWS Glue non supera mai la soglia di sicurezza, come illustrato nell'immagine seguente. Il flusso passa nelle righe dal database e vengono memorizzate nella cache solo 1.000 righe nel driver JDBC in un determinato momento. Non si è verificata un'eccezione di memoria esaurita.




Debug di fasi impegnative e attività in ritardo

È possibile usare la profilatura dei processi AWS Glue per identificare le fasi impegnative e le attività in ritardo nei processi ETL (Extract, Transform and Load, estrazione, trasformazione e caricamento). Un'attività in ritardo richiede molto più tempo rispetto al resto delle attività in una fase di un processo AWS Glue. Di conseguenza, il completamento della fase richiede più tempo e aumenta il tempo totale di esecuzione del processo.

Unione di file di input di piccole dimensioni in file di output di dimensioni maggiori

Un'attività in ritardo può verificarsi in caso di distribuzione non uniforme del lavoro tra attività diverse oppure in caso di un'asimmetria dei dati a causa della quale un'attività elabora più dati.

È possibile profilare il codice seguente, che rappresenta un modello comune in Apache Spark, per unire un numero elevato di piccoli file in file di output di dimensioni maggiori. Per questo esempio, il

set di dati di input è costituito da 32 GB di file compressi Gzip JSON. Il set di dati di output include circa 190 GB di file JSON non compressi.

Il codice profilato è il seguente:

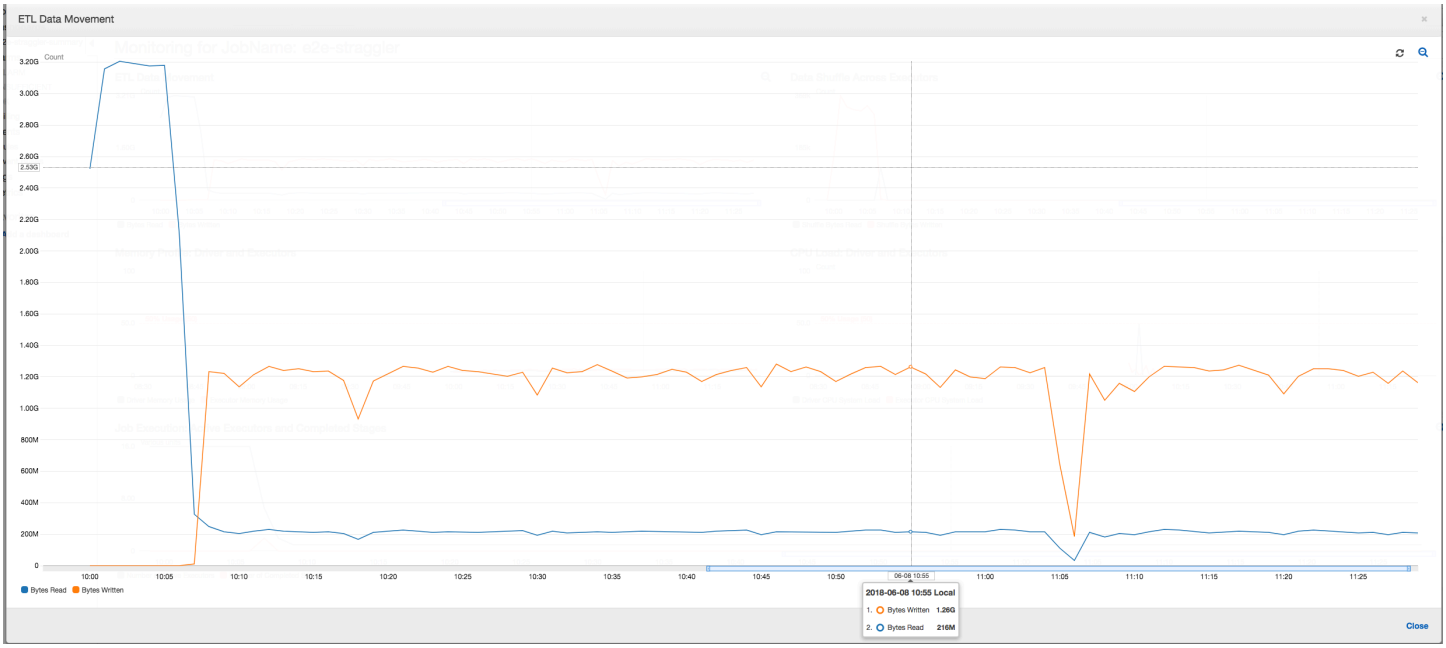
```
datasource0 = spark.read.format("json").load("s3://input_path")
df = datasource0.coalesce(1)
df.write.format("json").save(output_path)
```

Visualizzazione dei parametri profilati nella console AWS Glue

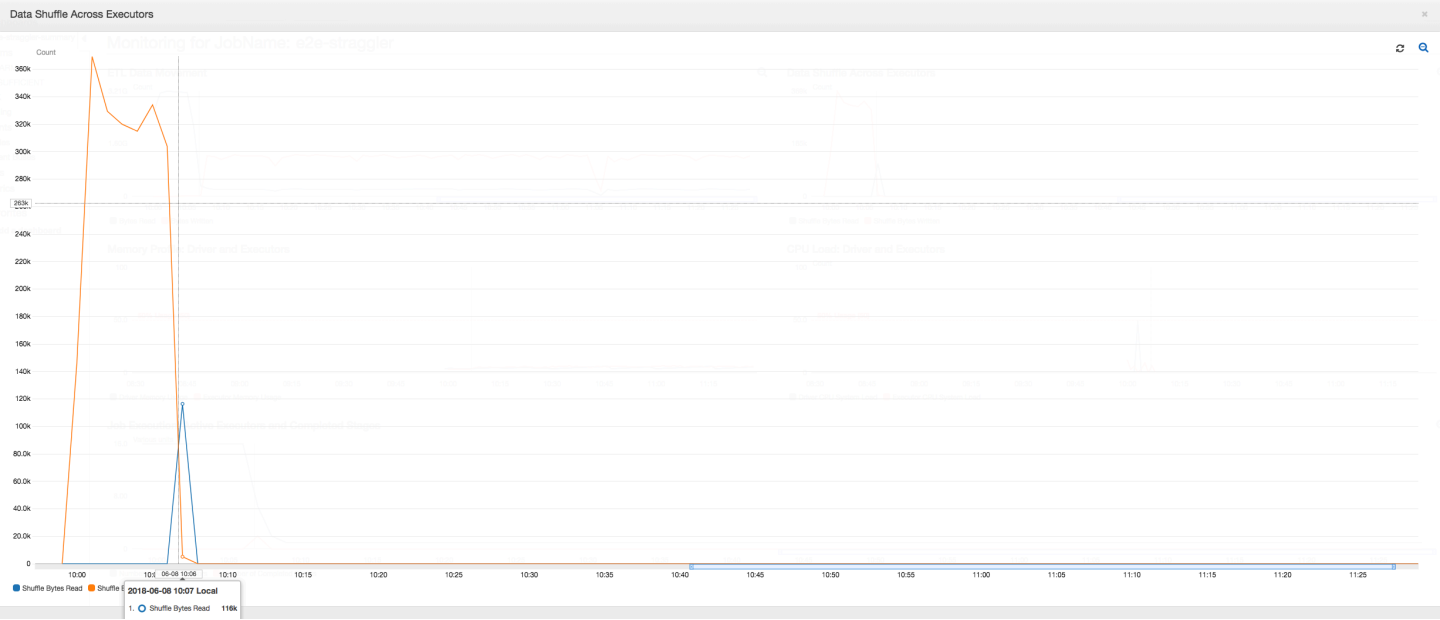
È possibile profilare il processo per esaminare quattro diversi set di parametri:

- Spostamento di dati ETL
- Distribuzione casuale dei dati tra executor
- Esecuzione del processo
- Profilo di memoria

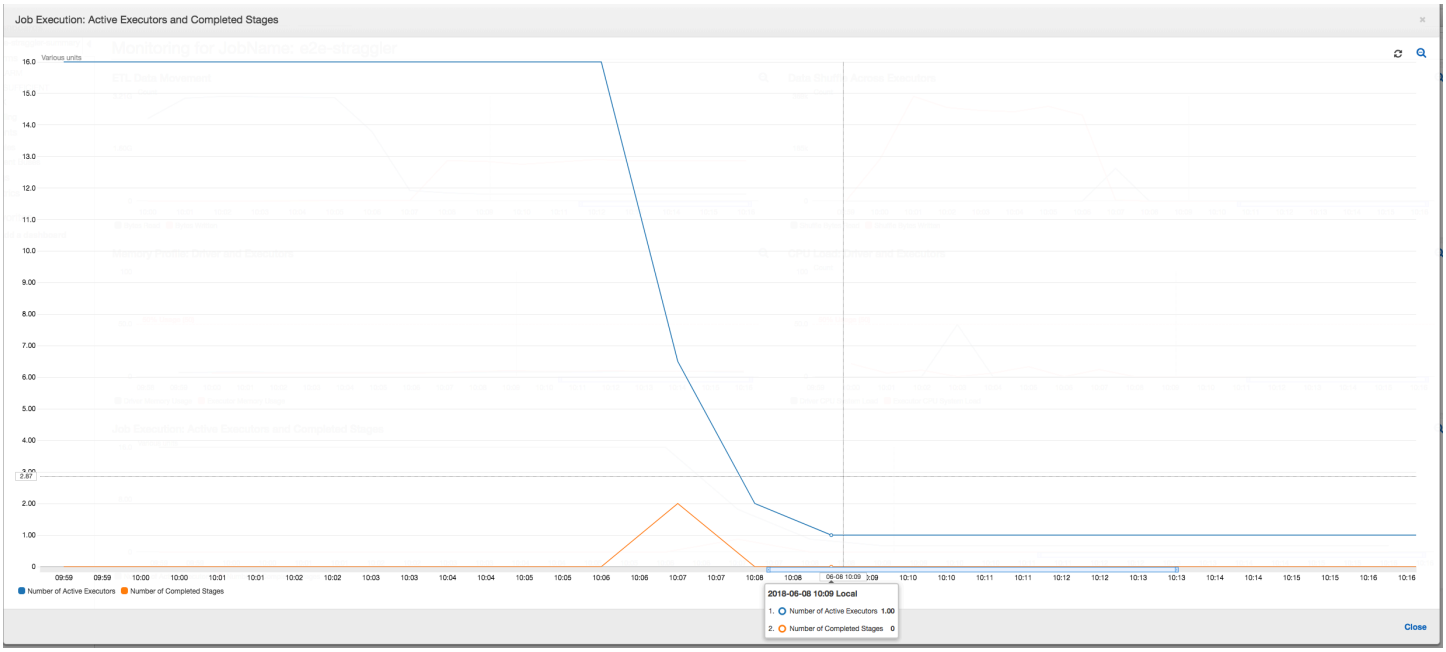
Spostamento di dati ETL: nel profilo ETL Data Movement (Spostamento di dati ETL) i byte vengono letti abbastanza rapidamente da tutti gli executor nella prima fase, che viene completata entro i primi sei minuti. Tuttavia, il tempo di esecuzione totale del processo è di circa un'ora, principalmente a causa delle operazioni di scrittura dei dati.



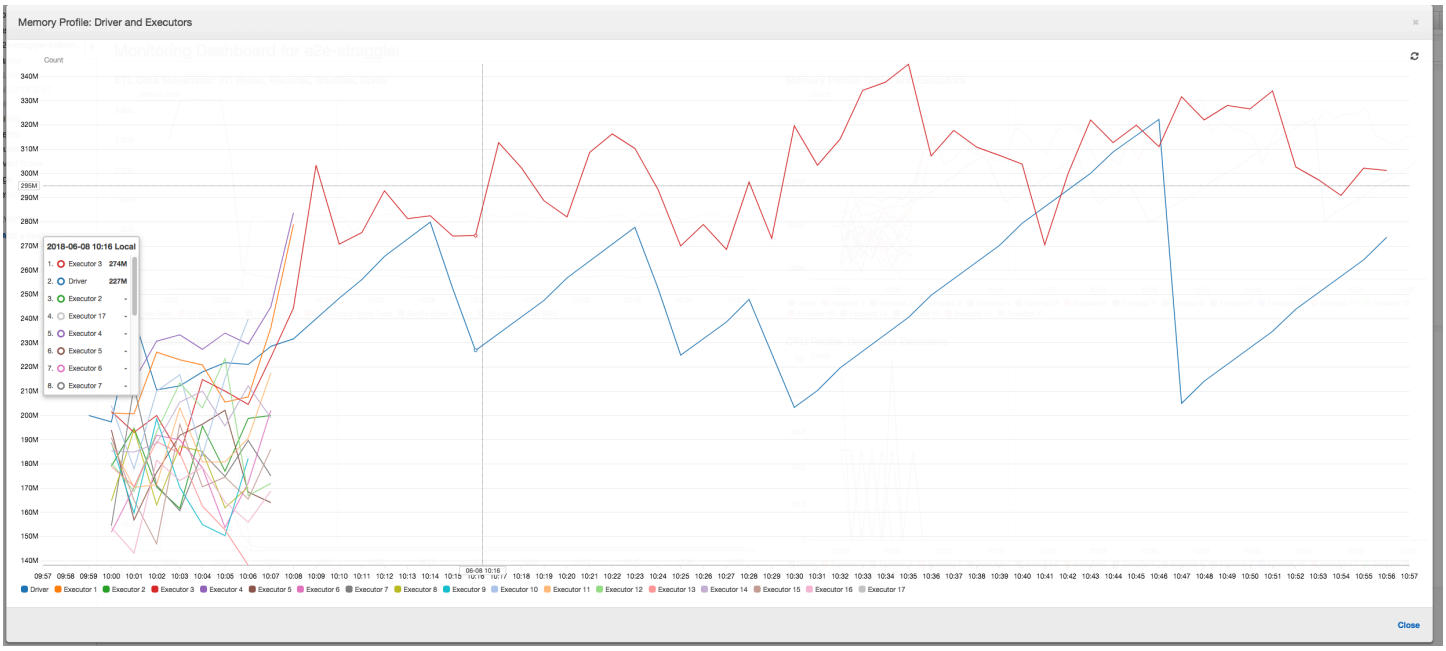
Distribuzione casuale dei dati tra executor: il numero di byte letti e scritti durante la distribuzione casuale mostra un picco prima del completamento della fase 2, come indicato dai parametri Job Execution (Esecuzione processo) e Data Shuffle (Distribuzione casuale dei dati). Dopo che i dati sono stati distribuiti in modo causale da tutti gli executor, le operazioni di lettura e scrittura procedono solo dall'executor numero 3.




Esecuzione del processo: come illustrato nel grafico seguente, tutti gli altri executor sono inattivi e vengono infine rilasciati entro le 10:09. A questo punto, il numero totale di executor scende a uno solo. Ciò mostra chiaramente che l'executor numero 3 è costituito dall'attività in ritardo che sta impiegando il tempo di esecuzione più lungo e che contribuisce in misura maggiore al tempo di esecuzione del processo.




Profilo di memoria: dopo le prime due fasi, solo l'executor numero 3 consuma attivamente memoria per elaborare i dati. Gli altri executor sono semplicemente inattivi o sono stati rilasciati poco dopo il completamento delle prime due fasi.




Correzione del calo degli executor tramite il raggruppamento

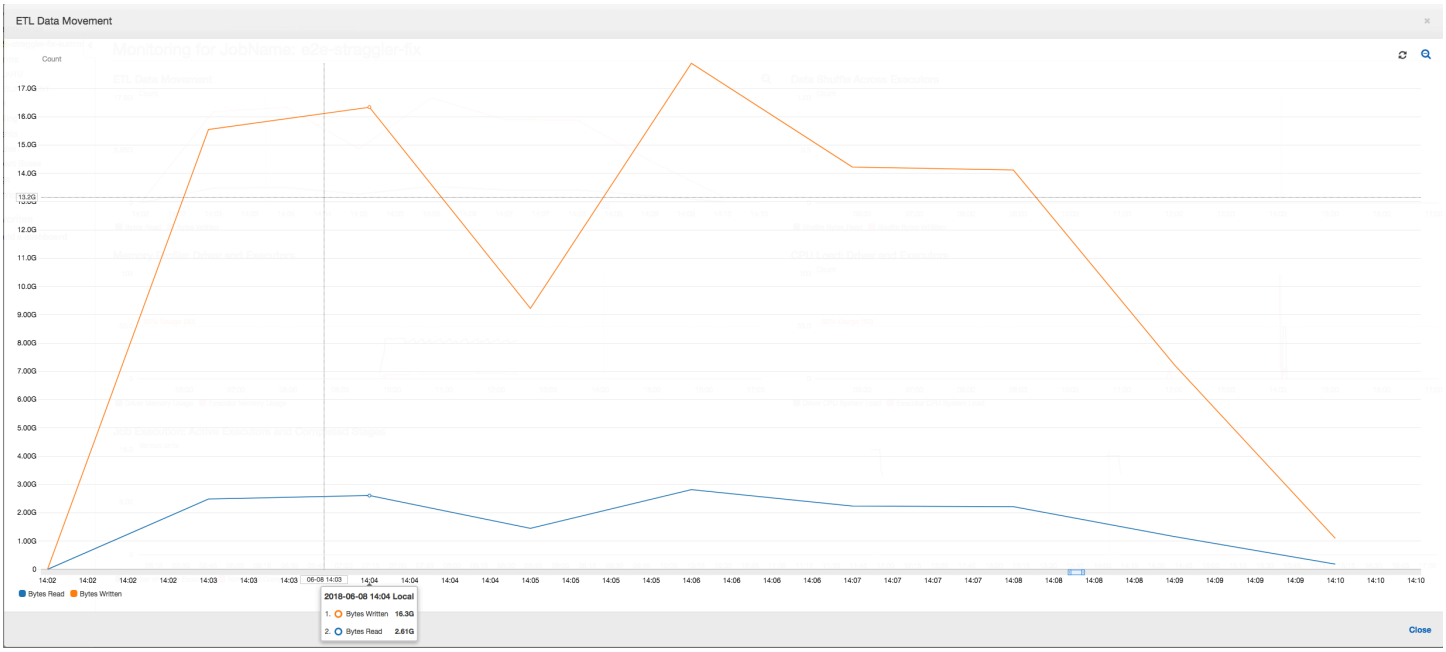
È possibile evitare il calo degli executor usando la caratteristica di raggruppamento in AWS Glue. Usa il raggruppamento per distribuire i dati in modo uniforme tra tutti gli executor e unire i file in file di

dimensioni maggiori usando tutti gli executor disponibili nel cluster. Per ulteriori informazioni, consulta Lettura di file di input in gruppi di grandi dimensioni.

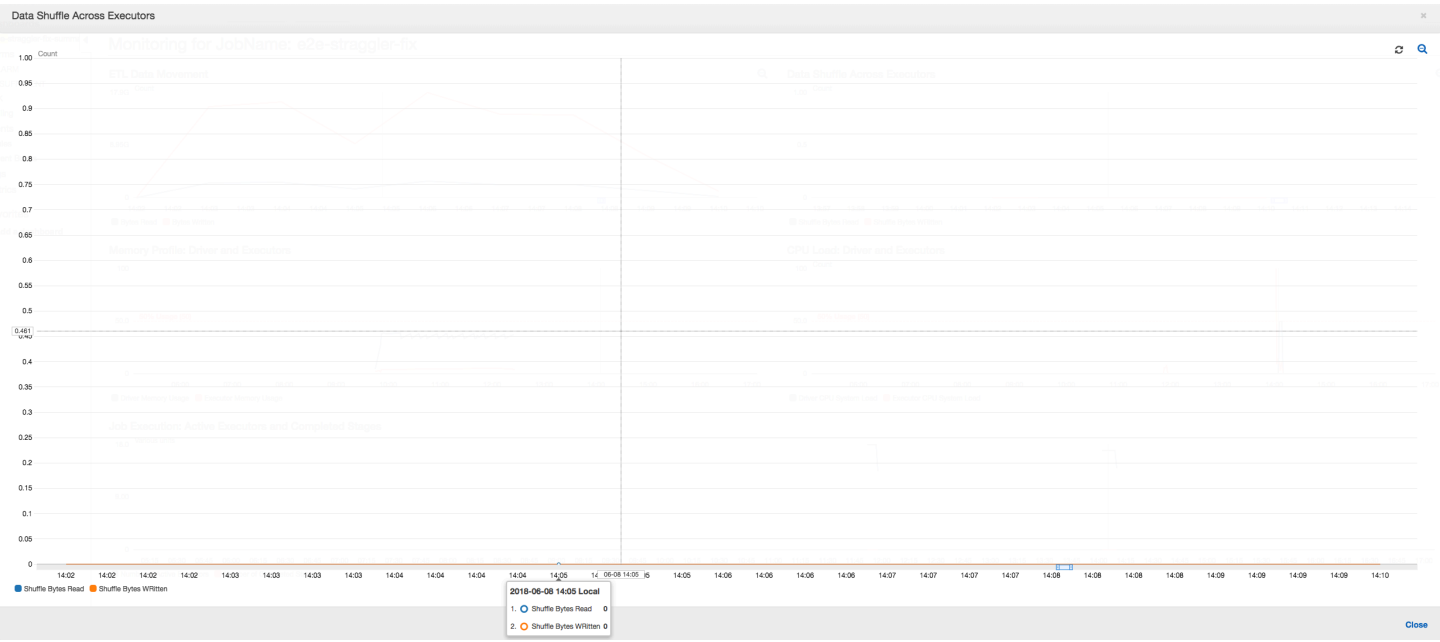
Per controllare gli spostamenti di dati ETL nel processo AWS Glue, profila il codice seguente con il raggruppamento abilitato:

```
df = glueContext.create_dynamic_frame_from_options("s3", {'paths': ["s3://input_path"],
 "recurse":True, 'groupFiles': 'inPartition'}, format="json")
datasink = glueContext.write_dynamic_frame.from_options(frame = df, connection_type =
 "s3", connection_options = {"path": output_path}, format = "json", transformation_ctx
 = "datasink4")
```

Spostamento di dati ETL: le operazioni di scrittura dei dati vengono ora trasmesse in parallelo alle operazioni di lettura dei dati per tutto il tempo di esecuzione del processo. Di conseguenza, il processo viene completato entro otto minuti, molto più rapidamente che in precedenza.



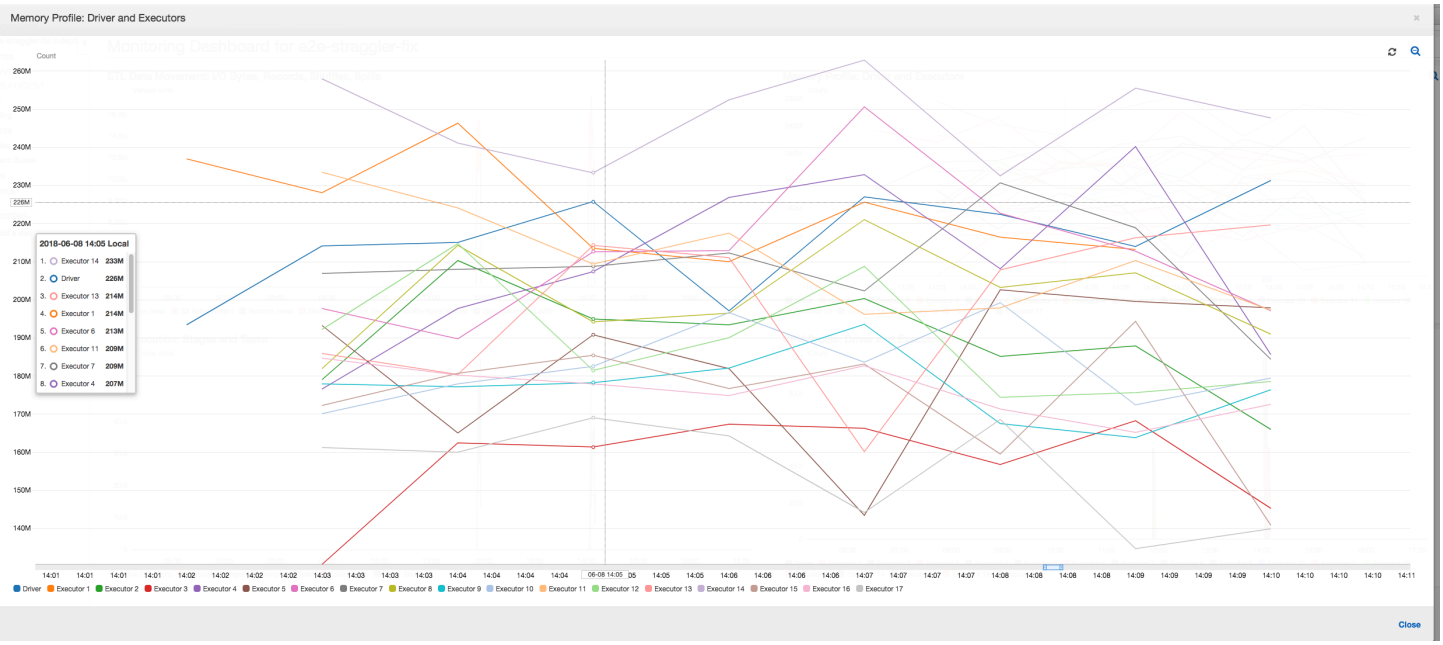

Distribuzione casuale dei dati tra executor: poiché i file di input vengono uniti durante le operazioni di lettura usando la caratteristica di raggruppamento, non vengono eseguite costose attività di distribuzione casuale dei dati dopo le operazioni di lettura dei dati.




Esecuzione del processo: i parametri relativi all'esecuzione del processo mostrano che il numero totale di executor attivi in esecuzione e che elaborano i dati rimane relativamente costante. Non vi sono singole attività in ritardo nel processo. Tutti gli executor sono attivi e non vengono rilasciati fino al completamento del processo. Poiché non vi è alcuna operazione intermedia di distribuzione casuale dei dati tra gli executor, il processo è costituito da un'unica fase.

Profilo di memoria: i parametri mostrano il consumo di memoria attivo tra tutti gli executor, riconfermando la presenza di attività in tutti gli executor. Poiché i dati vengono trasmessi e scritti in

parallelo, l'utilizzo totale di memoria di tutti gli executor è piuttosto uniforme e ben al di sotto della soglia di sicurezza per tutti gli executor.




Monitoraggio dell'avanzamento di processi multipli

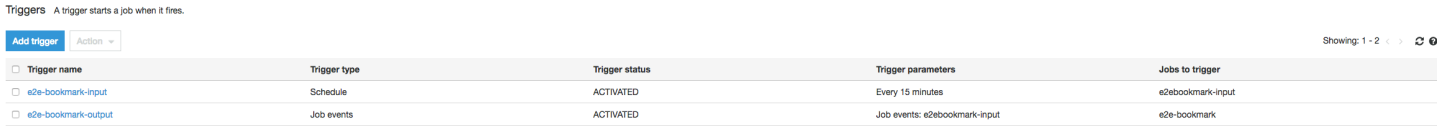
Puoi profilare più AWS Glue lavori insieme e monitorare il flusso di dati tra di essi. Si tratta di un modello di flusso di lavoro comune e richiede il monitoraggio per l'avanzamento del processo individuale, il backlog dell'elaborazione dei dati, la rielaborazione dei dati e i segnalibri dei processi.

Argomenti

- Codice profilato
- Visualizzazione dei parametri profilati nella console AWS Glue
- Correzione dell'elaborazione dei file

Codice profilato

In questo flusso di lavoro, ci sono due processi: un processo di input e uno di output. Il processo di input è pianificato per l'esecuzione ogni 30 minuti usando un trigger periodico. Il processo di output è pianificato per l'esecuzione dopo ogni esecuzione del processo di input. Questi processi pianificati sono controllati usando trigger dei processi.



Processo di input: questo processo legge i dati da una posizione Amazon Simple Storage Service (Amazon S3), li trasforma tramite `ApplyMapping` e li scrive in una posizione Amazon S3 di staging. Il codice seguente è il codice profilato per il processo di input:

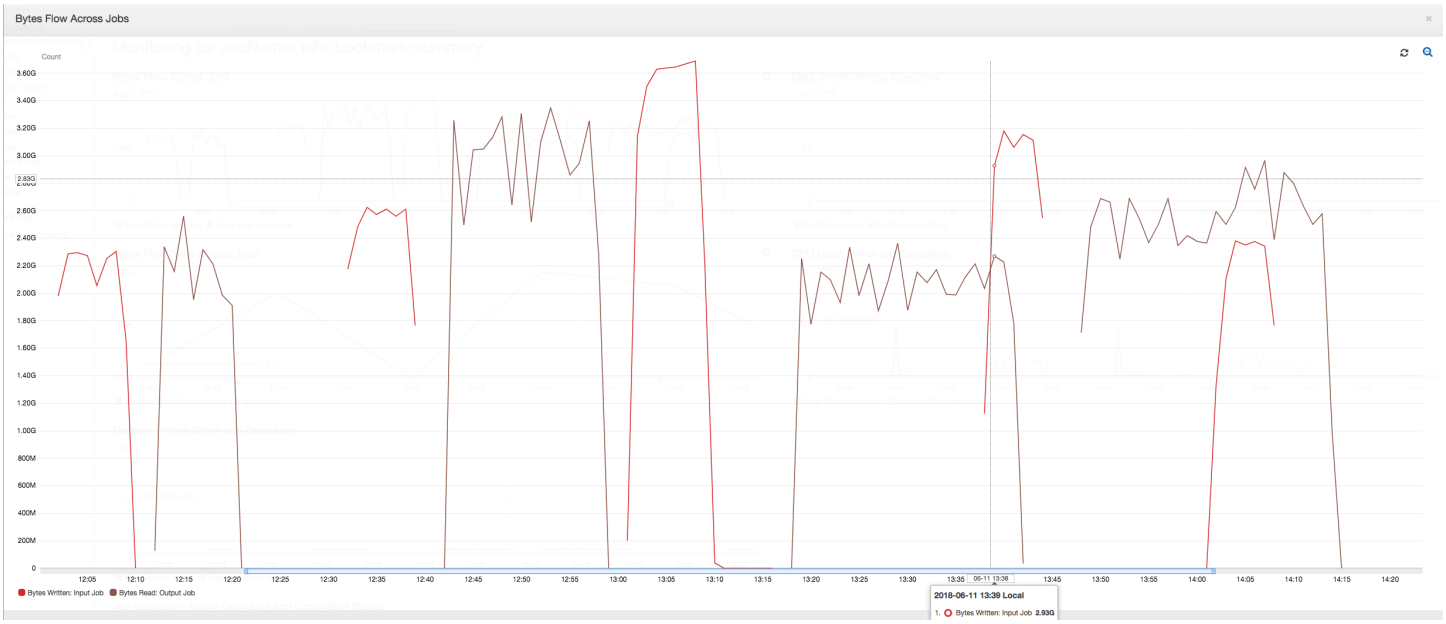
```
datasource0 = glueContext.create_dynamic_frame.from_options(connection_type="s3",
 connection_options = {"paths": ["s3://input_path"],
 "useS3ListImplementation":True,"recurse":True}, format="json")
applymapping1 = ApplyMapping.apply(frame = datasource0, mappings = [map_spec])
datasink2 = glueContext.write_dynamic_frame.from_options(frame = applymapping1,
 connection_type = "s3", connection_options = {"path": staging_path, "compression":
 "gzip"}, format = "json")
```

Processo di output: questo processo legge l'output del processo di input dalla posizione di staging in Amazon S3, lo trasforma nuovamente e lo scrive in una destinazione:

```

datasource0 = glueContext.create_dynamic_frame.from_options(connection_type="s3",
 connection_options = {"paths": [staging_path],
 "useS3ListImplementation":True,"recurse":True}, format="json")
applymapping1 = ApplyMapping.apply(frame = datasource0, mappings = [map_spec])
datasink2 = glueContext.write_dynamic_frame.from_options(frame = applymapping1,
 connection_type = "s3", connection_options = {"path": output_path}, format = "json")
```

Visualizzazione dei parametri profilati nella console AWS Glue

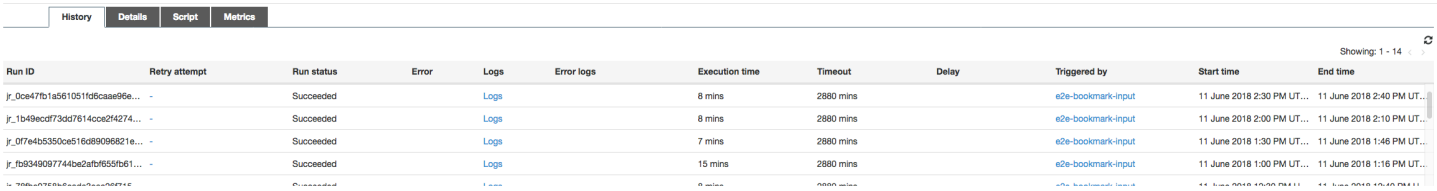
Il pannello di controllo seguente sovrappone il parametro Amazon S3 dei byte scritti dal processo di input al parametro Amazon S3 dei byte letti nella stessa sequenza temporale del processo di output. La sequenza temporale mostra diverse esecuzioni dei processi di input e di output. Il processo di input (mostrato in rosso) inizia ogni 30 minuti. Il processo di output (mostrato in marrone) viene avviato al completamento del processo di input, con una simultaneità massima di 1.



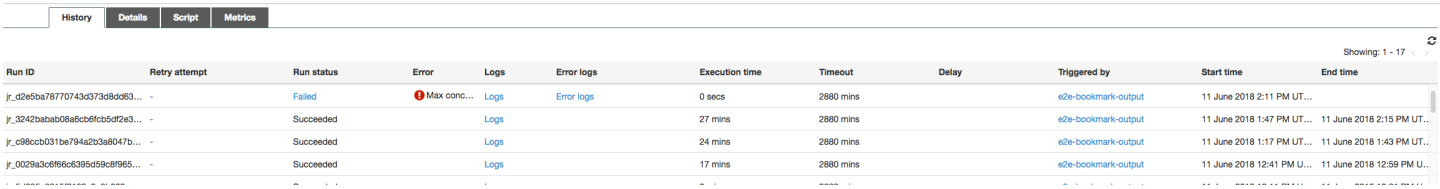

In questo esempio, i segnalibri dei processi non sono abilitati. Non vengono usati contesti di trasformazione per abilitare i segnalibri dei processi nel codice dello script.

Cronologia dei processi: i processi di input e di output hanno più esecuzioni, come illustrato nella scheda History (Cronologia), a partire dalle 12:00.

Il processo di input nella console AWS Glue ha il seguente aspetto:



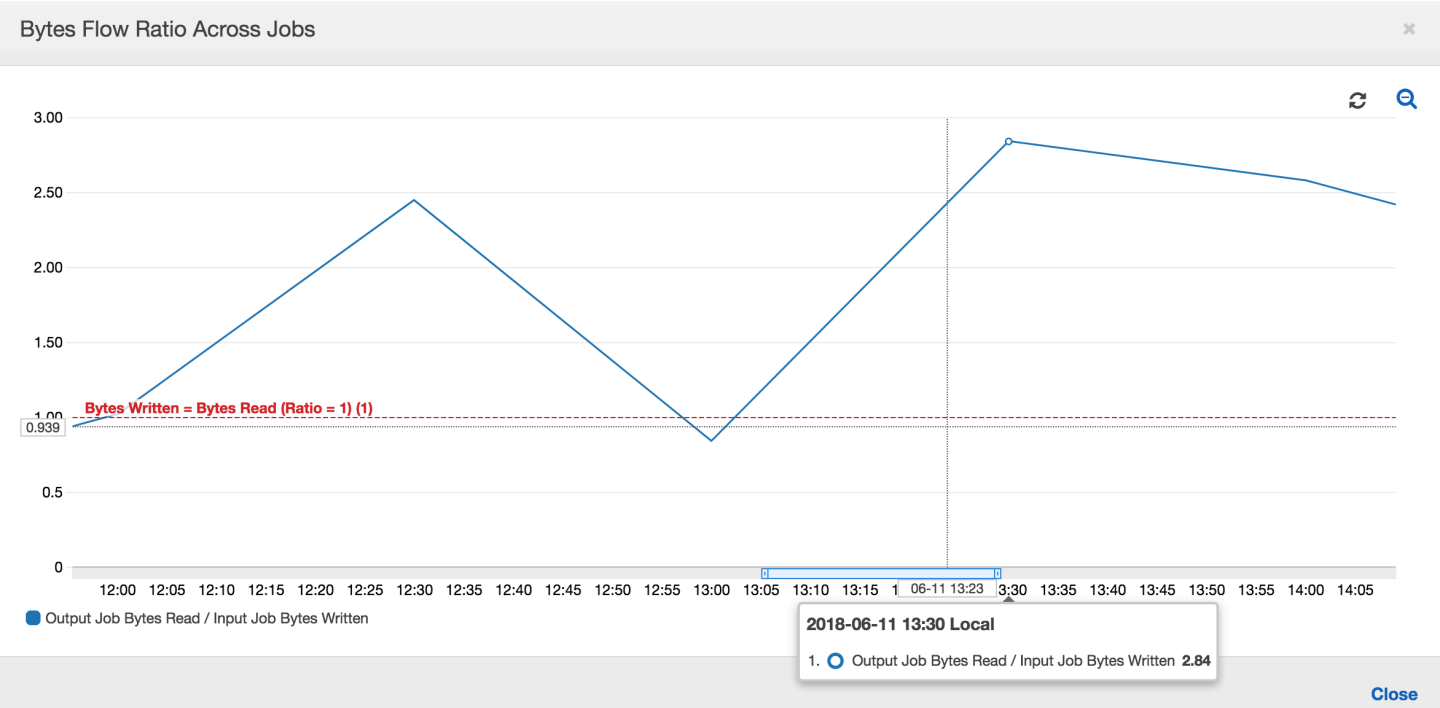

L'immagine seguente mostra il processo di output:




Prime esecuzioni dei processi: come illustrato nel grafico seguente dei byte di dati letti e scritti, le prime esecuzioni dei processi di input e di output tra le 12:00 e 12:30 mostrano pressappoco la stessa area sotto le curve. Tali aree rappresentano i byte Amazon S3 scritti dal processo di input e i byte Amazon S3 letti dal processo di output. Questi dati vengono inoltre confermati dal rapporto di byte Amazon S3 scritti (sommati nell'arco di 30 minuti, la frequenza del trigger dei processi per il

processo di input). Il punto dati del rapporto dell'esecuzione del processo di input iniziato alle 12:00 è 1.

Il grafico seguente mostra il rapporto del flusso di dati in tutte le esecuzioni dei processi:




Seconde esecuzioni dei processi: nella seconda esecuzione del processo, c'è una chiara differenza tra il numero di byte letti dal processo di output rispetto al numero di byte scritti dal processo di input. Confronta l'area sotto la curva tra le due esecuzioni del processo di output o confronta le aree nella seconda esecuzione dei processi di input e di output. Il rapporto tra i byte letti e scritti mostra che il processo di output ha letto 2,5 volte i dati scritti dal processo di input nel secondo intervallo di 30 minuti dalle 12:30 alle 13:00. Ciò è dovuto al fatto che il processo di output ha rielaborato l'output della prima esecuzione del processo di input perché i segnalibri non erano abilitati. Un rapporto superiore a 1 mostra che c'è un ulteriore backlog di dati che è stato elaborato dal processo di output.

Terze esecuzioni dei processi: il processo di input è abbastanza coerente in termini di numero di byte scritti (vedi l'area sotto le curve rosse). Tuttavia, la terza esecuzione del processo di input è durata più tempo del previsto (vedi la lunga coda della curva rossa). Di conseguenza, la terza esecuzione del processo di output è iniziata tardi. La terza esecuzione del processo ha elaborato solo una parte dei dati accumulati nella posizione di staging nei rimanenti 30 minuti tra le 13:00 e le 13:30. Il rapporto del flusso di byte mostra che ha elaborato solo un valore pari a 0,83 dei dati scritti dalla terza esecuzione del processo di input (vedi il rapporto alle 13:00).

Sovrapposizione dei processi di input e di output: la quarta esecuzione del processo di input è iniziata alle 13:30 in base alla pianificazione, prima del completamento della terza esecuzione del processo di output. C'è una sovrapposizione parziale tra queste due esecuzioni del processo. Tuttavia, la terza esecuzione del processo di output acquisisce solo i file elencati nella posizione di staging di Amazon S3 al momento dell'avvio, intorno alle 13:17. Ciò corrisponde a tutti i dati di output della prima esecuzione del processo di input. Il rapporto effettivo alle 13:30 è di circa 2,75. La terza esecuzione del processo di output ha elaborato circa 2,75 volte la quantità di dati scritti dalla quarta esecuzione del processo di input dalle 13:30 alle 14:00.

Come mostrano queste immagini, il processo di output sta rielaborando i dati dalla posizione di staging di tutte le esecuzioni precedenti del processo di input. Di conseguenza, la quarta esecuzione del processo di output è la più lunga e si sovrappone all'intera quinta esecuzione del processo di input.

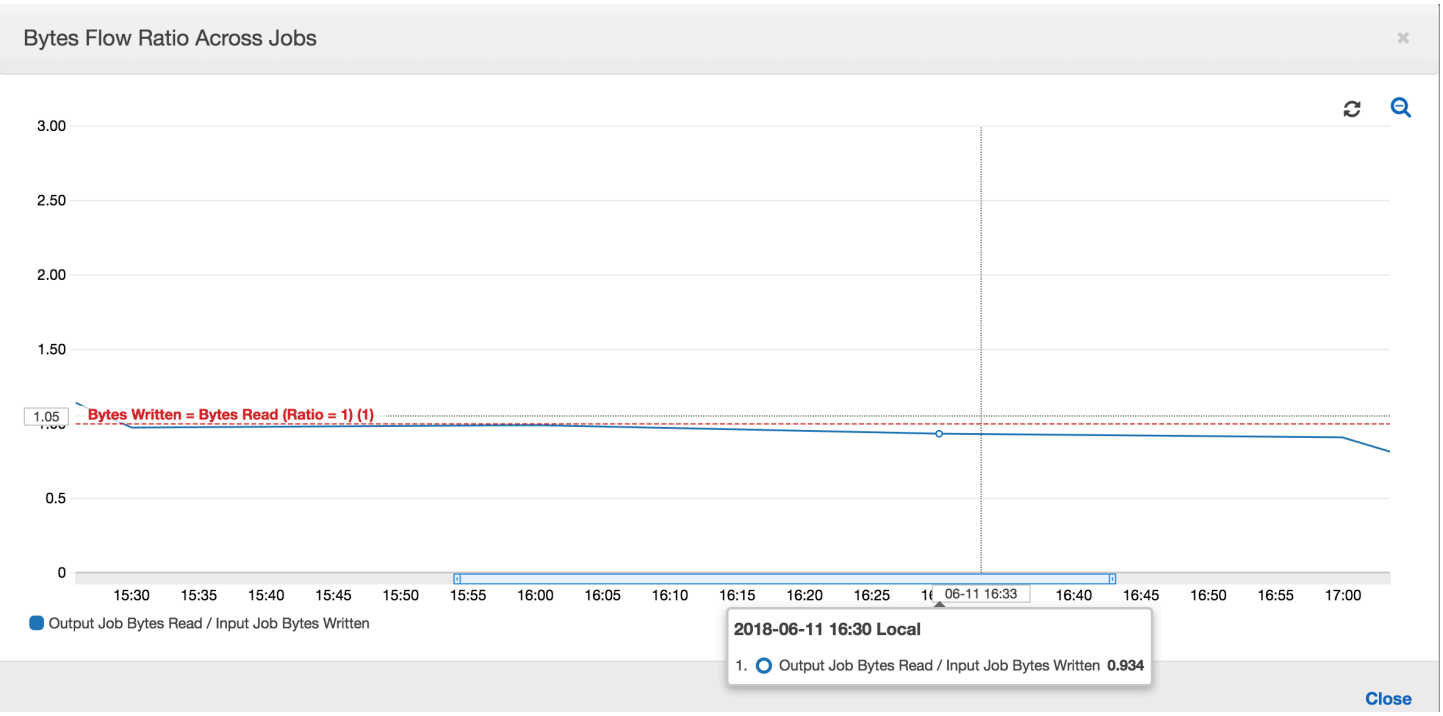
Correzione dell'elaborazione dei file

È necessario accertarsi che i processi di output elaborino solo i file che non sono stati elaborati da esecuzioni precedenti del processo di output. A tale scopo, abilita i segnalibri dei processi e imposta il contesto di trasformazione nel processo di output, come segue:

```
datasource0 = glueContext.create_dynamic_frame.from_options(connection_type="s3",
 connection_options = {"paths": [staging_path],
 "useS3ListImplementation":True,"recurse":True}, format="json", transformation_ctx =
 "bookmark_ctx")
```

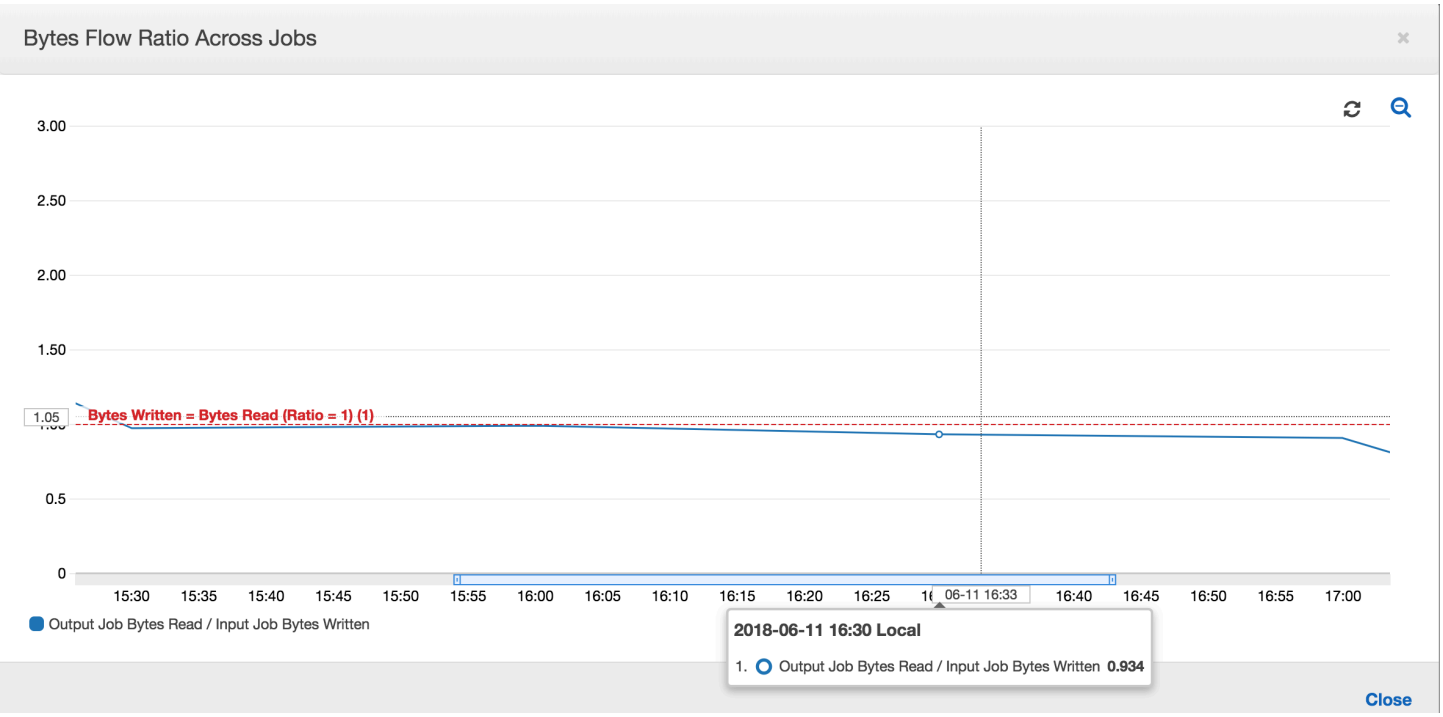
Con i segnalibri dei processi abilitati, il processo di output non rielabora i dati nella posizione di staging di tutte le precedenti esecuzioni del processo di input. Nell'immagine seguente, che mostra i dati letti e scritti, l'area sotto la curva marrone è abbastanza coerente e simile alle curve rosse.

I rapporti dei flussi di byte rimangono abbastanza vicino a 1 perché non ci sono dati aggiuntivi elaborati.




Un'esecuzione del processo di output viene avviata e acquisisce i file nella posizione di staging prima che la successiva esecuzione del processo di input inizi a inserire ulteriori dati nella posizione

di staging. Fino a quando ciò avviene, vengono elaborati solo i file acquisiti dall'esecuzione del processo di input precedente e il rapporto rimane vicino a 1.




Supponiamo che il processo di input richieda più tempo del previsto e, di conseguenza, il processo di output acquisisca i file nella posizione di staging da due esecuzioni del processo di input. Il rapporto è quindi superiore a 1 per l'esecuzione del processo di output. Tuttavia, le esecuzioni successive del processo di output non elaborano file già elaborati dalle esecuzioni precedenti del processo di output.

Monitoraggio per la pianificazione della capacità DPU

Puoi utilizzare le metriche del lavoro AWS Glue per stimare il numero di unità di elaborazione dati (DPUs) che possono essere utilizzate per scalare un AWS Glue lavoro.

> **Note**
>
> Questa pagina è applicabile solo a AWS Glue versioni 0.9 e 1.0. Le versioni successive di AWS Glue contengono funzionalità di riduzione dei costi che introducono considerazioni aggiuntive nella pianificazione della capacità.

Argomenti

- Codice profilato
- Visualizzazione dei parametri profilati nella console AWS Glue

- Determinazione della capacità DPU ottimale

Codice profilato

Lo script seguente legge una partizione Amazon Simple Storage Service (Amazon S3) contenente 428 file JSON GZIP. Lo script applica una mappatura per modificare i nomi dei campi, li converte e li scrive in Amazon S3 in un formato Apache Parquet. Si esegue il provisioning di 10 DPUs in base all'impostazione predefinita e si esegue questo processo.

```
datasource0 = glueContext.create_dynamic_frame.from_options(connection_type="s3",
 connection_options = {"paths": [input_path],
 "useS3ListImplementation":True,"recurse":True}, format="json")
applymapping1 = ApplyMapping.apply(frame = datasource0, mappings = [(map_spec])
datasink2 = glueContext.write_dynamic_frame.from_options(frame = applymapping1,
 connection_type = "s3", connection_options = {"path": output_path}, format =
 "parquet")
```

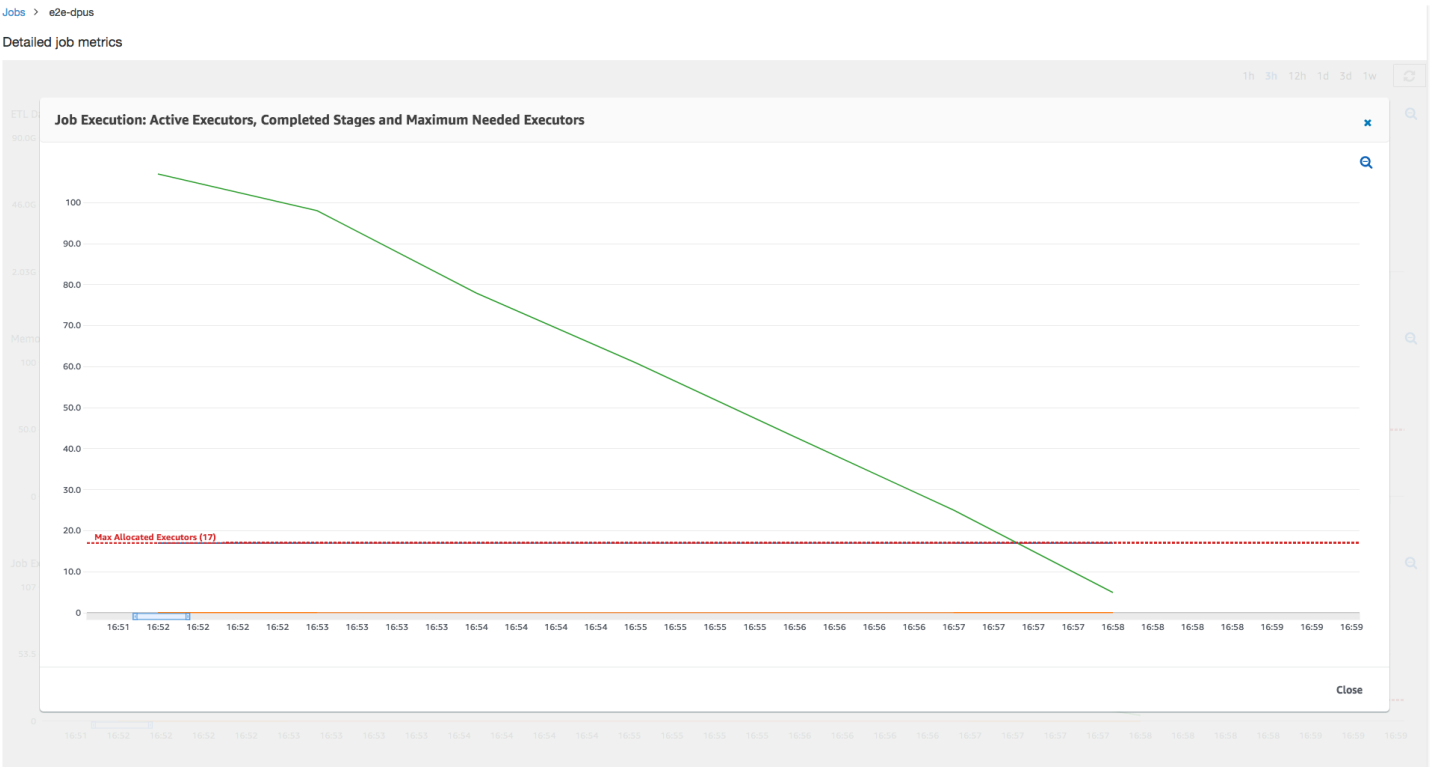
Visualizzazione dei parametri profilati nella console AWS Glue

Job run 1: In questo job run mostriamo come verificare se DPUs nel cluster non è disponibile un provisioning insufficiente. La funzionalità di esecuzione del processo in AWS Glue mostra il numero totale di esecutori attivi, il numero di fasi completate e il numero massimo di esecutori necessari.

Il numero massimo di executor necessari viene calcolato aggiungendo il numero totale di attività in esecuzione e attività in sospeso e dividendo per le attività per executor. Questo risultato è una misura del numero totale di executor necessari per soddisfare il carico corrente.

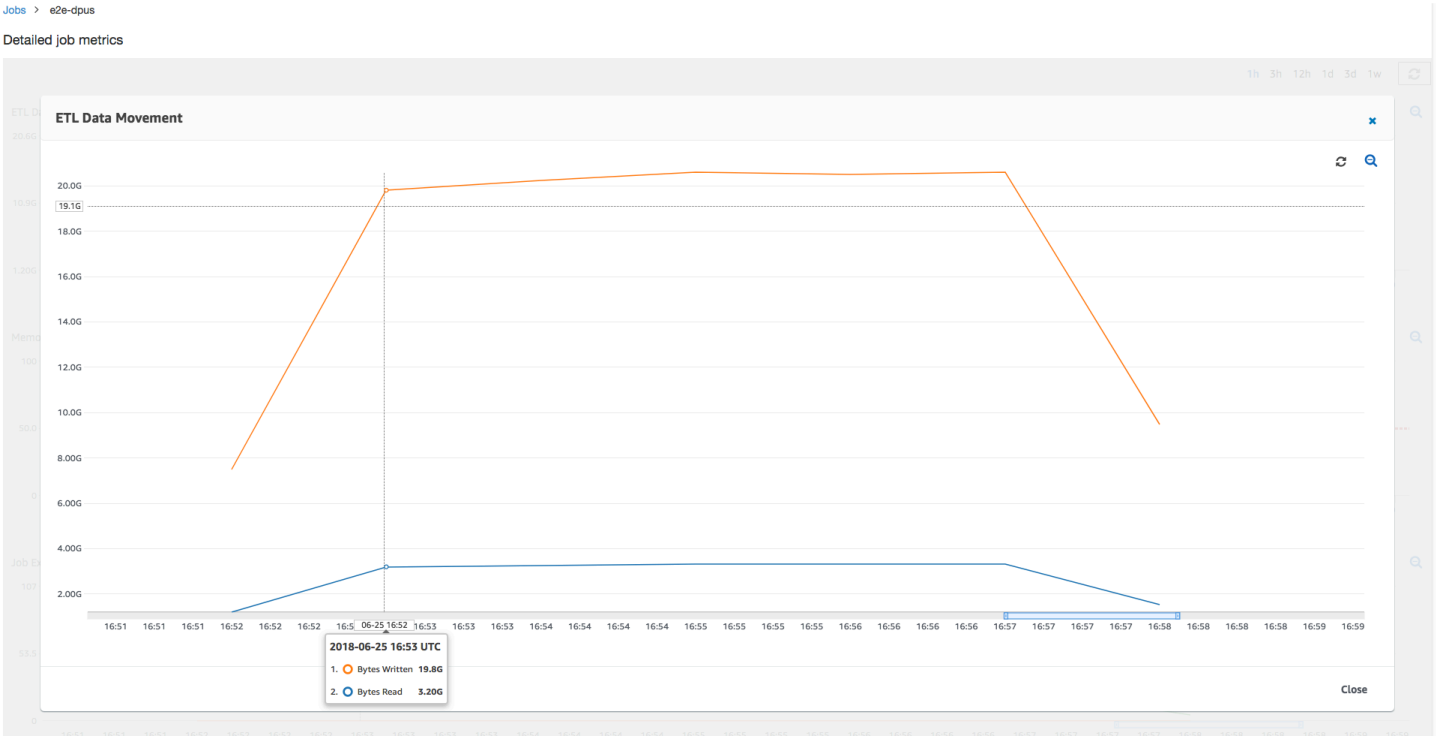
Al contrario, il numero di executor attivi misura quanti executor stanno eseguendo attivamente attività Apache Spark. Con l'avanzamento del processo, il numero massimo di executor necessari può cambiare e in genere diminuisce verso la fine del processo, in quanto le attività in coda si riducono.

La linea rossa orizzontale nel grafico seguente mostra il numero massimo di executor allocati, che dipende dal numero di esecutori allocati per il DPUs job. In questo caso, ne vengono allocati 10 DPUs per l'esecuzione del job. Una DPU è riservata alla gestione. Nove DPUs eseguono due executor ciascuno e un executor è riservato al driver Spark. Il driver Spark viene eseguito all'interno dell'applicazione principale. Pertanto, il numero massimo di executor allocati è 2*9 - 1 = 17 executor.




Come mostra il grafico, il numero massimo di executor parte da 107 all'inizio del processo, mentre il numero di executor attivi rimane 17. È lo stesso del numero massimo di executor allocati, pari a 10. DPUs Il rapporto tra il numero massimo di executor necessari e il numero massimo di executor allocati (aggiungendo 1 a entrambi per il driver Spark) indica il fattore di provisioning in difetto: 108/18 = 6x. È possibile eseguire il provisioning di 6 (in base al rapporto di provisioning) *9 (capacità DPU attuale - 1) + 1 DPUs = 55 DPUs per scalare il lavoro in modo da eseguirlo con il massimo parallelismo e terminarlo più velocemente.

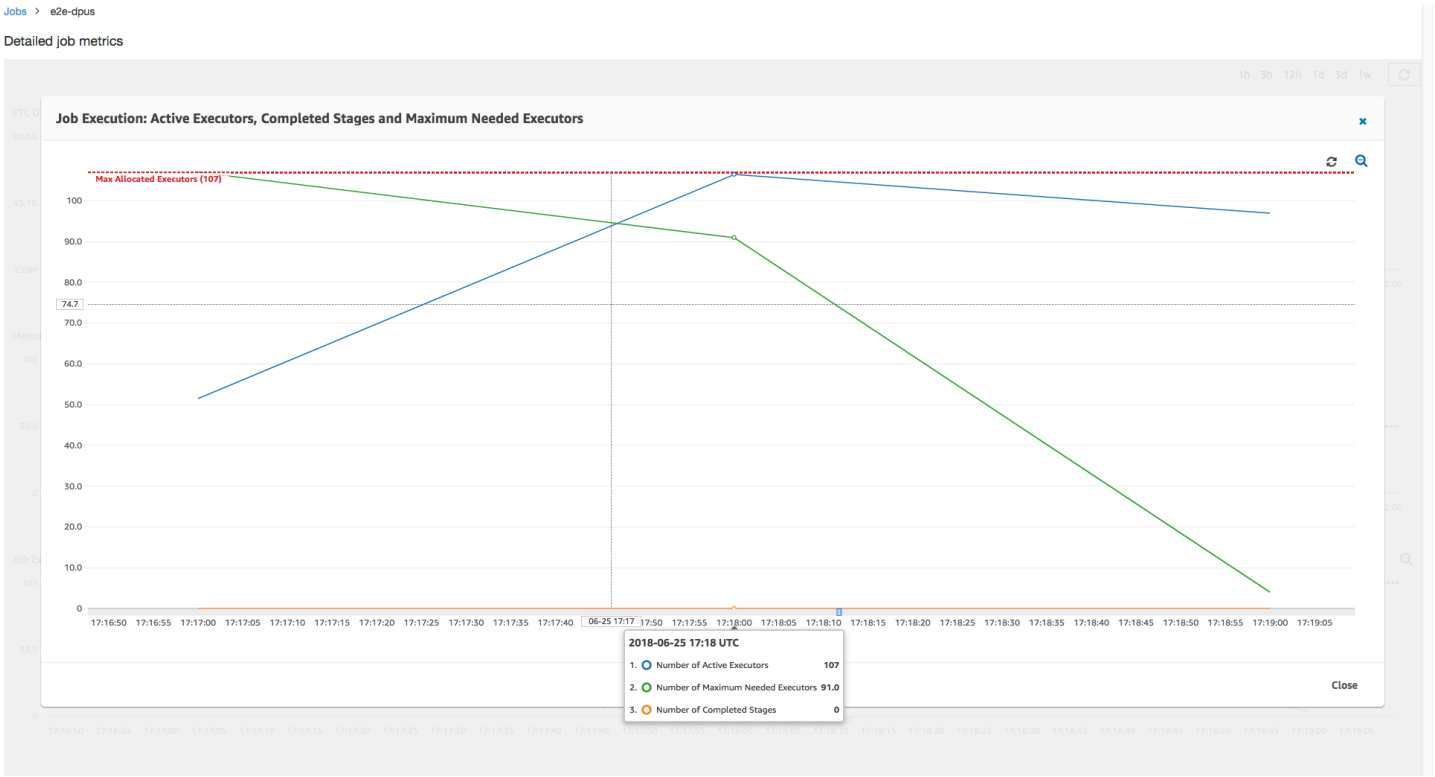
La console AWS Glue mostra i parametri dettagliati dei processi come una linea statica che rappresenta il numero massimo originale di esecutori allocati. La console calcola il numero massimo di esecutori allocati dalla definizione del processo per i parametri. Al contrario, per metriche dettagliate sull'esecuzione del processo, la console calcola il numero massimo di esecutori allocati dalla configurazione del job run, in particolare quello allocato per l'esecuzione del job. DPUs Per visualizzare i parametri dell'esecuzione di un singolo processo, seleziona l'esecuzione del processo e scegli View run metrics (Visualizza parametri di esecuzione).



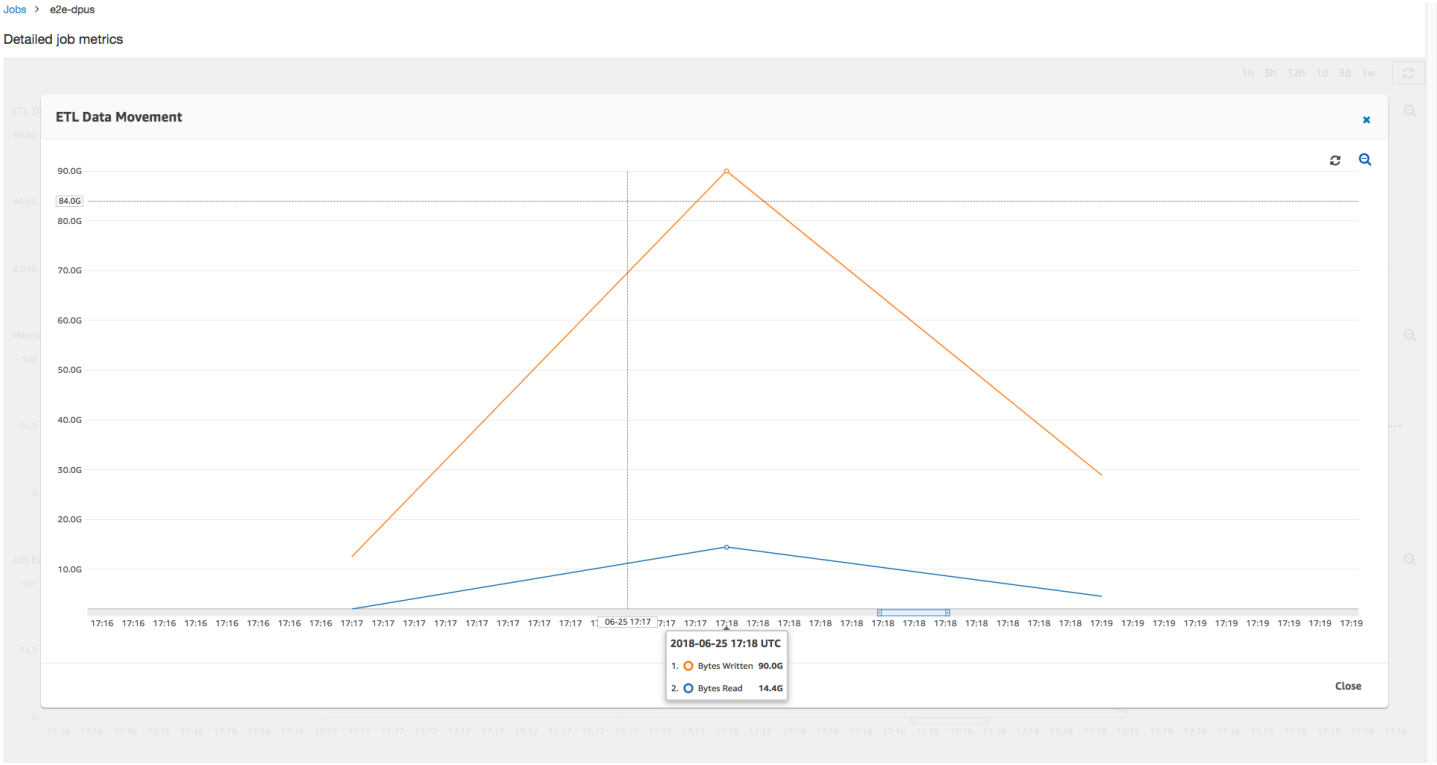

Osservando i byte Amazon S3 letti e scritti, si nota che il processo impiega tutti e sei i minuti per lo streaming in ingresso dei dati da Amazon S3 e la scrittura in uscita in parallelo. Tutti i core allocati DPUs sono in lettura e scrittura su Amazon S3. Il numero massimo di executor necessari (107), corrisponde anche al numero di file nel percorso di input Amazon S3 path428. Ogni executor può avviare quattro attività Spark per elaborare quattro file di input (JSON GZIP).

Determinazione della capacità DPU ottimale

In base ai risultati dell'esecuzione precedente, puoi aumentare il numero totale di job allocati DPUs a 55 e verificare le prestazioni del job. Il processo viene completato in meno di tre minuti, ossia in metà del tempo richiesto in precedenza. Il dimensionamento del processo non è lineare in questo caso, perché si tratta di un processo a esecuzione breve. I lavori con attività di lunga durata o con un numero elevato di attività (un numero elevato di esecutori massimi necessari) traggono vantaggio da un'accelerazione delle prestazioni con scalabilità orizzontale della close-to-linear DPU.



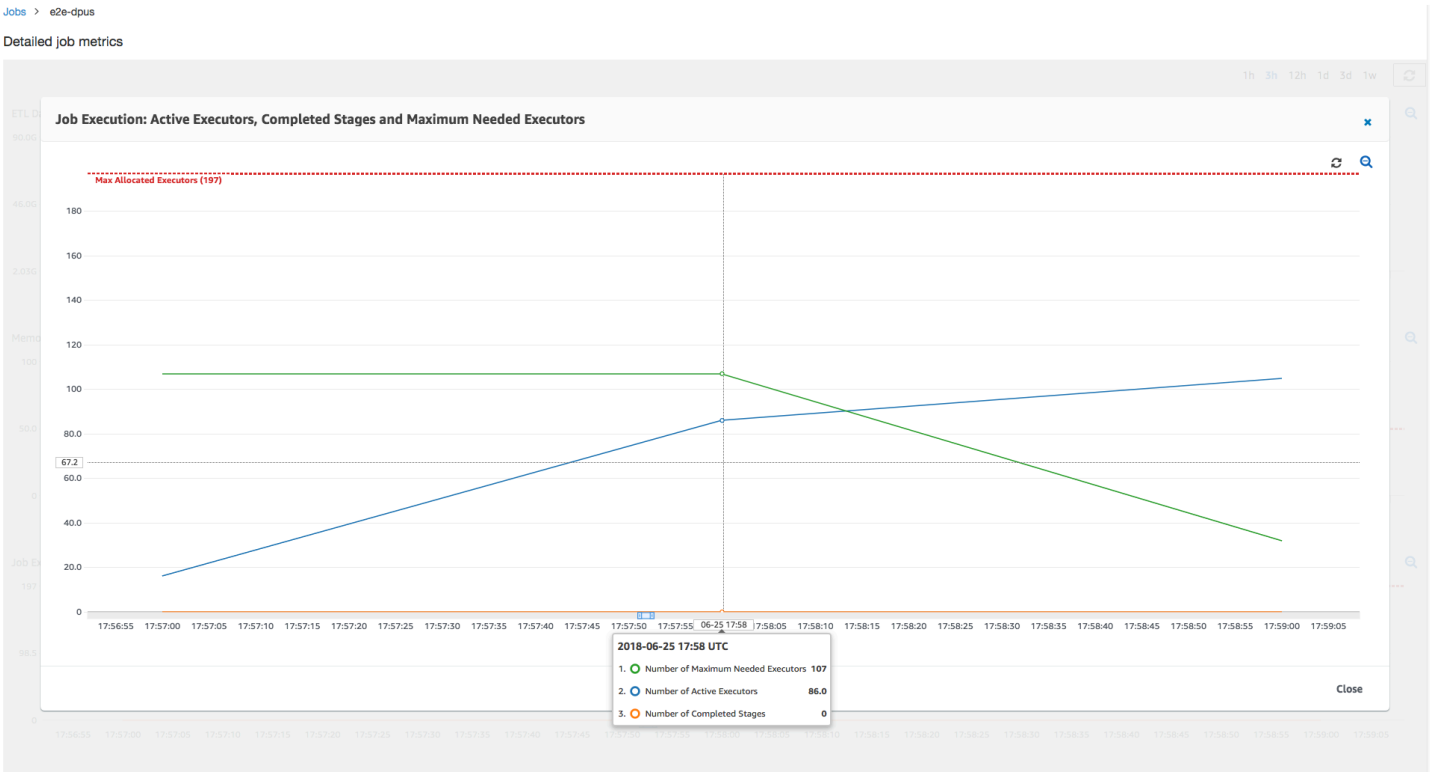

Come mostra l'immagine seguente, il numero totale di executor attivi raggiunge il numero massimo di executor allocati (107). Analogamente, il numero massimo di executor necessari non supera mai il numero massimo di executor allocati. Il numero massimo di executor necessari viene calcolato dai conteggi di attività in esecuzione e in attesa, quindi potrebbe essere inferiore al numero di executor attivi. Questo perché non ci possono essere executor che sono parzialmente o completamente inattivi per un breve periodo di tempo e non sono ancora stati rimossi.




Questa esecuzione del processo usa una quantità di executor sei volte maggiore per leggere e scrivere da Amazon S3 in parallelo. Di conseguenza, questa esecuzione del processo usa più larghezza di banda Amazon S3 per le operazioni di scrittura e lettura e termina più velocemente.

Identifica l'eccesso di approvvigionamento DPUs

Successivamente, è possibile determinare se la scalabilità orizzontale del lavoro con 100 DPUs (99 * 2 = 198 esecutori) contribuisca a un'ulteriore scalabilità orizzontale. Come mostra il grafico seguente, il processo richiede ancora tre minuti per giungere al termine. Analogamente, la scalabilità del job non supera i 107 executor (55 DPUs configurazioni) e i restanti 91 executor sono sovradimensionati e non vengono utilizzati affatto. Ciò dimostra che l'aumento del numero di esecutori DPUs potrebbe non sempre migliorare le prestazioni, come risulta evidente dal numero massimo di executor necessari.




Confronto tra differenze di tempo

Le tre esecuzioni di job mostrate nella tabella seguente riepilogano i tempi di esecuzione dei job per 10 DPUs DPUs, 55 e 100 DPUs. Puoi individuare la capacità DPU per migliorare il tempo di esecuzione del processo usando le stime definite monitorando la prima esecuzione del processo.

| Job ID | Numero di DPUs | Ora di esecuzione |
|---|---|---|
| jr_c894524c8ef5048a4d9... | 10 | 6 min. |
| jr_1a466cf2575e7ffe6856... | 55 | 3 min. |
| jr_34fa1ed4c6aa9ff0a814... | 100 | 3 min. |

## Risoluzione dei problemi di intelligenza artificiale generativa per Apache Spark in Glue AWS

La risoluzione dei problemi di intelligenza artificiale generativa per i lavori di Apache Spark in AWS Glue è una nuova funzionalità che aiuta i data engineer e gli scienziati a diagnosticare e risolvere i problemi nelle loro applicazioni Spark con facilità. Utilizzando tecnologie di machine learning e IA

generativa, questa funzionalità analizza i problemi nei processi Spark e fornisce un'analisi dettagliata delle cause principali insieme a consigli pratici per risolverli. La risoluzione dei problemi di intelligenza artificiale generativa per Apache Spark è disponibile per i lavori in esecuzione su AWS Glue versione 4.0 e successive.

Trasforma la risoluzione dei problemi di Apache Spark con il nostro agente di risoluzione dei problemi basato sull'intelligenza artificiale, che ora supporta tutte le principali modalità di implementazione tra cui Glue, Amazon EMR-, AWS Amazon EMR-Serverless e EC2 Amazon AI Notebooks. SageMaker Questo potente agente elimina i complessi processi di debug combinand o interazioni in linguaggio naturale, analisi del carico di lavoro in tempo reale e consigli sul codice intelligente in un'esperienza senza interruzioni. Per i dettagli sull'implementazione, consulta Cos'è l'agente di risoluzione dei problemi di Apache Spark per Amazon EMR. Guarda la seconda dimostrazione in Using the Troubleshooting Agent for AWS Glue: esempi di risoluzione dei problemi.

Come funzione la risoluzione dei problemi relativi all'IA generativa per Apache Spark?

Per i job Spark non riusciti, Generative AI Troubleshooting analizza i metadati del lavoro e le metriche e i log precisi associati alla firma di errore del job per generare un'analisi della causa principale e consiglia soluzioni e best practice specifiche per aiutare a risolvere i problemi del lavoro.

Configurare la risoluzione dei problemi relativi all'IA generativa per Apache Spark per i processi

Configurazione delle autorizzazioni IAM

La concessione delle autorizzazioni ai file APIs utilizzati da Spark Troubleshooting per i tuoi lavori in AWS Glue richiede le autorizzazioni IAM appropriate. Puoi ottenere le autorizzazioni allegando la seguente AWS policy personalizzata alla tua identità IAM (ad esempio un utente, un ruolo o un gruppo).

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
```

```
      "Action": [
        "glue:StartCompletion",
        "glue:GetCompletion"
      ],
      "Resource": [
        "arn:aws:glue:*:*:completion/*",
        "arn:aws:glue:*:*:job/*"
      ]
    }
  ]
}
```

> **Note**
>
> I due seguenti APIs vengono utilizzati nella policy IAM per abilitare questa esperienza tramite la console AWS Glue Studio: `StartCompletion` e`GetCompletion`.

Assegnare le autorizzazioni

Per fornire l'accesso, aggiungi autorizzazioni agli utenti, gruppi o ruoli:

- Per utenti e gruppi nel Centro identità IAM: Creazione di un set di autorizzazioni. Seguire le istruzioni riportate nella pagina Creazione di un set di autorizzazioni nella Guida per l'utente del Centro identità IAM.
- Per utenti gestiti in IAM tramite un provider di identità: crea un ruolo per la federazione delle identità. Seguire le istruzioni riportate nella pagina Creazione di un ruolo per un provider di identità di terze parti (federazione) nella Guida per l'utente di IAM.
- Per utenti IAM: Creare un ruolo che l'utente è in grado di assumere. Seguire le istruzioni riportate nella pagina Creazione di un ruolo per un utente IAM della Guida per l'utente di IAM.

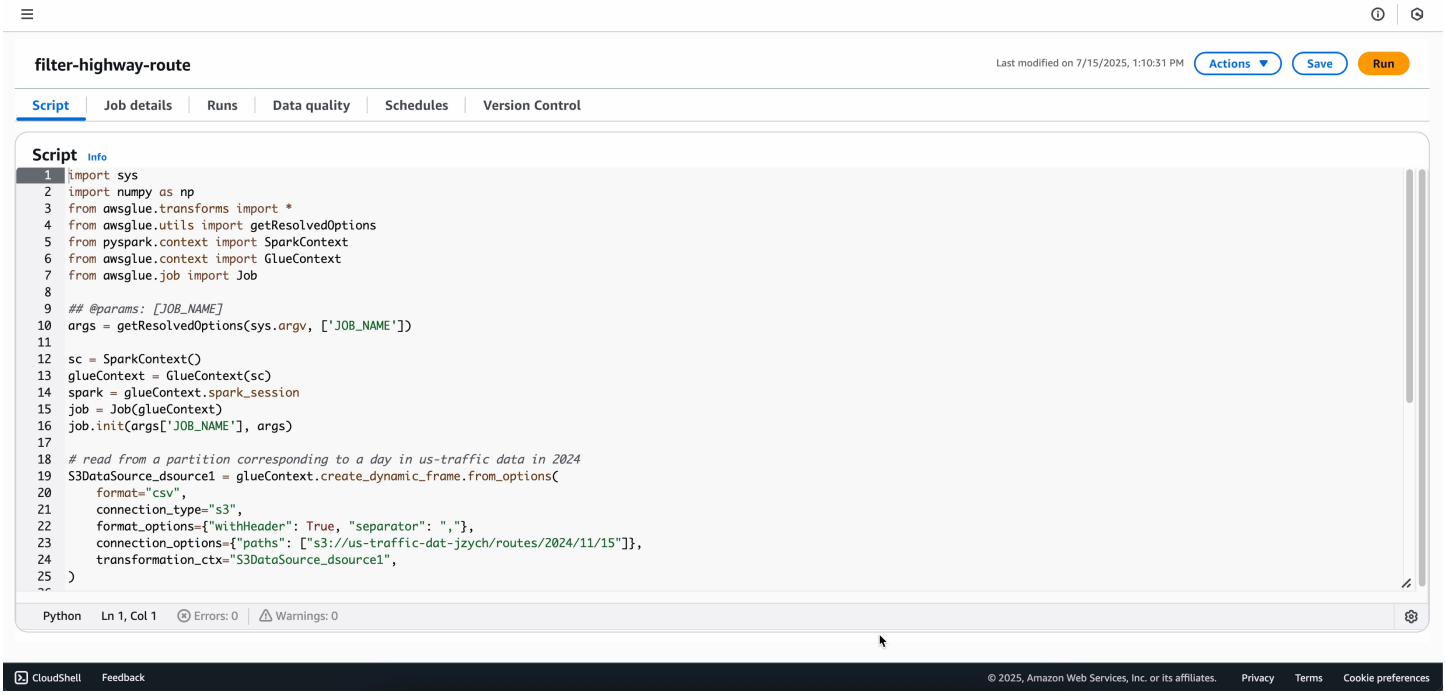
Esecuzione dell'analisi della risoluzione dei problemi da un'esecuzione non riuscita di un processo

È possibile accedere alla funzionalità di risoluzione dei problemi tramite più percorsi nella console AWS Glue. Ecco come iniziare:

Opzione 1: dalla pagina Elenco dei processi

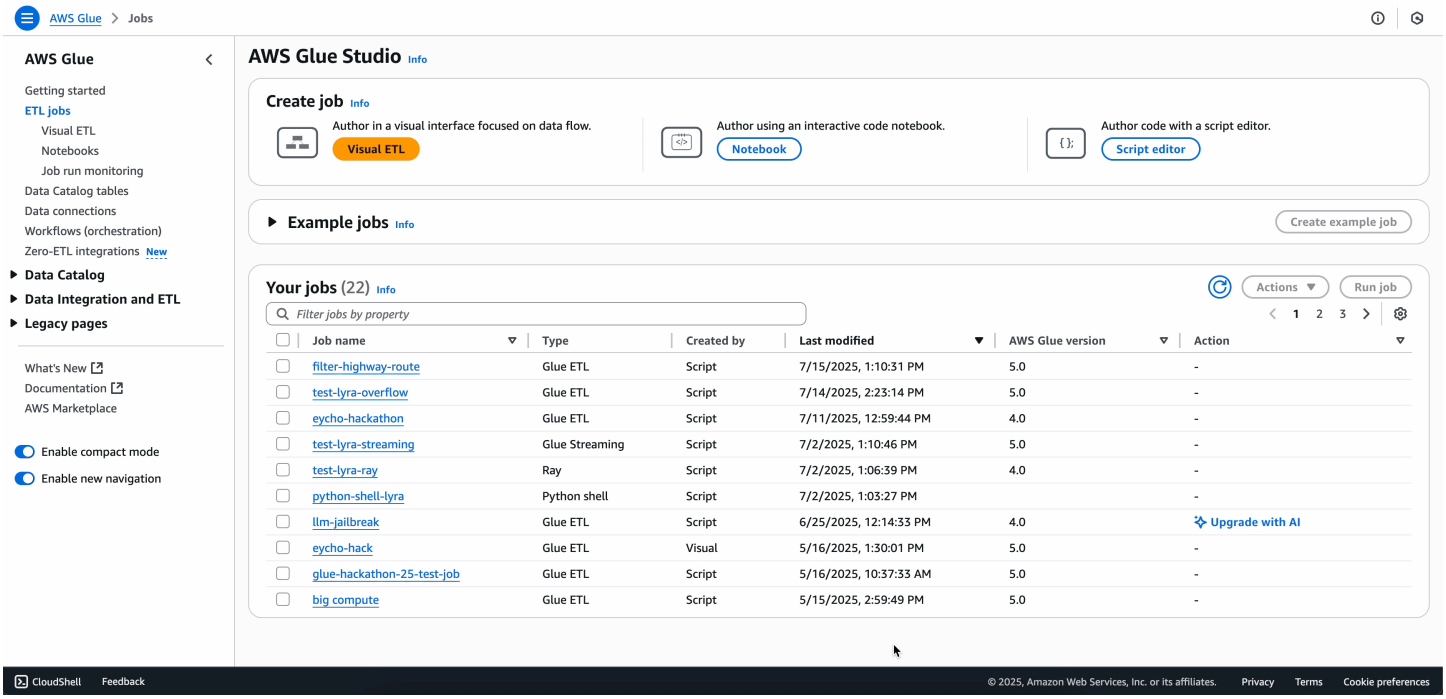
1. Apri la console AWS Glue all'indirizzo https://console.aws.amazon.com/glue/.

2. Nel riquadro di navigazione, scegliere Processi.
3. Individuare il processo non riuscito nell'elenco dei processi.
4. Selezionare la scheda Esecuzioni nella sezione dei dettagli del processo.
5. Fare clic sull'esecuzione del processo non riuscito che si desidera analizzare.
6. Scegliere Risoluzione dei problemi con IA per avviare l'analisi.
7. Una volta completata l'analisi della risoluzione dei problemi, è possibile visualizzare l'analisi della causa principale e i consigli nella scheda Analisi della risoluzione dei problemi nella parte inferiore dello schermo.




Opzione 2: usare la pagina Monitoraggio dell'esecuzione dei processi

1. Andare alla pagina Monitoraggio dell'esecuzione dei processi.
2. Individuare l'esecuzione del processo non riuscito.
3. Selezionare il menu a discesa Operazioni.
4. Scegliere Risoluzione dei problemi con IA.




Opzione 3: dalla pagina Dettagli dell'esecuzione dei processi

1. Andare sulla pagina dei dettagli dell'esecuzione del processo non riuscito facendo clic su Visualizza dettagli su un'esecuzione non riuscita dalla scheda Esecuzioni o selezionando il processo eseguito dalla pagina Monitoraggio dell'esecuzione dei processi.
2. Nella pagina dei dettagli dell'esecuzione del processo, individuare la scheda Analisi della risoluzione dei problemi.

## Categorie di risoluzione dei problemi supportate

Questo servizio si concentra su tre categorie principali di problemi che gli scienziati dei dati e gli sviluppatori affrontano frequentemente nelle loro applicazioni Spark:

- Errori di configurazione e accesso alle risorse: quando si eseguono applicazioni Spark in AWS Glue, gli errori di configurazione e accesso alle risorse sono tra i problemi più comuni ma difficili da diagnosticare. Questi errori si verificano spesso quando l'applicazione Spark tenta di interagire con AWS le risorse ma riscontra problemi di autorizzazione, risorse mancanti o problemi di configurazione.
- Problemi di memoria del driver Spark e dell'esecutore: gli errori relativi alla memoria nei processi di Apache Spark possono essere complessi da diagnosticare e risolvere. Questi errori si verificano

spesso quando i requisiti di elaborazione dei dati superano le risorse di memoria disponibili, sul nodo driver o sui nodi esecutori.

- Problemi di capacità del disco Spark: gli errori relativi allo storage nei job AWS Glue Spark spesso emergono durante le operazioni di shuffle, la fuoriuscita di dati o quando si ha a che fare con trasformazioni di dati su larga scala. Questi errori possono essere particolarmente complicati perché potrebbero verificarsi solo dopo un certo periodo di esecuzione del processo, con il rischio di sprecare risorse di elaborazione e tempo preziosi.
- Errori di esecuzione delle query: gli errori di query in Spark SQL e nelle DataFrame operazioni possono essere difficili da risolvere perché i messaggi di errore potrebbero non indicare chiaramente la causa principale e le query che funzionano bene con set di dati di piccole dimensioni possono improvvisamente fallire su larga scala. Questi errori diventano ancora più difficili quando si verificano all'interno di pipeline di trasformazione complesse, dove il vero problema può derivare da problemi di qualità dei dati nelle fasi precedenti piuttosto che dalla logica di interrogazione stessa.

> **Note**
>
> Prima di implementare le modifiche suggerite nell'ambiente di produzione, esaminare attentamente le modifiche suggerite. Il servizio fornisce consigli basati su modelli e best practice, ma il caso d'uso specifico potrebbe richiedere ulteriori considerazioni.

Regioni supportate

La risoluzione dei problemi di intelligenza artificiale generativa per Apache Spark è disponibile nelle seguenti regioni:

- Africa: Città del Capo (af-south-1)
- Asia Pacifico: Hong Kong (ap-east-1), Tokyo (ap-northeast-1), Seul (ap-northeast-2), Osaka (ap-northeast-3), Mumbai (ap-south-1), Singapore (ap-southeast-1)), Sydney (ap-southeast-2) e Giacarta (ap-southeast-3)
- Europa: Francoforte (eu-central-1), Stoccolma (eu-north-1), Milano (eu-south-1), Irlanda (eu-west-1), Londra (eu-west-2) e Parigi (eu-west-3)
- Medio Oriente: Bahrein (me-south-1) e Emirati Arabi Uniti (me-central-1)
- Nord America: Canada (ca-central-1)
- Sud America: San Paolo (sa-east-1)

- Stati Uniti: Virginia del Nord (us-east-1), Ohio (us-east-2), California del Nord (us-west-1) e Oregon (us-west-2)

## Utilizzo di viste materializzate con AWS Glue

AWS La versione 5.1 e successive di Glue supporta la creazione e la gestione delle viste materializzate di Apache Iceberg nel AWS Glue Data Catalog. Una vista materializzata è una tabella gestita che memorizza il risultato precalcolato di una query SQL in formato Apache Iceberg e si aggiorna in modo incrementale man mano che le tabelle di origine sottostanti cambiano. È possibile utilizzare le viste materializzate per semplificare le pipeline di trasformazione dei dati e accelerare le prestazioni delle query per carichi di lavoro analitici complessi.

Quando crei una vista materializzata utilizzando Spark in AWS Glue, la definizione della vista e i metadati vengono archiviati nel AWS Glue Data Catalog. I risultati precalcolati vengono archiviati come tabelle Apache Iceberg in bucket Amazon S3 Tables o bucket Amazon S3 per uso generico all'interno del tuo account. Il AWS Glue Data Catalog monitora automaticamente le tabelle dei sorgenti e aggiorna le viste materializzate utilizzando un'infrastruttura di elaborazione gestita.

Argomenti


- Come funzionano le viste materializzate con AWS Glue
- Prerequisiti
- Configurazione di Spark per l'utilizzo di viste materializzate
- Creazione di viste materializzate
- Invio di query alle viste materializzate
- Aggiornamento delle viste materializzate
- Gestione delle viste materializzate
- Autorizzazioni per le viste materializzate
- Monitoraggio delle operazioni di visualizzazione materializzata
- Esempio: flusso di lavoro completo
- Considerazioni e limitazioni


### Come funzionano le viste materializzate con AWS Glue

Le viste materializzate si integrano con AWS Glue tramite il supporto Iceberg di Apache Spark nei notebook Glue AWS job e AWS Glue Studio. Quando configuri la sessione Spark per utilizzare

il AWS Glue Data Catalog, puoi creare viste materializzate utilizzando la sintassi SQL standard. L'ottimizzatore Spark può riscrivere automaticamente le query per utilizzare le viste materializzate quando offrono prestazioni migliori, eliminando la necessità di modificare manualmente il codice dell'applicazione.

Il AWS Glue Data Catalog gestisce tutti gli aspetti operativi della manutenzione delle viste materializzate, tra cui:

- Rilevamento delle modifiche nelle tabelle dei sorgenti utilizzando il livello di metadati di Apache Iceberg
- Pianificazione ed esecuzione di operazioni di aggiornamento utilizzando il calcolo Spark gestito
- Determinare se eseguire l'aggiornamento completo o incrementale in base alle modifiche dei dati
- Memorizzazione dei risultati precalcolati in formato Apache Iceberg per l'accesso a più motori

Puoi interrogare le viste materializzate da AWS Glue usando le stesse interfacce SQL Spark che usi per le tabelle normali. I dati precalcolati sono accessibili anche da altri servizi, tra cui Amazon Athena e Amazon Redshift.

Prerequisiti

Per utilizzare le viste materializzate con AWS Glue, è necessario:

- Un account
- AWS Glue versione 5.1 o successiva
- Tabelle di origine in formato Apache Iceberg registrate nel AWS Glue Data Catalog
- AWS Lake Formation autorizzazioni configurate per le tabelle di origine e i database di destinazione
- Un bucket S3 Tables o un bucket S3 generico registrato per l'archiviazione dei dati delle viste materializzate AWS Lake Formation
- Un ruolo IAM con autorizzazioni per accedere a AWS Glue Data Catalog e Amazon S3

Configurazione di Spark per l'utilizzo di viste materializzate

Per creare e gestire le viste materializzate in AWS Glue, configura la tua sessione Spark con le estensioni Iceberg e le impostazioni del catalogo richieste. Il metodo di configurazione varia a seconda che si utilizzino i lavori AWS Glue o i notebook AWS Glue Studio.

Configurazione dei lavori AWS Glue

Durante la creazione o l'aggiornamento di un lavoro AWS Glue, aggiungete i seguenti parametri di configurazione come parametri del lavoro:

Per i bucket S3 Tables

```
job = glue.create_job(
    Name='materialized-view-job',
    Role='arn:aws:iam::111122223333:role/GlueServiceRole',
    Command={
        'Name': 'glueetl',
        'ScriptLocation': 's3://amzn-s3-demo-bucket/scripts/mv-script.py',
        'PythonVersion': '3'
    },
    DefaultArguments={
        '--enable-glue-datacatalog': 'true',
        '--conf':
 'spark.sql.extensions=org.apache.iceberg.spark.extensions.IcebergSparkSessionExtensions
 '
        '--conf spark.sql.catalog.glue_catalog=org.apache.iceberg.spark.SparkCatalog '
                  '--conf spark.sql.catalog.glue_catalog.type=glue '
                  '--conf spark.sql.catalog.glue_catalog.warehouse=s3://amzn-s3-demo-
bucket/warehouse '
                  '--conf spark.sql.catalog.glue_catalog.glue.region=us-east-1 '
                  '--conf spark.sql.catalog.glue_catalog.glue.id=111122223333 '
                  '--conf spark.sql.catalog.glue_catalog.glue.account-id=111122223333
 ',
                  '--conf spark.sql.catalog.glue_catalog.glue.lakeformation-
enabled=true ',
                  '--conf
 spark.sql.catalog.s3t_catalog=org.apache.iceberg.spark.SparkCatalog '
                  '--conf spark.sql.catalog.s3t_catalog.type=glue '
                  '--conf
 spark.sql.catalog.s3t_catalog.glue.id=111122223333:s3tablescatalog/my-table-bucket ',
                  '--conf spark.sql.catalog.s3t_catalog.glue.account-id=111122223333 ',
                  '--conf spark.sql.catalog.s3t_catalog.glue.lakeformation-enabled=true
 ',
                  '--conf spark.sql.catalog.s3t_catalog.warehouse=s3://amzn-s3-demo-
bucket/mv-warehouse '
                  '--conf spark.sql.catalog.s3t_catalog.glue.region=us-east-1 '
                  '--conf spark.sql.defaultCatalog=s3t_catalog '
                  '--conf spark.sql.optimizer.answerQueriesWithMVs.enabled=true '
```

```
                  '--conf spark.sql.materializedViews.metadataCache.enabled=true'
    },
    GlueVersion='5.1'
)
```

Per bucket S3 per uso generico

```
job = glue.create_job(
    Name='materialized-view-job',
    Role='arn:aws:iam::111122223333:role/GlueServiceRole',
    Command={
        'Name': 'glueetl',
        'ScriptLocation': 's3://amzn-s3-demo-bucket/scripts/mv-script.py',
        'PythonVersion': '3'
    },
    DefaultArguments={
        '--enable-glue-datacatalog': 'true',
        '--conf':
 'spark.sql.extensions=org.apache.iceberg.spark.extensions.IcebergSparkSessionExtensions
 '
                  '--conf
 spark.sql.catalog.glue_catalog=org.apache.iceberg.spark.SparkCatalog '
                  '--conf spark.sql.catalog.glue_catalog.type=glue '
                  '--conf spark.sql.catalog.glue_catalog.warehouse=s3://amzn-s3-demo-
bucket/warehouse '
                  '--conf spark.sql.catalog.glue_catalog.glue.region=us-east-1 '
                  '--conf spark.sql.catalog.glue_catalog.glue.id=111122223333 ',
                  '--conf spark.sql.catalog.glue_catalog.glue.account-id=111122223333
 ',
                  '--conf spark.sql.catalog.glue_catalog.glue.lakeformation-
enabled=true ',
                  '--conf spark.sql.defaultCatalog=glue_catalog '
                  '--conf spark.sql.optimizer.answerQueriesWithMVs.enabled=true '
                  '--conf spark.sql.materializedViews.metadataCache.enabled=true'
    },
    GlueVersion='5.1'
)
```

Configurazione dei notebook AWS Glue Studio

Nei taccuini AWS Glue Studio, configura la tua sessione Spark usando il comando magico % %configure all'inizio del notebook:

```
%%configure
{
    "conf": {
        "spark.sql.extensions":
 "org.apache.iceberg.spark.extensions.IcebergSparkSessionExtensions",
        "spark.sql.catalog.glue_catalog": "org.apache.iceberg.spark.SparkCatalog",
        "spark.sql.catalog.glue_catalog.type": "glue",
        "spark.sql.catalog.glue_catalog.warehouse": "s3://amzn-s3-demo-bucket/
warehouse",
        "spark.sql.catalog.glue_catalog.glue.region": "us-east-1",
        "spark.sql.catalog.glue_catalog.glue.id": "111122223333",
        "spark.sql.catalog.glue_catalog.glue.account-id": "111122223333",
        "spark.sql.catalog.glue_catalog.glue.lakeformation-enabled": "true",
        "spark.sql.defaultCatalog": "glue_catalog",
        "spark.sql.optimizer.answerQueriesWithMVs.enabled": "true",
        "spark.sql.materializedViews.metadataCache.enabled": "true"
    }
}
```

Attivazione dell'aggiornamento incrementale

Per abilitare l'ottimizzazione dell'aggiornamento incrementale, aggiungete le seguenti proprietà di configurazione ai parametri del lavoro o alla configurazione del notebook:

```
--conf spark.sql.optimizer.incrementalMVRefresh.enabled=true
--conf spark.sql.optimizer.incrementalMVRefresh.deltaThresholdCheckEnabled=false
```

Parametri di configurazione

I seguenti parametri di configurazione controllano il comportamento delle viste materializzate:

- `spark.sql.extensions`— Abilita le estensioni di sessione Iceberg Spark necessarie per il supporto delle viste materializzate.
- `spark.sql.optimizer.answerQueriesWithMVs.enabled`— Consente la riscrittura automatica delle query per utilizzare le viste materializzate. Imposta su true per attivare questa ottimizzazione.
- `spark.sql.materializedViews.metadataCache.enabled`— Abilita la memorizzazione nella cache dei metadati delle viste materializzate per l'ottimizzazione delle query. Impostato su true per migliorare le prestazioni di riscrittura delle query.

- `spark.sql.optimizer.incrementalMVRefresh.enabled`— Consente l'ottimizzazione incrementale dell'aggiornamento. Impostato su true per elaborare solo i dati modificati durante le operazioni di aggiornamento.
- `spark.sql.optimizer.answerQueriesWithMVs.decimalAggregateCheckEnabled`— Controlla la convalida delle operazioni di aggregazione decimale nella riscrittura delle query. Imposta su false per disabilitare determinati controlli di overflow decimale.

### Creazione di viste materializzate

È possibile creare viste materializzate utilizzando l'istruzione SQL CREATE MATERIALIZED VIEW nei lavori o nei AWS notebook Glue. La definizione della vista specifica la logica di trasformazione come una query SQL che fa riferimento a una o più tabelle di origine.

#### Creazione di una vista materializzata di base nei lavori AWS Glue

L'esempio seguente dimostra la creazione di una vista materializzata in uno script di lavoro AWS Glue, utilizza nomi di tabella completi con una convenzione di denominazione in tre parti nella definizione della vista:

```
from awsglue.context import GlueContext
from pyspark.context import SparkContext

sc = SparkContext()
glueContext = GlueContext(sc)
spark = glueContext.spark_session

# Create materialized view
spark.sql("""
    CREATE MATERIALIZED VIEW customer_orders
    AS
    SELECT
        customer_name,
        COUNT(*) as order_count,
        SUM(amount) as total_amount
    FROM glue_catalog.sales.orders
    GROUP BY customer_name
""")
```

Creazione di una vista materializzata con aggiornamento automatico

Per configurare l'aggiornamento automatico, specificate una pianificazione di aggiornamento durante la creazione della vista, utilizzando nomi di tabella completi con una convenzione di denominazione in tre parti nella definizione della vista:

```
spark.sql("""
    CREATE MATERIALIZED VIEW customer_orders
    SCHEDULE REFRESH EVERY 1 HOUR
    AS
    SELECT
        customer_name,
        COUNT(*) as order_count,
        SUM(amount) as total_amount
    FROM glue_catalog.sales.orders
    GROUP BY customer_name
""")
```

Creazione di una vista materializzata con riferimenti tra cataloghi

Quando le tabelle di origine si trovano in un catalogo diverso rispetto alla vista materializzata, utilizzate nomi di tabella completi con una convenzione di denominazione in tre parti sia nel nome della vista che nella definizione della vista:

```
spark.sql("""
    CREATE MATERIALIZED VIEW s3t_catalog.analytics.customer_summary
    AS
    SELECT
        customer_name,
        COUNT(*) as order_count,
        SUM(amount) as total_amount
    FROM glue_catalog.sales.orders
    GROUP BY customer_name
""")
```

Creazione di viste materializzate nei notebook AWS Glue Studio

Nei notebook AWS Glue Studio, è possibile utilizzare il comando magico %%sql per creare viste materializzate, utilizzando nomi di tabella completi con una convenzione di denominazione in tre parti nella definizione delle viste:

```
%%sql
```

```
CREATE MATERIALIZED VIEW customer_orders
AS
SELECT
    customer_name,
    COUNT(*) as order_count,
    SUM(amount) as total_amount
FROM glue_catalog.sales.orders
GROUP BY customer_name
```

Invio di query alle viste materializzate

Dopo aver creato una vista materializzata, è possibile interrogarla come qualsiasi altra tabella utilizzando le istruzioni SQL SELECT standard nei job o nei notebook AWS Glue.

Interrogazione nei lavori AWS Glue

```
from awsglue.context import GlueContext
from pyspark.context import SparkContext

sc = SparkContext()
glueContext = GlueContext(sc)
spark = glueContext.spark_session

# Query materialized view
result = spark.sql("SELECT * FROM customer_orders")
result.show()
```

Interrogazione nei notebook AWS Glue Studio

```
%%sql
SELECT * FROM customer_orders
```

Riscrittura automatica delle interrogazioni

Quando la riscrittura automatica delle query è abilitata, l'ottimizzatore Spark analizza le query e utilizza automaticamente le viste materializzate quando possono migliorare le prestazioni. Ad esempio, se esegui la seguente query:

```
result = spark.sql("""
    SELECT
        customer_name,
        COUNT(*) as order_count,
```

```
        SUM(amount) as total_amount
    FROM orders
    GROUP BY customer_name
""")
```

L'ottimizzatore Spark riscrive automaticamente questa query per utilizzare la vista materializzata customer_orders invece di elaborare la tabella degli ordini di base, a condizione che la vista materializzata sia aggiornata.

Verifica della riscrittura automatica delle query

Per verificare se una query utilizza la riscrittura automatica delle query, utilizzate il comando EXPLAIN EXTENDED:

```
spark.sql("""
    EXPLAIN EXTENDED
    SELECT customer_name, COUNT(*) as order_count, SUM(amount) as total_amount
    FROM orders
    GROUP BY customer_name
""").show(truncate=False)
```

Nel piano di esecuzione, cercate il nome della vista materializzata nell' BatchScan operazione. Se il piano mostra BatchScan glue_catalog.analytics.customer_orders anziché glue_catalog.sales.orders, la query è stata riscritta automaticamente per utilizzare la BatchScan vista materializzata.

Tieni presente che la riscrittura automatica delle query richiede del tempo prima che la cache dei metadati Spark venga compilata dopo la creazione di una vista materializzata. Questo processo viene in genere completato entro 30 secondi.

Aggiornamento delle viste materializzate

È possibile aggiornare le viste materializzate utilizzando due metodi: aggiornamento completo o aggiornamento incrementale. L'aggiornamento completo ricalcola l'intera vista materializzata da tutti i dati della tabella di base, mentre l'aggiornamento incrementale elabora solo i dati che sono stati modificati dall'ultimo aggiornamento.

Aggiornamento manuale completo nei lavori AWS Glue

Per eseguire un aggiornamento completo di una vista materializzata:

```
spark.sql("REFRESH MATERIALIZED VIEW customer_orders FULL")
```

```
# Verify updated results
result = spark.sql("SELECT * FROM customer_orders")
result.show()
```

Aggiornamento incrementale manuale nei lavori AWS Glue

Per eseguire un aggiornamento incrementale, assicurati che l'aggiornamento incrementale sia abilitato nella configurazione della sessione Spark, quindi esegui:

```
spark.sql("REFRESH MATERIALIZED VIEW customer_orders")

# Verify updated results
result = spark.sql("SELECT * FROM customer_orders")
result.show()
```

Il AWS Glue Data Catalog determina automaticamente se l'aggiornamento incrementale è applicabile in base alla definizione della vista e alla quantità di dati modificati. Se l'aggiornamento incrementale non è possibile, l'operazione torna all'aggiornamento completo.

Aggiornamento nei notebook AWS Glue Studio

Nei notebook, usa il comando magico %%sql:

```
%%sql
REFRESH MATERIALIZED VIEW customer_orders FULL
```

Verifica dell'esecuzione dell'aggiornamento incrementale

Per confermare che l'aggiornamento incrementale sia stato eseguito correttamente, abilita la registrazione di debug nel tuo job Glue: AWS

```
from awsglue.context import GlueContext
from pyspark.context import SparkContext
import logging

sc = SparkContext()
glueContext = GlueContext(sc)
spark = glueContext.spark_session

# Enable debug logging
logger = logging.getLogger('org.apache.spark.sql')
logger.setLevel(logging.DEBUG)
```

```
# Execute refresh
spark.sql("REFRESH MATERIALIZED VIEW customer_orders")
```

Cerca il seguente messaggio nei log dei lavori di AWS Glue:

```
DEBUG RefreshMaterializedViewExec: Executed Incremental Refresh
```

Gestione delle viste materializzate

AWS Glue fornisce comandi SQL per la gestione del ciclo di vita delle viste materializzate nei lavori e nei notebook.

Descrizione di una vista materializzata

Per visualizzare i metadati relativi a una vista materializzata, tra cui la definizione, lo stato di aggiornamento e il timestamp dell'ultimo aggiornamento:

```
spark.sql("DESCRIBE EXTENDED customer_orders").show(truncate=False)
```

Alterazione di una vista materializzata

Per modificare la pianificazione di aggiornamento di una vista materializzata esistente:

```
spark.sql("""
    ALTER MATERIALIZED VIEW customer_orders
    ADD SCHEDULE REFRESH EVERY 2 HOURS
""")
```

Per rimuovere l'aggiornamento automatico:

```
spark.sql("""
    ALTER MATERIALIZED VIEW customer_orders
    DROP SCHEDULE
""")
```

Eliminare una vista materializzata

Per eliminare una vista materializzata:

```
spark.sql("DROP MATERIALIZED VIEW customer_orders")
```

Questo comando rimuove la definizione della vista materializzata dal AWS Glue Data Catalog ed elimina i dati della tabella Iceberg sottostante dal bucket S3.

Elenco delle viste materializzate

Per elencare tutte le viste materializzate in un database:

```
spark.sql("SHOW VIEWS FROM analytics").show()
```

Autorizzazioni per le viste materializzate

Per creare e gestire le viste materializzate, è necessario configurare le autorizzazioni. AWS Lake Formation Il ruolo IAM che crea la vista materializzata (il ruolo definer) richiede autorizzazioni specifiche sulle tabelle di origine e sui database di destinazione.

Autorizzazioni richieste per il ruolo di definitore

Il ruolo che definisce deve avere i seguenti permessi di Lake Formation:

- Nelle tabelle di origine: autorizzazioni SELECT o ALL senza filtri di riga, colonna o cella
- Sul database di destinazione: autorizzazione CREATE_TABLE
- Sul AWS Glue Data Catalog GetTable e sulle autorizzazioni CreateTable API

Quando si crea una vista materializzata, l'ARN del ruolo definitore viene memorizzato nella definizione della vista. Il AWS Glue Data Catalog assume questo ruolo durante l'esecuzione di operazioni di aggiornamento automatico. Se il ruolo di definitore perde l'accesso alle tabelle di origine, le operazioni di aggiornamento falliranno fino al ripristino delle autorizzazioni.

Autorizzazioni IAM per i lavori AWS Glue

Il ruolo IAM del tuo lavoro AWS Glue richiede le seguenti autorizzazioni:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "glue:GetCatalog",
                "glue:GetCatalogs",
                "glue:GetTable",
```

```
                "glue:GetTables",
                "glue:CreateTable",
                "glue:UpdateTable",
                "glue:DeleteTable",
                "glue:GetDatabase",
                "glue:GetDatabases",
                "cloudwatch:PutMetricData"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:PutObject",
                "s3:DeleteObject"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": [
                "arn:aws:logs:*:*:*:/aws-glue/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
```

```
                "lakeformation:GetDataAccess"
            ],
            "Resource": "*"
        }
    ]
}
```

Il ruolo che usi per l'aggiornamento automatico di Materialized View deve avere l'PassRole autorizzazione iam: sul ruolo.

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": [
        "arn:aws:iam::111122223333:role/materialized-view-role-name"
      ]
    }
  ]
}
```

Affinché Glue aggiorni automaticamente la vista materializzata, il ruolo deve inoltre avere la seguente politica di attendibilità che consenta al servizio di assumere il ruolo.

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": [
        "arn:aws:iam::111122223333:role/materialized-view-role-name"
      ]
    }
```

```
  ]
}
```

Se la vista materializzata è archiviata in S3 Tables Bucket, devi anche aggiungere la seguente autorizzazione al ruolo.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3tables:PutTableMaintenanceConfiguration"
      ],
      "Resource": "arn:aws:s3tables:*:123456789012:*"
    }
  ]
}
```

Concessione dell'accesso alle viste materializzate

Per concedere ad altri utenti l'accesso all'interrogazione di una vista materializzata, utilizzare questa opzione AWS Lake Formation per concedere l'autorizzazione SELECT sulla tabella delle viste materializzate. Gli utenti possono interrogare la vista materializzata senza richiedere l'accesso diretto alle tabelle di origine sottostanti.

Per informazioni dettagliate sulla configurazione delle autorizzazioni di Lake Formation, consulta Concessione e revoca delle autorizzazioni sulle risorse di Data Catalog nella Developer Guide. AWS Lake Formation

Monitoraggio delle operazioni di visualizzazione materializzata

Il AWS Glue Data Catalog pubblica metriche e log per le operazioni di aggiornamento delle viste materializzate su Amazon. CloudWatch Puoi monitorare lo stato dell'aggiornamento, la durata e il volume di dati elaborati tramite metriche. CloudWatch

Visualizzazione dei registri dei lavori

Per visualizzare i log dei lavori AWS Glue che creano o aggiornano viste materializzate:

1. Apri la console AWS Glue.
2. Scegli Jobs dal pannello di navigazione.
3. Seleziona il tuo lavoro e scegli Esegue.
4. Seleziona una corsa specifica e scegli Registri per visualizzare i CloudWatch registri.

### Configurazione degli allarmi

Per ricevere notifiche quando le operazioni di aggiornamento falliscono o superano la durata prevista, crea CloudWatch allarmi sulle metriche delle viste materializzate. Puoi anche configurare EventBridge le regole di Amazon per attivare risposte automatiche agli eventi di aggiornamento.

### Esempio: flusso di lavoro completo

L'esempio seguente dimostra un flusso di lavoro completo per la creazione e l'utilizzo di una vista materializzata in AWS Glue.

### Esempio di script AWS Glue job

```
import sys
from awsglue.transforms import *
from awsglue.utils import getResolvedOptions
from pyspark.context import SparkContext
from awsglue.context import GlueContext
from awsglue.job import Job

args = getResolvedOptions(sys.argv, ['JOB_NAME'])
sc = SparkContext()
glueContext = GlueContext(sc)
spark = glueContext.spark_session
job = Job(glueContext)
job.init(args['JOB_NAME'], args)

# Create database and base table
spark.sql("CREATE DATABASE IF NOT EXISTS sales")
spark.sql("USE sales")

spark.sql("""
    CREATE TABLE IF NOT EXISTS orders (
        id INT,
        customer_name STRING,
        amount DECIMAL(10,2),
```

```
        order_date DATE
    )
""")

# Insert sample data
spark.sql("""
    INSERT INTO orders VALUES
        (1, 'John Doe', 150.00, DATE('2024-01-15')),
        (2, 'Jane Smith', 200.50, DATE('2024-01-16')),
        (3, 'Bob Johnson', 75.25, DATE('2024-01-17'))
""")

# Create materialized view
spark.sql("""
    CREATE MATERIALIZED VIEW customer_summary
    AS
    SELECT
        customer_name,
        COUNT(*) as order_count,
        SUM(amount) as total_amount
    FROM glue_catalog.sales.orders
    GROUP BY customer_name
""")

# Query the materialized view
print("Initial materialized view data:")
spark.sql("SELECT * FROM customer_summary").show()

# Insert additional data
spark.sql("""
    INSERT INTO orders VALUES
        (4, 'Jane Smith', 350.00, DATE('2024-01-18')),
        (5, 'Bob Johnson', 100.25, DATE('2024-01-19'))
""")

# Refresh the materialized view
spark.sql("REFRESH MATERIALIZED VIEW customer_summary FULL")

# Query updated results
print("Updated materialized view data:")
spark.sql("SELECT * FROM customer_summary").show()

job.commit()
```

## Esempio di notebook AWS Glue Studio

```
%%configure
{
    "conf": {
        "spark.sql.extensions":
 "org.apache.iceberg.spark.extensions.IcebergSparkSessionExtensions",
        "spark.sql.catalog.glue_catalog": "org.apache.iceberg.spark.SparkCatalog",
        "spark.sql.catalog.glue_catalog.type": "glue",
        "spark.sql.catalog.glue_catalog.warehouse": "s3://amzn-s3-demo-bucket/
warehouse",
        "spark.sql.catalog.glue_catalog.glue.region": "us-east-1",
        "spark.sql.catalog.glue_catalog.glue.id": "111122223333",
        "spark.sql.catalog.glue_catalog.glue.account-id": "111122223333",
        "spark.sql.catalog.glue_catalog.glue.lakeformation-enabled": "true",
        "spark.sql.defaultCatalog": "glue_catalog",
        "spark.sql.optimizer.answerQueriesWithMVs.enabled": "true",
        "spark.sql.materializedViews.metadataCache.enabled": "true"
    }
}
```

```
%%sql
CREATE DATABASE IF NOT EXISTS sales
```

```
%%sql
USE sales
```

```
%%sql
CREATE TABLE IF NOT EXISTS orders (
    id INT,
    customer_name STRING,
    amount DECIMAL(10,2),
    order_date DATE
)
```

```
%%sql
INSERT INTO orders VALUES
    (1, 'John Doe', 150.00, DATE('2024-01-15')),
    (2, 'Jane Smith', 200.50, DATE('2024-01-16')),
    (3, 'Bob Johnson', 75.25, DATE('2024-01-17'))
```

```
%%sql
CREATE MATERIALIZED VIEW customer_summary
AS
SELECT
    customer_name,
    COUNT(*) as order_count,
    SUM(amount) as total_amount
FROM glue_catalog.sales.orders
GROUP BY customer_name
```

```
%%sql
SELECT * FROM customer_summary
```

```
%%sql
INSERT INTO orders VALUES
    (4, 'Jane Smith', 350.00, DATE('2024-01-18')),
    (5, 'Bob Johnson', 100.25, DATE('2024-01-19'))
```

```
%%sql
REFRESH MATERIALIZED VIEW customer_summary FULL
```

```
%%sql
SELECT * FROM customer_summary
```

Considerazioni e limitazioni

Considerate quanto segue quando utilizzate viste materializzate con AWS Glue:

- Le viste materializzate richiedono AWS Glue versione 5.1 o successiva.
- Le tabelle di origine devono essere tabelle Apache Iceberg registrate nel AWS Glue Data Catalog. Le tabelle Apache Hive, Apache Hudi e Linux Foundation Delta Lake non sono supportate al momento del lancio.
- Le tabelle di origine devono risiedere nella stessa regione e nello stesso account della vista materializzata.
- Tutte le tabelle di origine devono essere governate da. AWS Lake Formation Le autorizzazioni solo IAM e l'accesso ibrido non sono supportati.
- Le viste materializzate non possono fare riferimento alle viste del AWS Glue Data Catalog, alle viste multidialettali o ad altre viste materializzate come tabelle di origine.

- Il ruolo View Definer deve disporre dell'accesso completo in lettura (autorizzazione SELECT o ALL) su tutte le tabelle di origine senza applicare filtri di riga, colonna o cella.
- Le viste materializzate alla fine sono coerenti con le tabelle di origine. Durante la finestra di aggiornamento, le query possono restituire dati non aggiornati. Esegui l'aggiornamento manuale per una coerenza immediata.
- L'intervallo minimo di aggiornamento automatico è di un'ora.
- L'aggiornamento incrementale supporta un sottoinsieme limitato di operazioni SQL. La definizione della vista deve essere un singolo blocco SELECT-FROM-WHERE-GROUP BY-HAVING e non può contenere operazioni di set, sottoquery, la parola chiave DISTINCT in SELECT o funzioni aggregate, funzioni di finestra o join diversi da INNER JOIN.
- L'aggiornamento incrementale non supporta funzioni definite dall'utente o determinate funzioni integrate. È supportato solo un sottoinsieme delle funzioni integrate di Spark SQL.
- La riscrittura automatica delle query considera solo le viste materializzate le cui definizioni appartengono a un sottoinsieme SQL limitato, simile alle restrizioni di aggiornamento incrementale.
- Gli identificatori contenenti caratteri speciali diversi dai caratteri alfanumerici e dai caratteri di sottolineatura non sono supportati nelle query CREATE MATERIALIZED VIEW. Questo vale per tutti i tipi di identificatori, inclusi catalog/namespace/table nomi, nomi di campi di colonne e strutture e alias. CTEs
- Le colonne delle viste materializzate che iniziano con il prefisso __ivm sono riservate all'uso del sistema. Amazon si riserva il diritto di modificare o rimuovere queste colonne nelle versioni future.
- Le clausole SORT BY, LIMIT, OFFSET, CLUSTER BY e ORDER BY non sono supportate nelle definizioni delle viste materializzate.
- Le tabelle di origine tra regioni e tra account non sono supportate.
- Le tabelle a cui si fa riferimento nella query di visualizzazione devono utilizzare una convenzione di denominazione in tre parti (ad esempio, glue_catalog.my_db.my_table) perché l'aggiornamento automatico non utilizza le impostazioni predefinite del catalogo e del database.
- Le operazioni di aggiornamento completo sovrascrivono l'intera tabella e rendono non disponibili le istantanee precedenti.
- Le funzioni non deterministiche come rand () o current_timestamp () non sono supportate nelle definizioni delle viste materializzate.

# AWS Glue tipi di lavoratori

## Panoramica di

AWS Glue offre diversi tipi di lavoratori per soddisfare diversi requisiti di carico di lavoro, da piccoli lavori in streaming a attività di elaborazione dati su larga scala e che richiedono molta memoria. Questa sezione fornisce informazioni complete su tutti i tipi di worker disponibili, le relative specifiche e i consigli di utilizzo.

Categorie di tipi di worker

AWS Glue offre due categorie principali di tipi di lavoratori:

- Tipi di worker G: elaboratori generici ottimizzati per carichi di lavoro ETL standard
- Tipi di worker R: worker ottimizzati per la memoria progettati per applicazioni Spark che richiedono molta memoria

Unità di elaborazione dati (DPUs)

Le risorse disponibili per i AWS Glue lavoratori sono misurate in DPUs. Una DPU è una misura relativa della potenza di elaborazione costituita da 4 V CPUs di capacità di elaborazione e 16 GB di memoria.

Ottimizzato per la memoria DPUs (M-DPUs): i lavoratori di tipo R utilizzano M-DPUs, che fornisce il doppio dell'allocazione di memoria per una determinata dimensione rispetto allo standard. DPUs Ciò significa che mentre una DPU standard fornisce 16 GB di memoria, una M-DPU di tipo R fornisce 32 GB di memoria ottimizzata per le applicazioni Spark a uso intensivo di memoria.

## Tipi di worker disponibili

G.1X

- DPU: 1 DPU (4 vCPUs, 16 GB di memoria)
- Memoria: disco da 94 GB (circa 44 GB liberi)
- Caso d'uso: trasformazioni, unioni e query di dati: scalabile e conveniente per la maggior parte dei processi

G.2X

- DPU: 2 DPU (8 vCPUs, 32 GB di memoria)
- Memoria: disco da 138 GB (circa 78 GB liberi)
- Caso d'uso: trasformazioni, unioni e query di dati: scalabile e conveniente per la maggior parte dei processi

G.4X

- DPU: 4 DPU (16 vCPUs, 64 GB di memoria)
- Memoria: disco da 256 GB (circa 230 GB liberi)
- Caso d'uso: trasformazioni, aggregazioni, join e query con i maggiori requisiti

G.8X

- DPU: 8 DPU (32 vCPUs, 128 GB di memoria)
- Memoria: disco da 512 GB (circa 485 GB liberi)
- Caso d'uso: trasformazioni, aggregazioni, join e query con i maggiori requisiti

G.12X

- DPU: 12 DPU (48 vCPUs, 192 GB di memoria)
- Memoria: disco da 768 GB (circa 741 GB liberi)
- Caso d'uso: carichi di lavoro molto grandi e che richiedono molte risorse con una notevole capacità di elaborazione

G.16X

- DPU: 16 DPU (64 vCPUs, 256 GB di memoria)
- Memoria: disco da 1024 GB (circa 996 GB liberi)
- Caso d'uso: carichi di lavoro più grandi in assoluto e che richiedono la maggiore capacità di elaborazione possibile

R.1X - Ottimizzato per la memoria*

- DPU: 1 M-DPU (4 v, 32 GB di memoria) CPUs
- Caso d'uso: carichi di lavoro a uso intensivo di memoria con errori frequenti o requisiti di rapporto elevato out-of-memory memory-to-CPU

R.2X - Ottimizzato per la memoria*

- DPU: 2 M-DPU (8 v, 64 GB di memoria) CPUs
- Caso d'uso: carichi di lavoro a uso intensivo di memoria con errori frequenti o requisiti di rapporto elevato out-of-memory memory-to-CPU

R.4X - Ottimizzato per la memoria*

- DPU: 4 M-DPU (16 v, 128 GB di memoria) CPUs
- Caso d'uso: grandi carichi di lavoro che richiedono molta memoria con errori frequenti o requisiti di rapporto elevato out-of-memory memory-to-CPU

R.8X - Ottimizzato per la memoria*

- DPU: 8 M-DPU (32 v, 256 GB di memoria) CPUs
- Caso d'uso: carichi di lavoro molto grandi che richiedono molta memoria con errori frequenti o requisiti di rapporto elevato out-of-memory memory-to-CPU

*È possibile riscontrare una maggiore latenza di avvio con questi worker. Per risolvere il problema, provare a eseguire queste operazioni:

- Attendere alcuni minuti, quindi inviare di nuovo il processo.
- Inviare un nuovo processo con un numero ridotto di worker.
- Inviare un nuovo processo utilizzando un tipo o una dimensione di worker diversi.

## Tabella delle specifiche del tipo di worker

Specifiche del tipo di worker

| Tipo di worker | DPU per nodo | VPCU | Memoria (GB) | Disco (GB) | Spazio libero su disco approssimativo (GB) | Esecutori Spark per nodo |
|---|---|---|---|---|---|---|
| G.1X | 1 | 4 | 16 | 94 | 44 | 1 |
| G.2X | 2 | 8 | 32 | 138 | 78 | 1 |
| G.4X | 4 | 16 | 64 | 256 | 230 | 1 |
| G.8X | 8 | 32 | 128 | 512 | 485 | 1 |
| G.12X | 12 | 48 | 192 | 768 | 741 | 1 |
| G.16X | 16 | 64 | 256 | 1.024 | 996 | 1 |
| R.1 X | 1 | 4 | 32 | 94 | 44 | 1 |
| R.2X | 2 | 8 | 64 | 138 | 78 | 1 |
| R. 4X | 4 | 16 | 128 | 256 | 230 | 1 |
| R.8X | 8 | 32 | 256 | 512 | 485 | 1 |

Nota: i tipi di worker R hanno configurazioni ottimizzate per la memoria con specifiche ottimizzate per carichi di lavoro che richiedono molta memoria.

## Considerazioni importanti

### Latenza di avvio

> **Important**
>
> I tipi di worker G.12X e G.16X, così come tutti i tipi di worker R (da R.1X a R.8X), possono riscontrare una latenza di avvio più elevata. Per risolvere il problema, provare a eseguire queste operazioni:
>
> - Attendere alcuni minuti, quindi inviare di nuovo il processo.
> - Inviare un nuovo processo con un numero ridotto di worker.
> - Inviare un nuovo processo utilizzando un tipo e una dimensione di worker diversi.

## Scegliere il tipo di worker giusto

### Per carichi di lavoro ETL standard

- G.1X o G.2X: la soluzione più conveniente per le trasformazioni, i join e le query tipiche dei dati
- G.4X o G.8X: per carichi di lavoro più impegnativi con set di dati più grandi

### Per carichi di lavoro su larga scala

- G.12X: set di dati molto grandi che richiedono risorse di elaborazione significative
- G.16X: capacità di elaborazione massima per i carichi di lavoro più impegnativi

### Per carichi di lavoro a uso intensivo di memoria

- R.1X o R.2X: processi di piccole e medie dimensioni con uso intensivo di memoria
- R.4X o R.8X: carichi di lavoro di grandi dimensioni a uso intensivo di memoria con frequenti errori OOM

## Considerazioni per l'ottimizzazione dei costi

- Worker G standard: offrono il giusto rapporto tra risorse di calcolo, memoria e rete e possono essere utilizzati per una varietà di carichi di lavoro diversi a costi inferiori

- Worker R: specializzati per attività a uso intensivo di memoria con prestazioni elevate per carichi di lavoro che elaborano set di dati di grandi dimensioni in memoria

### Best practice

Linee guida per la selezione dei worker

1. Avvio con worker standard (G.1X, G.2X) per la maggior parte dei carichi di lavoro
2. Usa R worker in caso di out-of-memory errori o carichi di lavoro frequenti con operazioni che richiedono molta memoria come caching, shuffling e aggregazione
3. Considerare G.12X/G.16X per carichi di lavoro ad alta intensità di calcolo che richiedono il massimo delle risorse
4. Considerare i vincoli di capacità per l'utilizzo di nuovi tipi di worker in flussi di lavoro urgenti

Ottimizzazione delle prestazioni

- CloudWatch Monitora le metriche per comprendere l'utilizzo delle risorse
- Utilizzare il numero di worker appropriato in base alla dimensione e alla complessità dei dati
- Considerare le strategie di partizionamento dei dati per ottimizzare l'efficienza dei worker

## Aggiunta di processi di streaming ETL in AWS Glue

È possibile creare operazioni in streaming di estrazione, trasformazione e caricamento (ETL) che vengono eseguite continuamente, consumano dati da origini di streaming come Amazon Kinesis Data Streams, Apache Kafka e Amazon Streaming gestito per Apache Kafka (Amazon MSK). I processi puliscono e trasformano i dati, quindi caricano i risultati in data lake Amazon S3 o datastore JDBC.

Inoltre, è possibile produrre dati per i flussi di dati Amazon Kinesis. Questa funzionalità è disponibile solo per la scrittura di AWS Glue script. Per ulteriori informazioni, consulta the section called "Connessioni Kinesis".

Per impostazione predefinita, AWS Glue elabora e scrive i dati in finestre di 100 secondi. Ciò consente di elaborare i dati in modo efficiente e di eseguire aggregazioni su dati che arrivano più tardi del previsto. È possibile modificare questa dimensione della finestra per aumentare la tempestività o la precisione dell'aggregazione. AWS Glue I processi di streaming utilizzano i checkpoint anziché i segnalibri di lavoro per tenere traccia dei dati letti.

> **Note**
>
> AWS Glue fatture orarie per lo streaming delle operazioni ETL mentre sono in esecuzione.

Questo video illustra le problematiche relative ai costi dello streaming ETL e le funzionalità di riduzione dei costi di. AWS Glue

La creazione di un processo di streaming ETL prevede i seguenti passaggi:

1. Per un'origine di streaming Apache Kafka, creare una connessione AWS Glue alla sorgente Kafka o al cluster Amazon MSK.
2. Creare manualmente un catalogo dati per l'origine di streaming.
3. Creare un processo ETL per l'origine dati di streaming. Definire le proprietà del processo specifiche dello streaming e fornire uno script personalizzato o, facoltativamente, modificare lo script generato.

Per ulteriori informazioni, consulta Streaming di ETL in AWS Glue.

Quando si crea un processo ETL di streaming per Amazon Kinesis Data Streams, non è necessario creare una connessione AWS Glue. Tuttavia, se è presente una connessione collegata al flusso di lavoro ETL AWS Glue con Kinesis Data Streams come origine, quindi è necessario un endpoint cloud privato virtuale (VPC) a Kinesis. Per ulteriori informazioni, consultare Creazione di un endpoint di interfaccia nella Guida per l'utente di Amazon VPC. Quando si specifica un flusso Amazon Kinesis Data Streams in un altro account, è necessario impostare i ruoli e le politiche per consentire l'accesso multi-account. Per ulteriori informazioni, consulta Esempio: lettura da un flusso Kinesis in un account diverso.

I processi ETL in streaming di AWS Glue possono rilevare automaticamente i dati compressi, decomprimere in modo trasparente i dati in streaming, eseguire le consuete trasformazioni sulla sorgente di ingresso e caricare nell'archivio di output.

AWS Glue supporta la decompressione automatica per i seguenti tipi di compressione, dato il formato di input:

| Tipo di compressione | File Avro | Dato Avro | JSON | CSV | Grok |
|---|---|---|---|---|---|
| BZIP2 | Sì | Sì | Sì | Sì | Sì |
| GZIP | No | Sì | Sì | Sì | Sì |
| SNAPPY | Sì (Snappy raw) | Sì (Snappy framed) | Sì (Snappy framed) | Sì (Snappy framed) | Sì (Snappy framed) |
| XZ | Sì | Sì | Sì | Sì | Sì |
| ZSTD | Sì | No | No | No | No |
| DEFLATE | Sì | Sì | Sì | Sì | Sì |

Argomenti


- Creazione di una connessione AWS Glue per un flusso di dati Apache Kafka
- Creazione di un catalogo dati per un'origine di streaming
- Note e restrizioni per le origini di streaming Avro
- Applicazione di pattern Grok alle origini di streaming
- Definizione delle proprietà di processo per un processo di streaming ETL
- Streaming di note e restrizioni ETL


## Creazione di una connessione AWS Glue per un flusso di dati Apache Kafka

Per leggere da un flusso Apache Kafka, è necessario creare una connessione AWS Glue.

Come creare una connessione AWS Glue per un'origine Kafka (Console)

1. Apri la console all'indirizzo. AWS Glue https://console.aws.amazon.com/glue/
2. Nel riquadro di navigazione, in Data catalog (Catalogo dati), seleziona Connections (Connessioni).
3. Scegliere Aggiungi connessione e, nella pagina Imposta proprietà della connessione, immettere un nome per la connessione.

> **Note**
>
> Per ulteriori informazioni sulla specifica delle proprietà della connessione, consulta Proprietà della connessione di AWS Glue.

4. Per Tipo di connessione, scegli Kafka.
5. Per i server bootstrap Kafka URLs, inserisci l'host e il numero di porta per i broker di bootstrap per il tuo cluster Amazon MSK o il cluster Apache Kafka. Utilizza solo endpoint Transport Layer Security (TLS) per stabilire la connessione iniziale al cluster Kafka. Gli endpoint in testo normale non sono supportati.

   Di seguito è riportato un elenco di esempio di coppie di nomi di host e numeri di porta per un cluster Amazon MSK.

   ```
   myserver1.kafka.us-east-1.amazonaws.com:9094,myserver2.kafka.us-
   east-1.amazonaws.com:9094,
   myserver3.kafka.us-east-1.amazonaws.com:9094
   ```

   Per ulteriori informazioni su come ottenere le informazioni del broker bootstrap, consulta Ottenere i broker bootstrap per un cluster Amazon MSK in Amazon Managed Streaming for Apache Kafka: Guida per gli sviluppatori.
6. Se si desidera una connessione sicura all'origine dati Kafka, seleziona Require SSL connection (Connessione SSL necessaria), e per Kafka private CA certificate location (Posizione del certificato emesso da una CA Kafka privata), inserisci un percorso Amazon S3 valido per un certificato SSL personalizzato.

   Per una connessione SSL a Kafka autogestito, il certificato personalizzato è obbligatorio. P Amazon MSK è facoltativo.

   Per ulteriori informazioni su come specificare un certificato personalizzato per Kafka, consulta the section called "Proprietà della connessione SSL".
7. Usa AWS Glue Studio o la AWS CLI per specificare un metodo di autenticazione del client Kafka. Per accedere, AWS Glue Studio seleziona AWS Gluedal menu ETL nel riquadro di navigazione a sinistra.

   Per ulteriori informazioni sui metodi di autenticazione client Kafka, consulta AWS Glue Proprietà di connessione Kafka per l'autenticazione client.

8. Opzionalmente, inserisci una descrizione, quindi scegli Next (Successivo).
9. Per un cluster Amazon MSK, specifica il cloud privato virtuale (VPC), la sottorete e il gruppo di sicurezza. Le informazioni VPC sono opzionali per Kafka autogestito.
10. Scegli Next (Successivo) per esaminare tutte le proprietà della connessione, quindi scegli Finish (Termina).

Per ulteriori informazioni sulle connessioni AWS Glue, consulta Connessione ai dati.

AWS Glue Proprietà di connessione Kafka per l'autenticazione client

Autenticazione SASL/GSSAPI (Kerberos)

La scelta di questo metodo di autenticazione consentirà di specificare le proprietà Kerberos.

Keytab Kerberos

Scegliere la posizione del file keytab. Un keytab memorizza le chiavi a lungo termine per uno o più principali. Per ulteriori informazioni, consulta la Documentazione di MIT Kerberos: keytab.

File Kerberos krb5.conf

Scegliere il file krb5.conf. Contiene l'area di autenticazione predefinita (una rete logica, simile a un dominio, che definisce un gruppo di sistemi sotto lo stesso KDC) e la posizione del server KDC. Per ulteriori informazioni, consulta la Documentazione di MIT Kerberos: krb5.conf.

Principale Kerberos e nome del servizio Kerberos

Immettere il nome del principale e il nome del servizio Kerberos. Per ulteriori informazioni, consulta Documentazione MIT Kerberos: principale Kerberos.

Autenticazione SASL/SCRAM-SHA-512

Scegliere questo metodo di autenticazione consentirà di specificare le credenziali di autenticazione.

AWS Secrets Manager

Cercare il token nella casella Cerca digitando il nome o l'ARN.

Nome utente e password del provider direttamente

Cercare il token nella casella Cerca digitando il nome o l'ARN.

Autenticazione client SSL

Scegliere questo metodo di autenticazione consente di selezionare la posizione del keystore client Kafka navigando su Amazon S3. Facoltativamente, puoi inserire la password del keystore del client Kafka e la password della chiave del client Kafka.

Autenticazione IAM

Questo metodo di autenticazione non richiede specifiche aggiuntive ed è applicabile solo quando la sorgente di streaming è MSK Kafka.

Autenticazione SASL/PLAIN

Scegliere questo metodo di autenticazione consente di specificare le credenziali di autenticazione.

## Creazione di un catalogo dati per un'origine di streaming

Una tabella del catalogo dati che specifica le proprietà del flusso dei dati di origine, incluso lo schema dei dati, può essere creata manualmente per una sorgente di streaming. Questa tabella viene utilizzata come origine dati per il processo di streaming ETL.

Se non si conosce lo schema dei dati nel flusso dei dati di origine, è possibile creare la tabella senza uno schema. Quindi, quando si crea il processo ETL di streaming, è possibile attivare la AWS Gluefunzione di rilevamento dello schema. AWS Glue determina lo schema dai dati di streaming.

Usa la AWS Glueconsole, il AWS Command Line Interface (AWS CLI) o l'API per creare la AWS Glue tabella. Per informazioni sulla creazione manuale di una tabella con la console AWS Glue, vedere the section called “Creazione di tabelle”.

> **Note**
>
> Non puoi usare la AWS Lake Formation console per creare la tabella; devi usare la AWS Glue console.

Considera inoltre le seguenti informazioni per le origini di streaming in formato Avro o per i dati di log a cui è possibile applicare i pattern Grok.

- the section called “Note e restrizioni per le origini di streaming Avro”
- the section called “Applicazione di pattern Grok alle origini di streaming”

Argomenti


- Origine dati Kinesis
- Origine dati Kafka
- Origine della tabella di AWS Glue Schema Registry


## Origine dati Kinesis

Durante la creazione della tabella, impostare le seguenti proprietà di streaming ETL (console).

Tipo di origine

Kinesis

Per una fonte Kinesis nello stesso account:

Region

La AWS regione in cui risiede il servizio Amazon Kinesis Data Streams. Il nome della regione e del flusso Kinesis sono tradotti insieme in un flusso ARN.

Esempio: https://kinesis.us-east-1.amazonaws.com

Nome del flusso Kinesis

Nome del flusso come descritto in Creazione di un flusso nella Guida per gli sviluppatori Amazon Kinesis Data Streams.

Per un'origine Kinesis in un altro account, fai riferimento a questo esempio per configurare i ruoli e i criteri per consentire l'accesso a più account. Configura queste impostazioni:

Flusso ARN

L'ARN del flusso dei dati Kinesis con il quale il consumatore è registrato. Per ulteriori informazioni, consulta Amazon Resource Names (ARNs) e AWS Service Namespaces nel. Riferimenti generali di AWS

ARN del ruolo assunto

L'Amazon Resource Name (ARN) del ruolo assegnato al ruolo da assumere.

Nome sessione (facoltativo)

Un identificatore della sessione del ruolo assunto.

Utilizza il nome della sessione del ruolo per identificare in modo univoco una sessione quando lo stesso ruolo viene assunto da diverse entità principali o per motivi diversi. In scenari multi-account, l'account proprietario del ruolo può vedere il nome della sessione del ruolo e può registrarlo. Il nome della sessione del ruolo viene utilizzato anche nell'ARN dell'entità ruolo assunto. Ciò significa che le successive richieste API tra più account che utilizzano le credenziali di sicurezza temporanee esporranno il nome della sessione del ruolo all'account esterno nei relativi log. AWS CloudTrail

Per impostare le proprietà di streaming ETL per Amazon Kinesis Data Streams (API AWS Glue o AWS CLI)

- Per impostare le proprietà di streaming ETL per un'origine Kinesis nello stesso account, specifica i parametri `streamName` e `endpointUrl` nella struttura `StorageDescriptor` dell'operazione API `CreateTable` o del comando CLI `create_table`.

```
"StorageDescriptor": {
 "Parameters": {
  "typeOfData": "kinesis",
  "streamName": "sample-stream",
  "endpointUrl": "https://kinesis.us-east-1.amazonaws.com"
 }
 ...
}
```

  In alternativa, specifica il `streamARN`.

  Example

```
"StorageDescriptor": {
 "Parameters": {
  "typeOfData": "kinesis",
  "streamARN": "arn:aws:kinesis:us-east-1:123456789:stream/sample-stream"
 }
 ...
}
```

- Per impostare le proprietà di streaming ETL per un'origine Kinesis nello stesso account, specifica i parametri `streamARN`, `awsSTSRoleARN` e `awsSTSSessionName` (facoltativo) nella struttura `StorageDescriptor` dell'operazione API `CreateTable` o del comando CLI `create_table`.

```
"StorageDescriptor": {
 "Parameters": {
  "typeOfData": "kinesis",
  "streamARN": "arn:aws:kinesis:us-east-1:123456789:stream/sample-stream",
  "awsSTSRoleARN": "arn:aws:iam::123456789:role/sample-assume-role-arn",
  "awsSTSSessionName": "optional-session"
 }
 ...
}
```

Origine dati Kafka

Durante la creazione della tabella, impostare le seguenti proprietà di streaming ETL (console).

Tipo di origine

Kafka

Per una fonte Kafka:

Nome dell'argomento

Nome argomento specificato in Kafka.

Connessione

Connessione AWS Glue che fa riferimento a un'origine Kafka, come descritto in the section called "Creazione di una connessione per un flusso di dati Kafka".

Origine della tabella di AWS Glue Schema Registry

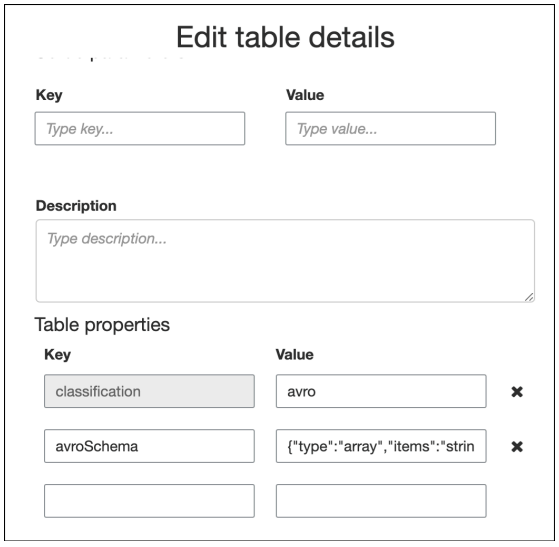
Per utilizzare AWS Glue Schema Registry per i processi di streaming, segui le istruzioni all'indirizzo Caso d'uso: AWS Glue Data Catalog su come creare o aggiornare una tabella Schema Registry.

Attualmente, lo streaming AWS Glue supporta solo il formato Glue Schema Registry Avro con inferenza dello schema impostata su `false`.

## Note e restrizioni per le origini di streaming Avro

Le seguenti note e restrizioni si applicano alle origini di streaming nel formato Avro:

- Quando il rilevamento dello schema è attivato, lo schema Avro deve essere incluso nel payload. Quando disattivato, il payload deve contenere solo dati.
- Alcuni tipi di dati Avro non sono supportati nei frame dinamici. Non è possibile specificare questi tipi di dati quando si definisce lo schema con la pagina Define a schema (Definire uno schema) nella procedura guidata per la creazione della tabella nella console AWS Glue. Durante il rilevamento dello schema, i tipi non supportati nello schema Avro vengono convertiti in tipi supportati come segue:
  - `EnumType => StringType`
  - `FixedType => BinaryType`
  - `UnionType => StructType`
- Se si definisce lo schema della tabella utilizzando la pagina Define a schema (Definire uno schema) nella console, il tipo di elemento root implicito per lo schema è `record`. Se si desidera un tipo di elemento root diverso da `record`, ad esempio `array` o `map`, non è possibile specificare lo schema utilizzando la pagina Define a schema (Definire uno schema). È invece necessario saltare quella pagina e specificare lo schema come proprietà di tabella o all'interno dello script ETL.
  - Per specificare lo schema nelle proprietà della tabella, completa la procedura guidata per la creazione della tabella, modifica i dettagli della tabella e aggiungi una nuova coppia chiave-valore in Table properties (Proprietà della tabella). Utilizza la chiave `avroSchema`, e inserisci un oggetto JSON dello schema per il valore, come mostrato nello screenshot seguente.




  - Per specificare lo schema nello script ETL, modifica l'istruzione di assegnazione `datasource0` e aggiungi la chiave `avroSchema` all'argomento `additional_options`, come mostrato nei seguenti esempi Python e Scala.

Python

```
SCHEMA_STRING = '{"type":"array","items":"string"}'
datasource0 = glueContext.create_data_frame.from_catalog(database =
 "database", table_name = "table_name", transformation_ctx = "datasource0",
 additional_options = {"startingPosition": "TRIM_HORIZON", "inferSchema":
 "false", "avroSchema": SCHEMA_STRING})
```

Scala

```
val SCHEMA_STRING = """{"type":"array","items":"string"}"""
val datasource0 = glueContext.getCatalogSource(database = "database", tableName
 = "table_name", redshiftTmpDir = "", transformationContext = "datasource0",
 additionalOptions = JsonOptions(s"""{"startingPosition": "TRIM_HORIZON",
 "inferSchema": "false", "avroSchema":"$SCHEMA_STRING"}""")).getDataFrame()
```

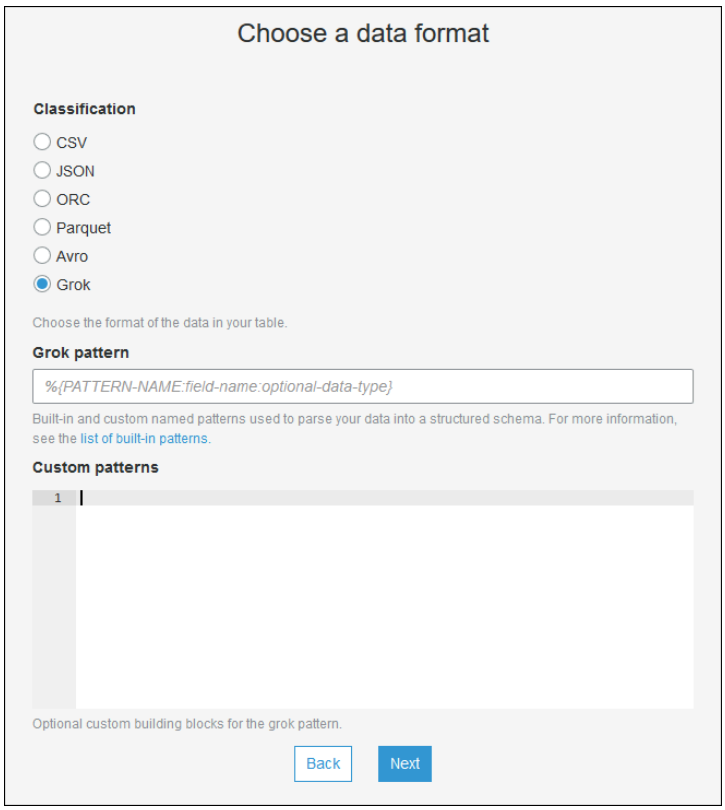
## Applicazione di pattern Grok alle origini di streaming

È possibile creare un processo di streaming ETL per un'origine dati dei log e utilizzare i pattern Grok per convertire i registri in dati strutturati. Il processo ETL elabora quindi i dati come origine dati strutturata. È possibile specificare i pattern Grok da applicare quando si crea la tabella Catalogo dati per l'origine di streaming.

Per informazioni sui pattern Grok e sui valori delle stringhe di pattern personalizzati, consulta Scrittura di classificatori personalizzati grok.

Aggiungere pattern Grok alla tabella Catalogo dati (console)

- Utilizza la procedura guidata per la creazione della tabella e crea la tabella con i parametri specificati in the section called "Creazione di un catalogo dati per un'origine di streaming". Specifica il formato dei dati come Grok, compila il pattern Grok e, facoltativamente, aggiungi pattern personalizzati in Custom patterns (optional) (Modelli personalizzati [facoltativo]).




Premi Invio dopo ogni pattern personalizzato.

Per aggiungere modelli grok alla tabella Data Catalog (API o) AWS Glue AWS CLI

- Aggiungi il parametro `GrokPattern` e, facoltativamente, il parametro `CustomPatterns` al processo API `CreateTable` o al comando CLI `create_table`.

```
 "Parameters": {
...
    "grokPattern": "string",
    "grokCustomPatterns": "string",
...
},
```

Esprimi `grokCustomPatterns` come stringa e usa "\n" come separatore tra i pattern.

Di seguito è riportato un esempio di specifica di questi parametri.

Example

```
"parameters": {
...
    "grokPattern": "%{USERNAME:username} %{DIGIT:digit:int}",
    "grokCustomPatterns": "digit \d",
...
}
```

## Definizione delle proprietà di processo per un processo di streaming ETL

Quando definisci un processo di streaming ETL nella console AWS Glue, fornisci le seguenti proprietà specifiche dei flussi. Per le descrizioni di proprietà aggiuntive, consulta Definire le proprietà di processo per i processi Spark.

Ruolo IAM

Specificate il ruolo AWS Identity and Access Management (IAM) utilizzato per l'autorizzazione alle risorse utilizzate per eseguire il processo, accedere alle sorgenti di streaming e accedere agli archivi dati di destinazione.

Per accedere ad Amazon Kinesis Data Streams, `AmazonKinesisFullAccess` AWS collega la policy gestita al ruolo o allega una policy IAM simile che consenta un accesso più dettagliato. Per i criteri di esempio, consulta Controllo dell'accesso alle risorse Amazon Kinesis Data Streams tramite IAM.

Per ulteriori informazioni sulle autorizzazioni per l'esecuzione di processi in AWS Glue, consulta Gestione delle identità e degli accessi per AWS Glue.

Tipo

Scegli Spark streaming.

Versione AWS Glue

La versione di AWS Glue determina le versioni di Apache Spark e Python che sono disponibili per il processo. Scegli una selezione che specifica la versione di Python o Scala disponibile per il processo. AWS Glue La versione 2.0 con supporto Python 3 è la versione predefinita per lo streaming delle operazioni ETL.

Finestra di manutenzione

Specifica una finestra in cui è possibile riavviare un processo di streaming. Per informazioni, consulta the section called "Finestre di manutenzione".

Timeout dei processi

È possibile inserire una durata in minuti. Il valore predefinito è vuoto.

- I processi di streaming devono avere un valore di timeout inferiore a 7 giorni o 10.080 minuti.
- Se il valore viene lasciato vuoto, il processo verrà riavviato dopo 7 giorni, se non è stata impostata una finestra di manutenzione. Se è stata impostata una finestra di manutenzione, il processo verrà riavviato durante la finestra di manutenzione dopo 7 giorni.

Origine dati

Rimuovi la tabella creata in the section called "Creazione di un catalogo dati per un'origine di streaming".

Destinazione dati

Esegui una delle seguenti operazioni:

- Scegliere Crea tabelle nell'oggetto dati e specificare le seguenti proprietà dell'oggetto dati.

  Datastore

  Seleziona Amazon S3 o JDBC.

  Formato

  Scegli un formato qualsiasi. Tutti sono supportati per lo streaming.
- Scegli Use tables in the data catalog and update your data target (Usa tabelle nel catalogo dati e aggiorna la destinazione dati) e scegli una tabella per un data store JDBC.

Definizione dello schema di output

Esegui una delle seguenti operazioni:

- Scegli Automatically detect schema of each record (Rileva automaticamente lo schema di ciascun record) per attivare il rilevamento dello schema. AWS Gluedetermina lo schema dai dati di streaming.
- Scegli Specify output schema for all records (Specificare lo schema di output per tutti i record) per utilizzare la trasformazione Apply Mapping (Applica mapping) per definire lo schema di output.

Script

È possibile fornire uno script personalizzato o modificare lo script generato per eseguire le operazioni supportate dal motore di Apache Spark Structured Streaming. Per informazioni sulle operazioni disponibili, consulta Operations on streaming/Datasets. DataFrames

## Streaming di note e restrizioni ETL

Tieni presente le seguenti note e restrizioni:

- La decompressione automatica per i processi ETL in streaming di AWS Glue è disponibile solo per i tipi di compressione supportati. Tieni presente quanto segue:
  - Snappy framed si riferisce al formato di framing ufficiale di Snappy.
  - Deflate è supportato in Glue versione 3.0, non Glue versione 2.0.
- Quando si utilizza il rilevamento dello schema, non è possibile eseguire join dei dati di streaming.
- I processi ETL di streaming di AWS Glue non supportano il tipo di dati Union per il registro degli schemi AWS Glue con il formato Avro.
- Lo script ETL può utilizzare le trasformazioni integrate di AWS Glue e le trasformazioni native di Apache Spark Structured Streaming. Per ulteriori informazioni, consulta Operazioni sullo streaming DataFrames /Datasets sul sito Web di Apache Spark oppure. AWS Glue PySpark trasforma il riferimento
- Le operazioni ETL di streaming AWS Glue utilizzano i checkpoint per tenere traccia dei dati letti. Pertanto, un processo arrestato e riavviato riprende da dove era stato interrotto nello stream. Se si desidera rielaborare i dati, è possibile eliminare la cartella di checkpoint a cui si fa riferimento nello script.
- I segnalibri delle operazioni non sono supportati.
- Per utilizzare la funzionalità di fan-out avanzato di Flusso di dati Kinesis, consulta la pagina the section called "Utilizzo del fan-out avanzato nei processi di flussi di dati Kinesis".
- Se si utilizza una tabella di catalogo dati creata da AWS Glue Schema Registry, quando diventa disponibile una nuova versione dello schema, per rifletterlo, è necessario effettuare le seguenti operazioni:
  1. Arrestare i processi associati alla tabella.
  2. Aggiornare lo schema per la tabella catalogo dati.
  3. Riavviare i processi associati alla tabella.

# Record di abbinamento con AWS Lake Formation FindMatches

> **Note**
>
> La corrispondenza dei record non è attualmente disponibile nelle seguenti regioni della AWS Glue console: Medio Oriente (Emirati Arabi Uniti), Europa (Spagna), Asia Pacifico (Giacarta) ed Europa (Zurigo).

AWS Lake Formation offre funzionalità di apprendimento automatico per creare trasformazioni personalizzate per ripulire i dati. Attualmente è disponibile una trasformazione denominata. FindMatches La FindMatches trasformazione consente di identificare i record duplicati o corrispondenti nel set di dati, anche quando i record non hanno un identificatore univoco comune e nessun campo corrisponde esattamente. Ciò non richiederà la scrittura di alcun codice o la conoscenza di come funziona l'apprendimento automatico. FindMatches può essere utile in molti problemi diversi, come ad esempio:

- Matching Customers (Corrispondenza di clienti): collegamento di record dei clienti tra diversi database, anche quando molti campi non corrispondono esattamente tra i database (ad es. diversa ortografia dei nomi, differenze di indirizzo, dati mancanti o imprecisi e così via).
- Matching Products (Corrispondenza di prodotti): abbinamento dei prodotti nel catalogo rispetto ad altre origini, ad esempio catalogo di prodotti rispetto al catalogo di un concorrente, in cui le voci sono strutturate in modo diverso.
- Improving Fraud Detection (Miglioramento del rilevamento delle frodi): identificazione di account dei clienti duplicati, determinazione di quando un nuovo account creato è (o potrebbe essere) una corrispondenza per un utente fraudolento noto.
- Other Matching Problems (Altri problemi di corrispondenza): abbinare indirizzi, film, elenchi di parti e così via. In generale, se un essere umano potesse esaminare le righe del database e determinare che coincidono, ci sono ottime probabilità che la FindMatches trasformazione possa aiutarvi.

È possibile creare queste trasformazioni al momento della creazione di un processo. La trasformazione creata si basa su uno schema del datastore di origine e su dati di esempio del set di dati di origine etichettato (questo processo viene denominato "insegnamento" di una trasformazione). I record etichettati devono essere presenti nel set di dati di origine. In questo processo viene generato un file etichettato e quindi ricaricato nel modo appreso dalla trasformazione. Dopo aver insegnato la

trasformazione, puoi richiamarla dal tuo AWS Glue job basato su Spark (PySpark o Scala Spark) e usarla in altri script con un data store di origine compatibile.

Dopo la creazione della trasformazione, questa viene memorizzata all'interno di AWS Glue. Nella console di AWS Glue è possibile gestire le trasformazioni create. Nel riquadro di navigazione in Integrazione dei dati ed ETL, Strumenti di classificazione dei dati > Corrispondenza dei record, puoi modificare e continuare a istruire la trasformazione del machine learning. Per ulteriori informazioni sulla gestione delle trasformazioni nella console, consultare Operare con le trasformazioni basate su machine learning.

> **Note**
>
> AWS Gluei FindMatches lavori della versione 2.0 utilizzano il bucket Amazon S3 `aws-glue-temp-<accountID>-<region>` per archiviare file temporanei mentre la trasformazione elabora i dati. Puoi eliminare questi dati dopo aver completato l'esecuzione, manualmente o impostando una regola del ciclo di vita di Amazon S3.

## Tipi di trasformazioni basate su machine learning

È possibile creare trasformazioni di machine learning per pulire i dati. Puoi chiamare queste trasformazioni dallo script ETL. I tuoi dati passano da una trasformazione all'altra in una struttura di dati chiamata a DynamicFrame, che è un'estensione di Apache Spark SQL. `DataFrame` `DynamicFrame` contiene i tuoi dati e il suo schema di riferimento per elaborare i dati.

Sono disponibili i seguenti tipi di trasformazioni basate su machine learning:

Rilevamento delle corrispondenze

Individua i record duplicati nei dati di origine. È possibile addestrare questa trasformazione basata su machine learning etichettando dei set di dati di esempio e indicando tra quali righe sono presenti delle corrispondenze. La trasformazione basata su machine learning apprende quali righe debbano essere abbinate man mano che vengono offerti dati di esempio etichettati. A seconda della configurazione della trasformazione, l'output è uno dei seguenti:

- Una copia della tabella di input con una colonna `match_id` aggiuntiva compilata con i valori che indicano insiemi di record corrispondenti. La colonna `match_id` è un identificatore arbitrario. Tutti i record con lo stesso `match_id` sono stati identificati come tra loro corrispondenti. I record con `match_id` diversi non corrispondono.

- Una copia della tabella di input con le righe duplicate rimosse. Se vengono rilevati molteplici duplicati, viene mantenuto il record con la chiave primaria minore.

Trova corrispondenze incrementali

La trasformazione Find matches può anche essere configurata per trovare le corrispondenze tra i frame esistenti e incrementali e restituire come output una colonna contenente un ID univoco per gruppo di corrispondenza.

Per ulteriori informazioni, consulta: Trovare corrispondenze incrementali

Utilizzo della FindMatches trasformazione

È possibile utilizzare la trasformazione `FindMatches` per individuare i record duplicati nei dati di origine. Viene generato o fornito un file di etichettatura che possa aiutare nell'addestramento della trasformazione.

> **Note**
>
> Attualmente, le trasformazioni `FindMatches` che usano una chiave di crittografia personalizzata non sono supportate nelle seguenti Regioni:
>
> - Asia Pacifico (Osaka): `ap-northeast-3`

Per iniziare con la FindMatches trasformazione, puoi seguire i passaggi seguenti. Per un esempio più avanzato e dettagliato, consulta il blog AWS Big Data: Harmonize data using AWS Glue and AWS Lake Formation FindMatches ML to build a customer 360.

Nozioni di base sull'utilizzo della trasformazione con rilevamento delle corrispondenze

Seguire questi passaggi per iniziare a usare la trasformazione `FindMatches`:

1. Crea una tabella AWS Glue Data Catalog per i dati di origine da pulire. Per informazioni su come creare un crawler, consultare Uso di crawler nella console AWS Glue.

   Se i dati di origine sono contenuti in un file di testo, ad esempio un file di valori separati da virgola (CSV), tenere conto delle seguenti considerazioni:

   - Mantenere il file CSV contenente i record di input e i file di etichettatura in cartelle separate. In caso contrario, il crawler di AWS Glue potrebbe considerarli come più componenti della stessa tabella e creare tabelle nel catalogo dati in modo non corretto.

- A meno che il file CSV includa solo caratteri ASCII, assicurarsi che per la codifica dei file CSV venga utilizzato UTF-8 senza BOM (Byte Order Mark). Microsoft Excel spesso aggiunge un BOM all'inizio dei file CSV UTF-8. Per rimuoverlo, aprire il file CSV con un editor di testo e salvare nuovamente il file in formato UTF-8 senza BOM.

2. Nella console di AWS Glue, creare un processo e scegliere il tipo di trasformazione Find matches (Rilevamento delle corrispondenze)

> **Important**
>
> La tabella dell'origine dati selezionata per il processo può contenere fino a un massimo di 100 colonne.

3. Indica a AWS Glue di generare un file di etichettatura scegliendo Generate labeling file (Genera file di etichettatura). AWS Glue utilizza il primo passaggio per raggruppare record simili per ciascun `labeling_set_id` in modo da poter rivedere tali raggruppamenti. Etichetta corrispondenze nella colonna `label`.
   - Se già disponi di un file di etichettatura, ossia di un esempio di record che indicano righe corrispondenti, carica il file su Amazon Simple Storage Service (Amazon S3). Per informazioni sul formato del file di etichettatura, consultare Formato del file di etichettatura. Continuare con la fase 4.
4. Scaricare il file di etichettatura ed etichettare il file come descritto nella sezione Etichettatura.
5. Caricare il file di etichettatura corretto. AWS Glue esegue delle attività per addestrare la trasformazione al riconoscimento delle corrispondenze.

   Nella pagina di elenco delle Machine learning transforms (Trasformazioni basate su machine learning), scegliere la scheda History (Cronologia). Questa pagina indica quando AWS Glue esegue le seguenti attività:
   - Import labels (Importa le etichette)
   - Export labels (Esporta le etichette)
   - Generate labels (Genera le etichette)
   - Estimate quality (Valuta la qualità)
6. Per creare una migliore trasformazione, è possibile scaricare, etichettare e caricare il file etichettato in modo iterativo. Nell'esecuzione iniziale, molti record potrebbero essere rilevati come non corrispondenti. Ma man mano che prosegue l'addestramento tramite la verifica del file di etichettatura, AWS Glue avanza nell'apprendimento.

7. Valutare e ottimizzare la trasformazione tramite la valutazione delle prestazioni e dei risultati della ricerca delle corrispondenze. Per ulteriori informazioni, consulta Ottimizzazione delle trasformazioni basate su machine learning in AWS Glue.

Etichettatura

Quando `FindMatches` genera un file di etichettatura, i record vengono selezionati dalla tabella di origine. Sulla base del training precedente, `FindMatches` identifica i record più importanti da cui apprendere.

L'atto di etichettatura consiste nella modifica di un file di etichettatura (ad esempio, un foglio di calcolo come Microsoft Excel) e l'aggiunta di identificatori, o etichette, nella colonna `label` che identifica i record con o senza corrispondenze. È importante avere una chiara e coerente definizione di corrispondenza nei dati di origine. `FindMatches` apprende sulla base dei record designati come corrispondenti (o meno) e utilizza le decisioni dell'utente per ricavare le informazioni necessarie all'individuazione dei record duplicati.

Quando il file di etichettatura viene generato da `FindMatches`, vengono generati circa 100 record. Questi 100 record sono in genere suddivisi in 10 set di etichettatura, dove ogni set di etichettatura è identificato da un unico `labeling_set_id` generato da `FindMatches`. Ogni set di etichettatura deve essere considerato come un'attività di etichettatura separata indipendente dagli altri set di etichettatura. Il tuo compito consiste nell'identificare i record corrispondenti e non corrispondenti all'interno di ciascun set di etichette.

Suggerimenti per la modifica dei file di etichettatura in un foglio di calcolo.

Quando si modifica il file di etichettatura in un foglio di calcolo, considerare i seguenti aspetti:

- Il file potrebbe non aprirsi con le colonne dei campi completamente espanse. Per visualizzare i contenuti di tali celle, potrebbe essere necessario espandere le colonne `labeling_set_id` e `label`.
- Se la colonna chiave primaria è un numero, ad esempio un tipo di dato `long`, il foglio di calcolo potrebbe interpretarlo come un numero e modificarne il valore. Questo valore chiave deve essere trattato come un testo. Per risolvere il problema, formattare tutte le celle nella colonna chiave primaria come Text data (Formato testo).

### Formato del file di etichettatura

Il file di etichettatura generato da AWS Glue per insegnare la trasformazione `FindMatches` utilizza il seguente formato. Se si genera il proprio file per AWS Glue, anch'esso deve seguire questo formato:

- Si tratta di un file di valori separati da virgola (CSV).
- Deve essere codificato in `UTF-8`. Se il file è stato modificato con Microsoft Windows, potrebbe essere codificato con `cp1252`.
- Affinché possa essere passato a AWS Glue è necessario che si trovi su Amazon S3.
- Utilizza un numero modesto di righe per ogni attività di etichettatura. Sono consigliate 10-20 righe per attività, anche se 2-30 righe per attività sono accettabili. Le attività superiori a 50 righe non sono consigliate e potrebbero causare risultati scadenti o errori di sistema.
- Se si dispone di dati già etichettati costituiti da coppie di record etichettati come "corrispondenza" o "nessuna corrispondenza", questo va bene. Queste coppie etichettate possono essere rappresentate come set di etichettatura di dimensione 2. In questo caso etichettare entrambi i record con, ad esempio, una lettera "A" se corrispondono, ma etichettare uno come "A" e uno come "B" se non corrispondono.

  > **Note**
  >
  > Poiché possiede delle colonne aggiuntive, il file di etichettatura presenta uno schema diverso da quello di un file che contiene i dati di origine. Posizionare il file di etichettatura in una cartella diversa da quella di qualsiasi altro file CSV di input della trasformazione in modo che il crawler di AWS Glue non lo prenda in considerazione al momento della creazione delle tabelle nel catalogo dati. In caso contrario, le tabelle create dal crawler di AWS Glue potrebbero non rappresentare correttamente i dati.

- Le prime due colonne (`labeling_set_id`, `label`) sono richieste obbligatoriamente da AWS Glue. Le colonne rimanenti devono corrispondere allo schema dei dati che devono essere elaborati.
- Per ogni `labeling_set_id`, è necessario identificare tutti i record corrispondenti utilizzando la stessa etichetta. Un'etichetta è una stringa univoca posizionata nella colonna `label`. Consigliamo di usare etichette contenenti caratteri semplici, ad esempio A, B, C e così via. Le etichette considerano in modo differente le maiuscole dalle minuscole e vengono inserite nella colonna `label`.

- Le righe che contengono lo stesso `labeling_set_id` e la stessa etichetta si intendono etichettate come corrispondenza.
- Le righe che contengono lo stesso `labeling_set_id` e un'etichetta diversa si intendono etichettate come non una corrispondenza
- Le righe che contengono un `labeling_set_id` diverso non trasmettono alcuna informazione a favore o contro la corrispondenza.

  Di seguito è riportato un esempio di etichettatura dei dati:

| labeling_set_id | etichetta | first_name | last_name | Compleanno |
|---|---|---|---|---|
| ABC123 | A | John | Doe | 04/01/1980 |
| ABC123 | B | Jane | Smith | 04/03/1980 |
| ABC123 | A | Johnny | Doe | 04/01/1980 |
| ABC123 | A | Jon | Doe | 04/01/1980 |
| DEF345 | A | Richard | Jones | 12/11/1992 |
| DEF345 | A | Rich | Jones | 11/12/1992 |
| DEF345 | B | Sara | Jones | 12/11/1992 |
| DEF345 | C | Richie | Jones Junior. | 05/06/2017 |
| DEF345 | B | Sara | Jones-Walker | 12/11/1992 |
| GHI678 | A | Roberto | Miller | 1/3/1999 |
| GHI678 | A | Bob | Miller | 1/3/1999 |
| XYZABC | A | Guglielmo | Robinson | 2/5/2001 |
| XYZABC | B | Andrea | Robinson | 2/5/1971 |

- Nell'esempio precedente identifichiamo John/Johnny/Jon Doe as being a match and we teach the system that these records do not match Jane Smith. Separately, we teach the system that Richard

and Rich Jones are the same person, but that these records are not a match to Sarah Jones/Jones-Walker e Richie Jones Jr.

- Come si può vedere, l'ambito delle etichette è limitato al `labeling_set_id`. Quindi le etichette non attraversano i limiti imposti dal `labeling_set_id`. Ad esempio, un'etichetta "A" nel `labeling_set_id` 1 non ha alcuna relazione con l'etichetta "A" nel `labeling_set_id` 2.
- Se un record non ha alcuna corrispondenza all'interno di un set di etichette, assegnargli un'etichetta univoca. Ad esempio, Jane Smith non corrisponde a nessun record nel set di etichette ABC123, quindi è l'unico disco di quel set di etichette con l'etichetta B.
- Il set di etichette "GHI678" mostra che un set di etichette può essere composto da soli due record a cui viene assegnata la stessa etichetta per dimostrare che corrispondono. Allo stesso modo, "XYZABC" mostra due record con etichette diverse per mostrare che non corrispondono.
- Si noti che a volte un set di etichette non può contenere corrispondenze (ovvero, si attribuisce a ogni record nel set di etichette un'etichetta diversa) o un set di etichette potrebbe essere "uguale" (ad essi è stata assegnata la stessa etichetta). Questo va bene fintanto che i set di etichettatura contengono collettivamente esempi di record "uguali" o "non uguali" secondo i propri criteri.

> **Important**
>
> Confermare che il ruolo IAM passato a AWS Glue abbia accesso al bucket Amazon S3 che contiene il file di etichettatura. Per convenzione, le policy di AWS Glue concedono le autorizzazioni sui bucket o sulle cartelle di Amazon S3 i cui nomi contengono il prefisso aws-glue-. Se i file di etichettatura si trovano in percorsi diversi, aggiungere al ruolo IAM l'autorizzazione di accesso a tale posizione.

## Ottimizzazione delle trasformazioni basate su machine learning in AWS Glue

È possibile ottimizzare le trasformazioni basate su machine learning in AWS Glue al fine di migliorare i risultati delle operazioni di pulizia dei dati affinché soddisfino gli obiettivi desiderati. Per migliorare la trasformazione, è possibile addestrarla generando un set di dati da etichettare, aggiungendo le etichette e quindi ripetendo questi passaggi diverse volte fino a ottenere i risultati desiderati. È inoltre possibile applicare l'ottimizzazione modificando alcuni parametri del sistema di machine learning.

Per ulteriori informazioni sulle trasformazioni basate su machine learning, consultare Record di abbinamento con AWS Lake Formation FindMatches.

Argomenti


- Misurazioni del machine learning
- Scelta tra precisione e recupero
- Scelta tra accuratezza e costo
- Stima della qualità delle corrispondenze utilizzando i punteggi di confidenza delle corrispondenze
- Addestramento della trasformazione di rilevamento delle corrispondenze


## Misurazioni del machine learning

Per comprendere le misurazioni che vengono utilizzate per ottimizzare una trasformazione basata su machine learning, è necessario avere familiarità con la seguente terminologia:

Vero positivo (True positive, TP)

: Una corrispondenza nei dati correttamente individuata dalla trasformazione, denominata anche colpo a segno.

Vero negativo (True negative , TN)

: Una mancata corrispondenza nei dati correttamente esclusa dalla trasformazione.

Falso positivo (False positive, FP)

: Una mancata corrispondenza nei dati che la trasformazione ha erroneamente classificato come una corrispondenza, denominata anche falso allarme.

Falso negativo (False negative, FN)

: Una corrispondenza nei dati non rilevata dalla trasformazione, denominata anche colpo mancato.

Per ulteriori informazioni sulla terminologia utilizzata nel campo del machine learning, consultare la voce Matrice di confusione su Wikipedia.

Per ottimizzare le trasformazioni basate su machine learning, è possibile modificare il valore delle seguenti misurazioni nella sezione Advanced properties (Proprietà avanzate) della trasformazione.

- Precision (Precisione) misura la capacità della trasformazione di individuare veri positivi sul numero totale di record che identifica come positivi (veri positivi e falsi positivi). Per ulteriori informazioni, consulta la voce Precisione e recupero su Wikipedia.

- Recall (Recupero) misura la capacità della trasformazione di individuare i veri positivi rispetto al totale dei record che compongono i dati di origine. Per ulteriori informazioni, consulta la voce Precisione e recupero su Wikipedia.
- Accuracy (Accuratezza) misura la capacità della trasformazione di individuare i veri positivi e i veri negativi. L'incremento dell'accuratezza implica maggiori risorse di elaborazione e costi superiori. Tuttavia permette di raggiungere anche un livello maggiore di recupero. Per ulteriori informazioni, consultare la voce Accuratezza e precisione su Wikipedia.
- Cost (Costo) misura la quantità di risorse di elaborazione (e quindi di denaro) consumate per l'esecuzione della trasformazione.

### Scelta tra precisione e recupero

Ogni trasformazione `FindMatches` contiene un parametro `precision-recall`. È possibile utilizzare questo parametro per specificare uno dei seguenti requisiti:

- Se la preoccupazione maggiore riguarda la possibilità che la trasformazione indichi la corrispondenza tra due record quando in effetti tale corrispondenza non esiste, allora è opportuno enfatizzare l'aspetto della precisione.
- Se la preoccupazione maggiore riguarda la mancata rilevazione di record tra i quali esiste in effetti una corrispondenza, allora è opportuno enfatizzare l'aspetto del recupero.

È possibile regolare questo compromesso all'interno della console di AWS Glue o utilizzando le operazioni API di machine learning di AWS Glue.

### Quando favorire la precisione

È opportuno favorire la precisione se la preoccupazione maggiore riguarda il rischio che `FindMatches` stabilisca una relazione tra due record quando in effetti tale corrispondenza non esiste. Per favorire la precisione, scegliere un valore più alto per il compromesso tra precisione e recupero. Con un valore più alto, la trasformazione `FindMatches` richiede un numero maggiore di elementi a sostegno per stabilire se una coppia di record deve essere è legata da una corrispondenza. Si incrementa la predisposizione della trasformazione a supporre che tra i record non esista una corrispondenza.

Ad esempio, si supponga di utilizzare `FindMatches` per rilevare gli elementi duplicati in un catalogo di video e di assegnare al parametro precisione-recupero della trasformazione un valore elevato. Se la trasformazione rileva erroneamente che Star Wars: Una nuova speranza è la stessa cosa di

Star Wars: L'impero colpisce ancora, a un cliente che desidera Una nuova speranza potrebbe essere mostrato L'impero colpisce ancora. Si tratterebbe di un'esperienza utente scadente.

Tuttavia, se la trasformazione non riesce a rilevare che Star Wars: Una nuova speranza e Star Wars: Episodio IV - Una nuova speranza sono lo stesso elemento, il cliente potrebbe essere confuso all'inizio ma potrebbe alla fine riconoscere i due elementi come lo stesso film. Sarebbe un errore, ma non così grave come nel caso precedente.

Quando favorire il recupero

È opportuno favorire il recupero se la preoccupazione maggiore riguarda il rischio che i risultati della trasformazione `FindMatches` possano non riuscire a rilevare una coppia di record tra i quali esiste un effetti una corrispondenza. Per favorire il recupero, scegliere un valore più basso per il compromesso tra precisione e recupero. Con un valore più basso, la trasformazione `FindMatches` richiede un numero minore di elementi a sostegno per decidere che una coppia di record è legata da una corrispondenza. Si incrementa la predisposizione della trasformazione a supporre che tra i record esista una corrispondenza.

Ad esempio, questa potrebbe essere una priorità per un'azienda che si occupa di sicurezza. Si supponga di confrontare l'elenco dei clienti con uno di noti frodatori e che sia importante determinare se un cliente è un frodatore. Si sta utilizzando `FindMatches` per trovare le corrispondenze tra l'elenco dei frodatori e quello dei clienti. Ogni volta che `FindMatches` rileva una corrispondenza tra i due elenchi, a un revisore umano viene assegnato il compito di verificare che la persona sia, in effetti, un frodatore. L'azienda potrebbe scegliere di favorire il recupero rispetto alla precisione. In altre parole, è preferibile che i verificatori debbano esaminare manualmente e rigettare alcuni casi in cui il cliente non è un frodatore piuttosto che fallire nell'identificazione di un cliente che si trova, in effetti, nell'elenco dei frodatori.

Come favorire sia la precisione che il recupero

Il modo migliore per migliorare la precisione e il recupero è etichettare una maggiore quantità di dati. Etichettando una maggiore quantità di dati, migliora la precisione globale della trasformazione `FindMatches`, con conseguenti miglioramenti sia della precisione che del recupero. Tuttavia, anche nel caso della trasformazione più accurata possibile, esiste sempre un'area grigia dove è necessario sperimentare se favorire precisione o recupero oppure scegliere un valore intermedio.

Scelta tra accuratezza e costo

Ogni trasformazione `FindMatches` contiene un parametro `accuracy-cost`. È possibile utilizzare questo parametro per specificare uno dei seguenti requisiti:

- Se la preoccupazione maggiore riguarda la possibilità che la trasformazione riveli con precisione la corrispondenza tra due record, allora è opportuno enfatizzare l'aspetto dell'accuratezza.
- Se la preoccupazione maggiore riguarda il costo o la velocità di esecuzione della trasformazione, allora è opportuno enfatizzare l'aspetto della riduzione del costo.

È possibile regolare questo compromesso all'interno della console di AWS Glue o utilizzando le operazioni API di machine learning di AWS Glue.

Quando favorire l'accuratezza

È opportuno favorire l'accuratezza se la preoccupazione maggiore riguarda il rischio che i risultati della trasformazione `find matches` non includano le corrispondenze. Per favorire l'accuratezza, scegliere un valore più alto per il compromesso tra accuratezza e costo. Con un valore più elevato, la trasformazione `FindMatches` richiede più tempo per approfondire la ricerca sui record tra i quali esiste una corrispondenza. Si noti che questo parametro non rende meno probabile la possibilità di indicare erroneamente corrispondenti due record tra i quali non esiste nessuna corrispondenza. Si incrementa la predisposizione della trasformazione a dedicare un tempo maggiore alla ricerca delle corrispondenze.

Quando favorire il costo

È opportuno favorire il costo se la preoccupazione maggiore riguarda il costo di esecuzione della trasformazione `find matches` rispetto al numero di corrispondenze rilevate. Per favorire il costo, scegliere un valore più basso per il compromesso tra accuratezza e costo. Con un valore più basso, la trasformazione `FindMatches` richiede una minore quantità di risorse per l'esecuzione. Si incrementa la predisposizione della trasformazione alla ricerca di un numero minore di corrispondenze. Utilizzare questa impostazione se, pur favorendo la ricerca di costi inferiori, i risultati sono comunque accettabili.

Come favorire sia l'accuratezza che il costo

Per esaminare un numero maggiore di coppie di record al fine di determinare la presenza di eventuali corrispondenze, serve un tempo di elaborazione maggiore. Se si desidera ridurre i costi senza ridurre la qualità, è possibile seguire la procedura illustrata qui di seguito:

- Eliminare i record dell'origine dati per i quali la presenza di una corrispondenza non è di interesse.
- Eliminare le colonne dell'origine dati che si è certi non siano utili ai fini della determinazione della presenza o meno di una corrispondenza. Un buon metodo per stabilirle quali siano è eliminare le

colonne che non sembrano influenzare la propria valutazione sul fatto che un insieme di record rappresentino "la stessa cosa".

Stima della qualità delle corrispondenze utilizzando i punteggi di confidenza delle corrispondenze

I punteggi di confidenza delle partite forniscono una stima della qualità delle corrispondenze rilevate FindMatches per distinguere tra record corrispondenti in cui il modello di apprendimento automatico è altamente sicuro, incerto o improbabile. Un punteggio di confidenza delle corrispondenze sarà compreso tra 0 e 1, dove il punteggio più alto significa una somiglianza più elevata. L'esame dei punteggi di confidenza delle corrispondenze consente di distinguere tra cluster di corrispondenze in cui il sistema è altamente sicuro (che potresti decidere di unire), cluster su cui il sistema è incerto (che potresti decidere di far esaminare da un essere umano) e cluster che il sistema ritiene improbabile (che potresti decidere di rifiutare).

Potresti dover modificare i tuoi dati di formazione in situazioni in cui vedi un punteggio di confidenza elevato, ma determinare che non ci sono corrispondenze, o dove vedi un punteggio basso determinare che ci sono, di fatto, corrispondenze.

I punteggi di fiducia sono particolarmente utili quando esistono set di dati industriali di grandi dimensioni, in cui è impossibile rivedere ogni decisione. FindMatches

I punteggi di confidenza delle corrispondenze sono disponibili in AWS Glue versione 2.0 o successive.

Generazione di punteggi di confidenza delle corrispondenze

È possibile generare punteggi di confidenza delle corrispondenze impostando il valore booleano di `computeMatchConfidenceScores` su Vero quando si chiama `FindMatches` o lAPI `FindIncrementalMatches`.

AWS Glue aggiunge una nuova `column match_confidence_score` all'output.

Esempi di punteggio di corrispondenza

Considera, ad esempio, le corrispondenze di registri seguenti:

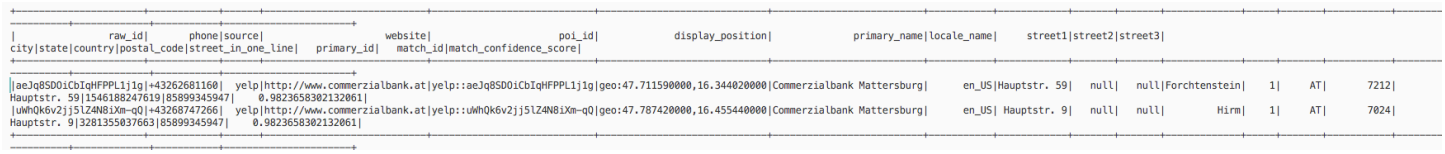
Punteggio >= 0,9

Riepilogo dei registri corrispondenti:

```
primary_id  |   match_id  | match_confidence_score
```

```
3281355037663    85899345947   0.9823658302132061
1546188247619    85899345947   0.9823658302132061
```

Informazioni:



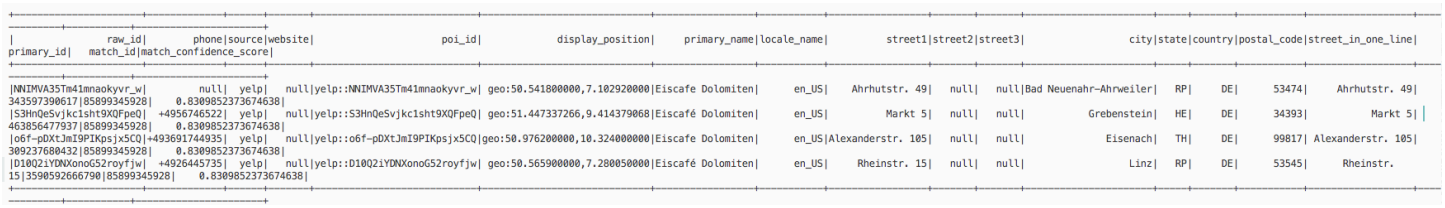
Da questo esempio, possiamo vedere che due registri sono molto simili e condividono `display_position`, `primary_name` e `street name`.

Punteggio >= 0,8 e punteggio < 0,9

Riepilogo dei registri corrispondenti:

```
  primary_id   |   match_id   | match_confidence_score

309237680432     85899345928   0.8309852373674638
3590592666790    85899345928   0.8309852373674638
343597390617     85899345928   0.8309852373674638
249108124906     85899345928   0.8309852373674638
463856477937     85899345928   0.8309852373674638
```

Informazioni:



Da questo esempio, possiamo vedere che questi registri condividono gli stessi `primary_name` e `country`.

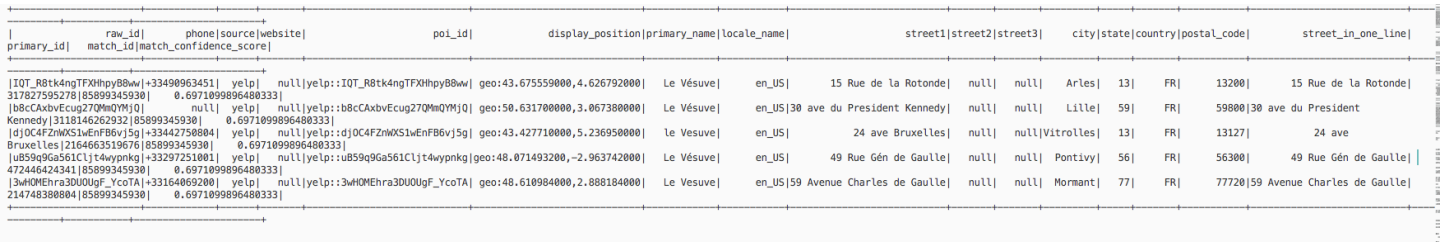
Punteggio >= 0,6 e punteggio < 0,7

Riepilogo dei registri corrispondenti:

```
  primary_id   |   match_id   | match_confidence_score
```

```
2164663519676    85899345930   0.6971099896480333
 317827595278    85899345930   0.6971099896480333
 472446424341    85899345930   0.6971099896480333
3118146262932    85899345930   0.6971099896480333
 214748380804    85899345930   0.6971099896480333
```

Informazioni:



```
+----------+--------------------+--------+-------+--------------------+--------------------+--------------------+--------------------+-----------+-----------+-----+-----+-----------+--------------------+
|primary_id|              raw_id|   phone| source|             website|              poi_id|    display_position|        primary_name|locale_name|    street1|street2|street3|       city|state|country|postal_code|    street_in_one_line|
+----------+--------------------+--------+-------+--------------------+--------------------+--------------------+--------------------+-----------+-----------+-------+-------+-----------+-----+-------+-----------+--------------------+
|IQT_R8tk4|IQT_R8tk4ngTFXHhpyB8ww|+33490963451|  yelp|                null|yelp::IQT_R8tk4ngTFXHhpyB8ww|geo:43.675559000,...|  Le Vésuve|      en_US|15 Rue de la Rotonde|   null|   null|      Arles|   13|     FR|      13200|15 Rue de la Rotonde|
|317827595278|85899345930|    0.6971099896480333|                                                
```

insiemi di attività, ricalcolare questi parametri e verificare il loro eventuale miglioramento. Se, dopo l'etichettatura di alcuni insiemi di attività, non si percepisce un miglioramento del parametro di interesse, la qualità della trasformazione potrebbe aver raggiunto uno stato stazionario.

Perché servono le etichette sia per gli eventi veri positivi che per quelli veri negativi?

La trasformazione `FindMatches` ha bisogno di esempi sia positivi che negativi per comprendere cosa intende l'utente per corrispondenza. Se si stanno etichettando dei dati di addestramento generati da `FindMatches` (ad esempio, utilizzando l'opzione I do not have labels (Non dispongo di etichette)), `FindMatches` prova a generare un set di "id di insiemi di etichette". All'interno di ciascuna attività, si assegna la stessa "etichetta" ad alcuni record e diverse "etichette" ad altri record. In altre parole, le attività generalmente non prevedono solo la presenza di elementi tutti uguali o tutti diversi (anche se è normale che una specifica attività comprenda elementi "tutti uguali" o "tutti diversi").

Se si sta addestrando la trasformazione `FindMatches` utilizzando l'opzione Upload labels from S3 (Caricamento delle etichette da S3), provare a includere sia esempi di record corrispondenti e che di record non corrispondenti. È accettabile averne di un solo tipo. Queste etichette consentono di creare una trasformazione `FindMatches` più accurata, ma è comunque necessario etichettare alcuni record generati utilizzando l'opzione Generate labeling file (Genera file di etichettatura).

Come posso fare in modo che la trasformazione rilevi le corrispondenze esattamente come è stata addestrata a fare?

La trasformazione `FindMatches` esegue un processo di apprendimento a partire dalle etichette fornite, perciò potrebbe generare coppie di record che non rispettano tali etichette. Per far sì che la `FindMatches` trasformazione rispetti le tue etichette, seleziona EnforceProvidedLabelsin. FindMatchesParameter

Quali tecniche è possibile utilizzare quando una trasformazione basata su ML identifica come corrispondenti degli elementi che non lo sono?

È possibile utilizzare le seguenti tecniche:

- Incrementare il valore di `precisionRecallTradeoff`. Questa operazione porterà all'individuazione di un numero minore di corrispondenze ma, al raggiungimento di un valore sufficientemente elevato, dovrebbe anche suddividere un cluster di grandi dimensioni in componenti più piccole.
- Selezionare le righe di output corrispondenti ai risultati errati e riformattarle sotto forma di insieme di dati per l'etichettatura (rimuovendo la colonna `match_id` e aggiungendo le colonne

`labeling_set_id` e `label`). Se necessario, spezzettarle (suddividerle) in più insiemi di dati per l'etichettatura al fine di assicurare che l'addetto all'etichettatura possa concentrarsi su ogni set di dati durante il processo di etichettatura. Quindi, etichettare correttamente i set corrispondenti e caricare il file di etichettatura accodandolo alle etichette esistenti. Queste informazioni potrebbero addestrare la trasformazione a sufficienza su cosa cercare per comprendere lo schema.

- (Avanzato) Infine, controllare i dati per verificare la presenza di uno schema che il sistema non sta rilevando. Pre-elaborare tali dati utilizzando le funzioni standard di AWS Glue per normalizzarli. Evidenziare gli elementi dai quali si desidera che l'algoritmo tragga insegnamenti separando i dati che l'utente ritiene importanti per la loro diversità nelle rispettive colonne. Oppure creare colonne combinate a partire dalle colonne i cui dati sono da ritenersi correlati.

## Operare con le trasformazioni basate su machine learning

Puoi utilizzarle AWS Glue per creare trasformazioni di machine learning personalizzate che possono essere utilizzate per pulire i dati. È possibile creare queste trasformazioni al momento della creazione di un processo nella console di AWS Glue .

Per informazioni su come creare una trasformazione basata su machine learning, consultare Record di abbinamento con AWS Lake Formation FindMatches.

Argomenti


- Proprietà della trasformazione
- Aggiunta e modifica della trasformazione basata su machine learning
- Visualizzazione dei dettagli della trasformazione
- Insegnamento delle trasformazioni utilizzando le etichette


### Proprietà della trasformazione

Per visualizzare una trasformazione di machine learning esistente, accedi a e apri la Console di gestione AWSAWS Glue console all'indirizzo. https://console.aws.amazon.com/glue/ Nel pannello di navigazione sotto Integrazione dati ed ETL, scegli Strumenti di classificazione dei dati > Corrispondenza dei record.

Le proprietà di ogni trasformazione:

Nome trasformazione

Il nome univoco che assegnato alla trasformazione al momento della creazione.

ID

Un identificatore unico della trasformazione.

Numero delle etichette

Il numero di etichette nel file di etichettatura fornito per l'addestramento della trasformazione.

Status

Indica se la trasformazione è Ready (Pronta) o Needs training (Ha bisogno di addestramento). Per eseguire correttamente una trasformazione basata su machine learning in un processo, questa deve trovarsi nello stato Ready (Pronta).

Creato

La data di creazione della trasformazione.

Modificato

La data dell'ultimo aggiornamento della trasformazione.

Description

La descrizione fornita per la trasformazione, se ne è stata fornita una.

Versione AWS Glue

La versione di AWS Glue utilizzata.

ID esecuzione

Il nome univoco che assegnato alla trasformazione al momento della creazione.

Tipo di attività

Il tipo di trasformazione basata su machine learning; ad esempio, Find matching records (Rilevamento record corrispondenti).

Status

Indica lo stato dell'esecuzione dell'attività. Gli stati possibili comprendono:

- Avvio in corso
- In esecuzione
- In arresto
- Arrestato
- Riuscito

- Non riuscito
- Timeout

Errore

Se lo stato è Non riuscito, viene visualizzato un messaggio di errore che descrive il motivo dell'errore.

Aggiunta e modifica della trasformazione basata su machine learning

Nella console AWS Glue è possibile visualizzare, eliminare, impostare e addestrare o ottimizzare una trasformazione. Selezionare la casella di controllo accanto alla trasformazione nell'elenco, scegliere Action (Operazione) e quindi scegliere l'operazione che si desidera eseguire.

Creazione di una nuova trasformazione ML

Per aggiungere una nuova trasformazione di machine learning, scegli Crea trasformazione. Segui le istruzioni nella procedura guidata Aggiungi crawler. Per ulteriori informazioni, consulta Record di abbinamento con AWS Lake Formation FindMatches.

Passaggio 1. Imposta le proprietà della trasformazione.

1. Inserisci il nome e la descrizione (facoltativo).
2. Facoltativamente, imposta la configurazione di sicurezza. Per informazioni, consulta Utilizzo della crittografia dati con le trasformazioni basate su machine learning.
3. Facoltativamente, configura le impostazioni di esecuzione delle attività. Le impostazioni di esecuzione delle attività consentono di personalizzare la modalità di esecuzione dell'attività. Seleziona il tipo di e il numero di worker, il timeout dell'attività (in minuti), il numero di nuovi tentativi e la versione di AWS Glue.
4. Facoltativamente, imposta i tag. I tag sono etichette che puoi assegnare a una AWS risorsa. Ciascun tag è formato da una chiave e da un valore facoltativo. I tag possono essere utilizzati per cercare e filtrare la risorsa o tenere traccia AWS dei costi.

Passaggio 2. Scegli la tabella e la chiave primaria.

1. Scegli il database e la tabella di Catalogo AWS Glue.
2. Scegli una chiave primaria dalla tabella selezionata. La colonna della chiave primaria contiene in genere un identificatore univoco per ogni record nell'origine dati.

Fase 3. Seleziona le opzioni di ottimizzazione.

1. Per Richiamo o precisione, scegli il valore di regolazione per ottimizzare la trasformazione in modo da favorire il richiamo o la precisione. Per impostazione predefinita, è selezionata l'opzione Bilanciato, ma puoi scegliere di favorire il richiamo o la precisione; puoi anche scegliere l'opzione Personalizzato e inserire un valore compreso tra 0,0 e 1,0 (inclusi).
2. Per Costo o precisione inferiore, scegli il valore di regolazione per favorire un costo o una precisione inferiori oppure scegli Personalizzato e inserisci un valore compreso tra 0,0 e 1,0 (inclusi).
3. Per Forza corrispondenza, scegli Forza l'output a corrispondere alle etichette se desideri addestrare la trasformazione ML forzando l'output a corrispondere alle etichette utilizzate.

Passaggio 4. Revisione e creazione.

1. Esamina le opzioni per i passaggi da 1 a 3.
2. Scegli Modifica per qualsiasi passaggio che desideri modificare. Scegli Crea trasformazione per completare la procedura guidata di creazione della trasformazione.

Utilizzo della crittografia dati con le trasformazioni basate su machine learning

Quando si aggiunge una trasformazione basata su machine learning a AWS Glue, è possibile specificare facoltativamente una configurazione di sicurezza associata all'origine dati o alla destinazione dati. Se il bucket Amazon S3 utilizzato per memorizzare i dati è crittografato con una configurazione di sicurezza, specifica la stessa configurazione di sicurezza durante la creazione della trasformazione.

Puoi anche scegliere di utilizzare la crittografia lato server con AWS KMS (SSE-KMS) per crittografare il modello e le etichette per impedire l'ispezione da parte di persone non autorizzate. Se scegli questa opzione, ti viene richiesto di scegliere il nome AWS KMS key per nome oppure puoi scegliere Inserisci una chiave ARN. Se si sceglie di inserire l'ARN per la chiave KMS, viene visualizzato un secondo campo in cui è possibile inserire l'ARN della chiave KMS.

> **Note**
>
> Attualmente, le trasformazioni ML che usano una chiave di crittografia personalizzata non sono supportate nelle seguenti Regioni:

- Asia Pacifico (Osaka): `ap-northeast-3`

### Visualizzazione dei dettagli della trasformazione

#### Visualizzazione delle proprietà della trasformazione

La pagina Proprietà della trasformazione include gli attributi della trasformazione. Mostra i dettagli relativi alla definizione della trasformazione, tra cui i seguenti:

- Transform name (Nome della trasformazione) mostra il nome della trasformazione.
- Tipo elenca il tipo della trasformazione.
- Stato indica se la trasformazione è pronta per essere utilizzata in uno script o un processo.
- Force output to match labels (Forza l'output affinché corrisponda alle etichette) mostra se la trasformazione esegue una forzatura affinché l'output corrisponda alle etichette indicate dall'utente.
- Versione Spark è correlato alla versione di AWS Glue che hai scelto nelle Proprietà esecuzione processo all'aggiunta della trasformazione. AWS Glue 1.0 e Spark 2.4 sono consigliati per la maggior parte dei clienti. Per ulteriori informazioni, consulta Versioni di AWS Glue.

#### Schede Cronologia, Stima qualità e Tag

I dettagli includono le informazioni definite al momento della creazione della trasformazione. Per visualizzare i dettagli di una trasformazione, selezionare la trasformazione nell'elenco delle Machine learning transforms (Trasformazioni basate su machine learning) e rivedere le informazioni contenute nelle seguenti schede:

- Cronologia
- Stima della qualità
- Tag

#### Cronologia

La scheda History (Cronologia) mostra la cronologia delle esecuzioni della trasformazione. Per addestrare una trasformazione, vengono eseguiti diversi tipi di attività. Per ogni attività, i parametri di esecuzione includono:

- Run ID (ID esecuzione) è un identificatore creato da AWS Glue per ogni esecuzione di questo processo.
- Task type (Tipo di attività) mostra il tipo di attività eseguita.
- Status (Stato) mostra la corretta conclusione di ogni esecuzione posizionando quella più recente in cima all'elenco.
- Errore mostra i dettagli di un messaggio di errore se l'esecuzione non riesce.
- Start time (Orario inizio) mostra la data e l'ora (ora locale) in cui è stato avviato il processo.
- Orario fine mostra la data e l'ora (ora locale) in cui il processo è finito.
- Log collega ai log scritti in `stdout` per questa esecuzione di processo.

  Il link Logs ti porta ad Amazon CloudWatch Logs. Qui puoi visualizzare i dettagli sulle tabelle create in AWS Glue Data Catalog e gli eventuali errori riscontrati. È possibile gestire il periodo di conservazione dei registri sulla CloudWatch console. Il periodo di conservazione log predefinito è `Never Expire`. Per ulteriori informazioni su come modificare il periodo di conservazione, consulta Change Log Data Retention in CloudWatch Logs nella Amazon CloudWatch Logs User Guide.
- File di etichettatura mostra un link ad Amazon S3 che permette di raggiungere un file di etichettatura generato.

### Stima della qualità

La scheda Estimate quality (Stima qualità) mostra i parametri utilizzati per misurare la qualità della trasformazione. Le stime vengono calcolate confrontando le previsioni di conversione utilizzando un sottoinsieme dei dati etichettati con le etichette fornite. Queste stime sono approssimative. Da questa scheda è possibile richiamare l'esecuzione dell'attività Estimate quality (Stima qualità).

La scheda Estimate quality (Stima qualità) mostra i parametri dell'ultima esecuzione Estimate quality (Stima qualità), incluse le seguenti proprietà:

- Area under the Precision-Recall curve (Area sotto la curva precisione-recupero) è un singolo numero che stima il limite superiore della qualità complessiva della trasformazione. È indipendente dalla scelta del parametro precisione-recupero. Valori più elevati indicano che si dispone di un compromesso precisione-recupero migliore.
- Precision (Precisione) stima la frequenza di correttezza della trasformazione quando prevede una corrispondenza.
- Recall upper limit (Limite superiore recupero) stima quanto spesso la trasformazione prevede una corrispondenza in caso di effettiva presenza.

- F1 stima l'accuratezza della trasformazione con un valore tra 0 e 1, dove 1 è la migliore precisione. Per ulteriori informazioni, consulta la voce F1 score su Wikipedia.
- La tabella Column importance (Importanza colonna) mostra i nomi delle colonne e il punteggio di importanza per ogni colonna. L'importanza delle colonne consente di comprendere il modo in cui queste contribuiscono al modello, identificando quali colonne nei record vengono maggiormente utilizzate per la corrispondenza. Questi dati possono richiedere di aggiungere o modificare il set di etichette per aumentare o diminuire l'importanza delle colonne.

  La colonna Importance (Importanza) fornisce un punteggio numerico per ogni colonna, come decimale non maggiore di 1,0.

Per ulteriori informazioni su come comprendere le stime della qualità rispetto alla vera qualità, consultare Stime sulla qualità rispetto alla qualità ( end-to-endvera).

Per ulteriori informazioni sull'ottimizzazione della trasformazione, consultare Ottimizzazione delle trasformazioni basate su machine learning in AWS Glue.

Stime sulla qualità rispetto alla qualità ( end-to-endvera)

AWS Glue stima la qualità della trasformazione passando al modello addestrato tramite machine learning interno un certo numero di coppie di record per i quali sono state fornite delle etichette corrispondenti ma che il modello non ha mai visto in precedenza. Queste stime di qualità sono una funzione della qualità del modello addestrato tramite machine learning (che dipende dal numero di record etichettati per "addestrare" la trasformazione). Il richiamo end-to-end, o vero, (che non viene calcolato automaticamente da`ML transform`) è influenzato anche dal meccanismo di `ML transform` filtraggio che propone un'ampia varietà di possibili corrispondenze al modello di apprendimento automatico.

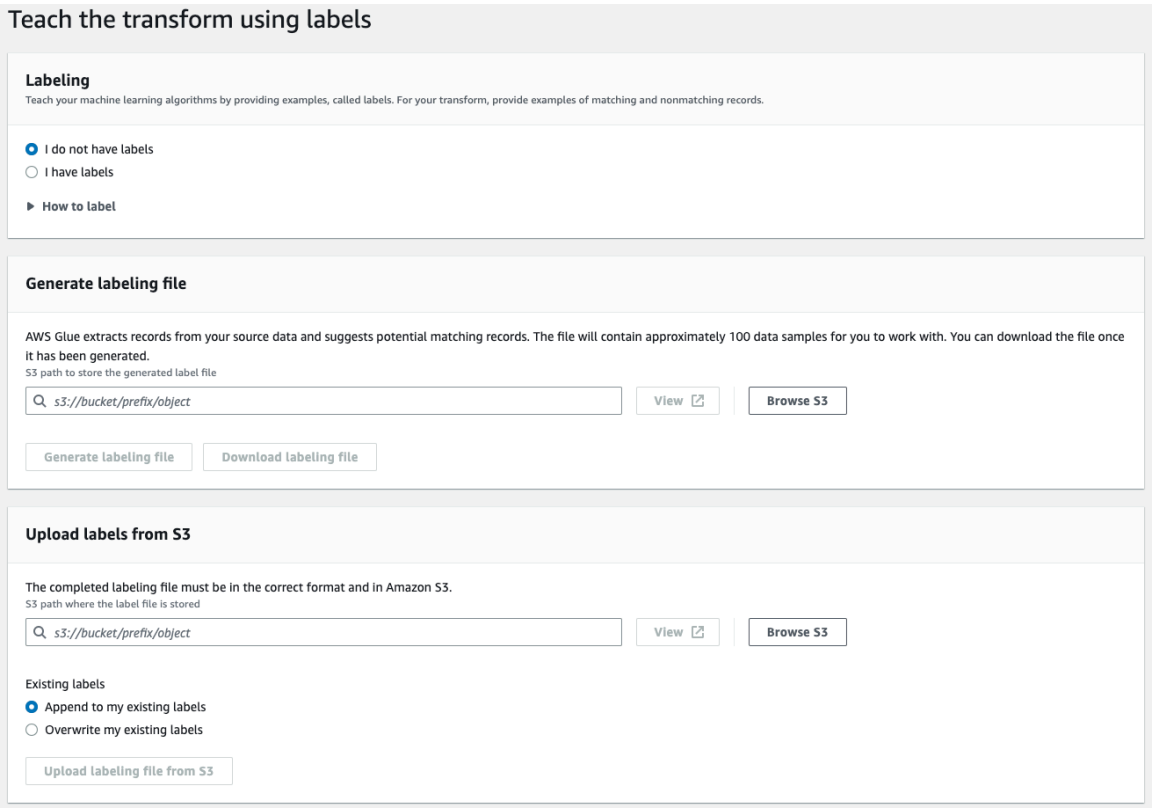
È possibile ottimizzare tale metodo di filtraggio principalmente utilizzando il cursore Costo o accuratezza inferiore. Spostando il cursore verso Accuratezza per favorire questo aspetto, il sistema esegue una ricerca più vasta e approfondita delle coppie di record che potrebbero rappresentare delle corrispondenze. Più coppie di record vengono inserite nel modello di apprendimento automatico e il tuo richiamo effettivo si avvicina alla metrica `ML transform` di end-to-end richiamo stimata. Di conseguenza, le variazioni nella end-to-end qualità delle partite dovute a variazioni del rapporto costo/precisione delle partite in genere non si riflettono nella stima della qualità.

## Tag

I tag sono etichette che puoi assegnare a una risorsa. AWS Ciascun tag è formato da una chiave e da un valore facoltativo. I tag possono essere utilizzati per cercare e filtrare la risorsa o tenere traccia AWS dei costi.

### Insegnamento delle trasformazioni utilizzando le etichette

È possibile insegnare la trasformazione ML tramite le etichette (esempi) scegliendo Insegna la trasformazione dalla pagina dei dettagli della trasformazione ML. Quando addestri l'algoritmo di machine learning fornendo esempi (chiamati etichette), puoi scegliere etichette esistenti da utilizzare o creare un file di etichettatura.




- Etichettatura: se hai delle etichette, scegli Ho delle etichette. Se non disponi di etichette, puoi comunque proseguire con il passaggio successivo per generare un file di etichettatura.
- Genera un file di etichettatura: AWS Glue estrae i record dai dati di origine e suggerisce potenziali record corrispondenti. Scegli il bucket Amazon S3 per archiviare il file di etichette generato. Scegli Genera file di etichettatura per avviare il processo. Al termine, scegli Scarica il file di etichettatura. Il file scaricato avrà una colonna per le etichette in cui potrai inserire le etichette.

- Carica etichette da Amazon S3: scegli il file di etichettatura completo dal bucket Amazon S3 in cui è archiviato il file di etichette. Quindi, scegli se aggiungere le etichette alle etichette esistenti o sovrascriverle. Scegli Carica file di etichettatura da Amazon S3.

## Tutorial: creazione di una trasformazione dell'apprendimento automatico con AWS Glue

Questa esercitazione guida l'utente nelle operazioni necessarie per creare e gestire una trasformazione basata su machine learning (ML) utilizzando AWS Glue. Prima di seguire questa esercitazione, è necessario sapere come utilizzare la console di AWS Glue per aggiungere crawler e processi e modificare script. È inoltre necessario avere familiarità con la ricerca e il download di file tramite la console Amazon Simple Storage Service (Amazon S3).

In questo esempio, viene creata una trasformazione `FindMatches` per identificare i record corrispondenti, insegnando ad essa come identificare i record con corrispondenze e quelli senza, e la si utilizza in un processo AWS Glue. Il processo AWS Glue genera un nuovo file Amazon S3 con una colonna aggiuntiva denominata `match_id`.

I dati di origine utilizzati da questa esercitazione cono contenuti in un file denominato `dblp_acm_records.csv`. Questo file è una versione modificata derivante da pubblicazioni accademiche (DBLP e ACM) disponibili presso la fonte originale set di dati DBLP ACM . Il file `dblp_acm_records.csv` è un file di valori separati da virgole (CSV) in formato UTF-8 senza BOM (Byte Order Mark).

Un secondo file, `dblp_acm_labels.csv`, è un esempio di file di etichettatura che contiene sia i record con corrispondenze che quelli senza utilizzato per addestrare la trasformazione come parte dell'esercitazione.

Argomenti


- Fase 1: crawling dei dati di origine
- Fase 2: aggiunta di una trasformazione basata su machine learning
- Fase 3: addestramento della trasformazione basata su machine learning
- Fase 4: stima della qualità della trasformazione basata su machine learning
- Fase 5: aggiunta ed esecuzione di un processo con la trasformazione basata su machine learning
- Fase 6: verifica dei dati di output da Amazon S3

### Fase 1: crawling dei dati di origine

In primo luogo, esegui il crawling del file CSV di origine su Amazon S3 per creare una tabella di metadati corrispondente nel catalogo dati.

> **Important**
>
> Per ottenere dal crawler la creazione di una tabella per il solo file CSV, archivia il file CSV dei dati di origine in una cartella Amazon S3 diversa da quella degli altri file.

1. Accedi a Console di gestione AWS e apri la AWS Glue console all'indirizzo https://console.aws.amazon.com/glue/.
2. Nel riquadro di navigazione, selezionare Crawlers (Crawler), Add crawler (Aggiungi Crawler).
3. Seguire la procedura guidata per creare ed eseguire un crawler denominato `demo-crawl-dblp-acm` con output indirizzato verso il database `demo-db-dblp-acm`. Se questo non esiste già, durante l'esecuzione della procedura guidata è necessario creare il database `demo-db-dblp-acm`. Scegli un percorso di inclusione di Amazon S3 per i dati di esempio nella regione corrente AWS . Ad esempio, per `us-east-1`, il percorso di inclusione per i dati di origine su Amazon S3 è `s3://ml-transforms-public-datasets-us-east-1/dblp-acm/records/dblp_acm_records.csv`.

   Se conclude l'attività con successo, il crawler crea la tabella `dblp_acm_records_csv` con le seguenti colonne: id, title, authors, venue, year e source.

### Fase 2: aggiunta di una trasformazione basata su machine learning

A questo punto, aggiungere una trasformazione basata su machine learning basata sullo schema dei dati della tabella di origine creata dal crawler e denominata `demo-crawl-dblp-acm`.

1. Nella console AWS Glue, nel riquadro di navigazione in Integrazione dati e ETL, scegli Strumenti di classificazione dei dati > Corrispondenza dei record, quindi Aggiungi trasformazione. Quindi segui la procedura guidata per creare una trasformazione `Find matches` con le seguenti proprietà.

   a. Alla voce Transform name (Nome trasformazione), immettere **`demo-xform-dblp-acm`**. Questo è il nome della trasformazione che viene utilizzato per trovare le corrispondenze nei dati di origine.

b. Alla voce IAM role (Ruolo IAM), scegli un ruolo IAM che disponga delle autorizzazioni per accedere ai dati di origine su Amazon S3, ai file di etichettatura e alle operazioni API di AWS Glue. Per ulteriori informazioni, consulta Creazione di un ruolo IAM per AWS Glue nella Guida per sviluppatori di AWS Glue .

c. Per l'origine dei dati, scegli la tabella denominata dblp_acm_records_csv nel database. demo-db-dblp-acm

d. Alla voce Primary key (Chiave primaria), scegliere la colonna chiave primaria della tabella, id.

2. Nella procedura guidata, scegliere Finish (Fine) e tornare all'elenco delle ML transforms (Trasformazioni basate su ML).

Fase 3: addestramento della trasformazione basata su machine learning

A questo punto è necessario addestrare la trasformazione basata su machine learning utilizzando il file di etichettatura di esempio del tutorial.

Non è possibile utilizzare una trasformazione basata su machine learning in un processo di estrazione, trasformazione e caricamento (ETL) finché il suo stato non è Ready for use (Pronta per l'uso). Affinché la trasformazione sia pronta, è necessario addestrarla a identificare i record con corrispondenze e quelli senza fornendo esempi di record con corrispondenza e di record senza corrispondenza. Per addestrare la trasformazione, è possibile scegliere Generate a label file (Genera un file di etichettatura), aggiungere le etichette e quindi selezionare Upload label file (Carica un file di etichettatura). In questa esercitazione è possibile utilizzare il file di etichettatura di esempio denominato `dblp_acm_labels.csv`. Per ulteriori informazioni sul processo di etichettatura, consultare Etichettatura.

1. Nella console AWS Glue, seleziona Corrispondenza dei record nel riquadro di navigazione.
2. Scegliere la trasformazione `demo-xform-dblp-acm` e quindi scegliere Action (Operazione), Teach (Addestra). Seguire la procedura guidata per addestrare la trasformazione `Find matches`.
3. Nella pagina delle proprietà della trasformazione, scegliere I have labels (Dispongo delle etichette). Scegli un percorso Amazon S3 per il file di etichettatura di esempio nella regione corrente. AWS Ad esempio, nel caso di `us-east-1`, caricare il file di etichettatura fornito dal percorso su Amazon S3 `s3://ml-transforms-public-datasets-us-east-1/dblp-acm/labels/dblp_acm_labels.csv` con l'opzione di overwrite (sovrascrivere) le etichette

esistenti. Il file di etichettatura deve essere situato su Amazon S3 nella stessa regione in cui si trova la console di AWS Glue.

Quando si carica un file di etichettatura, AWS Glue avvia un'attività per aggiungere o sovrascrivere le etichette utilizzate per addestrare la trasformazione sull'elaborazione dell'origine dati.

4. Nella pagina finale della procedura guidata scegliere Finish (Fine) e tornare all'elenco delle ML transforms (Trasformazioni basate su ML).

### Fase 4: stima della qualità della trasformazione basata su machine learning

Successivamente, è possibile stimare la qualità della propria trasformazione basata su machine learning. La qualità varia in base al numero di etichettature eseguite. Per ulteriori informazioni sulla stima della qualità, consultare Stima della qualità.

1. Nella console AWS Glue, nel riquadro di navigazione in Integrazione dati e ETL, scegli Strumenti di classificazione dei dati > Corrispondenza dei record.
2. Scegliere la trasformazione `demo-xform-dblp-acm` e scegliere la scheda Estimate quality (Stima della qualità). Questa scheda visualizza l'attuale stima di della qualità, se disponibile, per la trasformazione.
3. Scegliere Estimate quality (Stima della qualità) per avviare un'attività di stima della qualità della trasformazione. La precisione della stima della qualità si poggia sull'etichettatura dei dati di origine.
4. Passare alla scheda History (Cronologia). In questo riquadro sono elencate le esecuzioni di attività per ogni trasformazione, inclusa l'attività di Estimate quality (Stima della qualità). Per ulteriori informazioni sull'esecuzione, scegliere Logs (Log). Verificare che, al termine dell'operazione, lo stato di esecuzione sia Succeeded (Completata correttamente).

### Fase 5: aggiunta ed esecuzione di un processo con la trasformazione basata su machine learning

In questo passaggio si utilizza la trasformazione basata su machine learning per aggiungere ed eseguire un processo in AWS Glue. Quando la trasformazione `demo-xform-dblp-acm` è Ready for use (Pronta per l'uso) è possibile utilizzarla in un processo ETL.

1. Nel riquadro di navigazione della console di AWS Glue, scegliere Jobs (Processi).

2. Scegliere Add job (Aggiungi processo) e seguire la procedura guidata per creare un processo ETL Spark con uno script generato. Per le proprietà della trasformazione scegliere i seguenti valori:

   a. Per Nome, scegli il lavoro di esempio in questo tutorial,. demo-etl-dblp-acm

   b. Alla voce IAM role (Ruolo IAM), scegli un ruolo IAM che disponga delle autorizzazioni per accedere ai dati di origine su Amazon S3, ai file di etichettatura dei dati e alle operazioni API di AWS Glue. Per ulteriori informazioni, consulta Creazione di un ruolo IAM per AWS Glue nella Guida per sviluppatori di AWS Glue .

   c. Alla voce ETL language (Linguaggio ETL) scegli Scala. Questo è il linguaggio di programmazione dello script ETL.

   d. Per il nome del file di script, scegliete demo-etl-dblp-acm. Questo è il nome del file dello script Scala (uguale al nome del processo).

   e. Come Data source (Origine dati), scegliere dblp_acm_records_csv. L'origine dati scelta deve corrispondere allo schema dell'origine dati della trasformazione basata su machine learning.

   f. Alla voce Transform type (Tipo di trasformazione), scegliere Find matching records (Individuazione record corrispondenti) per creare un processo che utilizza una trasformazione basata su machine learning.

   g. Annullare la selezione di Remove duplicate records (Rimuovi record duplicati). Si sceglie di non rimuovere i record duplicati perché i record di output dispongono di un campo aggiuntivo `match_id` accodato.

   h. Per Transform demo-xform-dblp-acm, scegli la trasformazione di machine learning utilizzata dal job.

   i. Alla voce Create tables in your data target (Crea tabelle nella destinazioni dati), scegliere di creare tabelle con le seguenti proprietà:

      - Tipo di memorizzazione dei dati: **`Amazon S3`**
      - Formato: **`CSV`**
      - Tipo di compressione: **`None`**
      - Percorso di destinazione: il percorso Amazon S3 in cui viene scritto l'output del processo (nell'attuale regione della console AWS )

3. Scegliere Save job and edit script (Salva processo e modifica script) per visualizzare la pagina dell'editor dello script.

4. Modificare lo script per aggiungere un'istruzione che faccia sì che l'output del processo sia scritto sul Target path (Percorso di destinazione) in un file a singola partizione. Aggiungere questa

istruzione immediatamente dopo l'istruzione che esegue la trasformazione `FindMatches`. Le istruzioni sono simili alle seguenti.

```
val single_partition = findmatches1.repartition(1)
```

È necessario modificare l'istruzione `.writeDynamicFrame(findmatches1)` per scrivere l'output come `.writeDynamicFrame(single_partion)`.

5. Dopo aver modificato lo script, scegliere Save (Salva). Lo script modificato è simile al codice riportato qui di seguito, ma personalizzato in base al proprio tuo ambiente.

```
import com.amazonaws.services.glue.GlueContext
import com.amazonaws.services.glue.errors.CallSite
import com.amazonaws.services.glue.ml.FindMatches
import com.amazonaws.services.glue.util.GlueArgParser
import com.amazonaws.services.glue.util.Job
import com.amazonaws.services.glue.util.JsonOptions
import org.apache.spark.SparkContext
import scala.collection.JavaConverters._

object GlueApp {
  def main(sysArgs: Array[String]) {
    val spark: SparkContext = new SparkContext()
    val glueContext: GlueContext = new GlueContext(spark)
    // @params: [JOB_NAME]
    val args = GlueArgParser.getResolvedOptions(sysArgs, Seq("JOB_NAME").toArray)
    Job.init(args("JOB_NAME"), glueContext, args.asJava)
    // @type: DataSource
    // @args: [database = "demo-db-dblp-acm", table_name = "dblp_acm_records_csv",
 transformation_ctx = "datasource0"]
    // @return: datasource0
    // @inputs: []
    val datasource0 = glueContext.getCatalogSource(database = "demo-db-dblp-acm",
 tableName = "dblp_acm_records_csv", redshiftTmpDir = "", transformationContext =
 "datasource0").getDynamicFrame()
    // @type: FindMatches
    // @args: [transformId = "tfm-123456789012", emitFusion = false,
 survivorComparisonField = "<primary_id>", transformation_ctx = "findmatches1"]
    // @return: findmatches1
    // @inputs: [frame = datasource0]
```

```
    val findmatches1 = FindMatches.apply(frame = datasource0, transformId
 = "tfm-123456789012", transformationContext = "findmatches1",
 computeMatchConfidenceScores = true)


    // Repartition the previous DynamicFrame into a single partition.
    val single_partition = findmatches1.repartition(1)


    // @type: DataSink
    // @args: [connection_type = "s3", connection_options = {"path": "s3://aws-
glue-ml-transforms-data/sal"}, format = "csv", transformation_ctx = "datasink2"]
    // @return: datasink2
    // @inputs: [frame = findmatches1]
    val datasink2 = glueContext.getSinkWithFormat(connectionType =
 "s3", options = JsonOptions("""{"path": "s3://aws-glue-ml-transforms-
data/sal"}"""), transformationContext = "datasink2", format =
 "csv").writeDynamicFrame(single_partition)
    Job.commit()
  }
}
```

6. Scegliere Run job (Esegui processo) per avviare l'esecuzione del processo. Controllare lo stato del processo nell'elenco dei processi. Al termine del processo, nella finestra ML transform (Trasformazione ML), History (Cronologia), è disponibile una nuova riga Run ID (ID esecuzione) aggiunta di tipo  ETL job (Processo ETL).
7. Passare alla scheda Jobs (Processi), History (Cronologia). In questo riquadro vengono elencate le esecuzioni dei processi. Per ulteriori informazioni sull'esecuzione, scegliere Logs (Log). Verificare che, al termine dell'operazione, lo stato di esecuzione sia Succeeded (Completata correttamente).

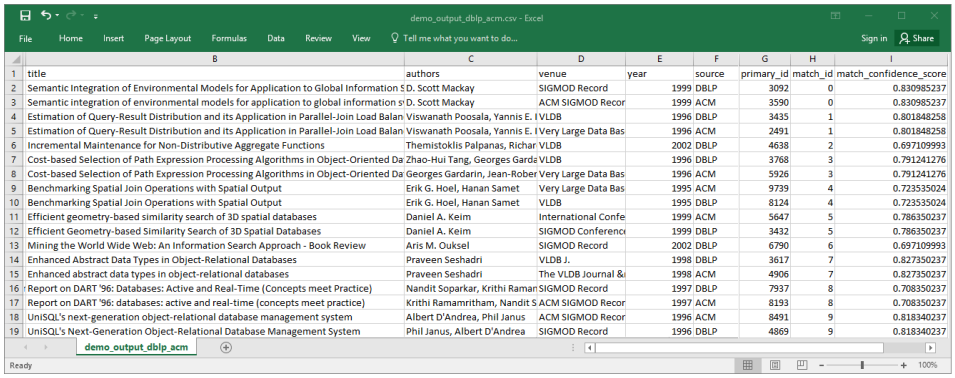
### Fase 6: verifica dei dati di output da Amazon S3

In questa fase si verifica l'output dell'esecuzione del processo nel bucket Amazon S3 scelto al momento dell'aggiunta del processo. È possibile scaricare il file di output sulla propria macchina locale e verificare che i record corrispondenti siano stati identificati.

1. Apri la console Amazon S3 all'indirizzo. https://console.aws.amazon.com/s3/

2. Scaricare il file di output di destinazione del processo `demo-etl-dblp-acm`. Aprire il file in un foglio di calcolo (per aprire il file correttamente, potrebbe essere necessario aggiungere al file l'estensione `.csv`).

   L'immagine seguente mostra un estratto dell'output in Microsoft Excel.



demo_output_dblp_acm.csv - Excel

| | B | C | D | E | F | G | H | I |
|---|---|---|---|---|---|---|---|---|
| 1 | title | authors | venue | year | source | primary_id | match_id | match_confidence_score |
| 2 | Semantic Integration of Environmental Models for Application to Global Information S | D. Scott Mackay | SIGMOD Record | 1999 | DBLP | 3092 | 0 | 0.830985237 |
| 3 | Semantic integration of environmental models for application to global information s | D. Scott Mackay | ACM SIGMOD Recor | 1999 | ACM | 3590 | 0 | 0.830985237 |
| 4 | Estimation of Query-Result Distribution and its Application in Parallel-Join Load Balan | Viswanath Poosala, Yannis E. I | VLDB | 1996 | DBLP | 3435 | 1 | 0.801848258 |
| 5 | Estimation of Query-Result Distribution and its Application in Parallel-Join Load Balan | Viswanath Poosala, Yannis E. I | Very Large Data Bas | 1996 | ACM | 2491 | 1 | 0.801848258 |
| 6 | Incremental Maintenance for Non-Distributive Aggregate Functions | Themistoklis Palpanas, Richar | VLDB | 2002 | DBLP | 4638 | 2 | 0.697109993 |
| 7 | Cost-based Selection of Path Expression Processing Algorithms in Object-Oriented Da | Zhao-Hui Tang, Georges Garda | VLDB | 1996 | DBLP | 3768 | 3 | 0.791241276 |
| 8 | Cost-based Selection of Path Expression Processing Algorithms in Object-Oriented Da | Georges Gardarin, Jean-Rober | Very Large Data Bas | 1996 | ACM | 5926 | 3 | 0.791241276 |
| 9 | Benchmarking Spatial Join Operations with Spatial Output | Erik G. Hoel, Hanan Samet | Very Large Data Bas | 1995 | ACM | 9739 | 4 | 0.723535024 |
| 10 | Benchmarking Spatial Join Operations with Spatial Output | Erik G. Hoel, Hanan Samet | VLDB | 1995 | DBLP | 8124 | 4 | 0.723535024 |
| 11 | Efficient geometry-based similarity search of 3D spatial databases | Daniel A. Keim | International Confe | 1999 | ACM | 5647 | 5 | 0.786350237 |
| 12 | Efficient Geometry-based Similarity Search of 3D Spatial Databases | Daniel A. Keim | SIGMOD Conferenc | 1999 | DBLP | 3432 | 5 | 0.786350237 |
| 13 | Mining the World Wide Web: An Information Search Approach - Book Review | Aris M. Ouksel | SIGMOD Record | 2002 | DBLP | 6790 | 6 | 0.697109993 |
| 14 | Enhanced Abstract Data Types in Object-Relational Databases | Praveen Seshadri | VLDB J. | 1998 | DBLP | 3617 | 7 | 0.827350237 |
| 15 | Enhanced abstract data types in object-relational databases | Praveen Seshadri | The VLDB Journal & | 1998 | ACM | 4906 | 7 | 0.827350237 |
| 16 | Report on DART '96: Databases: Active and Real-Time (Concepts meet Practice) | Nandit Soparkar, Krithi Raman | SIGMOD Record | 1997 | DBLP | 7937 | 8 | 0.708350237 |
| 17 | Report on DART '96: databases: active and real-time (concepts meet practice) | Krithi Ramamritham, Nandit S | ACM SIGMOD Recor | 1997 | ACM | 8193 | 8 | 0.708350237 |
| 18 | UniSQL's next-generation object-relational database management system | Albert D'Andrea, Phil Janus | ACM SIGMOD Recor | 1996 | ACM | 8491 | 9 | 0.818340237 |
| 19 | UniSQL's Next-Generation Object-Relational Database Management System | Phil Janus, Albert D'Andrea | SIGMOD Record | 1996 | DBLP | 4869 | 9 | 0.818340237 |

demo_output_dblp_acm

Ready 100%

   L'origine e la destinazione dei dati contano entrambe 4.911 record. Tuttavia, la trasformazione `Find matches` aggiunge un'altra colonna denominata `match_id` per identificare i record corrispondenti nell'output. Le righe con gli stessi `match_id` sono considerate record corrispondenti. La `match_confidence_score` è un numero compreso tra 0 e 1 che fornisce una stima della qualità delle corrispondenze trovate da `Find matches`.

3. Ordinare il file di output per `match_id` al fine di visualizzare facilmente i record corrispondenti. Confrontare i valori nelle altre colonne per confermare i risultati della trasformazione `Find matches`. Se i risultati non sono soddisfacenti, è possibile continuare ad addestrare la trasformazione aggiungendo ulteriori etichette.

   È anche possibile ordinare i file per un altro campo, ad esempio `title`, per vedere se record con titoli simili presentano lo stesso `match_id`.

## Trovare corrispondenze incrementali

La caratteristica FindMatches permette di identificare registri duplicati o corrispondenti nel set di dati, anche quando i registri non dispongono di un identificatore univoco comune e nessun campo corrisponde esattamente. La versione iniziale di Trova corrispondenze trasforma i registri corrispondenti identificati all'interno di un singolo set di dati. Quando si aggiungono nuovi dati al set, avrai già dovuto unirli con il set di dati pulito esistente e rieseguire la corrispondenza con il set di dati unito completo.

La funzione di corrispondenza incrementale semplifica la corrispondenza con i registri incrementali rispetto ai set di dati corrispondenti esistenti. Supponiamo che desideri abbinare i dati dei potenziali clienti con i set di dati esistenti dei clienti. La funzionalità di corrispondenza incrementale offre la flessibilità necessaria per abbinare centinaia di migliaia di nuovi prospect con un database esistente di prospect e potenziali clienti combinando i risultati in un unico database o tabella. Corrispondendo solo tra i set di dati nuovi ed esistenti, l'ottimizzazione delle corrispondenze incrementali di ricerca riduce i tempi di calcolo, riducendo anche i costi.

L'uso della corrispondenza incrementale è simile a Trova corrispondenze come descritto in Tutorial: creazione di una trasformazione dell'apprendimento automatico con AWS Glue. Questo argomento identifica solo le differenze con la corrispondenza incrementale.

Per ulteriori informazioni, leggi il post del blog su Corrispondenza incrementale dei dati.

Esecuzione di un processo di corrispondenza incrementale

Per la seguente procedura, supponiamo quanto segue:

- Hai eseguito il crawling del set di dati esistente nella tabella first_records. Il set di dati first_records deve essere un set di dati corrispondente o l'output del processo corrispondente.
- Hai creato e addestrato una trasformazione Find matches (Trova corrispondenze) con AWS Glue Glue versione 2.0. Questa è l'unica versione di AWS Glue che supporti le corrispondenze incrementali.
- Il linguaggio ETL è Scala. Si noti che anche Python è supportato.
- Il modello già generato viene chiamato `demo-xform`.

1. Esegui la scansione del set di dati incrementale nella tabella second_records.
2. Nel riquadro di navigazione della console di AWS Glue, scegliere Jobs (Processi).
3. Scegliere Add job (Aggiungi processo) e seguire la procedura guidata per creare un processo ETL Spark con uno script generato. Per le proprietà della trasformazione scegliere i seguenti valori:

   a. Per Name (Nome), scegli demo-etl.

   b. Alla voce IAM role (Ruolo IAM), scegli un ruolo IAM che disponga delle autorizzazioni per accedere ai dati di origine su Amazon S3, ai file di etichettatura dei dati e alle operazioni API di AWS Glue.

   c. Alla voce ETL language (Linguaggio ETL) scegli Scala.

d. Come Script file name (Nome del file di script), scegli demo-etl. Questo è il nome del file dello script Scala.

e. Per Data source (Origine dati), scegli first_records. L'origine dati scelta deve corrispondere allo schema dell'origine dati della trasformazione basata su machine learning.

f. Alla voce Transform type (Tipo di trasformazione), scegliere Find matching records (Individuazione record corrispondenti) per creare un processo che utilizza una trasformazione basata su machine learning.

g. Seleziona l'opzione di corrispondenza incrementale e per Data source (Origine dati) seleziona la tabella denominata second_records.

h. Alla voce Transform (Trasformazione), scegli demo-xform, la trasformazione basata su machine learning utilizzata del processo.

i. Scegli Create tables in your data target (Crea tabelle nella tua destinazione di dati) o Use tables in the catalogo dati and update your data target (Usa tabelle nel catalogo dati e aggiorna la destinazione dati).

4. Scegliere Save job and edit script (Salva processo e modifica script) per visualizzare la pagina dell'editor dello script.

5. Scegliere Run job (Esegui processo) per avviare l'esecuzione del processo.

## Utilizzo FindMatches in un lavoro visivo

Per utilizzare la FindMatchestrasformazione inAWS Glue Studio, puoi utilizzare il nodo Custom Transform che richiama l' FindMatches API. Per ulteriori informazioni su come utilizzare una trasformazione personalizzata, consulta la pagina Creating a custom transformation

> **Note**
>
> Attualmente, l' FindMatches API funziona solo con. `Glue 2.0` Per eseguire un lavoro con la trasformazione personalizzata che richiama l' FindMatches API, assicurati che la AWS Glue versione sia `Glue 2.0` nella scheda Dettagli del lavoro. Se la versione di non lo AWS Glue è`Glue 2.0`, il processo fallirà in fase di esecuzione con il seguente messaggio di errore: «impossibile importare il nome '' da FindMatches 'awsglueml.transforms'».

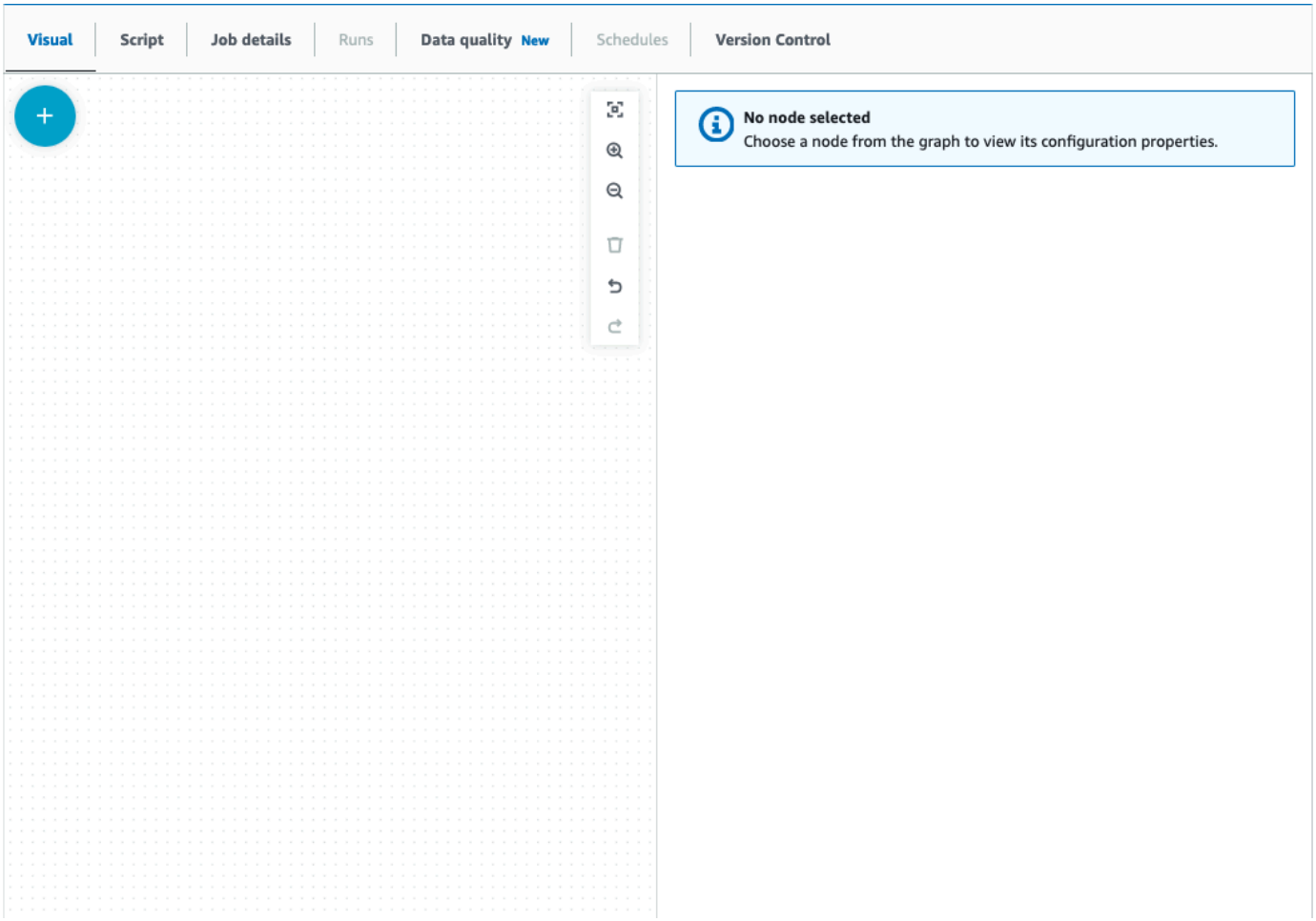
### Prerequisiti

- Per utilizzare la trasformazione Trova corrispondenze, apri la console AWS Glue Studio all'indirizzo https://console.aws.amazon.com/gluestudio/.
- Crea una trasformazione basata su machine learning. Una volta creata, viene generato un transformId. Avrai bisogno di questo ID per i passaggi successivi. Per ulteriori informazioni su come creare una trasformazione basata su machine learning, consulta la pagina Adding and editing machine learning transforms.

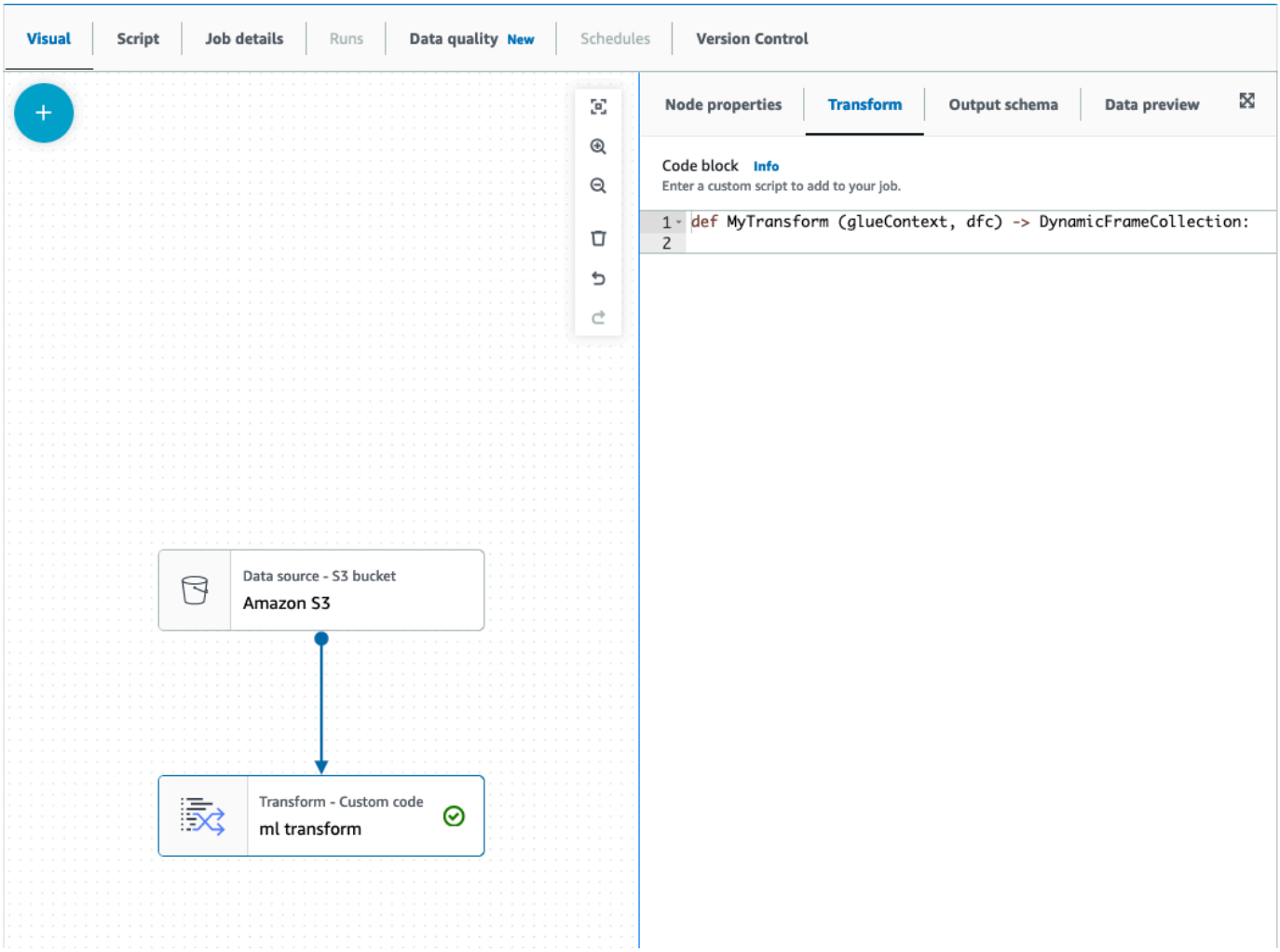
### Aggiungere una trasformazione FindMatches

Per aggiungere una FindMatches trasformazione:

1. Nell'editor dei processi AWS Glue Studio, apri il pannello Risorse facendo clic sul simbolo della croce nell'angolo in alto a sinistra del grafico visivo del processo e scegli un'origine dati selezionando la scheda Dati. Questa è l'origine dati nella quale verificare la presenza di corrispondenze.




2. Scegli il nodo dell'origine dati, quindi apri il pannello Risorse facendo clic sul simbolo della croce nell'angolo in alto a sinistra del grafico visivo del processo e cerca "trasformazione personalizzata". Scegli il nodo Trasformazione personalizzata per aggiungerlo al grafico. La Trasformazione personalizzata è collegata al nodo dell'origine dati. In caso contrario, puoi fare clic sul nodo Trasformazione personalizzata e scegliere la scheda Proprietà del nodo, quindi, in Padri del nodo, scegli l'origine dati.
3. Fai clic sul nodo Trasformazione personalizzata nel grafico visivo, quindi scegli la scheda Proprietà del nodo e assegna un nome alla trasformazione personalizzata. Ti consigliamo di rinominare la trasformazione assegnandole un nome facilmente identificabile nel grafico visivo.
4. Scegli la scheda Trasforma, dove puoi modificare il blocco di codice. Qui è possibile aggiungere il codice per richiamare l' FindMatches API.




Il blocco di codice contiene codice precompilato per aiutarti a iniziare. Sovrascrivi il codice precompilato con il modello seguente. Il modello ha un segnaposto per transformId, che puoi sostituire con il tuo valore.

```
def MyTransform (glueContext, dfc) -> DynamicFrameCollection:
    dynf = dfc.select(list(dfc.keys())[0])
    from awsglueml.transforms import FindMatches
    findmatches = FindMatches.apply(frame = dynf, transformId = "<your id>")
    return(DynamicFrameCollection({"FindMatches": findmatches}, glueContext))
```

5. Fai clic sul nodo Trasformazione personalizzata nel grafico visivo, quindi apri il pannello Risorse facendo clic sul simbolo della croce nell'angolo in alto a sinistra del grafico visivo del processo e cerca "Seleziona dalla raccolta". Non è necessario modificare la selezione predefinita poiché ce n'è solo una DynamicFrame nella raccolta.

6. È possibile continuare ad aggiungere trasformazioni o archiviare il risultato, che ora è arricchito con le colonne aggiuntive delle corrispondenze trovate. Se vuoi fare riferimento a quelle nuove colonne nelle trasformazioni a valle, devi aggiungerle allo schema di output della trasformazione. Il modo più semplice per farlo è scegliere la scheda Anteprima dati e quindi, nella scheda Schema, scegliere "Utilizza schema di anteprima dati".
7. Per personalizzare FindMatches, puoi aggiungere parametri aggiuntivi da passare al metodo 'applica'. Vedi FindMatches classe.

Aggiungere una trasformazione FindMatches incrementale

Nel caso di corrispondenze incrementali, il processo è lo stesso dell'aggiunta di una FindMatches trasformazione con le seguenti differenze:

- Per la trasformazione personalizzata sono necessari due nodi padri anziché un solo nodo.
- Il primo nodo padre dovrebbe essere il set di dati.
- Il secondo nodo padre dovrebbe essere il set di dati incrementale.

  Sostituisci il valore `transformId` con il tuo `transformId` nel blocco di codice del modello:

```
def MyTransform (glueContext, dfc) -> DynamicFrameCollection:
    dfs = list(dfc.values())
    dynf = dfs[0]
    inc_dynf = dfs[1]
    from awsglueml.transforms import FindIncrementalMatches
    findmatches = FindIncrementalMatches.apply(existingFrame = dynf, incrementalFrame
 = inc_dynf,
                                    transformId = "<your id>")
    return(DynamicFrameCollection({"FindMatches": findmatches}, glueContext))
```

- Per i parametri opzionali, vedete FindIncrementalMatches class.

## Esegui la migrazione dei programmi Apache Spark a AWS Glue

Apache Spark è una piattaforma open source per carichi di lavoro di calcolo distribuiti eseguiti su set di dati di grandi dimensioni. AWS Glue sfrutta le capacità di Spark per fornire un'esperienza ottimizzata per ETL. Puoi migrare i programmi Spark per sfruttare AWS Glue le nostre funzionalità.

AWS Glue offre gli stessi miglioramenti delle prestazioni che ti aspetteresti da Apache Spark su Amazon EMR.

## Esegui codice Spark

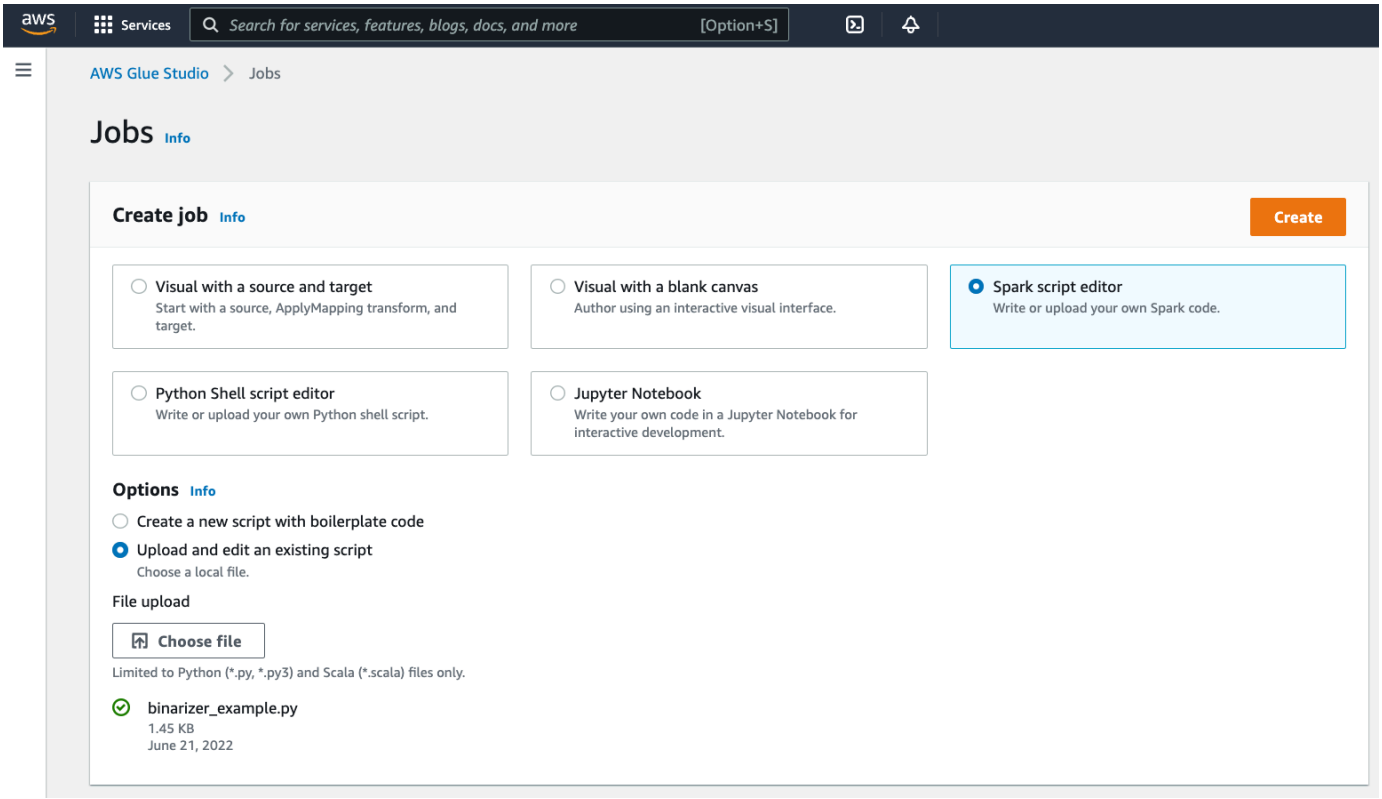
Il codice Spark nativo può essere eseguito in un AWS Glue ambiente pronto all'uso. Gli script sono spesso sviluppati modificando iterativamente un pezzo di codice, un flusso di lavoro adatto per una sessione interattiva. Tuttavia, il codice esistente è più adatto all'esecuzione in un AWS Glue job, il che consente di pianificare e ottenere in modo coerente log e metriche per ogni esecuzione di script. Puoi caricare e modificare uno script esistente tramite la console.

1. Acquisisci la fonte del tuo script. Per questo esempio, si utilizzerà uno script di esempio dal repository Apache Spark. Esempio di Binarizer
2. Nella AWS Glue console, espandi il riquadro di navigazione a sinistra e seleziona ETL > Jobs

   Nel panello Create job (crea processo), seleziona Spark script editor (editor di script di Spark). Apparirà una sezione Options (opzioni). Sotto Options (opzioni) , seleziona Upload and edit an existing script (carica e modifica uno script esistente).

   Apparirà una sezione File upload (caricamento file). Sotto a File upload (caricamento file), fai clic su Choose file (seleziona file). Apparirà il tuo selettore di file di sistema. Accedi al percorso in cui hai effettuato il salvataggio di `binarizer_example.py`, selezionalo e conferma della selezione.

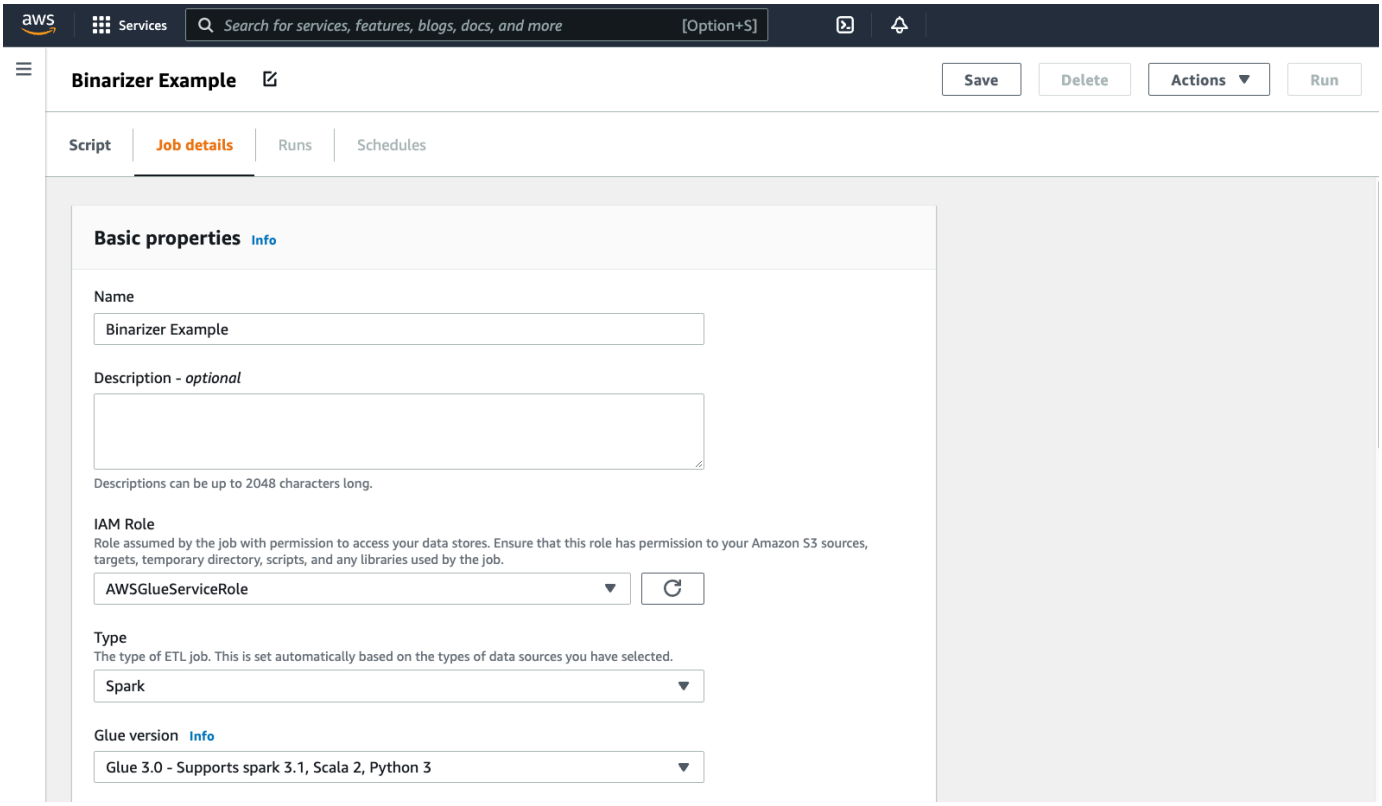
   Verrà visualizzato un pulsante Create (crea) nell'intestazione del pannello Create job (crea processo). Fai clic sul pulsante.



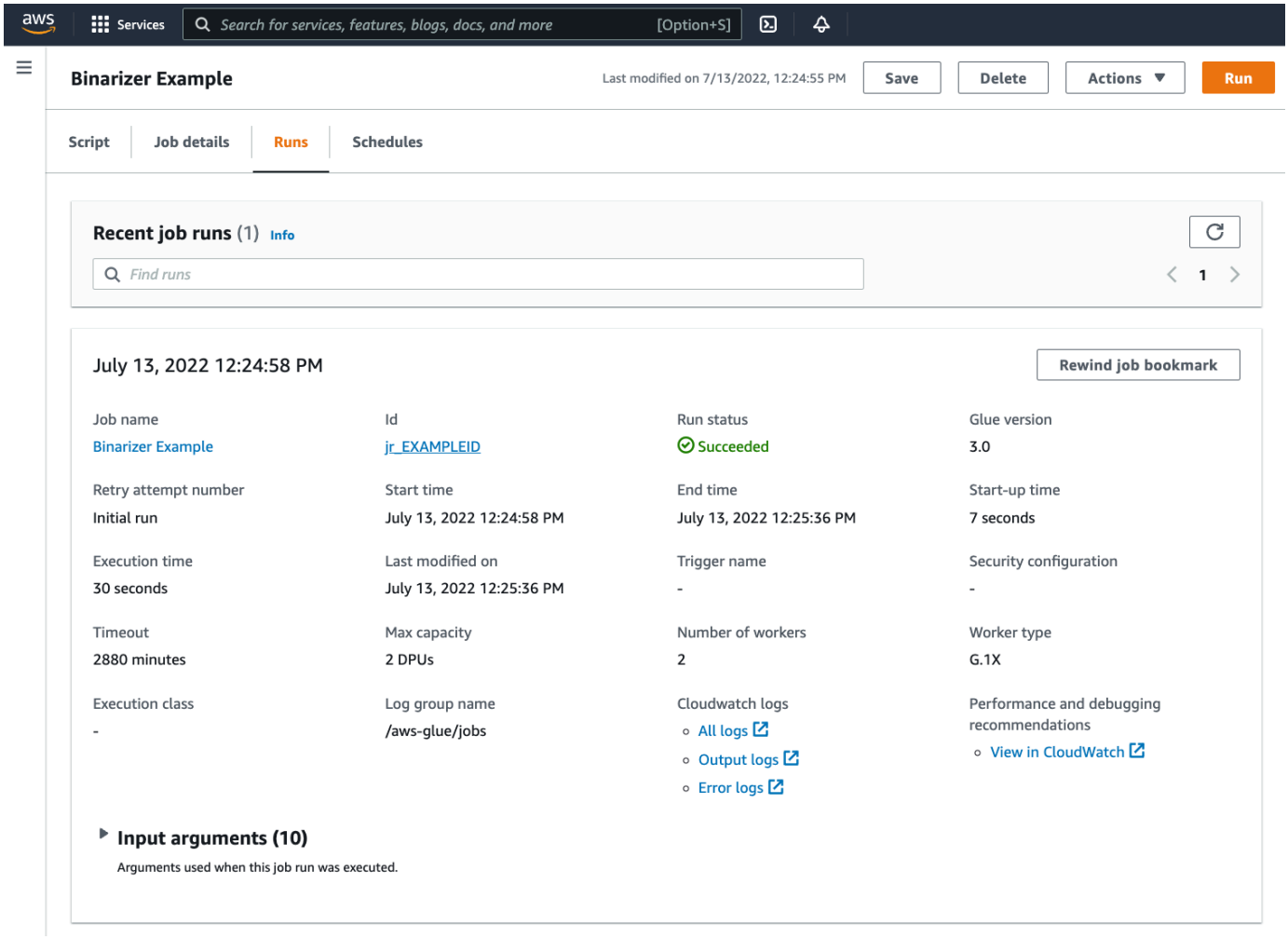

3. Il tuo browser accederà all'editor di script. Nell'intestazione, fai clic sul sulla scheda Job details (dettagli processo). Imposta il Nome e il Ruolo IAM. Per indicazioni sui ruoli AWS Glue IAM, consultathe section called "Impostazione delle autorizzazioni IAM".

   Facoltativo: imposta Requested number of workers (numero di worker richiesti) su 2 e Number of retries (numero di tentativi) su 1. Queste opzioni sono utili quando si eseguono lavori di produzione, ma la loro riduzione semplificherà la tua esperienza durante il test di una funzionalità.

   Nella barra del titolo, fai clic su Save (salva), quindi Run (esegui)

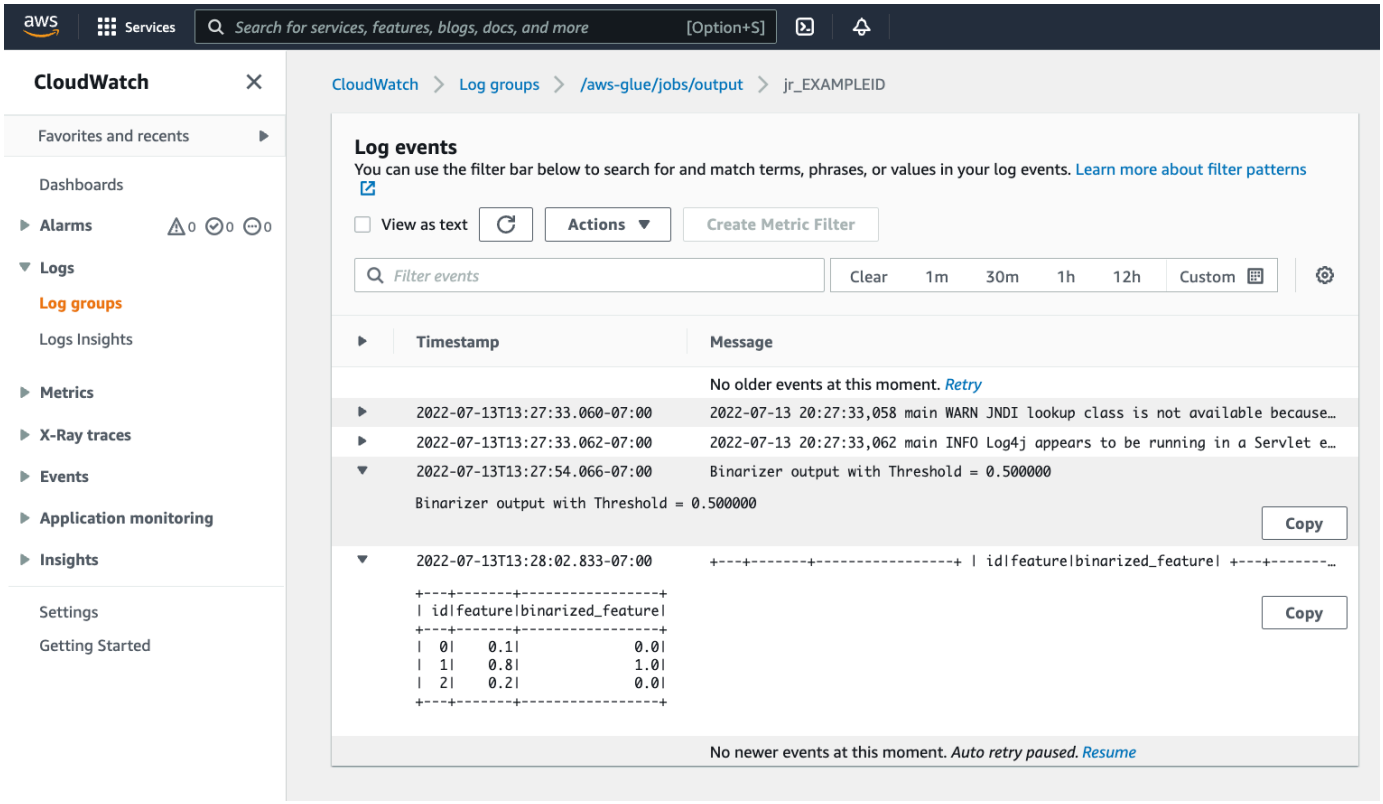



4. Passa alla scheda Runs (esecuzioni). Vedrai un pannello corrispondente all'esecuzione del processo. Attendi alcuni minuti e la pagina dovrebbe essere aggiornata automaticamente per mostrare Succeeded (elaborazione riuscita) sotto a Run status (stato dell'esecuzione).




5. Dovrai esaminare l'output per confermare che lo script Spark è stato eseguito come previsto. Questo script di esempio di Apache Spark dovrebbe scrivere una stringa nel flusso di output. Puoi trovarlo navigando su Output logs (registri di output) sotto a Cloudwatch logs(registri di Cloudwatch) nel pannello dell'esecuzione del processo riuscito. Ricorda l'id di esecuzione del processo, un id generato sotto l'etichetta Id che inizia con `jr_`.

   Si aprirà la CloudWatch console, impostata per visualizzare il contenuto del gruppo di AWS Glue log predefinito`/aws-glue/jobs/output`, filtrato in base al contenuto dei flussi di log per l'ID di esecuzione del processo. Ogni worker avrà generato un flusso di log, mostrato in righe sotto Log streams (Flussi di log). Un worker dovrebbe aver eseguito il codice richiesto. Sarà necessario aprire tutti i flussi di log per identificare il worker corretto. Una volta trovato il worker giusto, dovresti vedere l'output dello script, come mostrato nell'immagine seguente:




## Procedure comuni necessarie per la migrazione dei programmi Spark

### Convalida il supporto della versione di Spark

AWS Glue le versioni di rilascio definiscono la versione di Apache Spark e Python disponibile per il job. AWS Glue Puoi trovare AWS Glue le nostre versioni e il loro supporto all'indirizzo. the section called “AWS Glue versioni” Potrebbe essere necessario aggiornare il programma Spark per renderlo compatibile con una versione più recente di Spark per accedere a determinate caratteristiche di AWS Glue .

### Includi librerie di terze parti

Molti programmi Spark esistenti avranno dipendenze, sia da artefatti privati che pubblici. AWS Glue supporta le dipendenze in stile JAR per processi Scala così come le dipendenze Wheel e pure-Python per i processi Python.

Python : per informazioni sulle dipendenze Python, consulta the section called “Librerie Python”

Le dipendenze Python comuni sono fornite nell' AWS Glue ambiente, inclusa la libreria Pandas comunemente richiesta. Le dipendenze sono incluse in AWS Glue versione 2.0+. Per ulteriori

informazioni sui moduli disponibili, consulta the section called "Moduli Python già forniti in Glue AWS". Se è necessario fornire un processo con una versione diversa da quella di una dipendenza inclusa per impostazione predefinita, è possibile utilizzare `--additional-python-modules`. Per informazioni su questi argomenti, consulta the section called "Parametri Job".

È possibile fornire dipendenze Python aggiuntive con l'argomento del processo `--extra-py-files`. Se state migrando un job da un programma Spark, questo parametro è una buona opzione perché è funzionalmente equivalente al `--py-files` flag in ed è soggetto alle stesse PySpark limitazioni. Per ulteriori informazioni sul parametro `--extra-py-files`, consulta the section called "Inclusione di file Python con funzionalità native PySpark".

Per i nuovi processi, è possibile gestire le dipendenze Python con l'argomento del processo `--additional-python-modules`. L'utilizzo di questo argomento consente un'esperienza di gestione delle dipendenze più approfondita. Questo parametro supporta le dipendenze dello stile Wheel, incluse quelle con associazioni di codice native compatibili con Amazon Linux 2.

Scala

È possibile fornire dipendenze Scala aggiuntive con l'argomento del processo `--extra-jars`. Le dipendenze devono essere ospitate in Amazon S3 e il valore dell'argomento deve essere un elenco delimitato da virgole di percorsi Amazon S3 senza spazi. Potresti trovare più semplice gestire la configurazione riorganizzando le dipendenze prima di ospitarle e configurarle. AWS Glue Le dipendenze JAR contengono bytecode Java, che può essere generato da qualsiasi linguaggio JVM. È possibile utilizzare altri linguaggi JVM, come Java, per scrivere dipendenze personalizzate.

Gestisci le credenziali dell'origine dei dati.

I programmi Spark esistenti possono avere una configurazione complessa o personalizzata per estrarre dati dalle loro origini dati. I flussi di autenticazione delle origini dati comuni sono supportati dalle connessioni. AWS Glue Per ulteriori informazioni sulle connessioni AWS Glue , consulta Connessione ai dati.

AWS Glue le connessioni facilitano la connessione del Job a una varietà di tipi di archivi dati in due modi principali: tramite chiamate di metodo alle nostre librerie e impostando la connessione di rete aggiuntiva nella AWS console. Puoi anche chiamare l' AWS SDK dall'interno del tuo job per recuperare informazioni da una connessione.

Chiamate di metodo: AWS Glue le connessioni sono strettamente integrate con AWS Glue Data Catalog, un servizio che consente di gestire le informazioni sui set di dati, e i metodi disponibili per interagire con le connessioni lo riflettono. AWS Glue Se disponi di una configurazione di

autenticazione esistente che desideri riutilizzare, per le connessioni JDBC, puoi accedere alla configurazione della AWS Glue connessione tramite il metodo su. `extract_jdbc_conf` `GlueContext` Per ulteriori informazioni, consulta the section called "extract_jdbc_conf"

Configurazione della console: i AWS Glue job utilizzano AWS Glue le connessioni associate per configurare le connessioni alle sottoreti Amazon VPC. Se gestisci direttamente i tuoi materiali di sicurezza, potrebbe essere necessario fornire un `NETWORK` tipo «Connessione di rete aggiuntiva» nella AWS console per configurare il routing. Per ulteriori informazioni sull'API di connessione AWS Glue , consulta the section called "Connessioni".

Se i tuoi programmi Spark hanno un flusso di autenticazione personalizzato o non comune, potresti dover gestire i materiali di sicurezza in modo pratico. Se AWS Glue le connessioni non sembrano adatte, puoi ospitare in modo sicuro i materiali di sicurezza in Secrets Manager e accedervi tramite boto3 o AWS SDK, forniti nel job.

Configurazione di Apache Spark

Le migrazioni complesse spesso alterano la configurazione di Spark per adattarsi ai loro carichi di lavoro. Le versioni moderne di Apache Spark consentono di impostare la configurazione di runtime con. `SparkSession` AWS Glue Sono disponibili oltre 3.0 job`SparkSession`, che possono essere modificati per impostare la configurazione di runtime. Configurazione di Apache Spark. L'ottimizzazione di Spark è complessa e AWS Glue non garantisce il supporto per l'impostazione di tutte le configurazioni di Spark. Se la migrazione richiede una configurazione sostanziale a livello di Spark, contatta l'assistenza.

Impostazione della configurazione personalizzata

I programmi Spark migrati possono essere progettati per accettare configurazioni personalizzate. AWS Glue consente di impostare la configurazione a livello di job e job run, tramite gli argomenti del job. Per informazioni su questi argomenti, consulta the section called "Parametri Job". Puoi accedere agli argomenti relativi al lavoro nel contesto di un lavoro tramite le nostre librerie. AWS Glue fornisce una funzione di utilità per fornire una visualizzazione coerente tra gli argomenti impostati nel job e gli argomenti impostati durante l'esecuzione del job. Consulta the section called "getResolvedOptions" in Python e the section called "GlueArgParser" in Scala.

Migrazione del codice Java

Come spiegato in the section called "Librerie di terze parti", le dipendenze possono contenere classi generate da linguaggi JVM, come Java o Scala. Le dipendenze possono includere un metodo `main`. È possibile utilizzare un `main` metodo in una dipendenza come punto di ingresso per un AWS Glue

processo Scala. Questo permette di scrivere il proprio metodo `main` in Java o riutilizzare un metodo `main` elaborato in pacchetti secondo gli standard della tua libreria.

Per utilizzare un metodo `main` da una dipendenza, esegui le seguenti operazioni: cancella il contenuto del riquadro di modifica fornendo il valore predefinito dell'oggetto `GlueApp`. Fornisci il nome completo di una classe in una dipendenza come argomento di processo con la chiave `--class`. A questo punto dovresti essere in grado di attivare un'esecuzione del processo.

Non è possibile configurare l'ordine o la struttura degli argomenti passati al AWS Glue metodo. `main` Se il codice esistente deve leggere la configurazione impostata AWS Glue, ciò potrebbe causare incompatibilità con il codice precedente. Se utilizzi `getResolvedOptions`, non avrai nemmeno un luogo ideale per chiamare questo metodo. Prendi in considerazione la possibilità di invocare la tua dipendenza direttamente da un metodo principale generato da AWS Glue. Il seguente script AWS Glue ETL ne mostra un esempio.

```
import com.amazonaws.services.glue.util.GlueArgParser

object GlueApp {
  def main(sysArgs: Array[String]) {
    val args = GlueArgParser.getResolvedOptions(sysArgs, Seq("JOB_NAME").toArray)

    // Invoke static method from JAR. Pass some sample arguments as a String[], one
 defined inline and one taken from the job arguments, using getResolvedOptions
    com.mycompany.myproject.MyClass.myStaticPublicMethod(Array("string parameter1",
 args("JOB_NAME")))

    // Alternatively, invoke a non-static public method.
    (new com.mycompany.myproject.MyClass).someMethod()
  }
}
```

# Lavorare con Ray Jobs in AWS Glue

> **Important**
>
> AWS Glue for Ray non è più aperto a nuovi clienti. I clienti esistenti possono continuare a utilizzare il servizio normalmente. Per ulteriori informazioni, consulta AWS Glue Ray End of Support.

Questa sezione fornisce informazioni sull'utilizzo AWS Glue per i lavori Ray. Per ulteriori informazioni sulla scrittura di script AWS Glue per Ray, consulta la the section called "AWS Glue per Ray" sezione.

Argomenti


- Guida introduttiva AWS Glue per Ray
- Ambienti di runtime Ray supportati
- Contabilità per i worker nei processi Ray
- Utilizzo dei parametri di processo nei processi Ray
- Monitoraggio dei processi di Ray con i parametri


## Guida introduttiva AWS Glue per Ray

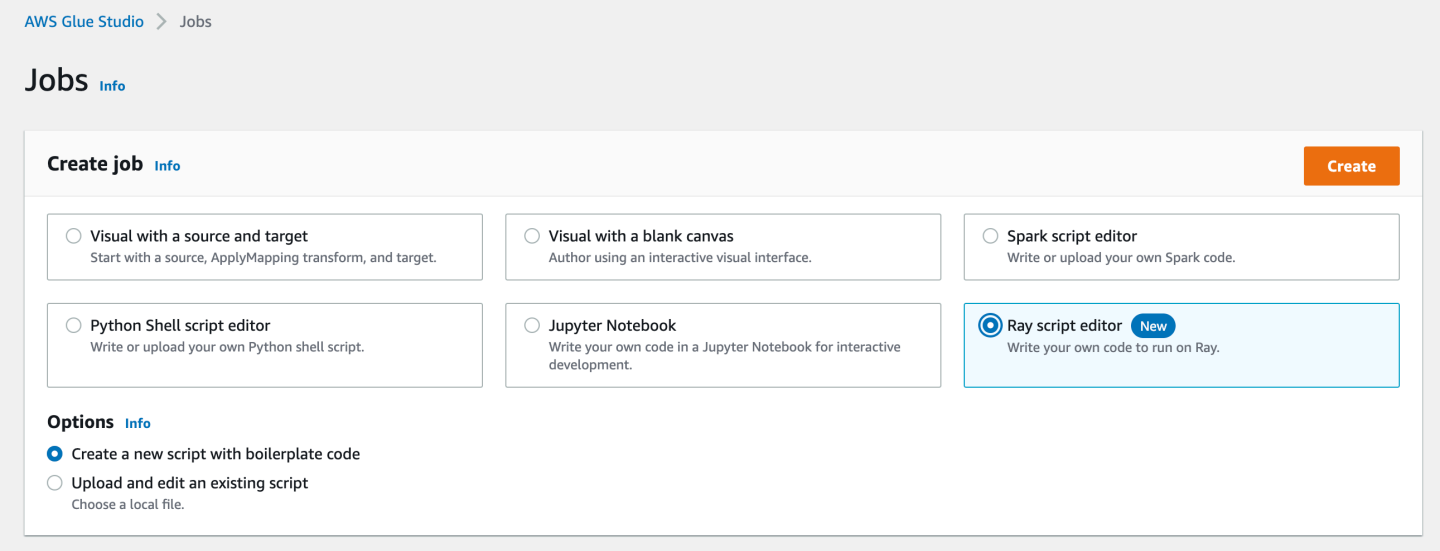
Per lavorare con AWS Glue for Ray, usi gli stessi AWS Glue job e le stesse sessioni interattive che usi AWS Glue per Spark. AWS Glue i job sono progettati per eseguire lo stesso script con una cadenza ricorrente, mentre le sessioni interattive sono progettate per consentire di eseguire frammenti di codice in sequenza sulle stesse risorse a cui sono state assegnate.

AWS Glue ETL e Ray sono fondamentalmente diversi, quindi nello script è possibile accedere a diversi strumenti, funzionalità e configurazioni. Essendo un nuovo framework di calcolo gestito da AWS Glue, Ray ha un'architettura diversa e utilizza un vocabolario diverso per descrivere ciò che fa. Per ulteriori informazioni, consulta Whitepaper sull'architettura nella documentazione di Ray.

> **Note**
>
> AWS Glue for Ray è disponibile negli Stati Uniti orientali (Virginia settentrionale), Stati Uniti orientali (Ohio), Stati Uniti occidentali (Oregon), Asia Pacifico (Tokyo) ed Europa (Irlanda).

### Ray Jobs nella console AWS Glue Studio

Nella pagina Jobs della AWS Glue Studio console, puoi selezionare una nuova opzione quando crei un lavoro in AWS Glue Studio: Ray script editor. Scegli questa opzione per creare un processo Ray nella console. Per ulteriori informazioni sui processi e sul relativo utilizzo, consulta la pagina Creazione di processi ETL visivi.




## Lavori Ray nell'SDK AWS CLI e

I lavori Ray in AWS CLI uso utilizzano le stesse azioni e parametri SDK degli altri job. AWS Glue for Ray introduce nuovi valori per determinati parametri. Per ulteriori informazioni sull'API Processi, consulta la pagina the section called "Jobs".

# Ambienti di runtime Ray supportati

Nei processi Spark, `GlueVersion` determina le versioni di Apache Spark e Python disponibili in un processo AWS Glue per Spark. La versione di Python indica la versione supportata per i processi di tipo Spark. Questo non è il modo in cui sono configurati gli ambienti di runtime Ray.

Per i processi Ray, è necessario impostare `GlueVersion` su `4.0` o superiore. Tuttavia, le versioni di Ray, Python e le librerie aggiuntive disponibili nel processo Ray sono determinate dal campo `Runtime` nella definizione del processo.

L'ambiente di runtime `Ray2.4` sarà disponibile per un minimo di 6 mesi dopo il rilascio. Di pari passo con la rapida evoluzione di Ray, potrai incorporare aggiornamenti e miglioramenti di Ray nelle future versioni dell'ambiente di runtime.

Valori validi: `Ray2.4`

| Valore di runtime | Versioni di Ray e Python |
| --- | --- |
| `Ray2.4` (per AWS Glue 4.0 e versioni successive) | Ray 2.4.0 |

| Valore di runtime | Versioni di Ray e Python |
| --- | --- |
| | Python 3.9 |

Informazioni aggiuntive

- Per le note di rilascio che accompagnano AWS Glue le versioni di Ray, vedithe section called "AWS Glue versioni".
- Per le librerie Python disponibili in un ambiente di runtime, consulta la pagina the section called "Moduli disponibili con i processi Ray".

## Contabilità per i worker nei processi Ray

AWS Glue esegue lavori Ray su nuovi tipi di worker EC2 basati su Graviton, che sono disponibili solo per i lavori Ray. Per fornire in modo appropriato questi worker per i carichi di lavoro per cui Ray è progettato, forniamo un rapporto diverso tra risorse di calcolo e risorse di memoria rispetto alla maggior parte dei worker. Per tenere conto di queste risorse, utilizziamo l'unità di elaborazione dati ottimizzata per la memoria (M-DPU) anziché l'unità di elaborazione dati standard (DPU).

- Una M-DPU corrisponde a 4 v CPUs e 32 GB di memoria.
- Una DPU corrisponde a 4 v CPUs e 16 GB di memoria. DPUs vengono utilizzati per contabilizzare le risorse relative AWS Glue ai job Spark e ai lavoratori corrispondenti.

I processi Ray attualmente hanno accesso a un tipo di worker, `Z.2X`. Il `Z.2X` worker esegue la mappatura su 2 M- DPUs (8 vCPUs, 64 GB di memoria) e dispone di 128 GB di spazio su disco. Una macchina `Z.2X` fornisce 8 worker Ray (uno per vCPU).

Il numero di M- DPUs che è possibile utilizzare contemporaneamente in un account è soggetto a una quota di servizio. Per ulteriori informazioni sui limiti del tuo AWS Glue account, consulta AWS Glue endpoint e quote.

Nella definizione del processo si specifica il numero di nodi worker disponibili per un processo Ray con `--number-of-workers (NumberOfWorkers)`. Per ulteriori informazioni sui valori di Ray nell'API Processi, consulta la pagina the section called "Jobs".

È possibile specificare ulteriormente un numero minimo di worker che un processo Ray deve allocare con il parametro di processo `--min-workers`. Per ulteriori informazioni sui parametri di processo, consulta the section called "Documentazione di riferimento".

## Utilizzo dei parametri di processo nei processi Ray

> **Important**
>
> AWS Glue for Ray non è più aperto a nuovi clienti. I clienti esistenti possono continuare a utilizzare il servizio normalmente. Per ulteriori informazioni, consulta AWS Glue Ray End of Support.

Gli argomenti per i lavori AWS Glue Ray vengono impostati nello stesso modo in cui si impostano gli argomenti AWS Glue per i lavori Spark. Per ulteriori informazioni sull' AWS Glue API, consultathe section called "Jobs". È possibile configurare i lavori AWS Glue Ray con diversi argomenti, elencati in questo riferimento. È anche possibile fornire i propri argomenti.

È possibile configurare un processo tramite la console, nella scheda Job details (Dettagli del processo), sotto l'intestazione Job Parameters (Parametri del processo). È inoltre possibile configurare un lavoro tramite AWS CLI `DefaultArguments` impostando un lavoro o impostando l'`Arguments`esecuzione di un lavoro. Gli argomenti e i parametri dei processi predefiniti resteranno gli stessi nel processo anche dopo più esecuzioni.

Ad esempio, la seguente è la sintassi per l'esecuzione di un processo utilizzando `--arguments` per impostare un parametro speciale.

```
$ aws glue start-job-run --job-name "CSV to CSV" --arguments='--scriptLocation="s3://
my_glue/libraries/test_lib.py",--test-environment="true"'
```

Dopo aver impostato gli argomenti, è possibile accedere ai parametri di processo dall'interno del processo Ray tramite le variabili di ambiente. Questo ti consente di configurare il processo per ogni esecuzione. Il nome della variabile di ambiente sarà il nome dell'argomento del processo senza il prefisso `--`.

Ad esempio, nell'esempio precedente, i nomi delle variabili sarebbero `scriptLocation` e `test-environment`. Pertanto, l'argomento dovrebbe essere recuperato tramite i metodi disponibili nella libreria standard: `test_environment = os.environ.get('test-environment')`. Per

ulteriori informazioni sull'accesso alle variabili di ambiente con Python, consulta la sezione OS module nella documentazione di Python.

### Configurazione delle modalità di generazione dei log da parte dei processi Ray

Per impostazione predefinita, i lavori Ray generano log e metriche che vengono inviati ad Amazon CloudWatch S3. È possibile utilizzare il parametro `--logging_configuration` per modificare la modalità di generazione dei log; attualmente è possibile utilizzarlo per impedire ai processi Ray di generare vari tipi di log. Questo parametro accetta un oggetto JSON, le cui chiavi corrispondono a quelle logs/behaviors che desideri modificare. Supporta le seguenti chiavi:

- `CLOUDWATCH_METRICS`— Configura serie di CloudWatch metriche che possono essere utilizzate per visualizzare lo stato del lavoro. Per ulteriori informazioni sui parametri, consulta the section called "Parametri dei processi Ray".
- `CLOUDWATCH_LOGS`— Configura i CloudWatch log che forniscono dettagli a livello di applicazione Ray sullo stato di esecuzione del job. Per ulteriori informazioni sui log, consulta the section called "Risoluzione degli errori relativi ai processi Ray".
- `S3`— Configura ciò che viene AWS Glue scritto su Amazon S3, principalmente informazioni CloudWatch simili nei log ma come file anziché come flussi di log.

Per disabilitare un comportamento di registrazione di Ray, fornisci il valore `{\"IS_ENABLED\": \"False\"}`. Ad esempio, per disabilitare CloudWatch metriche e CloudWatch log, fornisci la seguente configurazione:

```
"--logging_configuration": "{\"CLOUDWATCH_METRICS\": {\"IS_ENABLED\": \"False\"},
 \"CLOUDWATCH_LOGS\": {\"IS_ENABLED\": \"False\"}}"
```

### Documentazione di riferimento

I processi Ray riconoscono i seguenti nomi di argomenti che possono essere utilizzati per configurare l'ambiente di script per i processi Ray e le esecuzioni di processo:

- `--logging_configuration`: viene utilizzato per interrompere la generazione di vari log creati dai processi Ray. Questi log vengono generati per impostazione predefinita su tutti i processi Ray. Formato: oggetto JSON con escape di stringhe. Per ulteriori informazioni, consulta the section called "Configurazione delle modalità di generazione dei log da parte dei processi Ray".
- `--min-workers`: il numero minimo di nodi worker allocati a un processo Ray. Un nodo worker può eseguire più repliche, una per CPU virtuale. Formato: numero intero. Minimo: 0 Massimo:

valore specificato in `--number-of-workers (NumberOfWorkers)` nella definizione di processo. Per ulteriori informazioni su come allocare adeguatamente i nodi worker, consulta la pagina the section called "Contabilità per i worker nei processi Ray".

- `--object_spilling_config`— AWS Glue for Ray supporta l'utilizzo di Amazon S3 per estendere lo spazio disponibile per l'object store di Ray. Per abilitare questo comportamento, è possibile fornire a Ray un oggetto di configurazione JSON per il riversamento di oggetti con questo parametro. Per ulteriori informazioni sulla configurazione del riversamento di oggetti in Ray, consulta la pagina Object Spilling nella documentazione di Ray. Formato: oggetto JSON.

  AWS Glue for Ray supporta solo la fuoriuscita su disco o la trasmissione su Amazon S3 contemporaneamente. È possibile fornire più punti di riversamento, purché rispettino questa limitazione. In caso di riversamento su Amazon S3, sarà necessario aggiungere al processo anche le autorizzazioni IAM per questo bucket.

  Quando si fornisce un oggetto JSON come configurazione con la CLI, è necessario fornirlo come stringa, specificando l'oggetto JSON con escape di stringa. Ad esempio, un valore di stringa per il riversamento su un percorso Amazon S3 apparirebbe come: `"{\"type\": \"smart_open\", \"params\": {\"uri\":\"s3path\"}}"`. In AWS Glue Studio, fornisci questo parametro come oggetto JSON senza formattazioni aggiuntive.

- `--object_store_memory_head`: la memoria allocata all'archivio di oggetti Plasma sul nodo principale di Ray. Questa istanza esegue i servizi di gestione dei cluster e le repliche dei worker. Il valore rappresenta una percentuale di memoria libera sull'istanza dopo un avvio a caldo. Questo parametro viene utilizzato per ottimizzare i carichi di lavoro che richiedono un uso intensivo della memoria: i valori predefiniti sono accettabili per la maggior parte dei casi d'uso. Formato: numero intero positivo. Minimo: 1. Massimo: 100

  Per ulteriori informazioni su Plasma, consulta L'archivio oggetti in memoria di Plasma nella documentazione di Ray.

- `--object_store_memory_worker`: la memoria allocata all'archivio di oggetti Plasma sui nodi worker di Ray. Queste istanze eseguono solo repliche worker. Il valore rappresenta una percentuale di memoria libera sull'istanza dopo un avvio a caldo. Questo parametro viene utilizzato per ottimizzare i carichi di lavoro che richiedono un uso intensivo della memoria: i valori predefiniti sono accettabili per la maggior parte dei casi d'uso. Formato: numero intero positivo. Minimo: 1. Massimo: 100

  Per ulteriori informazioni su Plasma, consulta L'archivio oggetti in memoria di Plasma nella documentazione di Ray.

- `--pip-install`: un set di pacchetti Python da installare. È possibile installare pacchetti da PyPI utilizzando questo argomento. Formato: elenco delimitato da virgole.

  Una voce del pacchetto PyPI sarà nel formato `package==version`, con il nome e la versione di PyPI del pacchetto di destinazione. Le voci usano la corrispondenza della versione Python per abbinare il pacchetto e la versione, come ==, non il singolo uguale a =. Esistono altri operatori di corrispondenza delle versioni. Per ulteriori informazioni, consulta PEP 440 sul sito Web di Python. È inoltre possibile fornire moduli personalizzati con `--s3-py-modules`.

- `--s3-py-modules`: un set di percorsi Amazon S3 che ospitano le distribuzioni di moduli Python. Formato: elenco delimitato da virgole.

  Puoi utilizzarlo per distribuire i tuoi moduli al tuo processo di Ray. I moduli possono essere forniti anche da PyPI con `--pip-install`. A differenza di AWS Glue ETL, i moduli personalizzati non vengono configurati tramite pip, ma vengono passati a Ray per la distribuzione. Per ulteriori informazioni, consulta the section called "Moduli Python aggiuntivi per i processi Ray".

- `--working-dir`: un percorso verso un file .zip ospitato in Amazon S3 che contiene file da distribuire a tutti i nodi che eseguono il processo Ray. Formato: stringa. Per ulteriori informazioni, consulta the section called "Fornitura di file al processo Ray".

## Monitoraggio dei processi di Ray con i parametri

> **Important**
>
> AWS Glue for Ray non è più aperto a nuovi clienti. I clienti esistenti possono continuare a utilizzare il servizio normalmente. Per ulteriori informazioni, consulta AWS Glue Ray End of Support.

Puoi monitorare i lavori Ray utilizzando AWS Glue Studio e Amazon CloudWatch. CloudWatch raccoglie ed elabora le metriche non elaborate AWS Glue con Ray, che le rende disponibili per l'analisi. Queste metriche vengono visualizzate nella AWS Glue Studio console, in modo da poter monitorare il lavoro mentre viene eseguito.

Per una panoramica generale su come monitorare AWS Glue, consulta. the section called "Utilizzo delle metriche CloudWatch " Per una panoramica generale su come utilizzare le CloudWatch metriche pubblicate da AWS Glue, consultathe section called "Monitoraggio con CloudWatch".

## Monitoraggio dei job di Ray nella console AWS Glue

Nella pagina dei dettagli dell'esecuzione di un lavoro, sotto la sezione Dettagli dell'esecuzione, puoi visualizzare grafici aggregati predefiniti che visualizzano le metriche dei job disponibili. AWS Glue Studio invia le metriche dei job a per ogni job eseguito. CloudWatch Con questi, è possibile creare un profilo del cluster e delle attività, nonché accedere a informazioni dettagliate su ciascun nodo.

Per ulteriori informazioni sui grafici di parametri disponibili, consulta the section called "Visualizzazione delle Amazon CloudWatch metriche per l'esecuzione di un job Ray".

## Panoramica delle metriche di Ray Jobs in CloudWatch

Pubblichiamo le metriche Ray quando è abilitato il monitoraggio dettagliato. CloudWatch Le metriche vengono pubblicate nel namespace. `Glue/Ray` CloudWatch

- Parametri dell'istanza

  Pubblichiamo i parametri sull'utilizzo della CPU, della memoria e del disco delle istanze assegnate a un processo. Questi parametri sono identificati da funzionalità quali `ExecutorId`, `ExecutorType` e `host`. Queste metriche sono un sottoinsieme delle metriche standard degli agenti Linux. CloudWatch Puoi trovare informazioni sui nomi e le funzionalità delle metriche nella documentazione. CloudWatch Per ulteriori informazioni, consulta Metriche raccolte dall' CloudWatch agente.
- Parametri del cluster Ray

  Inoltriamo i parametri dai processi Ray che eseguono lo script a questo spazio dei nomi, quindi ti trasmettiamo quelli più rilevanti per te. I parametri disponibili potrebbero differire in base alla versione di Ray. Per ulteriori informazioni sulla versione di Ray utilizzata dal processo, consulta the section called "AWS Glue versioni".

  Ray raccoglie i parametri a livello di istanza. Inoltre, fornisce parametri per le attività e il cluster. Per ulteriori informazioni sulla strategia dei parametri di base di Ray, consulta la pagina Metrics nella documentazione di Ray.

> **Note**
>
> Non pubblichiamo i parametri Ray nello spazio dei nomi `Glue/Job Metrics/`; questo viene utilizzato solo per i processi AWS Glue ETL.

# Configurazione delle proprietà del processo per i processi shell di Python in AWS Glue

Puoi utilizzare un processo shell di Python per eseguire gli script di Python come una shell in AWS Glue. Con un processo shell di Python puoi eseguire script compatibili con Python 3.6 o Python 3.9.

> **Note**
>
> Il supporto per Pyshell v3.6 terminerà il 1° marzo 2026. Per migrare i carichi di lavoro, consulta Migrare dai job della shell Python AWS Glue. Se desideri continuare con la shell di Python v3.9, consulta Migrazione dalla shell 3.6 di Python alla 3.9.

Argomenti


- Limitazioni
- Ambiente di esecuzione
- Definire le proprietà del processo per i processi shell di Python
- Librerie supportate dai processi shell di Python
- Fornire la propria libreria Python
- Utilizzabile AWS CloudFormation con i lavori della shell Python in AWS Glue
- Migrazione dalla shell 3.6 di Python alla 3.9
- Migrazione dai job della AWS shell Glue Python


## Limitazioni

Tieni presente le seguenti limitazioni dei processi shell Python:

- Non è possibile utilizzare i segnalibri nei processi di shell di Python.
- In Python 3.9 e versioni successive non puoi pacchettizzare alcuna libreria Python come file `.egg`. Utilizza invece `.whl`.
- Per via di una limitazione sulle copie temporanee dei dati di S3, l'opzione `--extra-files` non può essere utilizzata.

## Ambiente di esecuzione

I job della shell Python vengono eseguiti in un ambiente di esecuzione gestito che fornisce l'accesso all'archiviazione locale per l'elaborazione temporanea dei dati:

Archiviazione temporanea locale

La `/tmp` directory è disponibile per l'archiviazione temporanea durante l'esecuzione del lavoro. Questa directory fornisce circa 14 GiB di spazio libero che è possibile utilizzare per:

- Elaborazione di file temporanei
- Archiviazione intermedia dei dati
- Memorizzazione nella cache di piccoli set di dati

> **Note**
>
> La `/tmp` directory è effimera e viene ripulita dopo il completamento del lavoro. Non utilizzarlo per l'archiviazione persistente di dati importanti.

## Definire le proprietà del processo per i processi shell di Python

Queste sezioni descrivono la definizione delle proprietà del lavoro nella AWS Glue Studio o l'utilizzo della AWS CLI.

### AWS Glue Studio

Quando definisci il processo della shell Python in AWS Glue Studio, fornisci alcune delle seguenti proprietà:

Ruolo IAM

Specificate il ruolo AWS Identity and Access Management (IAM) utilizzato per l'autorizzazione alle risorse utilizzate per eseguire il processo e accedere agli archivi dati. Per ulteriori informazioni sulle autorizzazioni per l'esecuzione di processi in AWS Glue, consulta Gestione delle identità e degli accessi per AWS Glue.

Tipo

Scegli Python shell (Shell di Python) per eseguire uno script Python con il comando di processo denominato `pythonshell`.

Versione di Python

Scegli la versione di Python. La versione predefinita è Python 3.9. Le versioni valide sono Python 3.6 e Python 3.9.

Carica librerie di analisi comuni (scelta consigliata)

Scegli questa opzione per includere le librerie comuni per Python 3.9 nella shell Python.

Se le tue librerie sono personalizzate o sono in conflitto con quelle preinstallate, puoi scegliere di non installare librerie comuni. Tuttavia, oltre alle librerie comuni puoi installare librerie aggiuntive.

Quando selezioni questa opzione, l'opzione `library-set` è impostata su `analytics`. Quando deselezioni questa opzione, l'opzione `library-set` è impostata su `none`.

Nome file e percorso dello script

Il codice nello script definisce la logica procedurale del processo. Puoi fornire il nome e la posizione dello script in Amazon Simple Storage Service (Amazon S3). Conferma che non esiste un file con lo stesso nome della directory di script nel percorso. Per ulteriori informazioni sull'uso degli script, consulta AWS Glue guida alla programmazione.

Script

Il codice nello script definisce la logica procedurale del processo. Puoi codificare lo script in Python 3.6 o Python 3.9. Puoi modificare uno script in AWS Glue Studio.

Unità di elaborazione dati (DPU)

Il numero massimo di unità di elaborazione AWS Glue dati (DPUs) che possono essere allocate durante l'esecuzione di questo processo. Una DPU è una misura relativa della potenza di elaborazione costituita da 4 V di capacità CPUs di elaborazione e 16 GB di memoria. Per ulteriori informazioni, consultare Prezzi di AWS Glue.

Puoi impostare il valore su 0,0625 o 1. Il valore predefinito è 0.0625. In entrambi i casi, il disco locale per l'istanza sarà di 20 GB.

## CLI

Puoi anche creare un job della shell Python usando AWS CLI, come nell'esempio seguente.

```
aws glue create-job --name python-job-cli --role Glue_DefaultRole
```

```
    --command '{"Name" :  "pythonshell", "PythonVersion": "3.9", "ScriptLocation" :
"s3://amzn-s3-demo-bucket/scriptname.py"}'
    --max-capacity 0.0625
```

> **Note**
>
> Non devi specificare la versione di AWS Glue poiché il parametro `--glue-version` non si applica ai processi shell di AWS Glue. Qualsiasi versione specificata verrà ignorata.

Lavori che crei con l' AWS CLI impostazione predefinita di Python 3. Le versioni valide di Python sono 3 (corrispondenti a 3.6) e 3.9. Per specificare Python 3.6, aggiungi questa tupla al parametro `--command`: `"PythonVersion":"3"`

Per specificare Python 3.9, aggiungi questa tupla al parametro `--command`: `"PythonVersion":"3.9"`

Per impostare la capacità massima utilizzata da un processo shell di Python, fornire il parametro `--max-capacity`. Per i processi di shell di Python non è possibile utilizzare il parametro `--allocated-capacity`.

## Librerie supportate dai processi shell di Python

Nella shell Python con Python 3.9 puoi scegliere il set di librerie per utilizzare set di librerie preconfezionati per le tue esigenze. Puoi utilizzare l'opzione `library-set` per scegliere il set di librerie. I valori validi sono `analytics` e `none`.

L'ambiente per l'esecuzione di un processo shell di Python supporta le seguenti librerie:

| Versione di Python | Python 3.6 | Python 3.9 | |
|---|---|---|---|
| Set di librerie | N/D | analytics | nessuno |
| avro | | 1.11.0 | |
| awscli | 116,242 | 1,23,5 | 1,23,5 |
| awswrangler | | 2,15,1 | |
| botocore | 1,12.232 | 1,24,21 | 1,23,5 |

| Versione di Python | Python 3.6 | Python 3.9 | |
|---|---|---|---|
| boto3 | 1,9203 | 1,21,21 | |
| elasticsearch | | 8.2.0 | |
| numpy | 1,16,2 | 1.22.3 | |
| pandas | 0,24,2 | 1.4.2 | |
| psycopg2 | | 2,9,3 | |
| pyathena | | 2.5.3 | |
| PyGreSQL | 5.0.6 | | |
| PyMySQL | | 1.0.2 | |
| pyodbc | | 4.0.32 | |
| pyorc | | 0,6,0 | |
| redshift-connector | | 2.0,907 | |
| richieste | 2.22.0 | 2,27,1 | |
| scikit-learn | 0,20,3 | 1.0.2 | |
| scipy | 1.2.1 | 1.8.0 | |
| SQLAlchemy | | 1,4,36 | |
| s3fs | | 2022,3,0 | |

Puoi utilizzare la libreria `NumPy` in un processo shell di Python per il calcolo scientifico. Per ulteriori informazioni, consulta NumPy. L'esempio seguente mostra uno NumPy script che può essere usato in un job della shell Python. Lo script visualizza "Hello world" e i risultati di numerosi calcoli matematici.

```
import numpy as np
```

```
print("Hello world")

a = np.array([20,30,40,50])
print(a)

b = np.arange( 4 )

print(b)

c = a-b

print(c)

d = b**2

print(d)
```

# Fornire la propria libreria Python

## Utilizzo di PIP

La shell Python che utilizza Python 3.9 consente di fornire moduli Python aggiuntivi o versioni diverse a livello di processo. Puoi utilizzare l'opzione `--additional-python-modules` con un elenco di moduli Python separati da virgole per aggiungere un nuovo modulo o modificare la versione di un modulo esistente. Non è possibile fornire moduli Python personalizzati ospitati su Amazon S3 con questo parametro quando si utilizzano processi di shell Python.

Ad esempio, per aggiornare o aggiungere un nuovo modulo `scikit-learn` usa la seguente coppia di chiave-valore: `"--additional-python-modules", "scikit-learn==0.21.3"`.

AWS Glue utilizza Python Package Installer (pip3) per installare i moduli aggiuntivi. Puoi passare opzioni pip3 aggiuntive all'interno del valore di `--additional-python-modules`. Ad esempio, `"scikit-learn==0.21.3 -i https://pypi.python.org/simple/"`. Si applicano eventuali incompatibilità o limitazioni da pip3.

> **Note**
>
> Per evitare incompatibilità in futuro, si consiglia di utilizzare le librerie create per Python 3.9.

## Utilizzo di un file Egg o Whl

È possibile che uno o più pacchetti di librerie Python siano disponibili come un file `.whl` o `.egg`. In questo caso, puoi specificarli nel tuo processo utilizzando AWS Command Line Interface (AWS CLI) sotto il flag "`--extra-py-files`", come mostrato nell'esempio seguente.

```
aws glue create-job --name python-redshift-test-cli --role role --command '{"Name" :
 "pythonshell", "ScriptLocation" : "s3://MyBucket/python/library/redshift_test.py"}'
    --connections Connections=connection-name --default-arguments '{"--extra-py-
files" : ["s3://amzn-s3-demo-bucket/EGG-FILE", "s3://amzn-s3-demo-bucket/WHEEL-FILE"]}'
```

In caso di dubbi su come creare un file `.egg` o `.whl` da una libreria Python, utilizza la procedura seguente. Questo esempio si applica su sistemi macOS, Linux e Windows Subsystem for Linux (WSL).

Per creare un file .egg o .whl Python

1. Crea un cluster Amazon Redshift in un cloud privato virtuale (VPC, Virtual Private Cloud) e aggiungi alcuni dati a una tabella.
2. Crea una AWS Glue connessione per la VPC-SecurityGroup-Subnet combinazione che hai usato per creare il cluster. Verifica che la connessione funzioni.
3. Crea una directory denominata `redshift_example` e crea un file denominato `setup.py`. Incollare il codice seguente in `setup.py`.

   ```
   from setuptools import setup

   setup(
       name="redshift_module",
       version="0.1",
       packages=['redshift_module']
   )
   ```

4. Nella directory `redshift_example` crea una directory `redshift_module`. Nella directory `redshift_module` crea i file `__init__.py` e `pygresql_redshift_common.py`.
5. Lascia il file `__init__.py` vuoto. Incolla il codice seguente in `pygresql_redshift_common.py`. Sostituisci *`portdb_name`*,*`user`*, e *`password_for_user`* con dettagli specifici del tuo cluster Amazon Redshift. Sostituisci *`table_name`* con il nome della tabella in Amazon Redshift.

```
import pg


def get_connection(host):
    rs_conn_string = "host=%s port=%s dbname=%s user=%s password=%s" % (
        host, port, db_name, user, password_for_user)

    rs_conn = pg.connect(dbname=rs_conn_string)
    rs_conn.query("set statement_timeout = 1200000")
    return rs_conn


def query(con):
    statement = "Select * from table_name;"
    res = con.query(statement)
    return res
```

6. Se non sei ancora in tale directory, passa alla directory `redshift_example`.
7. Esegui una delle seguenti operazioni:

   - Per creare un file `.egg`, esegui il comando seguente.

     ```
     python setup.py bdist_egg
     ```

   - Per creare un file `.whl`, esegui il comando seguente.

     ```
     python setup.py bdist_wheel
     ```

8. Installa le dipendenze necessarie per il comando precedente.
9. Il comando crea un file nella directory `dist`:

   - Se hai creato un file egg, viene denominato `redshift_module-0.1-py2.7.egg`.
   - Se hai creato un file wheel, viene denominato `redshift_module-0.1-py2.7-none-any.whl`.

   Carica questo file in Amazon S3.

   In questo esempio, il percorso del file caricato è *s3://amzn-s3-demo-bucket/EGG-FILE* o *s3://amzn-s3-demo-bucket/WHEEL-FILE*.

10. Crea un file Python da utilizzare come script per il processo AWS Glue e aggiungi al file il codice seguente.

```
from redshift_module import pygresql_redshift_common as rs_common

con1 = rs_common.get_connection(redshift_endpoint)
res = rs_common.query(con1)

print "Rows in the table cities are: "

print res
```

11. Carica il file precedente in Amazon S3. In questo esempio, il percorso del file caricato è *s3://amzn-s3-demo-bucket/scriptname.py*.
12. Crea un processo shell di Python utilizzando questo script. Nella console AWS Glue, nella pagina Job properties (Proprietà processo), specifica il percorso al file `.egg/.whl` nella casella Python library path (Percorso libreria Python). Se sono presenti più file `.egg/.whl` e file Python, occorre fornire un elenco separato da virgole in questa casella.

    Quando si modificano o si rinominano i file `.egg`, i nomi dei file devono utilizzare i nomi predefiniti generati dal comando "python setup.py bdist_egg" o devono rispettare le convenzioni di denominazione del modulo Python. Per ulteriori informazioni, vedere la Guida di stile per il codice Python.

    Utilizzando AWS CLI, crea un lavoro con un comando, come nell'esempio seguente.

```
aws glue create-job --name python-redshift-test-cli --role Role --command
 '{"Name" :  "pythonshell", "ScriptLocation" : "s3://amzn-s3-demo-bucket/
scriptname.py"}'
     --connections Connections="connection-name" --default-arguments '{"--extra-
py-files" : ["s3://amzn-s3-demo-bucket/EGG-FILE", "s3://amzn-s3-demo-bucket/WHEEL-
FILE"]}'
```

    Quando il processo viene eseguito, lo script stampa le righe create nella *table_name* tabella nel cluster Amazon Redshift.

## Utilizzabile AWS CloudFormation con i lavori della shell Python in AWS Glue

È possibile utilizzare AWS CloudFormation con i job della shell Python in. AWS Glue Di seguito è riportato un esempio:

```
AWSTemplateFormatVersion: 2010-09-09
Resources:
  Python39Job:
    Type: 'AWS::Glue::Job'
    Properties:
      Command:
        Name: pythonshell
        PythonVersion: '3.9'
        ScriptLocation: 's3://bucket/location'
      MaxRetries: 0
      Name: python-39-job
      Role: RoleName
```

L'output del gruppo Amazon CloudWatch Logs per i job della shell Python è. `/aws-glue/python-jobs/output` Per gli errori, consulta il gruppo di log `/aws-glue/python-jobs/error`.

## Migrazione dalla shell 3.6 di Python alla 3.9

Per migrare i lavori della shell Python alla versione più recente: AWS Glue

1. Nella AWS Glue console (https://console.aws.amazon.com/glue/), scegli il tuo job di shell Python esistente.
2. Nella scheda Dettagli del processo, imposta la versione di Python su `Python 3.9` e scegli Salva.
3. Assicurati che lo script del processo sia compatibile con Python 3.9 e che funzioni correttamente.

## Migrazione dai job della AWS shell Glue Python

AWS ha lanciato i job della shell AWS Glue Python nel 2018 per offrire ai clienti un modo semplice per eseguire script Python per lavori ETL di small-to-medium grandi dimensioni e per attivare query SQL. Tuttavia, ora sono disponibili opzioni più moderne e flessibili per affrontare i carichi di lavoro attualmente in esecuzione. PythonShell Questo argomento spiega come migrare i carichi di lavoro dai

job della shell Glue AWS Python a una di queste opzioni alternative per sfruttare le nuove funzionalità disponibili.

## Migrazione del carico di lavoro verso i job AWS Glue Spark

AWS Glue Spark e PySpark jobs ti consentono di eseguire i tuoi carichi di lavoro in modo distribuito. Poiché sia i job AWS Glue Python Shell che i job AWS Glue Spark vengono eseguiti sulla stessa piattaforma, la migrazione è facile e puoi continuare a utilizzare le funzionalità di Glue esistenti che usi con i job di Python Shell, come AWS Glue AWS Workflows, Glue AWS Triggers, l' Amazon EventBridge integrazione di Glue, l'installazione di pacchetti basata su\ PIP e così via. AWS

Tuttavia, i job AWS Glue Spark sono progettati per eseguire carichi di lavoro Spark e il numero minimo di lavoratori è 2. Se si effettua la migrazione dai processi shell di Python senza modificare gli script, verrà effettivamente utilizzato solo un worker e gli altri worker rimarranno inattivi. Questa operazione aumenterà i costi.

Per aumentare l'efficienza, occorre riscrivere lo script del processo di Python per utilizzare le funzionalità di Spark e distribuire il carico di lavoro tra più worker. Se lo script di Python si basa su Pandas, è facile eseguire la migrazione usando la nuova API Pandas su Spark. Scopri di più su questo argomento nel blog AWS Big Data: approfondisci la conoscenza di AWS Glue 4.0 for Apache Spark.

## Migrazione del carico di lavoro a AWS Lambda

AWS Lambda è un servizio di elaborazione serverless che consente di eseguire codice senza effettuare il provisioning o la gestione di server. Poiché AWS Lambda offre tempi di startup inferiori e opzioni più flessibili per la capacità di elaborazione, è possibile beneficiare di questi vantaggi. Per gestire librerie Python aggiuntive, i processi shell di AWS Glue Python utilizzano l'installazione basata su PIP. Tuttavia, per AWS Lambda, devi scegliere una delle seguenti opzioni: un archivio zip, un'immagine del contenitore o Lambda Layers.

D'altra parte, il timeout massimo AWS Lambdaè di 900 secondi (15 minuti). Se la durata del tuo attuale carico di lavoro di AWS Glue Python Shell è superiore a quella, o se il tuo carico di lavoro presenta uno schema a picchi che può causare durate di lavoro più lunghe, ti consigliamo di esplorare altre opzioni invece di. AWS Lambda

## Migrazione del carico di lavoro su Amazon ECS/Fargate

Amazon Elastic Container Service (Amazon ECS) è un servizio completamente gestito che semplifica la distribuzione, la gestione e la scalabilità delle applicazioni containerizzate. AWS Fargate è un

motore di elaborazione serverless per carichi di lavoro containerizzati in esecuzione su Amazon ECS e Amazon Elastic Kubernetes Service (Amazon EKS). Non è previsto un timeout massimo su Amazon ECS e Fargate, quindi questa è una buona opzione per i processi di lunga durata. Poiché si ha il pieno controllo sull'immagine del container, è possibile portare lo script Python e librerie Python supplementari nel container e usarle. Tuttavia, è necessario containerizzare lo script Python per utilizzare questo approccio.

### Migrazione del carico di lavoro verso Flussi di lavoro gestiti da Amazon per operatore Python di Apache Airflow

Amazon Managed Workflows for Apache Airflow (Managed Workflows for Apache Airflow) è un servizio di orchestrazione gestito per Apache Airflow che semplifica la configurazione e la gestione di pipeline di dati nel cloud su larga scala. end-to-end Se disponi già di un ambiente MWAA, sarà semplice utilizzare l'operatore Python anziché i job AWS Glue Python Shell. L'operatore Python è un operatore che esegue il codice Python all'interno di un flusso di lavoro Airflow. Tuttavia, se non si dispone di un ambiente MWAA esistente, consigliamo di valutare altre opzioni.

### Migrazione del carico di lavoro verso i lavori di formazione sull'intelligenza artificiale Amazon SageMaker AI

Amazon SageMaker AI La formazione è un servizio di machine learning (ML) completamente gestito offerto da Amazon SageMaker AI che consente di addestrare in modo efficiente un'ampia gamma di modelli di machine learning su larga scala. Il fulcro dei lavori di Amazon SageMaker AI intelligenza artificiale è la containerizzazione dei carichi di lavoro ML e la capacità di gestire AWS le risorse di elaborazione. Se preferisci un ambiente serverless in cui non è previsto un timeout massimo, i lavori di formazione sull' Amazon SageMaker AI intelligenza artificiale potrebbero essere la soluzione giusta per te. Tuttavia, la latenza di avvio tende ad essere più lunga rispetto ai job di Glue AWS Python Shell. Per i processi sensibili alla latenza, consigliamo di valutare altre opzioni.

## Monitoraggio AWS Glue

Il monitoraggio è un elemento importante per mantenere l'affidabilità, la disponibilità e le prestazioni delle AWS Glue altre AWS soluzioni. AWS fornisce strumenti di monitoraggio che è possibile utilizzare per osservare AWS Glue, segnalare quando qualcosa non va e agire automaticamente quando appropriato:

Per controllare AWS Glue e segnalare l'eventuale presenza di problemi, puoi usare gli strumenti di monitoraggio automatici seguenti:

- Amazon CloudWatch Events offre un flusso quasi in tempo reale di eventi di sistema che descrivono i cambiamenti nelle AWS risorse. CloudWatch Events consente l'elaborazione automatizzata basata sugli eventi. È possibile scrivere regole che controllano determinati eventi e attivano azioni automatiche in altri AWS servizi quando si verificano tali eventi. Per ulteriori informazioni, consulta la Amazon CloudWatch Events User Guide.
- Amazon CloudWatch Logs ti consente di monitorare, archiviare e accedere ai tuoi file di registro da EC2 istanze Amazon e altre fonti. AWS CloudTrail CloudWatch I log possono monitorare le informazioni contenute nei file di registro e avvisarti quando vengono raggiunte determinate soglie. Puoi inoltre archiviare i dati del log in storage estremamente durevole. Per ulteriori informazioni, consulta la Amazon CloudWatch Logs User Guide.
- AWS CloudTrailacquisisce le chiamate API e gli eventi correlati effettuati da o per conto del tuo AWS account e invia i file di log a un bucket Amazon S3 da te specificato. Puoi identificare quali utenti e account effettuano le chiamate AWS, l'indirizzo IP di origine da cui vengono effettuate le chiamate e quando vengono effettuate le chiamate. Per ulteriori informazioni, consulta la Guida per l'utente AWS CloudTrail.

Inoltre, hai accesso alle seguenti informazioni nella AWS Glue console per aiutarti a eseguire il debug e profilare i lavori:

- Lavori Spark: puoi visualizzare una visualizzazione di serie di CloudWatch metriche selezionate e i lavori più recenti hanno accesso all'interfaccia utente di Spark. Per ulteriori informazioni, consulta the section called “Monitoraggio dei processi Spark”.
- Ray jobs: puoi vedere una visualizzazione di serie di metriche selezionate. CloudWatch Per ulteriori informazioni, consulta the section called “Parametri dei processi Ray”.

Argomenti


- AWS tag in AWS Glue
- Automatizzazione con AWS Glue EventBridge
- Risorse di monitoraggio AWS Glue
- Registrazione delle chiamate AWS Glue API con AWS CloudTrail


## AWS tag in AWS Glue

Per facilitare la gestione delle risorse AWS Glue puoi facoltativamente assegnare i tag ad alcuni tipi di risorse AWS Glue. Un tag è un'etichetta che si assegna a una AWS risorsa. Ogni tag è

composto da una chiave e da un valore opzionale, entrambi personalizzabili. Puoi usare i tag in AWS Glue per organizzare e identificare le risorse. I tag possono essere utilizzati per creare report di contabilità dei costi e limitare l'accesso alle risorse. Se lo utilizzi AWS Identity and Access Management, puoi controllare quali utenti del tuo AWS account sono autorizzati a creare, modificare o eliminare i tag. Oltre alle autorizzazioni per chiamare i tag relativi ai tag APIs, ti serve anche l'`glue:GetConnection`autorizzazione per richiamare i tag APIs sulle connessioni e l'`glue:GetDatabase`autorizzazione per richiamare i tag sui database. APIs Per ulteriori informazioni, consulta ABAC con Glue AWS.

In AWS Glue è possibile applicare tag alle seguenti risorse:

- Connessione
- Database
- Crawler
- Sessione interattiva
- Endpoint di sviluppo
- Processo
- Trigger
- Flusso di lavoro
- Blueprint
- Trasformazione basata su machine learning
- Set di regole sulla qualità dei dati
- Schemi di flussi di dati
- Registri degli schemi di flussi di dati

> **Note**
>
> Come best practice, per consentire il tagging di queste risorse AWS Glue, includi sempre l'operazione `glue:TagResource` nelle policy.

Tieni in considerazione le informazioni seguenti quando usi i tag con AWS Glue.

- Il numero massimo di tag supportati per entità è 50.

- In AWS Glue, i tag vengono specificati come elenco di coppie chiave-valore nel formato `{"string": "string" ...}`
- Quando crei un tag su un oggetto, la chiave di tag è obbligatoria e il valore di tag è facoltativo.
- La chiave di tag e il valore di tag fanno distinzione tra maiuscole e minuscole.
- La chiave di tag e il valore di tag non devono contenere il prefisso aws. Non sono consentite operazioni su questi tag.
- La lunghezza massima delle chiavi di tag è 128 caratteri Unicode in UTF-8. La chiave di tag non deve essere vuota o nulla.
- Il valore massimo dei tag è 256 caratteri Unicode in UTF-8. Il valore di tag può essere vuoto o nullo.

## Supporto per l'etichettatura delle connessioni AWS Glue

È possibile limitare le autorizzazioni alle operazioni `CreateConnection`, `UpdateConnection`, `GetConnection` e `DeleteConnection` basate sull'assegnazione di tag alla risorsa. Ciò consente di implementare il controllo degli accessi con privilegi minimi sui AWS Glue lavori con origini dati JDBC che devono recuperare le informazioni di connessione JDBC dal Data Catalog.

Esempio di utilizzo

Crea una AWS Glue connessione con il tag ["connection-category», «dev-test"].

Specifica la condizione del tag per l'operazione `GetConnection` nella policy IAM.

```
{
   "Effect": "Allow",
   "Action": [
     "glue:GetConnection"
     ],
   "Resource": "*",
   "Condition": {
     "ForAnyValue:StringEquals": {
       "aws:ResourceTag/tagKey": "dev-test"
     }
   }
 }
```

## Esempi

I seguenti esempi creano un processo con tag assegnati.

### AWS CLI

```
aws glue create-job --name job-test-tags --role MyJobRole --command
 Name=glueetl,ScriptLocation=S3://aws-glue-scripts//prod-job1
--tags key1=value1,key2=value2
```

### CloudFormation JSON

```
{
  "Description": "AWS Glue Job Test Tags",
  "Resources": {
    "MyJobRole": {
      "Type": "AWS::IAM::Role",
      "Properties": {
        "AssumeRolePolicyDocument": {
          "Version": "2012-10-17",
          "Statement": [
            {
              "Effect": "Allow",
              "Principal": {
                "Service": [
                  "glue.amazonaws.com"
                ]
              },
              "Action": [
                "sts:AssumeRole"
              ]
            }
          ]
        },
        "Path": "/",
        "Policies": [
          {
            "PolicyName": "root",
            "PolicyDocument": {
              "Version": "2012-10-17",
              "Statement": [
                {
                  "Effect": "Allow",
```

```
                  "Action": "*",
                  "Resource": "*"
                }
              ]
            }
          }
        ]
      }
    },
    "MyJob": {
      "Type": "AWS::Glue::Job",
      "Properties": {
        "Command": {
          "Name": "glueetl",
          "ScriptLocation": "s3://aws-glue-scripts//prod-job1"
        },
        "DefaultArguments": {
          "--job-bookmark-option": "job-bookmark-enable"
        },
        "ExecutionProperty": {
          "MaxConcurrentRuns": 2
        },
        "MaxRetries": 0,
        "Name": "cf-job1",
        "Role": {
           "Ref": "MyJobRole",
           "Tags": {
                "key1": "value1",
                "key2": "value2"
           }
        }
      }
    }
  }
}
```

## CloudFormation YAML

```
Description: AWS Glue Job Test Tags
Resources:
  MyJobRole:
    Type: AWS::IAM::Role
    Properties:
```

```
      AssumeRolePolicyDocument:
        Version: '2012-10-17'
        Statement:
          - Effect: Allow
            Principal:
              Service:
                - glue.amazonaws.com
            Action:
              - sts:AssumeRole
      Path: "/"
      Policies:
        - PolicyName: root
          PolicyDocument:
            Version: '2012-10-17'
            Statement:
              - Effect: Allow
                Action: "*"
                Resource: "*"
  MyJob:
    Type: AWS::Glue::Job
    Properties:
      Command:
        Name: glueetl
        ScriptLocation: s3://aws-glue-scripts//prod-job1
      DefaultArguments:
        "--job-bookmark-option": job-bookmark-enable
      ExecutionProperty:
        MaxConcurrentRuns: 2
      MaxRetries: 0
      Name: cf-job1
      Role:
        Ref: MyJobRole
        Tags:
           key1: value1
           key2: value2
```

Per ulteriori informazioni, consulta Strategie di tagging di AWS.

Per informazioni su come controllare l'accesso tramite i tag, consulta ABAC con Glue AWS.

# Automatizzazione con AWS Glue EventBridge

Puoi utilizzare Amazon EventBridge per automatizzare AWS i tuoi servizi e rispondere automaticamente a eventi di sistema come problemi di disponibilità delle applicazioni o modifiche delle risorse. Gli eventi AWS relativi ai servizi vengono forniti quasi EventBridge in tempo reale. Puoi compilare regole semplici che indichino quali eventi sono considerati di interesse per te e quali azioni automatizzate intraprendere quando un evento corrisponde a una regola. Le azioni che possono essere attivate automaticamente includono le seguenti:

- Invocare una funzione AWS Lambda
- Richiamo del comando Amazon EC2 Run
- Inoltro dell'evento a Amazon Kinesis Data Streams
- Attivazione di una macchina a stati AWS Step Functions
- Notifica di un argomento Amazon SNS o di una coda Amazon SQS

Alcuni esempi di utilizzo di EventBridge with AWS Glue includono quanto segue:

- Attivazione di una funzione Lambda in caso di esito positivo di un processo ETL
- Notifica di un argomento Amazon SNS quando un processo ETL ha esito negativo

Quanto segue EventBridge è generato daAWS Glue.

- Gli eventi per `"detail-type":"Glue Job State Change"` vengono generati per `SUCCEEDED`, `FAILED`, `TIMEOUT` e `STOPPED`.
- Eventi per `"detail-type":"Glue Job Run Status"` vengono generati per l'esecuzione dei processi `RUNNING`, `STARTING`e `STOPPING` quando superano la soglia di notifica di ritardo del processo. È necessario impostare la proprietà della soglia di notifica del ritardo del processo per ricevere questi eventi.

  Quando viene superata la soglia di notifica del ritardo del processo, viene generato un solo evento per ciascuno stato di esecuzione del processo.
- Eventi per `"detail-type":"Glue Crawler State Change"` vengono generati per `Started`, `Succeeded` e `Failed`.
- Gli eventi per `“detail_type”:“Glue Scheduled Crawler Invocation Failure”` vengono generati quando il crawler pianificato non si avvia. Nei dettagli della notifica:

  - `customerId` contiene l'ID account del cliente.
  - `crawlerName` contiene il nome del crawler che non è stato avviato.
  - `errorMessage` contiene il messaggio di eccezione dell'errore di invocazione.
- Gli eventi per `"detail_type":"Glue Auto Statistics Invocation Failure"` vengono generati quando l'esecuzione dell'attività di statistica delle colonne gestita automaticamente non viene avviata. Nei dettagli della notifica:
  - `catalogId` contiene l'ID associato a un catalogo.
  - `databaseName` contiene il nome del database interessato.
  - `tableName` contiene il nome della tabella interessata.
  - `errorMessage` contiene il messaggio di eccezione dell'errore di invocazione.
- Gli eventi per `"detail_type":"Glue Scheduled Statistics Invocation Failure"` vengono generati quando l'esecuzione dell'attività di statistica delle colonne pianificate (cron) non viene avviata. Nei dettagli della notifica:
  - `catalogId` contiene l'ID associato a un catalogo.
  - `databaseName` contiene il nome del database interessato.
  - `tableName` contiene il nome della tabella interessata.
  - `errorMessage` contiene il messaggio di eccezione dell'errore di invocazione.
- Gli eventi per `"detail_type":"Glue Statistics Task Started"` vengono generati quando l'esecuzione dell'attività di statistica delle colonne viene avviata.
- Gli eventi per `"detail_type":"Glue Statistics Task Succeeded"` vengono generati quando l'esecuzione dell'attività di statistica delle colonne ha esito positivo.
- Gli eventi per `"detail_type":"Glue Statistics Task Failed"` vengono generati quando l'esecuzione dell'attività di statistica delle colonne ha esito negativo.
- Gli eventi per `"detail-type":"Glue Data Catalog Database State Change"` vengono generati per `CreateDatabase`, `DeleteDatabase`, `CreateTable`, `DeleteTable` e `BatchDeleteTable`. Ad esempio, se viene creata o eliminata una tabella, viene inviata una notifica a EventBridge. Si noti che non è possibile scrivere un programma che dipende dall'ordine o dall'esistenza di eventi di notifica, poiché potrebbero essere fuori sequenza o mancanti. Gli eventi vengono emessi secondo il principio del massimo sforzo. Nei dettagli della notifica:
  - `typeOfChange` contiene il nome dell'operazione API.
  - `databaseName` contiene il nome del database interessato.

  - `changedTables` contiene fino a 100 nomi di tabelle interessate per ogni notifica. Quando i nomi di tabella sono lunghi, potrebbero essere create più notifiche.
- Gli eventi per `"detail-type":"Glue Data Catalog Table State Change"` vengono generati per `UpdateTable`, `CreatePartition`, `BatchCreatePartition`, `UpdatePartition`, `DeletePartition`, `BatchUpdatePartition` e `BatchDeletePartition`. Ad esempio, se una tabella o una partizione viene aggiornata, viene inviata una notifica a EventBridge. Si noti che non è possibile scrivere un programma che dipende dall'ordine o dall'esistenza di eventi di notifica, poiché potrebbero essere fuori sequenza o mancanti. Gli eventi vengono emessi secondo il principio del massimo sforzo. Nei dettagli della notifica:
  - `typeOfChange` contiene il nome dell'operazione API.
  - `databaseName` contiene il nome del database contenente le risorse interessate.
  - `tableName` contiene il nome della tabella interessata.
  - `changedPartitions` specifica fino a 100 partizioni interessate in una notifica. Quando i nomi di partizione sono lunghi, potrebbero essere create più notifiche.

    Ad esempio, se ci sono due chiavi di partizione, `Year` e `Month`, `"2018,01", "2018,02"` modifica la partizione dove `"Year=2018" and "Month=01"` e la partizione dove `"Year=2018" and "Month=02"`.

```
{
    "version":"0",
    "id":"abcdef00-1234-5678-9abc-def012345678",
    "detail-type":"Glue Data Catalog Table State Change",
    "source":"aws.glue",
    "account":"123456789012",
    "time":"2017-09-07T18:57:21Z",
    "region":"us-west-2",
    "resources":["arn:aws:glue:us-west-2:123456789012:database/default/foo"],
    "detail":{
        "changedPartitions": [
            "2018,01",
            "2018,02"
        ],
        "databaseName": "default",
        "tableName": "foo",
        "typeOfChange": "BatchCreatePartition"
        }
```

```
}
```

Per ulteriori informazioni, consulta la Amazon CloudWatch Events User Guide. Per eventi specifici di AWS Glue, consulta Eventi AWS Glue.

## Risorse di monitoraggio AWS Glue

AWS Glue prevede dei limiti di servizio per proteggere i clienti da forniture eccessive e impreviste e da operazioni dannose volte ad aumentare le spese. I limiti proteggono anche il servizio. Accedendo alla console AWS Service Quota, i clienti possono visualizzare i limiti attuali delle risorse e richiederne un aumento (se del caso).

AWS Glueconsente di visualizzare l'utilizzo delle risorse del servizio in percentuale in Amazon CloudWatch e di configurare CloudWatch allarmi su di esso per monitorare l'utilizzo. Amazon CloudWatch fornisce il monitoraggio AWS delle risorse e delle applicazioni dei clienti in esecuzione sull'infrastruttura Amazon. I parametri sono gratuiti. Sono supportati i parametri seguenti:

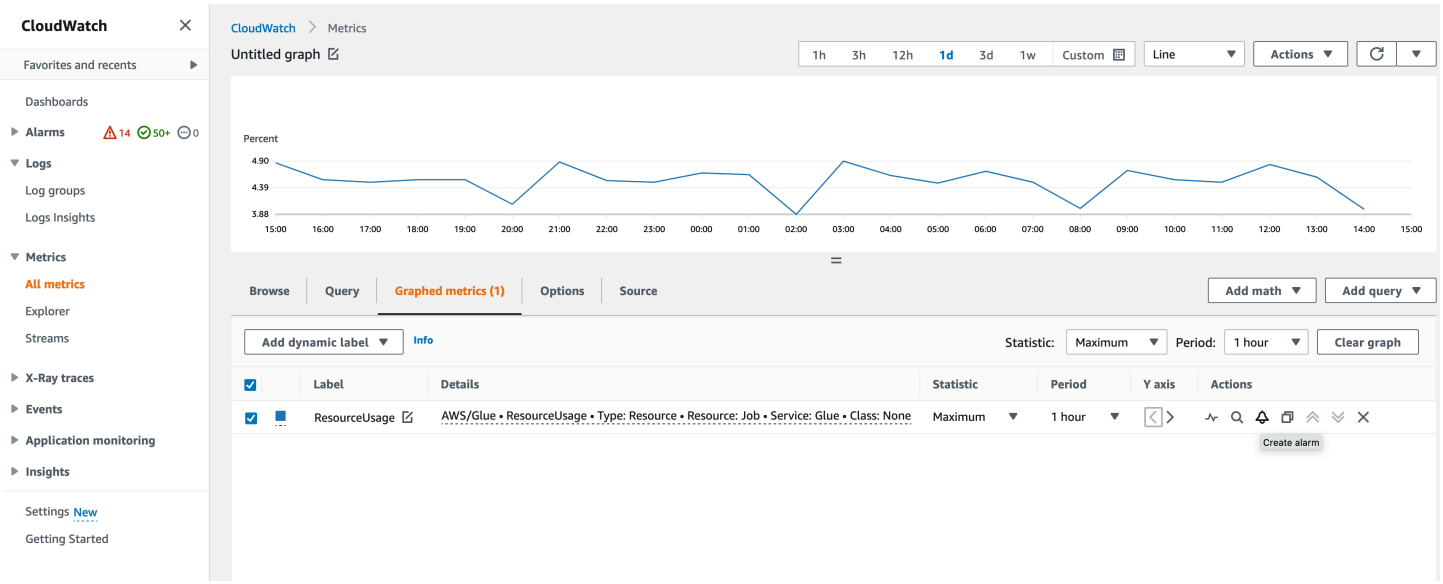
- Numero di flussi di lavoro per account
- Numero di trigger per account
- Numero di processi per account
- Numero di esecuzioni processo simultanee per account
- Numero di schemi per account
- Numero di sessioni interattive per account

### Configurazione e utilizzo dei parametri delle risorse

Per utilizzare questa funzionalità, puoi accedere alla CloudWatch console Amazon per visualizzare le metriche e configurare gli allarmi. Le metriche si trovano nello spazio dei nomi AWS/Glue e rappresentano una percentuale del conteggio effettivo dell'utilizzo delle risorse diviso per la quota di risorse. Le CloudWatch metriche vengono inviate ai tuoi account, il che non comporta alcun costo per te. Ad esempio, se hai creato 10 flussi di lavoro e la tua quota di servizio ti consente di avere un massimo di 200 flussi di lavoro, l'utilizzo è 10/200 = 5% e nel grafico vedrai un punto dati di 5 come percentuale. Per maggiore specificità:

```
Namespace: AWS/Glue
Metric name: ResourceUsage
Type: Resource
```

```
Resource: Workflow (or Trigger, Job, JobRun, Blueprint, InteractiveSession)
Service: Glue
Class: None
```




Per creare un allarme in base a una metrica nella CloudWatch console:

1. Una volta individuato il parametro, vai a Parametri definiti.
2. Fai clic su Crea allarme in Operazioni.
3. Configura l'allarme secondo necessità.

Emettiamo dei parametri ogni volta che l'utilizzo delle risorse cambia, ad esempio in caso di aumento o diminuzione. Ma se l'utilizzo delle risorse non cambia, emettiamo le metriche ogni ora, in modo da avere un grafico continuo. CloudWatch Per evitare la perdita di punti dati, sconsigliamo di configurare un periodo inferiore a 1 ora.

Puoi anche configurare gli allarmi usando AWS CloudFormation come nell'esempio seguente. In questo esempio, quando l'utilizzo delle risorse del flusso di lavoro raggiunge l'80%, viene attivato un allarme per inviare un messaggio all'argomento SNS esistente, a cui è possibile abbonarsi per ricevere notifiche.

```
{
 "Type": "AWS::CloudWatch::Alarm",
 "Properties": {
  "AlarmName": "WorkflowUsageAlarm",
  "ActionsEnabled": true,
```

```
  "OKActions": [],
  "AlarmActions": [
   "arn:aws:sns:af-south-1:085425700061:Default_CloudWatch_Alarms_Topic"
  ],
  "InsufficientDataActions": [],
  "MetricName": "ResourceUsage",
  "Namespace": "AWS/Glue",
  "Statistic": "Maximum",
  "Dimensions": [{
    "Name": "Type",
    "Value": "Resource"
   },
   {
    "Name": "Resource",
    "Value": "Workflow"
   },
   {
    "Name": "Service",
    "Value": "Glue"
   },
   {
    "Name": "Class",
    "Value": "None"
   }
  ],
  "Period": 3600,
  "EvaluationPeriods": 1,
  "DatapointsToAlarm": 1,
  "Threshold": 80,
  "ComparisonOperator": "GreaterThanThreshold",
  "TreatMissingData": "notBreaching"
 }
}
```

## Registrazione delle chiamate AWS Glue API con AWS CloudTrail

AWS Glueè integrato con AWS CloudTrail, un servizio che fornisce una registrazione delle azioni intraprese da un utente, un ruolo o un AWS servizio in. AWS Glue CloudTrail acquisisce tutte le chiamate API AWS Glue come eventi. Le chiamate acquisite includono le chiamate dalla console di AWS Glue e le chiamate di codice alle operazioni delle API AWS Glue. Se crei un trail, puoi abilitare la distribuzione continua di CloudTrail eventi a un bucket Amazon S3, inclusi gli eventi per. AWS Glue Se non configuri un percorso, puoi comunque visualizzare gli eventi più recenti nella

CloudTrail console nella cronologia degli eventi. Utilizzando le informazioni raccolte da CloudTrail, puoi determinare a quale richiesta è stata inviataAWS Glue, l'indirizzo IP da cui è stata effettuata la richiesta, chi ha effettuato la richiesta, quando è stata effettuata e dettagli aggiuntivi.

Per ulteriori informazioni CloudTrail, consulta la Guida AWS CloudTrail per l'utente.

## AWS Glueinformazioni in CloudTrail

CloudTrail è abilitato sul tuo AWS account al momento della creazione dell'account. Quando si verifica un'attività inAWS Glue, tale attività viene registrata in un CloudTrail evento insieme ad altri eventi AWS di servizio nella cronologia degli eventi. Puoi visualizzare, cercare e scaricare gli eventi recenti nel tuo AWS account. Per ulteriori informazioni, consulta Visualizzazione degli eventi con la cronologia degli CloudTrail eventi.

Per una registrazione continua degli eventi nel tuo AWS account, inclusi gli eventi diAWS Glue, crea un percorso. Un trail consente di CloudTrail inviare file di log a un bucket Amazon S3. Per impostazione predefinita, quando crei un percorso nella console, il percorso si applica a tutte le AWS regioni. Il trail registra gli eventi di tutte le regioni della AWS partizione e consegna i file di log al bucket Amazon S3 specificato. Inoltre, puoi configurare altri AWS servizi per analizzare ulteriormente e agire in base ai dati sugli eventi raccolti nei log. CloudTrail Per ulteriori informazioni, consulta gli argomenti seguenti:

- Creare un percorso per il tuo account AWS
- CloudTrail servizi e integrazioni supportati
- Configurazione delle notifiche Amazon SNS per CloudTrail
- Ricezione di file di CloudTrail registro da più regioni e ricezione di file di CloudTrail registro da più account

Tutte AWS Glue le azioni vengono registrate CloudTrail e documentate in. AWS Glue API Ad esempio, le chiamate a `CreateTable` e `CreateDatabase` le `CreateScript` azioni generano voci nei file di CloudTrail registro.

Ogni evento o voce di log contiene informazioni sull'utente che ha generato la richiesta. Le informazioni di identità consentono di determinare quanto segue:

- Se la richiesta è stata effettuata con le credenziali dell'utente IAM o root.
- Se la richiesta è stata effettuata con le credenziali di sicurezza temporanee per un ruolo o un utente federato.

- Se la richiesta è stata effettuata da un altro AWS servizio.

Per ulteriori informazioni, consulta Elemento CloudTrail userIdentity.

Tuttavia, CloudTrail non registra tutte le informazioni relative alle chiamate. Ad esempio, non registra determinate informazioni sensibili, come quelle `ConnectionProperties` utilizzate nelle richieste di connessione, e registra `null` invece delle risposte restituite da quanto segue: APIs

```
BatchGetPartition       GetCrawlers          GetJobs          GetTable
CreateScript            GetCrawlerMetrics    GetJobRun        GetTables
GetCatalogImportStatus  GetDatabase          GetJobRuns       GetTableVersions
GetClassifier           GetDatabases         GetMapping       GetTrigger
GetClassifiers          GetDataflowGraph     GetObjects       GetTriggers
GetConnection           GetDevEndpoint       GetPartition     GetUserDefinedFunction
GetConnections          GetDevEndpoints      GetPartitions    GetUserDefinedFunctions
GetCrawler              GetJob               GetPlan
```

## Informazioni sulle voci dei file di log di AWS Glue

Un trail è una configurazione che consente la distribuzione di eventi come file di log in un bucket Amazon S3 specificato dall'utente. CloudTrail i file di registro contengono una o più voci di registro. Un evento rappresenta una singola richiesta proveniente da qualsiasi fonte e include informazioni sull'azione richiesta, la data e l'ora dell'azione, i parametri della richiesta e così via. CloudTrail i file di registro non sono una traccia ordinata dello stack delle chiamate API pubbliche, quindi non vengono visualizzati in un ordine specifico.

L'esempio seguente mostra una voce di CloudTrail registro che illustra l'`DeleteCrawler`azione.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AKIAIOSFODNN7EXAMPLE",
    "arn": "arn:aws:iam::123456789012:user/johndoe",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "johndoe"
  },
  "eventTime": "2017-10-11T22:29:49Z",
  "eventSource": "glue.amazonaws.com",
  "eventName": "DeleteCrawler",
```

```
  "awsRegion": "us-east-1",
  "sourceIPAddress": "72.21.198.64",
  "userAgent": "aws-cli/1.11.148 Python/3.6.1 Darwin/16.7.0 botocore/1.7.6",
  "requestParameters": {
    "name": "tes-alpha"
  },
  "responseElements": null,
  "requestID": "b16f4050-aed3-11e7-b0b3-75564a46954f",
  "eventID": "e73dd117-cfd1-47d1-9e2f-d1271cad838c",
  "eventType": "AwsApiCall",
  "recipientAccountId": "123456789012"
}
```

Questo esempio mostra una voce di CloudTrail registro che illustra un'`CreateConnection`azione.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AKIAIOSFODNN7EXAMPLE",
    "arn": "arn:aws:iam::123456789012:user/johndoe",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "johndoe"
  },
  "eventTime": "2017-10-13T00:19:19Z",
  "eventSource": "glue.amazonaws.com",
  "eventName": "CreateConnection",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "72.21.198.66",
  "userAgent": "aws-cli/1.11.148 Python/3.6.1 Darwin/16.7.0 botocore/1.7.6",
  "requestParameters": {
    "connectionInput": {
      "name": "test-connection-alpha",
      "connectionType": "JDBC",
      "physicalConnectionRequirements": {
        "subnetId": "subnet-323232",
        "availabilityZone": "us-east-1a",
        "securityGroupIdList": [
          "sg-12121212"
        ]
      }
    }
```

```
  },
  "responseElements": null,
  "requestID": "27136ebc-afac-11e7-a7d6-ab217e5c3f19",
  "eventID": "e8b3baeb-c511-4597-880f-c16210c60a4a",
  "eventType": "AwsApiCall",
  "recipientAccountId": "123456789012"
}
```

# AWS Glue Streaming

AWS Glue Lo streaming, un componente di AWS Glue, consente di gestire in modo efficiente i dati in streaming quasi in tempo reale, consentendoti di svolgere attività cruciali come l'acquisizione, l'elaborazione e l'apprendimento automatico dei dati. Utilizzando il framework Apache Spark Streaming, Streaming fornisce un servizio serverless in grado di gestire AWS Glue lo streaming di dati su larga scala. AWS Glue offre varie ottimizzazioni oltre ad Apache Spark, come infrastruttura serverless, auto-scaling, sviluppo di lavori visivi, notebook ad accensione istantanea per lavori di streaming e altri miglioramenti delle prestazioni.

## Casi d'uso per lo streaming

Alcuni casi d'uso comuni per lo streaming includono: AWS Glue

Near-real-time elaborazione dei dati: AWS Glue lo streaming consente alle organizzazioni di elaborare i dati in streaming quasi in tempo reale, consentendo loro di ricavare informazioni e prendere decisioni tempestive sulla base delle informazioni più recenti.

Rilevamento delle frodi: puoi utilizzare AWS Glue Streaming per l'analisi in tempo reale dei dati di streaming, rendendolo utile per rilevare attività fraudolente, come frodi con carte di credito, intrusioni di rete o truffe online. Elaborando e analizzando continuamente i dati in entrata, è possibile identificare rapidamente anomalie o sequenze sospette.

Analisi dei social media: AWS Glue lo streaming può elaborare dati sui social media in tempo reale, come tweet, post o commenti, consentendo alle organizzazioni di monitorare le tendenze, l'analisi del sentiment e gestire la reputazione del marchio in tempo reale.

Analisi dell'Internet of Things (IoT): AWS Glue lo streaming è adatto per gestire e analizzare flussi di dati ad alta velocità generati da dispositivi IoT, sensori e macchinari connessi. Consente il monitoraggio in tempo reale, il rilevamento delle anomalie, la manutenzione predittiva e altri casi d'uso di analisi IoT.

Analisi clickstream: lo AWS Glue streaming può elaborare e analizzare i dati clickstream in tempo reale provenienti da siti Web o applicazioni mobili. In tal modo, le aziende possono ottenere approfondimenti sul comportamento degli utenti, personalizzare le loro esperienze e ottimizzare le campagne di marketing sulla base di dati di clickstream in tempo reale.

Monitoraggio e analisi dei log: AWS Glue lo streaming può elaborare e analizzare continuamente i dati di registro da server, applicazioni o dispositivi di rete in tempo reale. Ciò contribuisce a rilevare le anomalie, risolvere i problemi e monitorare lo stato e le prestazioni del sistema.

Sistemi di raccomandazione: AWS Glue lo streaming può elaborare i dati sulle attività degli utenti in tempo reale e aggiornare i modelli di raccomandazione in modo dinamico. Ciò consente di fornire consigli personalizzati e in tempo reale in base al comportamento e alle preferenze degli utenti.

Questi sono alcuni esempi della vasta gamma di casi d'uso in cui è possibile applicare AWS Glue lo Streaming. La sua integrazione con l' AWS ecosistema e i servizi gestiti lo rendono una scelta conveniente per l'elaborazione e l'analisi dei flussi in tempo reale nel cloud.

## Quali sono i vantaggi dell'utilizzo AWS Glue dello streaming?

I vantaggi dell'utilizzo AWS Glue dello streaming sono i seguenti:

- Serverless: AWS Glue lo streaming è serverless, il che elimina la necessità di gestire l'infrastruttura. Ciò riduce il sovraccarico operativo e consente agli utenti di concentrarsi sulle attività di elaborazione e analisi dei dati anziché sulla gestione dell'infrastruttura.
- Scalabilità automatica: AWS Glue lo streaming offre funzionalità di scalabilità automatica, regolando dinamicamente la capacità di elaborazione in base al carico di lavoro. È scalabile automaticamente in orizzontale o in verticale per gestire le fluttuazioni del volume di dati, garantendo livelli ottimali di prestazioni e utilizzo delle risorse.
- Sviluppo visivo: lo sviluppo di lavori in streaming può essere complesso. AWS Glue Streaming affronta questa sfida offrendo AWS Glue Studio, uno strumento di creazione visiva. AWS Glue Studio semplifica il processo di creazione di flussi di lavoro in streaming e consente agli sviluppatori di progettare e gestire visivamente le applicazioni di streaming, riducendo la curva di apprendimento e aumentando la produttività.
- Conveniente: in quanto servizio serverless, AWS Glue Streaming offre efficienza in termini di costi eliminando la necessità di fornire e mantenere l'infrastruttura. Agli utenti vengono fatturate le risorse utilizzate durante l'esecuzione dei processi di streaming, contribuendo all'ottimizzazione dei costi e a un dimensionamento in base all'utilizzo effettivo.
- Gestisce carichi di lavoro complessi: AWS Glue lo streaming è progettato per gestire carichi di lavoro di streaming complessi. Può elaborare e analizzare grandi volumi di dati in tempo reale, supportare trasformazioni avanzate e integrarsi con altri AWS servizi, abilitando sofisticate pipeline di dati di streaming e flussi di lavoro di analisi.

- Nessun vincolo: AWS Glue lo streaming offre flessibilità ed evita il vincolo del fornitore. Gli utenti possono sfruttare AWS Glue lo streaming come parte di un AWS ecosistema più ampio, integrandolo senza problemi con altri servizi. AWS Ciò consente una facile integrazione con le origini dati, le applicazioni e i servizi esistenti senza vincolare a una tecnologia o piattaforma specifica.

# Quando usare lo streaming? AWS Glue

I casi d'uso dello streaming includono numerose opzioni. Consigliamo AWS Glue lo streaming nei seguenti scenari.

1. Se stai già utilizzando AWS Glue Spark per l'elaborazione in batch, AWS Glue Streaming è la scelta ideale per te. Fornisce una transizione ottimale alla creazione di processi di streaming senza la necessità di imparare un nuovo linguaggio o framework. Sfruttando le conoscenze e l'infrastruttura esistenti, AWS Glue Streaming semplifica il processo di sviluppo del lavoro e consente di estendere facilmente le capacità di elaborazione dei dati a scenari di streaming in tempo reale.
2. Se hai bisogno di un servizio o di un prodotto unificato per gestire carichi di lavoro in batch, in streaming e basati sugli eventi, Streaming è la soluzione che fa per te. AWS Glue Con AWS Glue Streaming, puoi consolidare le tue esigenze di elaborazione dei dati in un unico framework, eliminando la complessità della gestione di più sistemi. Ciò consente di sviluppare e mantenere flussi di lavoro di dati diversi in modo efficiente, garantendo al contempo la coerenza e la compatibilità tra diversi tipi di carichi di lavoro.
3. AWS Glue Lo streaming è ideale per scenari che prevedono volumi di dati di streaming estremamente grandi e trasformazioni complesse, come giunzioni tra flussi o database relazionali. È in grado di elaborare e analizzare in modo efficiente enormi flussi di dati, consentendoti di affrontare con facilità carichi di lavoro impegnativi. Che si tratti di ingestione di dati ad alta velocità o di complesse manipolazioni dei dati, la scalabilità e le funzionalità di elaborazione avanzate di AWS Glue Streaming garantiscono prestazioni ottimali e risultati accurati.
4. Se preferisci un approccio visivo alla creazione di lavori in streaming, AWS Glue offre AWS Glue Studio, con cui puoi progettare e gestire visivamente le tue applicazioni di streaming, semplificando il processo di sviluppo. Questa interfaccia intuitiva consente agli sviluppatori di creare, configurare e monitorare i flussi di lavoro di streaming utilizzando un'interfaccia visiva, riducendo la curva di apprendimento e aumentando la produttività.
5. AWS Glue Lo streaming è una scelta eccellente per i casi near-real-time d'uso in cui esistono rigorosi SLAs (Service Level Agreement) superiori a 10 secondi.

6. Se stai creando un data lake transazionale utilizzando Apache Iceberg, Apache Hudi o Delta Lake, AWS Glue Streaming fornisce il supporto nativo per questi formati di tabelle aperte. Questa perfetta integrazione consente di elaborare i dati in streaming direttamente da questi data lake transazionali, garantendo la coerenza, l'integrità e la compatibilità dei dati.
7. Quando è necessario importare dati di streaming per una varietà di destinazioni di dati: AWS Glue lo streaming fornisce destinazioni native a una varietà di destinazioni di dati come Amazon Redshift, Amazon RDS, Amazon Aurora, Oracle, SQL Server e altre destinazioni.

# Origini dati supportate

AWS Glue Lo streaming supporta le seguenti fonti di dati:

- Amazon Kinesis
- Amazon MSK (Streaming gestito per Apache Kafka)
- Apache Kafka gestito dal cliente

# Destinazioni di dati supportate

AWS Glue Lo streaming supporta una varietà di target di dati come:

- Target di dati supportati da AWS Glue Data Catalog
- Simple Storage Service (Amazon S3)
- Amazon Redshift
- MySQL
- PostgreSQL
- Oracle
- Microsoft SQL Server
- Snowflake
- Qualsiasi database che possa essere collegato tramite JDBC
- Apache Iceberg, Delta e Apache Hudi
- AWS Glue Connettori Marketplace

# Tutorial: crea il tuo primo carico di lavoro in streaming con Studio AWS Glue

In questo tutorial, imparerai come creare un lavoro in streaming utilizzando AWS Glue Studio. AWS Glue Studio è un'interfaccia visiva per creare AWS Glue posti di lavoro.

È possibile creare processi in streaming di estrazione, trasformazione e caricamento (ETL) che vengono eseguiti continuamente e utilizzano dati da origini di streaming in Flusso di dati Amazon Kinesis, Apache Kafka e Streaming gestito da Amazon per Apache Kafka (Amazon MSK).

## Prerequisiti

Per seguire questo tutorial avrai bisogno di un utente con le autorizzazioni di utilizzo AWS della console, Amazon Kinesis AWS Glue, Amazon S3, Amazon Athena, AWS CloudFormation AWS Lambda e Amazon Cognito.

## Utilizzo dei dati in streaming da Amazon Kinesis

Argomenti

- Generazione di dati fittizi con Kinesis Data Generator
- Creazione di un lavoro AWS Glue in streaming con Studio AWS Glue
- Esecuzione di una trasformazione e archiviazione del risultato della trasformazione in Amazon S3

### Generazione di dati fittizi con Kinesis Data Generator

È possibile generare sinteticamente dati di esempio in formato JSON utilizzando Kinesis Data Generator (KDG). Puoi trovare le istruzioni complete e i dettagli nella documentazione dello strumento.

1. Per iniziare, fai clic per eseguire un modello

   nel tuo ambiente. AWS CloudFormation AWS

   > **Note**
   >
   > Potresti riscontrare un errore nel CloudFormation modello perché alcune risorse, come l'utente Amazon Cognito per Kinesis Data Generator, esistono già nel tuo account. AWS

> Ciò potrebbe essere dovuto al fatto che l'hai già configurato in un altro tutorial o da un post di un blog. Per risolvere questo problema, puoi provare il modello in un nuovo AWS account per ricominciare da capo, oppure esplorare un'altra AWS regione. Queste opzioni consentono di eseguire il tutorial senza entrare in conflitto con le risorse esistenti.

Il modello fornisce un flusso di dati Kinesis e un account Kinesis Data Generator. Crea anche un bucket Amazon S3 per contenere i dati e un ruolo di servizio Glue con l'autorizzazione richiesta per questo tutorial.

2. Immetti un Nome utente e una Password che KDG utilizzerà per l'autenticazione. Prendi nota del nome utente e della password per utilizzarli in seguito.
3. Seleziona Avanti fino all'ultimo passaggio. Esprimi il consenso alla creazione di risorse IAM. Verifica la presenza di eventuali errori nella parte superiore dello schermo, ad esempio la password che non soddisfa i requisiti minimi, e implementa il modello.
4. Vai alla scheda Output dello stack. Una volta distribuito, il modello mostrerà la proprietà KinesisDataGeneratorUrlgenerata. Fai clic su quell'URL.
5. Inserisci il Nome utente e la Password di cui hai preso nota.
6. Seleziona la regione che stai utilizzando e seleziona il flusso Kinesis `GlueStreamTest-{AWS::AccountId}`.
7. Immetti il seguente modello:

```
{
    "ventilatorid": {{random.number(100)}},
    "eventtime": "{{date.now("YYYY-MM-DD HH:mm:ss")}}",
    "serialnumber": "{{random.uuid}}",
    "pressurecontrol": {{random.number(
        {
            "min":5,
            "max":30
        }
    )}},
    "o2stats": {{random.number(
        {
            "min":92,
            "max":98
        }
    )}},
    "minutevolume": {{random.number(
```

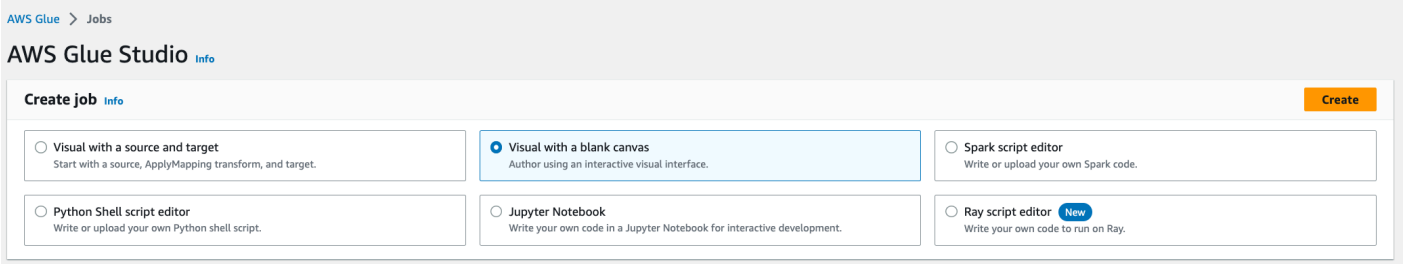
```
        {
            "min":5,
            "max":8
        }
    )}},
    "manufacturer": "{{random.arrayElement(
        ["3M", "GE","Vyaire", "Getinge"]
    )}}"
}
```

Ora puoi visualizzare i dati fittizi con Modello di prova e importare i dati fittizi in Kinesis con Invia dati.

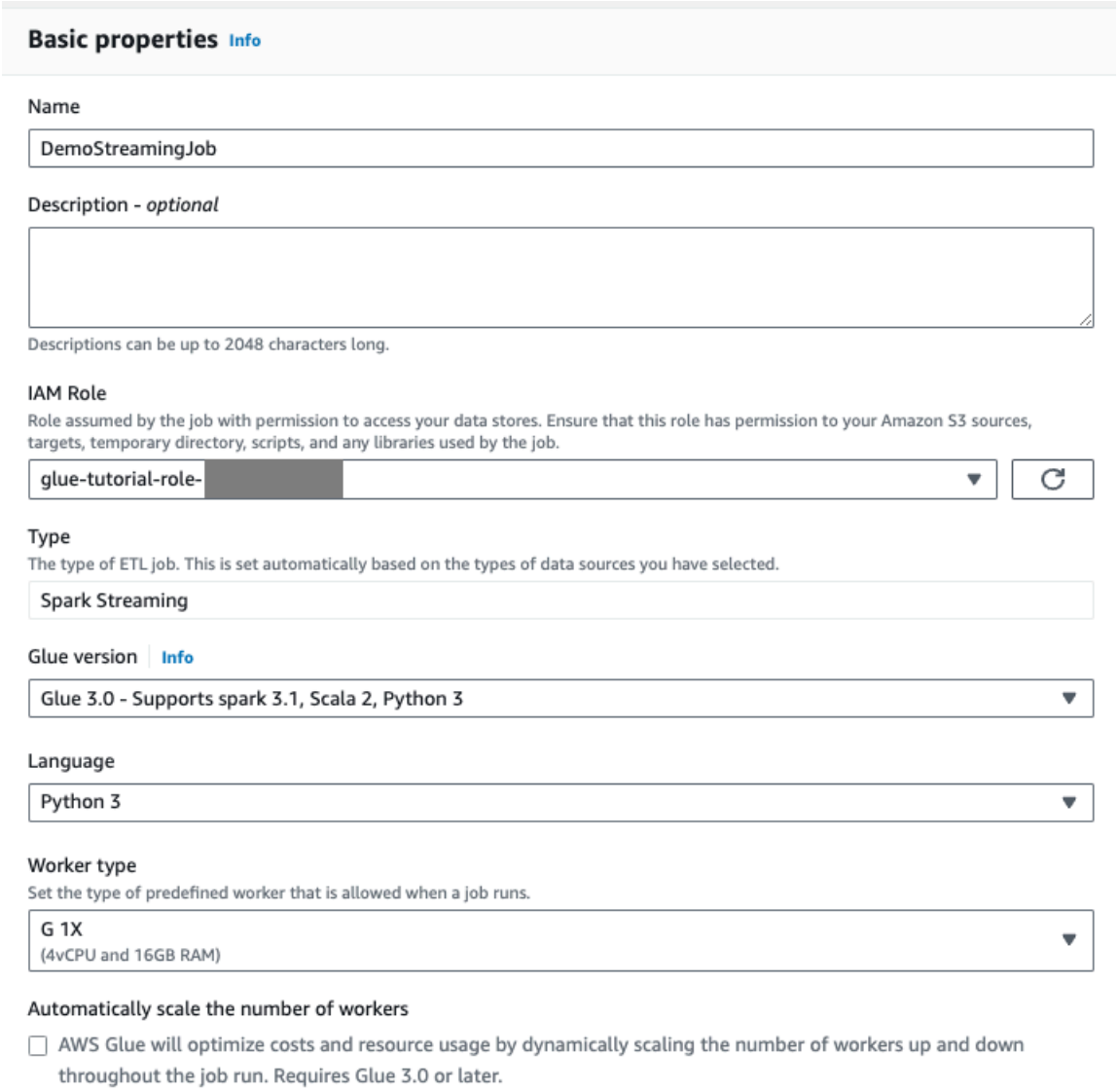
8. Fai clic su Invia dati e genera 5-10.000 record su Kinesis.

## Creazione di un lavoro AWS Glue in streaming con Studio AWS Glue

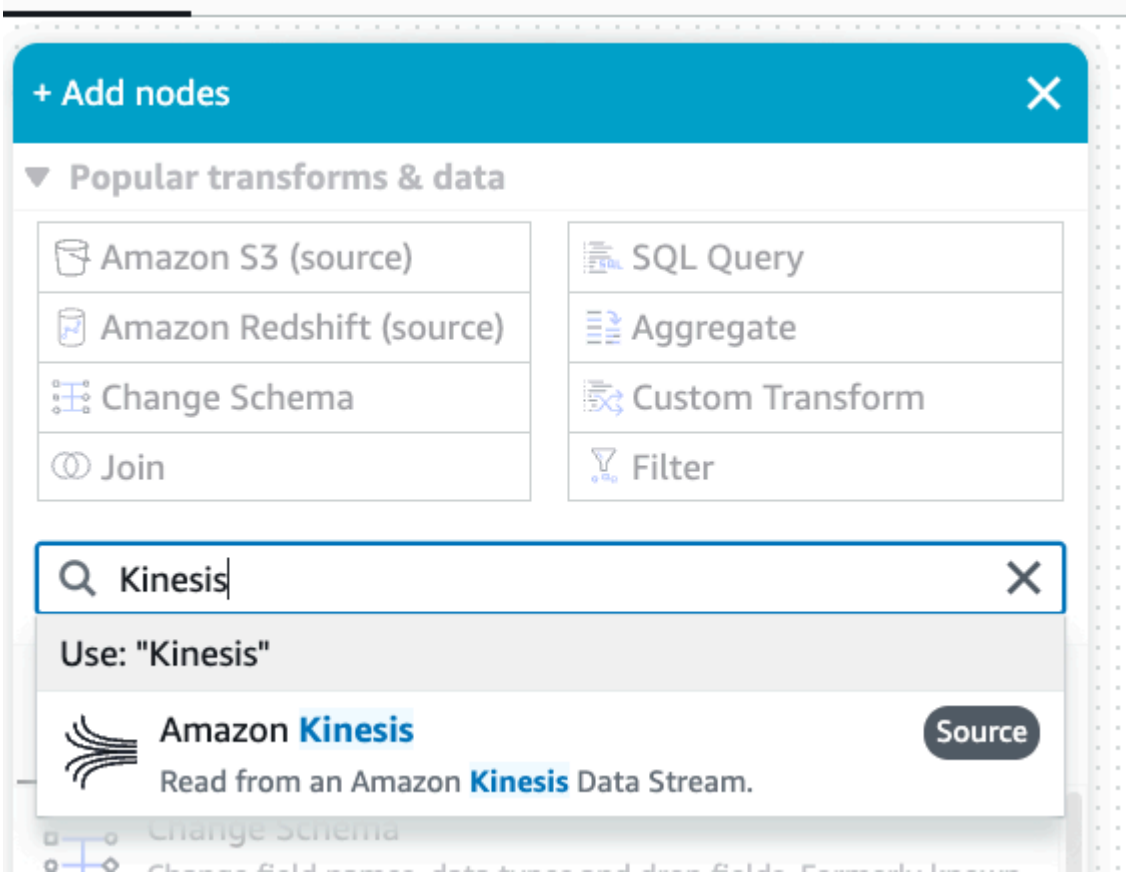
1. Passa alla AWS Glue console nella stessa regione.
2. Seleziona Processi ETL nella barra di navigazione a sinistra in Integrazione dati ed ETL.
3. Crea un AWS Glue Job tramite Visual con una tela bianca.



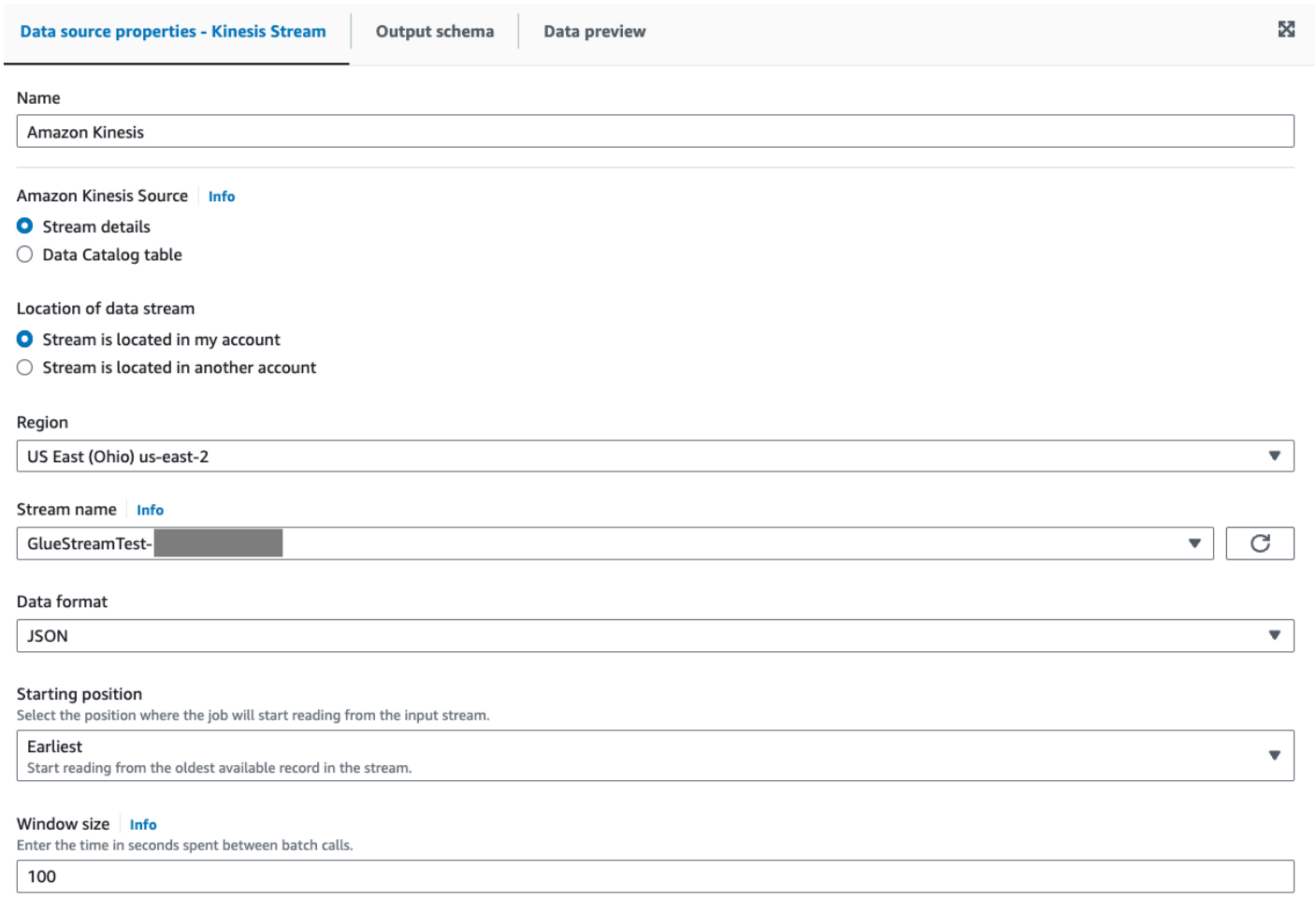

4. Passa alla scheda Dettagli del processo.
5. Per il nome del AWS Glue lavoro, immettere`DemoStreamingJob`.
6. Per IAM Role, seleziona il ruolo assegnato dal CloudFormation modello,`glue-tutorial-role-${AWS::AccountId}`.
7. Per Versione Glue, seleziona Glue 3.0. Mantieni tutte le altre opzioni predefinite.




8. Vai alla scheda Visivo.
9. Fai clic sull'icona del segno più. Immetti Kinesis nella barra di ricerca. Seleziona l'origine dati Amazon Kinesis.



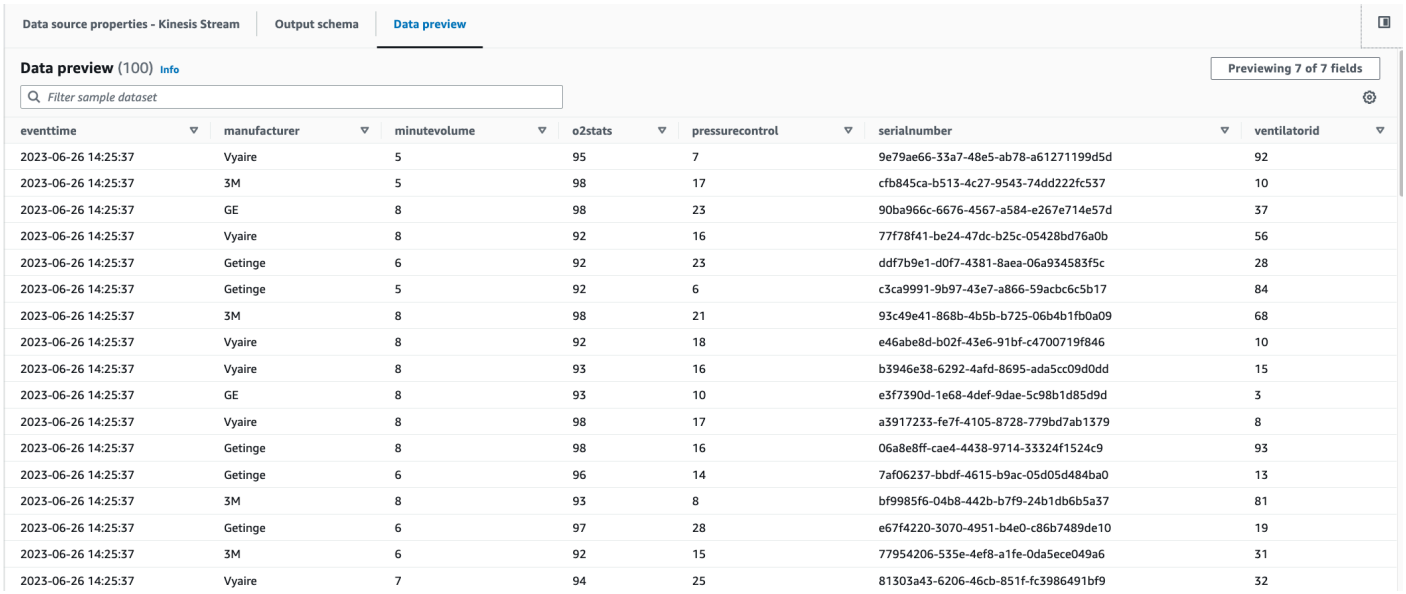

10. Seleziona Dettagli del flusso per Origine Amazon Kinesis nella scheda Proprietà dell'origine dati - Flusso Kinesis.
11. Seleziona Il flusso si trova nel mio account per Posizione del flusso di dati.
12. Seleziona la regione che stai utilizzando.
13. Seleziona il flusso `GlueStreamTest-{AWS::AccountId}`.
14. Mantieni tutte le altre impostazioni predefinite.




15. Vai alla scheda Anteprima dei dati.
16. Fai clic su Avvia sessione di anteprima dei dati, che visualizza in anteprima i dati fittizi generati da KDG. Scegli il ruolo del servizio Glue che hai creato in precedenza per il lavoro AWS Glue Streaming.

    Occorrono 30-60 secondi prima che i dati di anteprima vengano visualizzati. Se compare Nessun dato da visualizzare, fai clic sull'icona a forma di ingranaggio e imposta il Numero di righe in base al quale campionare su `100`.

    Puoi visualizzare i dati di esempio come segue:



Data source properties - Kinesis Stream | Output schema | Data preview

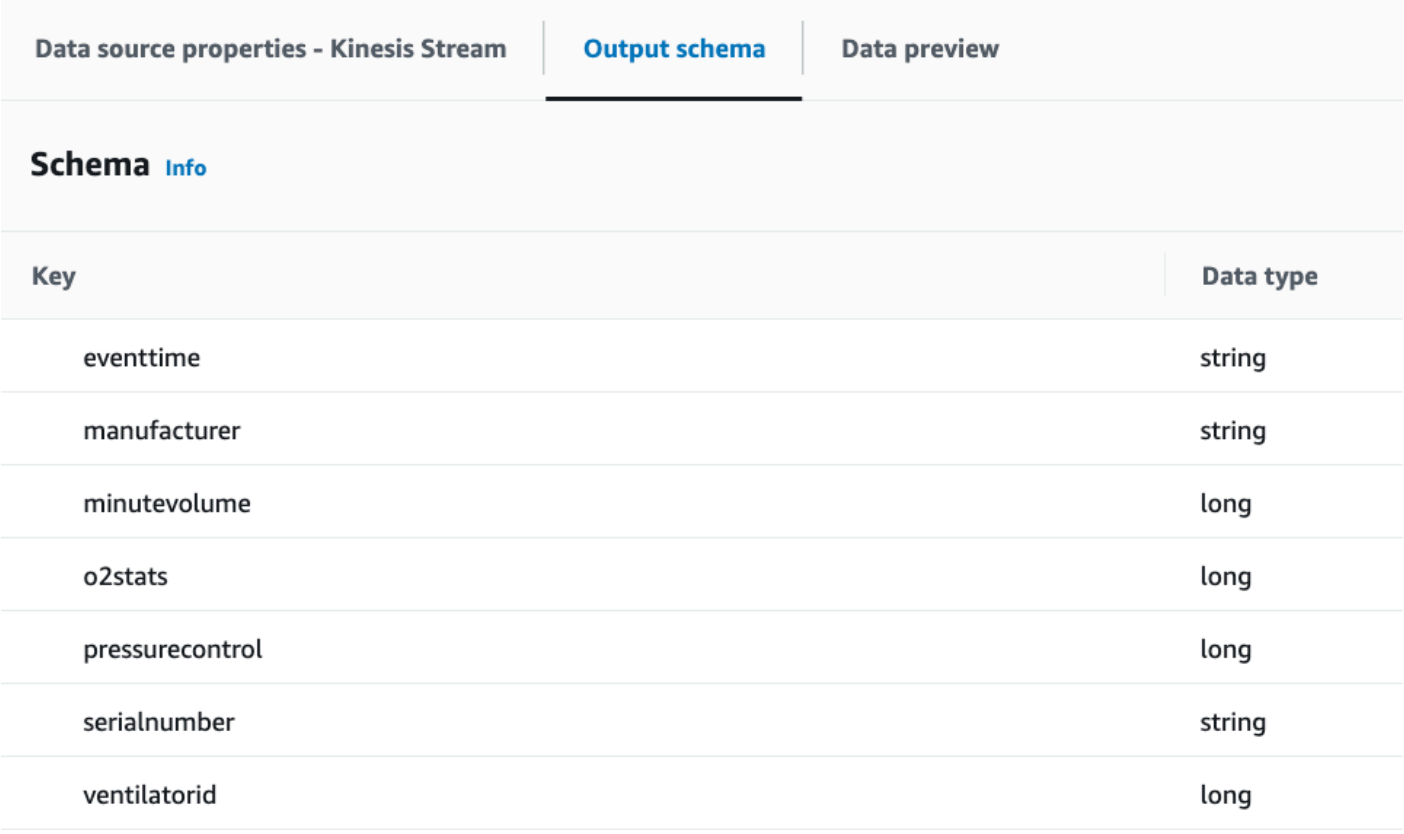
**Data preview** (100) Info

Previewing 7 of 7 fields

Filter sample dataset

| eventtime | manufacturer | minutevolume | o2stats | pressurecontrol | serialnumber | ventilatorid |
|---|---|---|---|---|---|---|
| 2023-06-26 14:25:37 | Vyaire | 5 | 95 | 7 | 9e79ae66-33a7-48e5-ab78-a61271199d5d | 92 |
| 2023-06-26 14:25:37 | 3M | 5 | 98 | 17 | cfb845ca-b513-4c27-9543-74dd222fc537 | 10 |
| 2023-06-26 14:25:37 | GE | 8 | 98 | 23 | 90ba966c-6676-4567-a584-e267e714e57d | 37 |
| 2023-06-26 14:25:37 | Vyaire | 8 | 92 | 16 | 77f78f41-be24-47dc-b25c-05428bd76a0b | 56 |
| 2023-06-26 14:25:37 | Getinge | 6 | 92 | 23 | ddf7b9e1-d0f7-4381-8aea-06a934583f5c | 28 |
| 2023-06-26 14:25:37 | Getinge | 5 | 92 | 6 | c3ca9991-9b97-43e7-a866-59acbc6c5b17 | 84 |
| 2023-06-26 14:25:37 | 3M | 8 | 98 | 21 | 93c49e41-868b-4b5b-b725-06b4b1fb0a09 | 68 |
| 2023-06-26 14:25:37 | Vyaire | 8 | 92 | 18 | e46abe8d-b02f-43e6-91bf-c4700719f846 | 10 |
| 2023-06-26 14:25:37 | Vyaire | 8 | 93 | 16 | b3946e38-6292-4afd-8695-ada5cc09d0dd | 15 |
| 2023-06-26 14:25:37 | GE | 8 | 93 | 10 | e3f7390d-1e68-4def-9dae-5c98b1d85d9d | 3 |
| 2023-06-26 14:25:37 | Vyaire | 8 | 98 | 17 | a3917233-fe7f-4105-8728-779bd7ab1379 | 8 |
| 2023-06-26 14:25:37 | Getinge | 8 | 98 | 16 | 06a8e8ff-cae4-4438-9714-33324f1524c9 | 93 |
| 2023-06-26 14:25:37 | Getinge | 6 | 96 | 14 | 7af06237-bbdf-4615-b9ac-05d05d484ba0 | 13 |
| 2023-06-26 14:25:37 | 3M | 8 | 93 | 8 | bf9985f6-04b8-442b-b7f9-24b1db6b5a37 | 81 |
| 2023-06-26 14:25:37 | Getinge | 6 | 97 | 28 | e67f4220-3070-4951-b4e0-c86b7489de10 | 19 |
| 2023-06-26 14:25:37 | 3M | 6 | 92 | 15 | 77954206-535e-4ef8-a1fe-0da5ece049a6 | 31 |
| 2023-06-26 14:25:37 | Vyaire | 7 | 94 | 25 | 81303a43-6206-46cb-851f-fc3986491bf9 | 32 |

È inoltre possibile visualizzare lo schema dedotto nella scheda Schema di output.



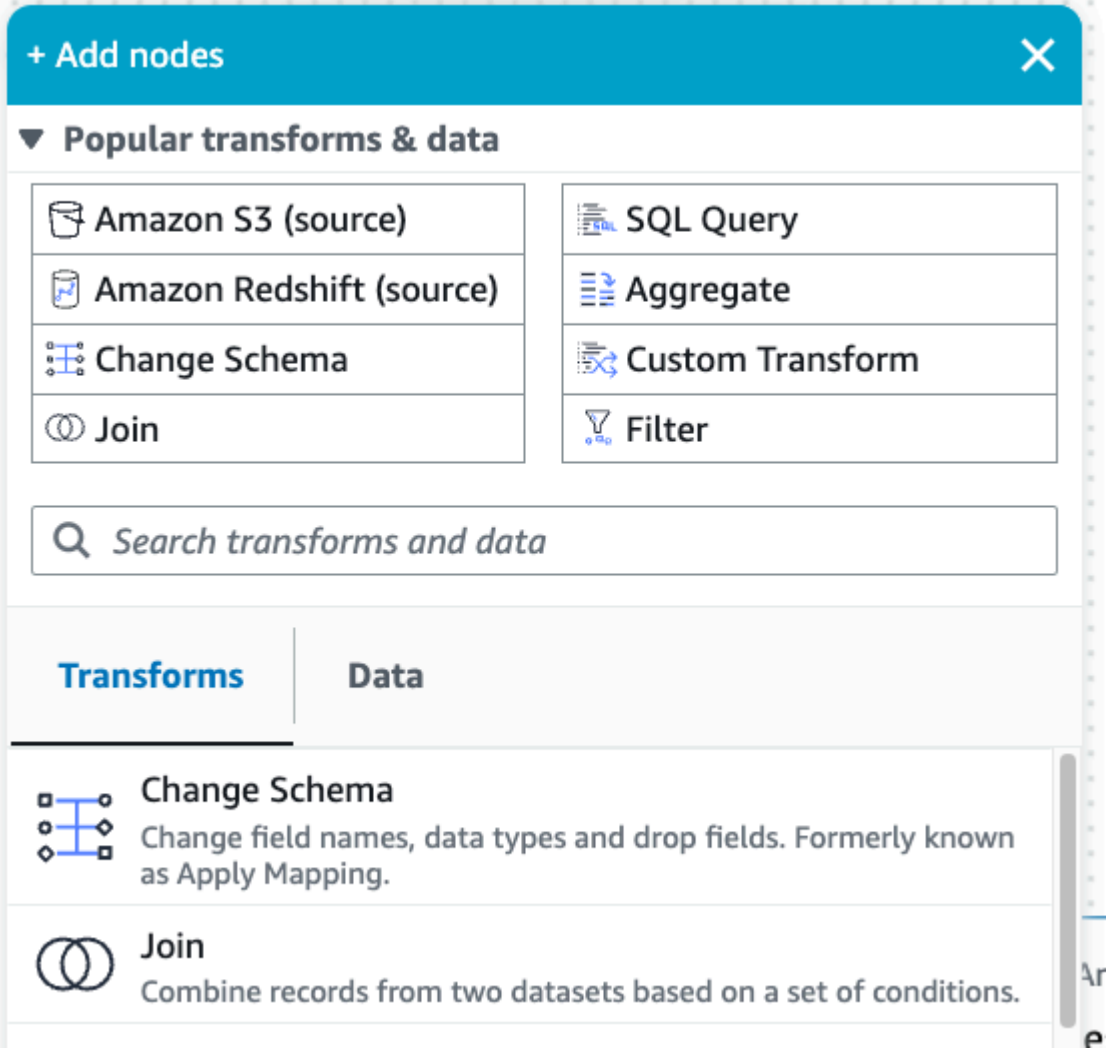
Data source properties - Kinesis Stream | Output schema | Data preview

**Schema** Info

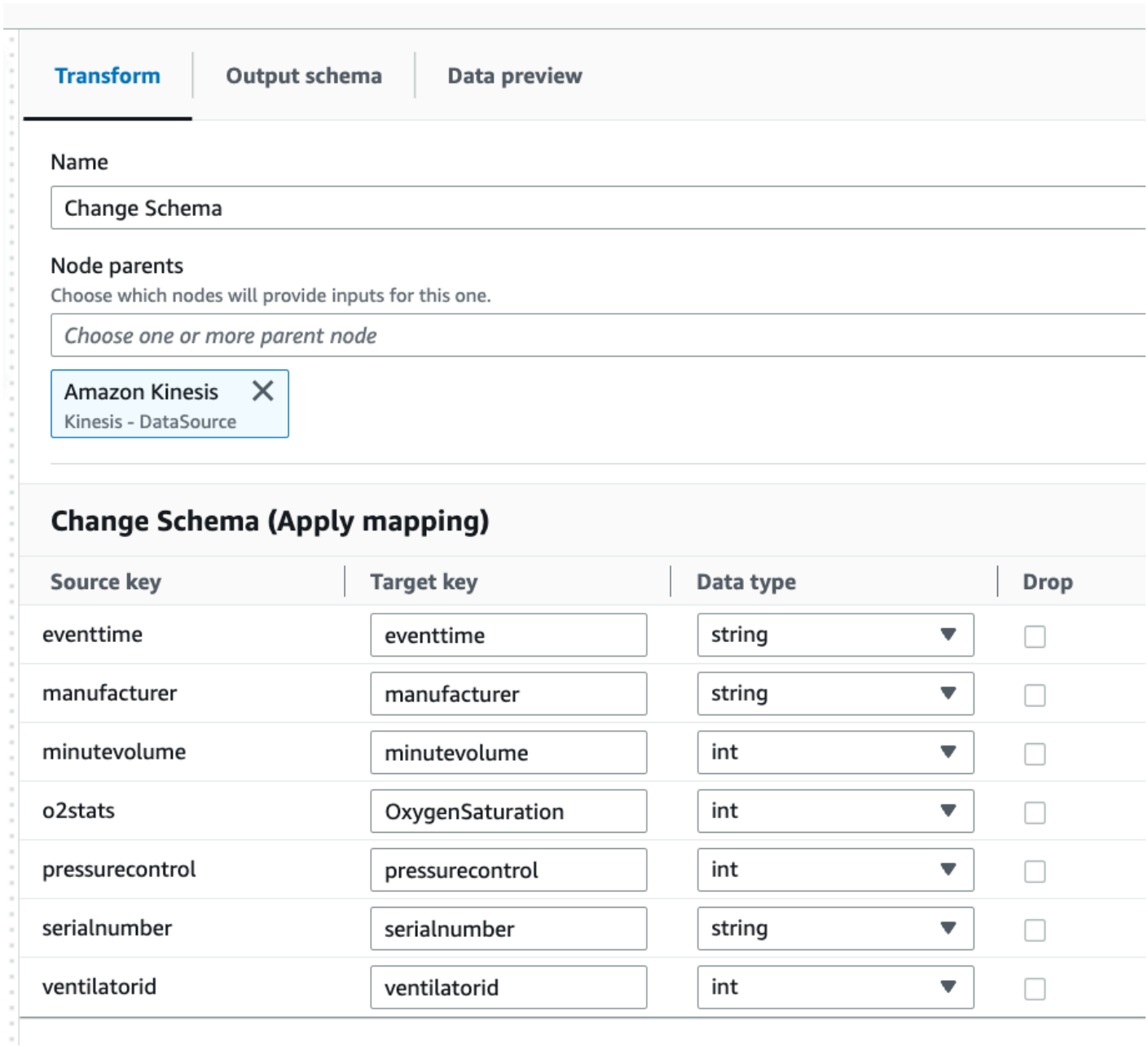
| Key | Data type |
|---|---|
| eventtime | string |
| manufacturer | string |
| minutevolume | long |
| o2stats | long |
| pressurecontrol | long |
| serialnumber | string |
| ventilatorid | long |

## Esecuzione di una trasformazione e archiviazione del risultato della trasformazione in Amazon S3

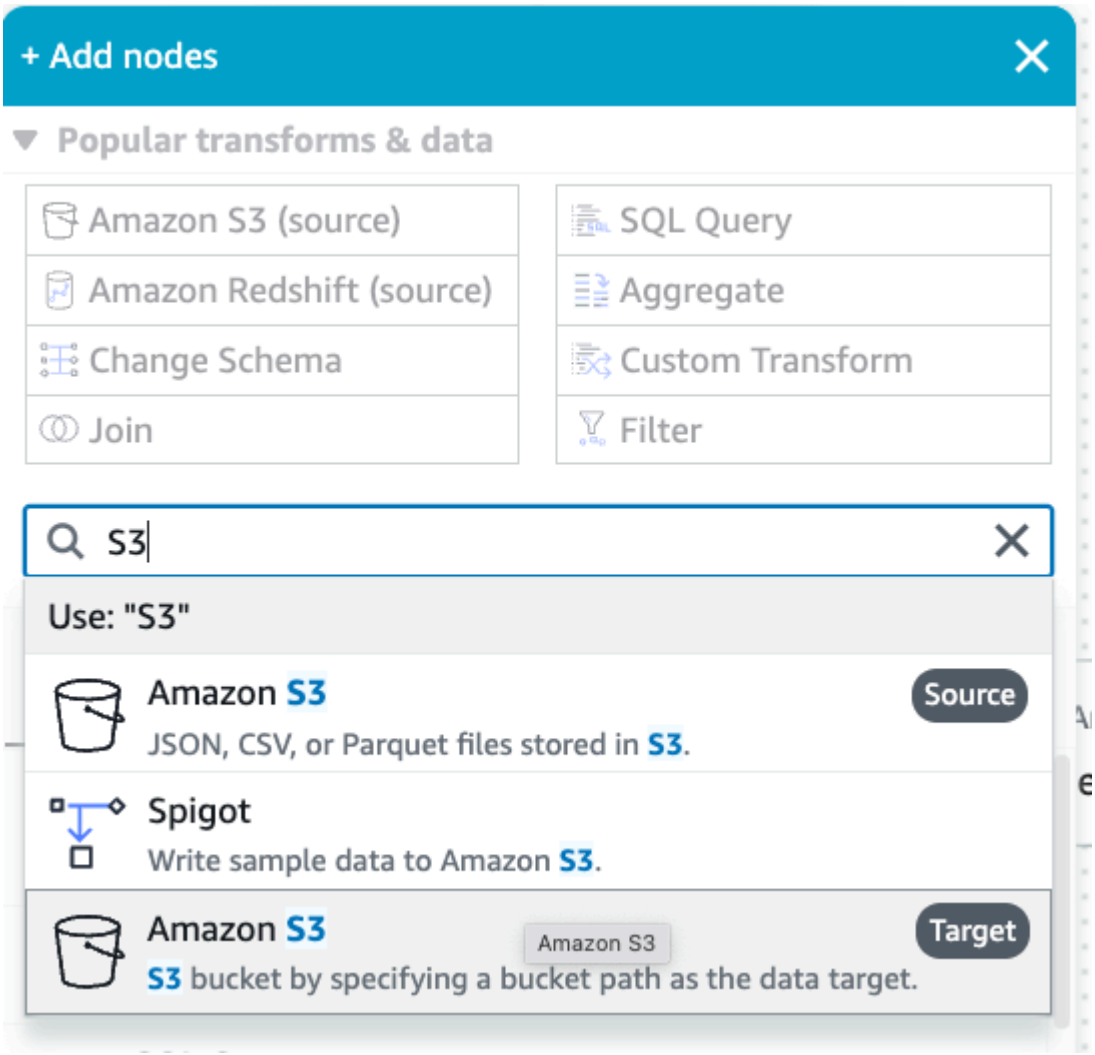
1. Con il nodo di origine selezionato, fai clic sull'icona del segno più in alto a sinistra per aggiungere un passaggio Trasformazioni.
2. Seleziona il passaggio Modifica schema.



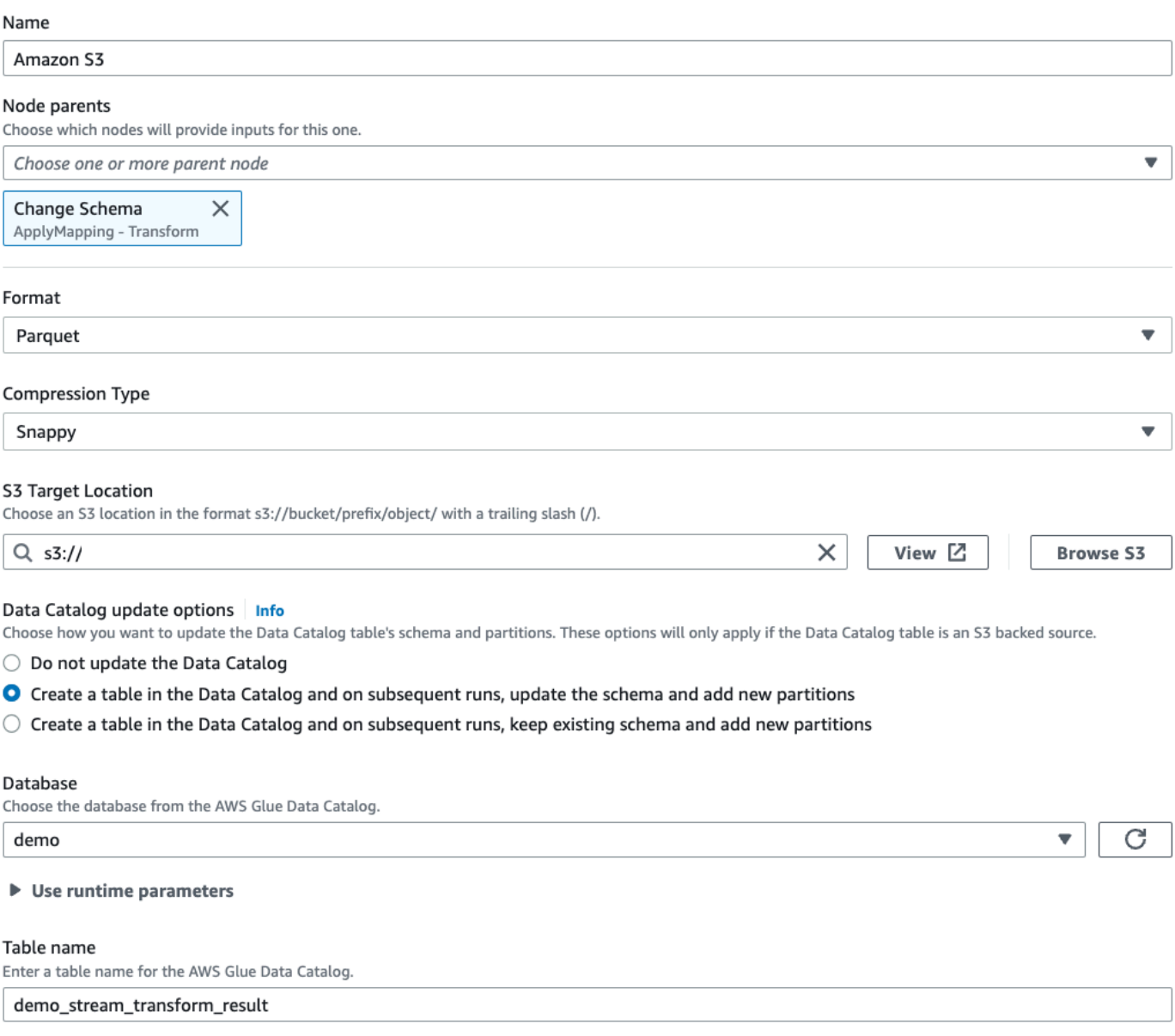

3. In questo passaggio è possibile rinominare i campi e convertire il tipo di dati dei campi. Rinomina la colonna `o2stats` in `OxygenSaturation` e converti tutti i tipi di dati `long` in `int`.




4. Fai clic sull'icona del segno più per aggiungere una destinazione Amazon S3. Immetti S3 nella casella di ricerca e seleziona la fase di trasformazione di Amazon S3 - Destinazione.

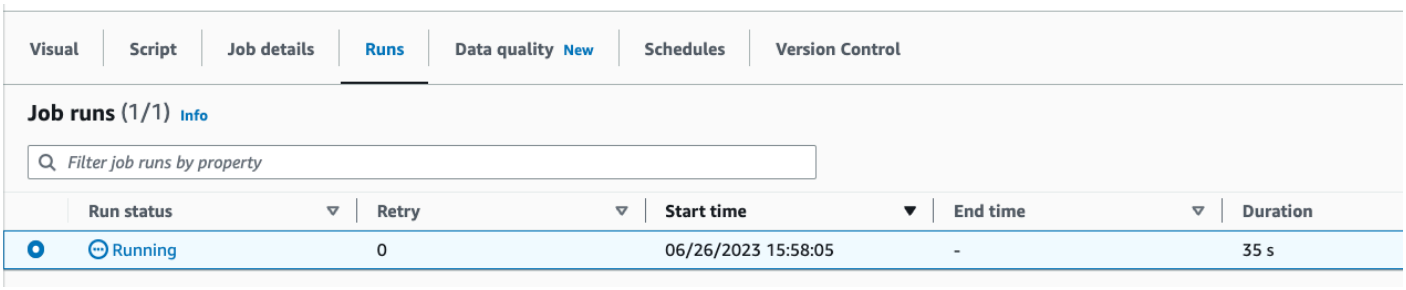



5. Seleziona Parquet come formato del file di destinazione.
6. Seleziona Snappy come tipo di compressione.
7. Inserisci una posizione di destinazione S3 creata dal CloudFormation modello,`streaming-tutorial-s3-target-{AWS::AccountId}`.
8. Seleziona Crea una tabella nel Catalogo dati e, nelle esecuzioni successive, aggiorna lo schema e aggiungi nuove partizioni.
9. Inserisci il nome del Database e della Tabella di destinazione per archiviare lo schema della tabella di destinazione Amazon S3.

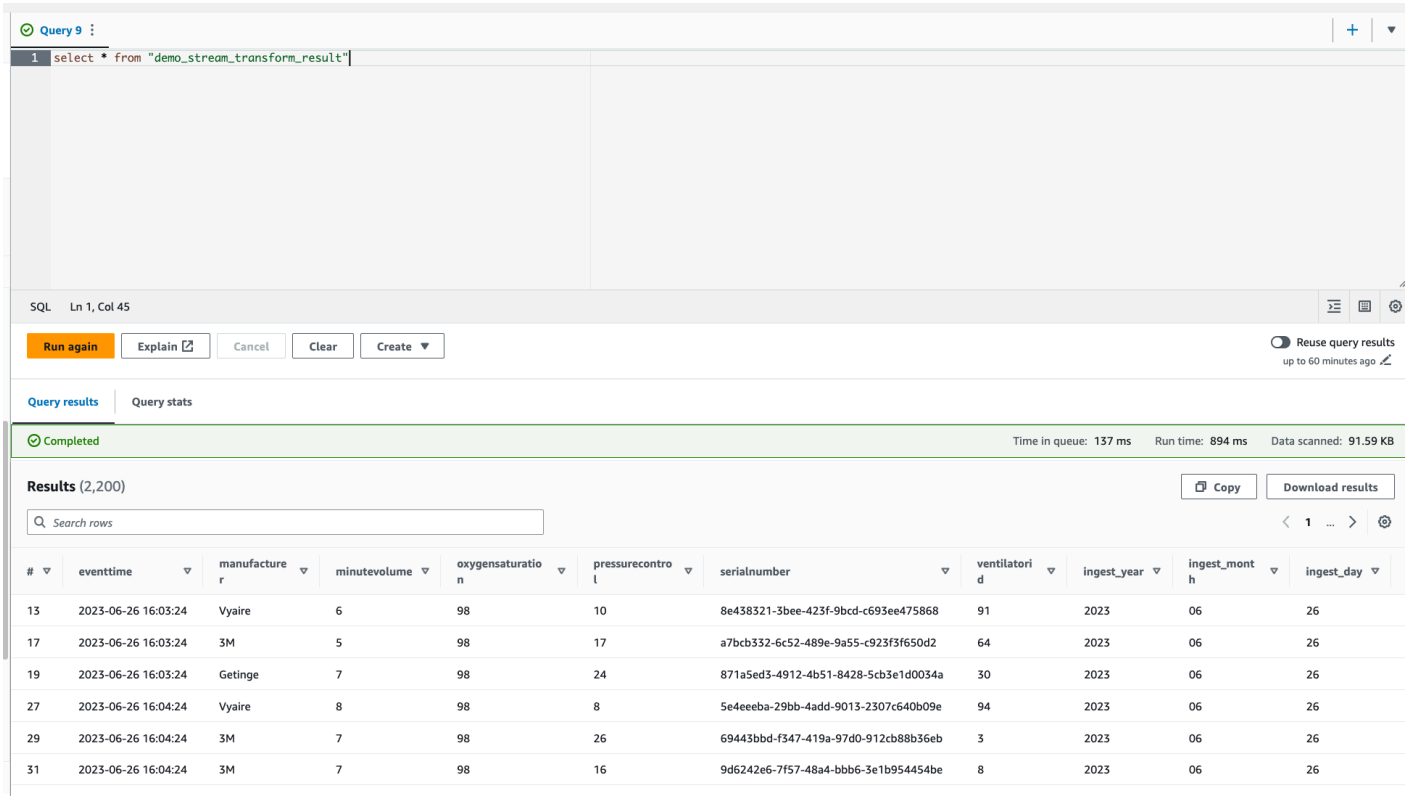



10. Fai clic sulla scheda Script per visualizzare il codice generato.
11. Fai clic su Salva in alto a destra per salvare il codice ETL, quindi fai clic su Esegui per avviare il processo di streaming. AWS Glue

    Puoi trovare lo Stato di esecuzione nella scheda Esecuzioni. Lascia che il processo venga eseguito per 3-5 minuti, quindi interrompilo.




12.Verifica la nuova tabella creata in Amazon Athena.




# Tutorial: crea il tuo primo carico di lavoro in streaming utilizzando i notebook AWS Glue Studio

In questo tutorial, scoprirai come sfruttare i notebook AWS Glue Studio per creare e perfezionare in modo interattivo i tuoi job ETL per un'elaborazione dei dati quasi in tempo reale. Che siate alle prime armi AWS Glue o che vogliate migliorare le vostre competenze, questa guida vi guiderà attraverso il processo, consentendovi di sfruttare tutto il potenziale dei taccuini interattivi con sessioni. AWS Glue

Con AWS Glue Streaming, puoi creare processi di estrazione, trasformazione e caricamento (ETL) in streaming che vengono eseguiti in modo continuo e utilizzano dati da fonti di streaming come

Amazon Kinesis Data Streams, Apache Kafka e Amazon Managed Streaming for Apache Kafka (Amazon MSK).

## Prerequisiti

Per seguire questo tutorial avrai bisogno di un utente con le autorizzazioni di utilizzo AWS della console, Amazon Kinesis AWS Glue, Amazon S3, Amazon Athena, AWS CloudFormation AWS Lambda e Amazon Cognito.

## Utilizzo dei dati in streaming da Amazon Kinesis

Argomenti

- Generazione di dati fittizi con Kinesis Data Generator
- Creazione di un lavoro AWS Glue in streaming con Studio AWS Glue
- Eliminazione
- Conclusioni

### Generazione di dati fittizi con Kinesis Data Generator

> **Note**
>
> Se hai già completato i passaggi del precedente Tutorial: crea il tuo primo carico di lavoro in streaming con Studio AWS Glue e hai già installato Kinesis Data Generator sull'account, puoi saltare i passaggi da 1 a 8 riportati di seguito e andare direttamente alla sezione Creazione di un lavoro AWS Glue in streaming con Studio AWS Glue.

È possibile generare sinteticamente dati di esempio in formato JSON utilizzando Kinesis Data Generator (KDG). Puoi trovare le istruzioni complete e i dettagli nella documentazione dello strumento.

1. Per iniziare, fai clic per eseguire un modello

   nel tuo ambiente. AWS CloudFormation AWS

> **Note**
>
> Potresti riscontrare un errore nel CloudFormation modello perché alcune risorse, come l'utente Amazon Cognito per Kinesis Data Generator, esistono già nel tuo account. AWS Ciò potrebbe essere dovuto al fatto che l'hai già configurato in un altro tutorial o da un post di un blog. Per risolvere questo problema, puoi provare il modello in un nuovo AWS account per ricominciare da capo, oppure esplorare un'altra AWS regione. Queste opzioni consentono di eseguire il tutorial senza entrare in conflitto con le risorse esistenti.

Il modello fornisce un flusso di dati Kinesis e un account Kinesis Data Generator.

2. Immetti un Nome utente e una Password che KDG utilizzerà per l'autenticazione. Prendi nota del nome utente e della password per utilizzarli in seguito.
3. Seleziona Avanti fino all'ultimo passaggio. Esprimi il consenso alla creazione di risorse IAM. Verifica la presenza di eventuali errori nella parte superiore dello schermo, ad esempio la password che non soddisfa i requisiti minimi, e implementa il modello.
4. Vai alla scheda Output dello stack. Una volta distribuito, il modello mostrerà la proprietà KinesisDataGeneratorUrlgenerata. Fai clic su quell'URL.
5. Inserisci il Nome utente e la Password di cui hai preso nota.
6. Seleziona la regione che stai utilizzando e seleziona il flusso Kinesis `GlueStreamTest-{AWS::AccountId}`.
7. Immetti il seguente modello:

```
{
    "ventilatorid": {{random.number(100)}},
    "eventtime": "{{date.now("YYYY-MM-DD HH:mm:ss")}}",
    "serialnumber": "{{random.uuid}}",
    "pressurecontrol": {{random.number(
        {
            "min":5,
            "max":30
        }
    )}},
    "o2stats": {{random.number(
        {
            "min":92,
            "max":98
```

```
        }
    )}},
    "minutevolume": {{random.number(
        {
            "min":5,
            "max":8
        }
    )}},
    "manufacturer": "{{random.arrayElement(
        ["3M", "GE","Vyaire", "Getinge"]
    )}}"
}
```

Ora puoi visualizzare i dati fittizi con Modello di prova e importare i dati fittizi in Kinesis con Invia dati.

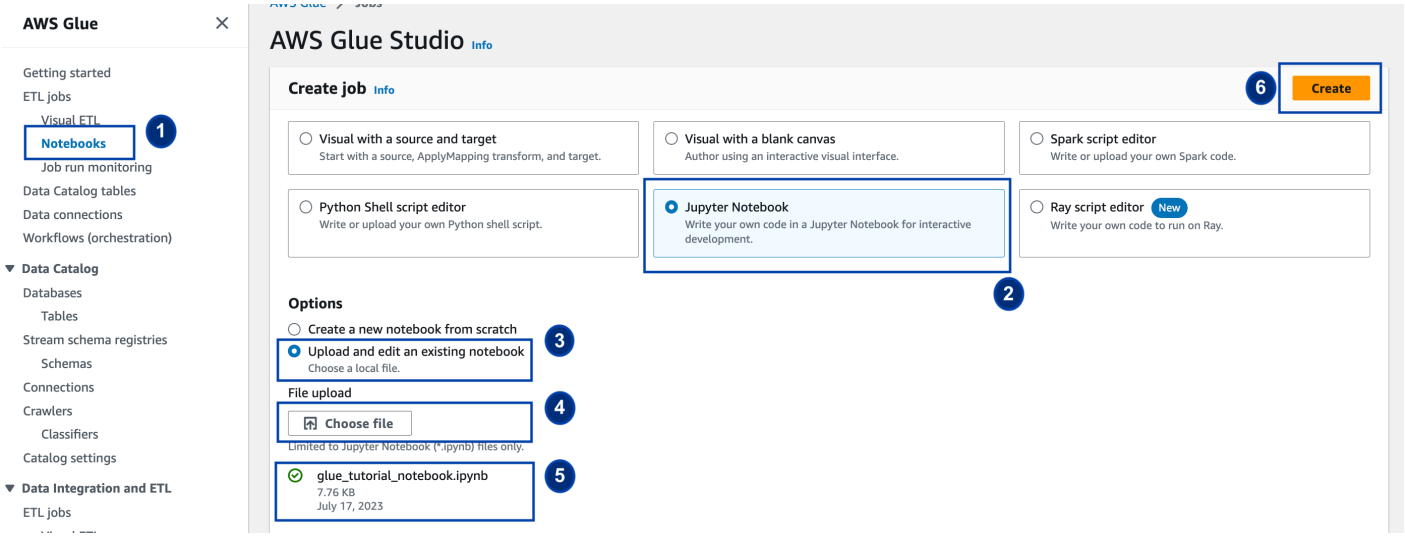
8. Fai clic su Invia dati e genera 5-10.000 record su Kinesis.

## Creazione di un lavoro AWS Glue in streaming con Studio AWS Glue

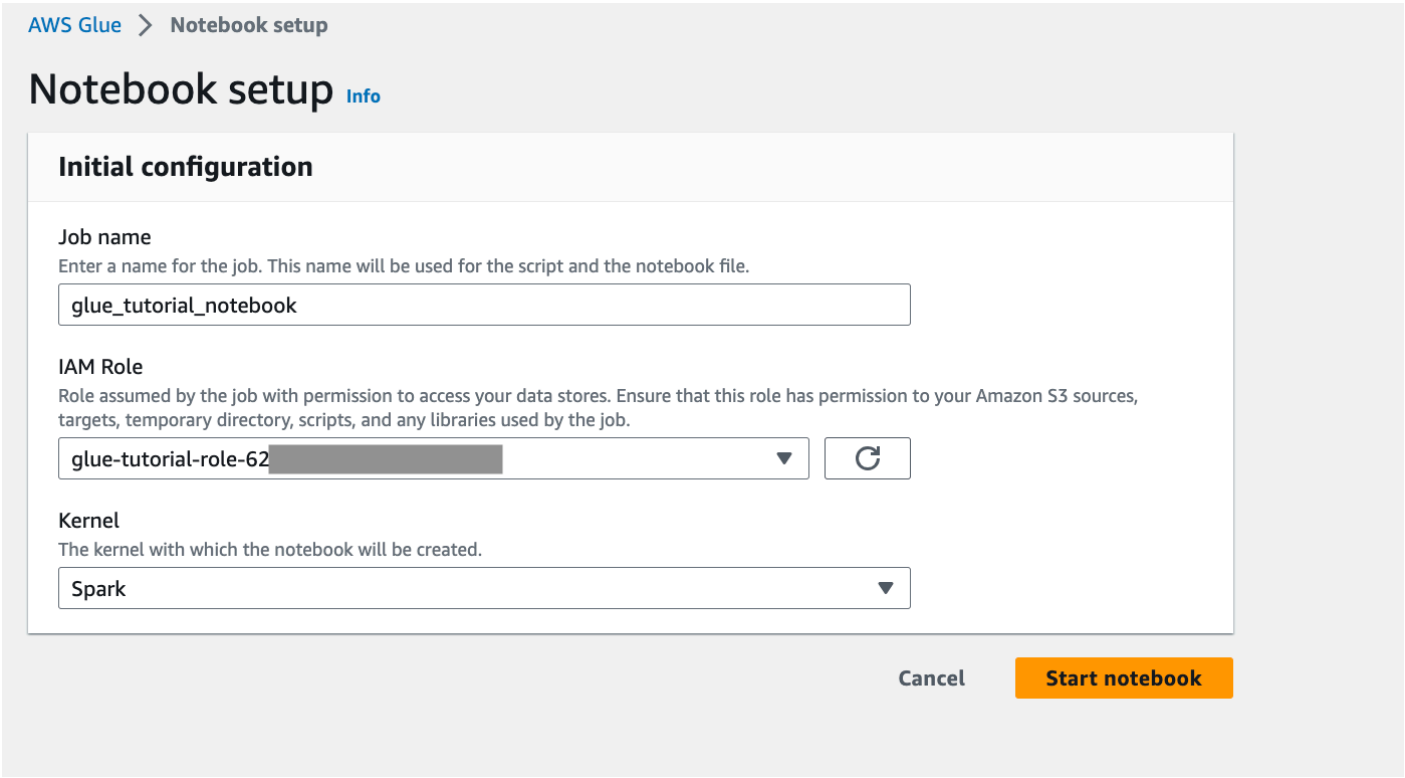
AWS Glue Studio è un'interfaccia visiva che semplifica il processo di progettazione, orchestrazione e monitoraggio delle pipeline di integrazione dei dati. Consente agli utenti di creare pipeline di trasformazione dei dati senza scrivere codice esteso. Oltre all'esperienza di creazione visiva dei lavori, AWS Glue Studio include anche un notebook Jupyter supportato da sessioni AWS Glue interattive, che utilizzerai nel resto di questo tutorial.

Configura il processo di streaming delle sessioni interattive AWS Glue

1. Scarica il file del notebook fornito e salvalo in una directory locale
2. Apri la AWS Glue console e nel riquadro sinistro fai clic su Notebooks > Jupyter Notebook > Carica e modifica un taccuino esistente. Carica il notebook dal passaggio precedente e fai clic su Crea.



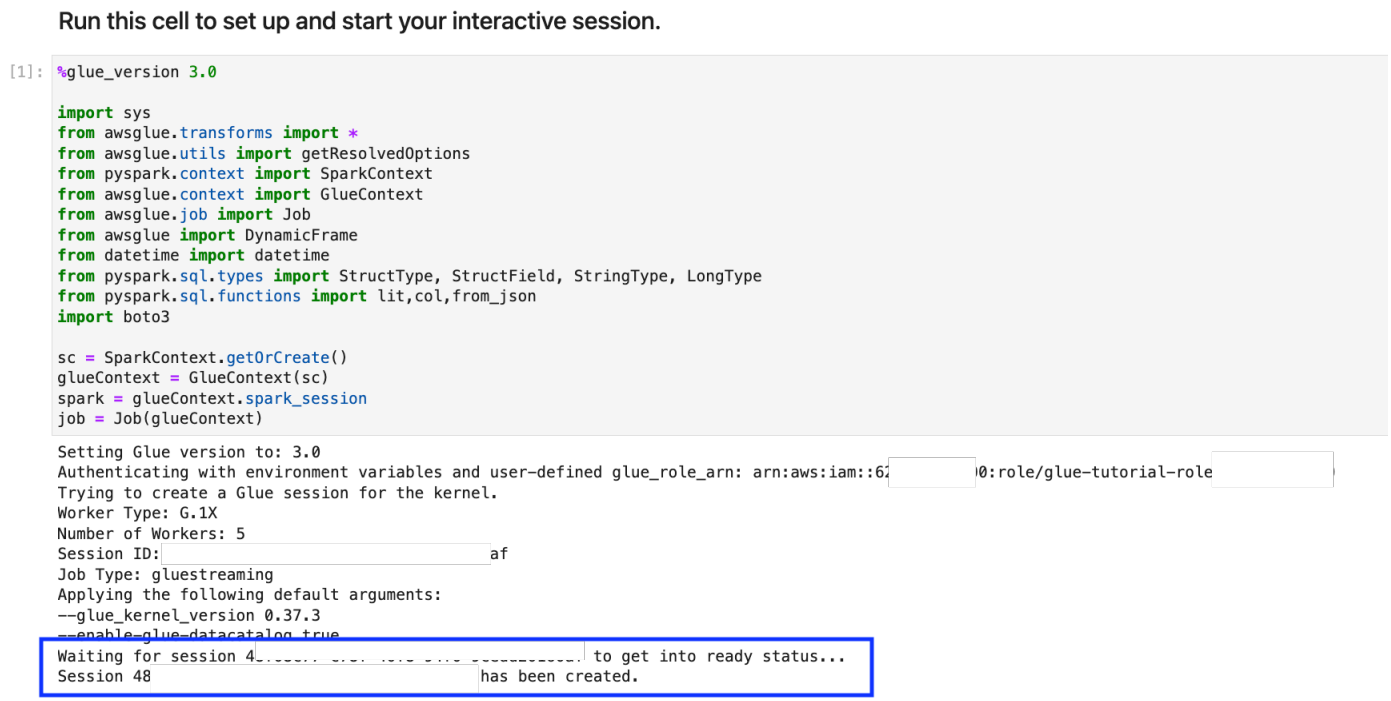

3. Fornisci un nome e un ruolo per il processo e seleziona il kernel Spark predefinito. Quindi fai clic su Avvia notebook. Per il ruolo IAM, seleziona il ruolo assegnato dal modello. CloudFormation Puoi vederlo nella scheda Output di. CloudFormation



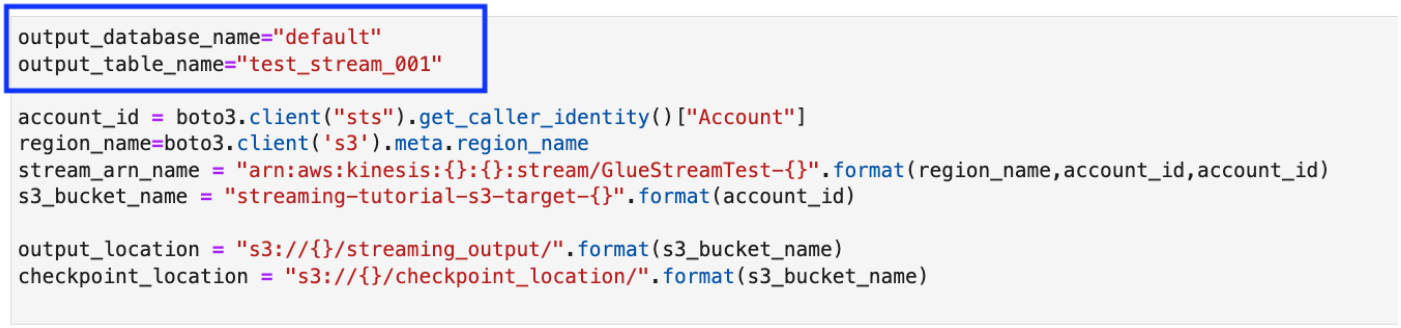

Il notebook contiene tutte le istruzioni necessarie per continuare il tutorial. Puoi eseguire le istruzioni sul notebook o seguire questo tutorial per continuare con lo sviluppo del processo.

Esecuzione delle celle del notebook

1. (Facoltativo) La prima cella di codice, `%help`, elenca tutte le funzioni magic disponibili per il notebook. Per ora puoi saltare questa cella, ma se desideri puoi esplorarla.
2. Inizia con il blocco di codice successivo, `%streaming`. Questa magia imposta il tipo di lavoro sullo streaming, che consente di sviluppare, eseguire il debug e distribuire un lavoro ETL AWS Glue in streaming.
3. Esegui la cella successiva per creare una AWS Glue sessione interattiva. La cella di output contiene un messaggio che conferma la creazione della sessione.




4. La cella successiva definisce le variabili. Sostituisci i valori con quelli appropriati per il tuo processo ed esegui la cella. Esempio:



5. Poiché i dati vengono già trasmessi in streaming a Flussi di dati Kinesis, la cella successiva utilizzerà i risultati del flusso. Esegui la cella successiva. Poiché non ci sono istruzioni di stampa, non è previsto alcun output per questa cella.
6. Nella cella seguente, esplori il flusso in entrata prelevando un set di esempio e stampandone lo schema e i dati effettivi. Esempio:

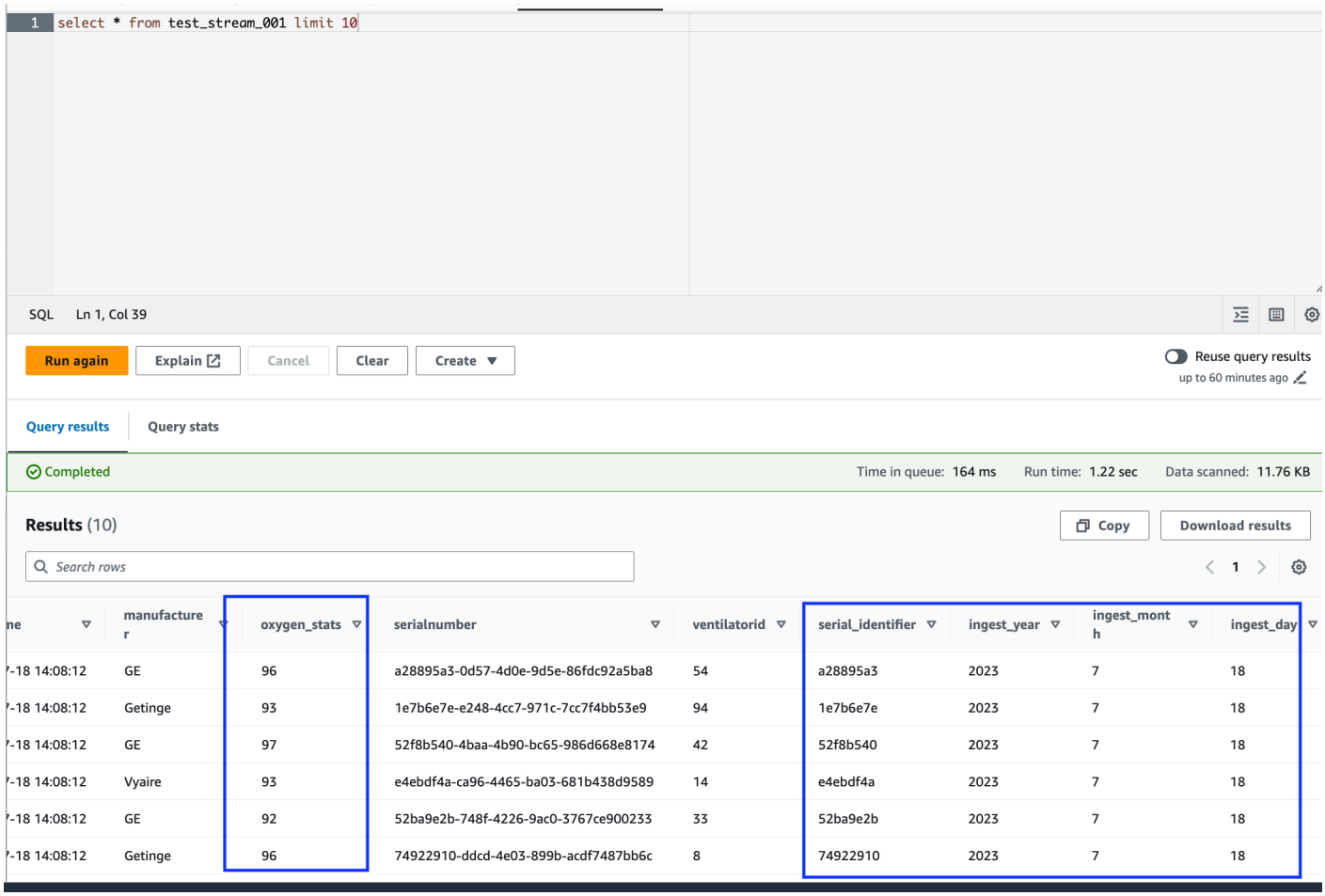
**Sample and print the incoming records**

**the sampling is for debugging purpose. You may comment off the entire code cell below, before deploying the actual code**

```
[4]: options = {
        "pollingTimeInMs": "20000",
        "windowSize": "5 seconds"
     }
     sampled_dynamic_frame = glueContext.getSampleStreamingDynamicFrame(data_frame, options, None)

     count_of_sampled_records = sampled_dynamic_frame.count()

     print(count_of_sampled_records)

     sampled_dynamic_frame.printSchema()

     sampled_dynamic_frame.toDF().show(10,False)
```

```
100
root
|-- eventtime: string
|-- manufacturer: string
|-- minutevolume: long
|-- o2stats: long
|-- pressurecontrol: long
|-- serialnumber: string
|-- ventilatorid: long

+-------------------+------------+------------+-------+---------------+------------------------------------+------------+
|eventtime          |manufacturer|minutevolume|o2stats|pressurecontrol|serialnumber                        |ventilatorid|
+-------------------+------------+------------+-------+---------------+------------------------------------+------------+
|2023-07-18 10:20:11|3M          |6           |92     |24             |a3e860ba-24b9-41c4-bc10-91c6b35e1406|6           |
|2023-07-18 10:20:11|Vyaire      |6           |95     |6              |96101dca-3e88-457f-b390-e3291df48a81|26          |
|2023-07-18 10:20:12|Getinge     |8           |96     |24             |18f3d448-1dee-4c80-835b-1a0daa818915|22          |
|2023-07-18 10:20:12|Getinge     |7           |98     |30             |25f425cd-b978-4953-9a03-4d607a639364|91          |
|2023-07-18 10:20:12|GE          |5           |93     |25             |2cd7cdc2-f5f5-4ff2-ae32-45e5a8922d53|93          |
```

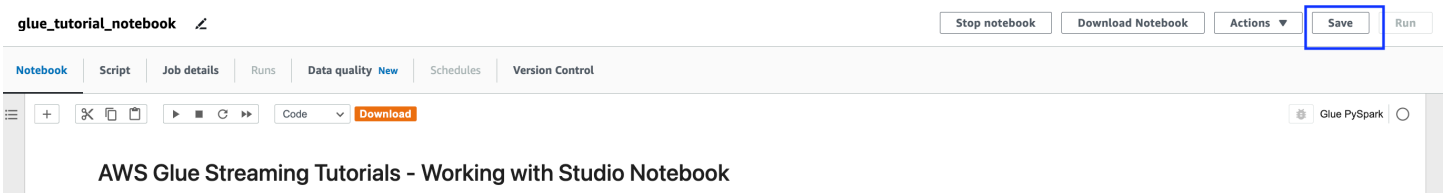
7. Successivamente, definisci la logica di trasformazione dei dati effettiva. La cella è costituita dal metodo `processBatch` che viene attivato durante ogni microbatch. Esegui la cella. Ad alto livello, eseguiamo le operazioni seguenti per il flusso in entrata:
   a. Seleziona un sottoinsieme delle colonne di input.
   b. Rinomina una colonna (o2stats in oxygen_stats).
   c. Ricava nuove colonne (serial_identifier, ingest_year, ingest_month e ingest_day).
   d. Archivia i risultati in un bucket Amazon S3 e crea anche una tabella di catalogo partizionata AWS Glue
8. Nell'ultima cella, il batch di processo si attiva ogni 10 secondi. Esegui la cella e attendi circa 30 secondi affinché compili il bucket Amazon S3 e AWS Glue la tabella del catalogo.
9. Infine, esplora i dati archiviati utilizzando l'editor di query di Amazon Athena. Puoi visualizzare la colonna rinominata e le nuove partizioni.




Il notebook contiene tutte le istruzioni necessarie per continuare il tutorial. Puoi eseguire le istruzioni sul notebook o seguire questo tutorial per continuare con lo sviluppo del processo.

## Salva ed esegui il lavoro AWS Glue

Una volta completato lo sviluppo e il test dell'applicazione utilizzando il notebook delle sessioni interattive, fai clic su Salva nella parte superiore dell'interfaccia del notebook. Una volta salvata l'applicazione, puoi anche eseguirla come processo.



### Eliminazione

Per evitare addebiti aggiuntivi sul tuo account, interrompi il processo di streaming che hai avviato seguendo le istruzioni. Puoi farlo arrestando il notebook, operazione che termina la sessione. Svuota il bucket Amazon S3 ed elimina lo AWS CloudFormation stack che hai fornito in precedenza.

### Conclusioni

In questo tutorial, abbiamo dimostrato come eseguire le seguenti operazioni utilizzando il notebook Studio AWS Glue

- Creazione di un processo di ETL in streaming utilizzando i notebook
- Visualizzazione in anteprima dei flussi di dati in entrata
- Codifica e risolvi i problemi senza dover pubblicare AWS Glue lavori
- Esamina il codice end-to-end funzionante, rimuovi eventuali errori di debug e stampa le istruzioni o le celle dal taccuino
- Pubblica il codice come lavoro AWS Glue

L'obiettivo di questo tutorial è darti un'esperienza pratica di lavoro con AWS Glue lo streaming e le sessioni interattive. Ti invitiamo a utilizzarlo come riferimento per i tuoi casi d'uso individuali di AWS Glue Streaming. Per ulteriori informazioni, consulta Nozioni di base sulle sessioni interattive AWS Glue.

# AWS Glue Concetti di streaming

Le seguenti sezioni forniscono informazioni sui concetti di AWS Glue Streaming.

Argomenti

- Anatomia di un lavoro AWS Glue in streaming

## Anatomia di un lavoro AWS Glue in streaming

AWS Glue i lavori di streaming si basano sul paradigma dello streaming Spark e sfruttano lo streaming strutturato del framework Spark. I processi di streaming effettuano costantemente il polling dall'origine dati di streaming a un intervallo di tempo specifico per recuperare i record sotto forma di microbatch. Le seguenti sezioni esaminano le diverse parti di un processo di streaming. AWS Glue

## forEachBatch

Il `forEachBatch` metodo è il punto di ingresso dell'esecuzione di un processo di streaming. AWS Glue AWS Glue streaming jobs utilizza il `forEachBatch` metodo per eseguire il polling dei dati funzionando come un iteratore che rimane attivo durante il ciclo di vita del processo di streaming e interroga regolarmente la fonte di streaming alla ricerca di nuovi dati ed elabora i dati più recenti in microbatch.

```
glueContext.forEachBatch(
    frame=dataFrame_AmazonKinesis_node1696872487972,
    batch_function=processBatch,
    options={
        "windowSize": "100 seconds",
        "checkpointLocation": args["TempDir"] + "/" + args["JOB_NAME"] + "/
checkpoint/",
    },
)
```

Configura la proprietà `frame` di `forEachBatch` per specificare un'origine di streaming. In questo esempio, il nodo di origine creato nell'area di disegno vuota durante la creazione del lavoro viene popolato con l'impostazione predefinita del lavoro. DataFrame Imposta la proprietà `batch_function` come `function` che decidi di richiamare per ogni operazione di microbatch. Per gestire la trasformazione in batch sui dati in entrata è necessario definire una funzione.

## Origine

Nella prima fase della `processBatch` funzione, il programma verifica il conteggio dei DataFrame record definiti come proprietà frame di. `forEachBatch` Il programma aggiunge un timestamp di inserimento a un valore non vuoto. DataFrame La clausola `data_frame.count()>0` determina se l'ultimo microbatch non è vuoto ed è pronto per un'ulteriore elaborazione.

```
def processBatch(data_frame, batchId):
    if data_frame.count() >0:
        AmazonKinesis_node1696872487972 = DynamicFrame.fromDF(
            glueContext.add_ingestion_time_columns(data_frame, "hour"),
            glueContext,
            "from_data_frame",
        )
```

## Mapping

La sezione successiva del programma consiste nell'applicare la mappatura. Il `Mapping.apply` metodo su Spark DataFrame consente di definire una regola di trasformazione relativa agli elementi di dati. In genere è possibile rinominare, modificare il tipo di dati o applicare una funzione personalizzata alla colonna di dati di origine e mapparli alle colonne di destinazione.

```
    #Script generated for node ChangeSchema
    ChangeSchema_node16986872679326 = ApplyMapping.apply(
        frame =  AmazonKinesis_node1696872487972,
        mappings = [
            ("eventtime", "string", "eventtime", "string"),
            ("manufacturer", "string", "manufacturer", "string"),
            ("minutevolume", "long", "minutevolume", "int"),
            ("o2stats", "long", "OxygenSaturation", "int"),
            ("pressurecontrol", "long", "pressurecontrol", "int"),
            ("serialnumber", "string", "serialnumber", "string"),
```

```
            ("ventilatorid", "long", "ventilatorid", "long"),
            ("ingest_year", "string", "ingest_year", "string"),
            ("ingest_month", "string", "ingest_month", "string"),
            ("ingest_day", "string", "ingest_day", "string"),
            ("ingest_hour", "string", "ingest_hour", "string"),
        ],
        transformation_ctx="ChangeSchema_node16986872679326",
    )
        )
```

## Sink

In questa sezione, il set di dati in entrata dall'origine di streaming viene archiviato in una posizione di destinazione. In questo esempio scriveremo i dati in una posizione Amazon S3. I dettagli della proprietà `AmazonS3_node_path` sono precompilati in base alle impostazioni utilizzate durante la creazione del processo dal canvas. È possibile impostare `updateBehavior` in base al proprio caso d'uso e decidere di non aggiornare la tabella del Catalogo dati, creare il Catalogo dati e aggiornare il relativo schema nelle esecuzioni successive oppure creare una tabella di catalogo e non aggiornare la definizione dello schema nelle esecuzioni successive.

La proprietà `partitionKeys` definisce l'opzione della partizione di archiviazione. Il comportamento predefinito consiste nel partizionare i dati in base al valore `ingestion_time_columns` fornito nella sezione di origine. La proprietà `compression` consente di impostare l'algoritmo di compressione da applicare durante la scrittura della destinazione. È possibile impostare la tecnica di compressione su Snappy, LZO o GZIP. La proprietà `enableUpdateCatalog` controlla se la tabella del catalogo AWS Glue deve essere aggiornata. Le opzioni disponibili per questa proprietà sono `True` o `False`.

```
    #Script generated for node Amazon S3
    AmazonS3_node1696872743449 = glueContext.getSink(
        path =  AmazonS3_node1696872743449_path,
        connection_type = "s3",
        updateBehavior = "UPDATE_IN_DATABASE",
        partitionKeys = ["ingest_year", "ingest_month", "ingest_day", "ingest_hour"],
        compression = "snappy",
        enableUpdateCatalog = True,
        transformation_ctx = "AmazonS3_node1696872743449",
    )
```

### AWS Glue Lavello da catalogo

Questa sezione del lavoro controlla il comportamento di aggiornamento della tabella del AWS Glue catalogo. Imposta `catalogDatabase` una `catalogTableName` proprietà in base al nome del database del AWS Glue catalogo e al nome della tabella associata al AWS Glue lavoro che stai progettando. È possibile definire il formato di file dei dati di destinazione tramite la proprietà `setFormat`. Per questo esempio, i dati verranno archiviati in formato Parquet.

Una volta configurato ed eseguito il processo di AWS Glue streaming che fa riferimento a questo tutorial, i dati di streaming prodotti Amazon Kinesis Data Streams verranno archiviati nella sede di Amazon S3 in un formato parquet con compressione rapida. Una volta eseguito correttamente il processo di streaming, potrai interrogare i dati tramite Amazon Athena.

```
AmazonS3_node1696872743449 = setCatalogInfo(
    catalogDatabase =  "demo", catalogTableName = "demo_stream_transform_result"
)
AmazonS3_node1696872743449.setFormat("glueparquet")
AmazonS3_node1696872743449.writeFormat("ChangeSchema_node16986872679326")
)
```

## AWS Glue Connessioni in streaming

Le seguenti sezioni forniscono informazioni su come utilizzare le connessioni in AWS Glue Streaming.

Argomenti

- Utilizzo delle connessioni Kafka
- Utilizzo delle connessioni Kinesis

### Utilizzo delle connessioni Kafka

È possibile usare una connessione Kafka per leggere e scrivere su flussi di dati Kafka utilizzando le informazioni archiviate in una tabella del Catalogo dati o fornendo informazioni per accedere direttamente al flusso di dati. La connessione supporta un cluster Kafka o un cluster Streaming

gestito da Amazon per Apache Kafka. Puoi leggere le informazioni di Kafka in uno Spark DataFrame, quindi convertirle in un Glue. AWS DynamicFrame Puoi scrivere su Kafka DynamicFrames in formato JSON. Se si accede direttamente al flusso di dati, utilizzare queste opzioni per fornire le informazioni su come accedere al flusso di dati.

Se si utilizza `getCatalogSource` o `create_data_frame_from_catalog` per consumare i record da una sorgente di streaming Kafka oppure `getCatalogSink` o `write_dynamic_frame_from_catalog` per scrivere i record su Kafka, il processo avrà le informazioni sul database del Catalogo dati e sul nome della tabella, e potrà usarle per ottenere alcuni parametri di base per la lettura dall'origine di streaming Kafka. Se si utilizza `getSource`, `getCatalogSink`, `getSourceWithFormat`, `getSinkWithFormat`, `createDataFrameFromOptions`, `create_data_frame_from_options` o `write_dynamic_frame_from_catalog`, sarà necessario specificare questi parametri di base utilizzando le opzioni di connessione descritte qui.

È possibile specificare le opzioni di connessione per Kafka utilizzando gli argomenti per i metodi specificati nella classe `GlueContext` descritti di seguito.

- Scala
  - `connectionOptions`: utilizza con `getSource`, `createDataFrameFromOptions` e `getSink`
  - `additionalOptions`: utilizza con `getCatalogSource`, `getCatalogSink`
  - `options`: utilizza con `getSourceWithFormat`, `getSinkWithFormat`
- Python
  - `connection_options`: utilizza con `create_data_frame_from_options`, `write_dynamic_frame_from_options`
  - `additional_options`: utilizza con `create_data_frame_from_catalog`, `write_dynamic_frame_from_catalog`
  - `options`: utilizza con `getSource`, `getSink`

Per osservazioni e restrizioni sui processi ETL dei flussi di dati, consultare la pagina the section called “Streaming di note e restrizioni ETL”.

Argomenti


- Configurazione di Kafka
- Esempio: lettura di flussi da Kafka
- Esempio: scrittura in flussi Kafka

- Indicazioni di riferimento alle opzioni di connessione a Kafka

## Configurazione di Kafka

Non ci sono AWS prerequisiti per la connessione agli stream di Kafka disponibili su Internet.

Puoi creare una connessione AWS Glue Kafka per gestire le tue credenziali di connessione. Per ulteriori informazioni, consulta the section called "Creazione di una connessione per un flusso di dati Kafka". Nella configurazione del processo AWS Glue, fornisci *`connectionName`* una connessione di rete aggiuntiva, quindi, nella chiamata *`connectionName`* al metodo, fornisci il `connectionName` parametro.

In alcuni casi, è necessario configurare ulteriori prerequisiti:

- Se utilizzi Streaming gestito da Amazon per Apache Kafka con l'autenticazione IAM, avrai bisogno di una configurazione appropriata di IAM.
- Se utilizzi Streaming gestito da Amazon per Apache Kafka con un Amazon VPC, avrai bisogno di una configurazione appropriata di Amazon VPC. Dovrai creare una connessione AWS Glue che fornisca informazioni sulla connessione Amazon VPC. È necessaria la configurazione del lavoro per includere la connessione AWS Glue come connessione di rete aggiuntiva.

Per ulteriori informazioni sui prerequisiti dei processi ETL dei flussi di dati, consulta la pagina the section called "Aggiunta di processi di streaming ETL".

## Esempio: lettura di flussi da Kafka

Usato in combinazione con the section called "forEachBatch".

Esempio per l'origine di streaming Kafka:

```
kafka_options =
    { "connectionName": "ConfluentKafka",
      "topicName": "kafka-auth-topic",
      "startingOffsets": "earliest",
      "inferSchema": "true",
      "classification": "json"
    }
data_frame_datasource0 =
 glueContext.create_data_frame.from_options(connection_type="kafka",
 connection_options=kafka_options)
```

## Esempio: scrittura in flussi Kafka

Esempi di scrittura in Kafka:

Esempio con il metodo `getSink`:

```
data_frame_datasource0 =
glueContext.getSink(
 connectionType="kafka",
 connectionOptions={
  JsonOptions("""{
   "connectionName": "ConfluentKafka",
   "classification": "json",
   "topic": "kafka-auth-topic",
   "typeOfData": "kafka"}
  """)},
 transformationContext="dataframe_ApacheKafka_node1711729173428")
 .getDataFrame()
```

Esempio con il metodo `write_dynamic_frame.from_options`:

```
kafka_options =
    { "connectionName": "ConfluentKafka",
      "topicName": "kafka-auth-topic",
      "classification": "json"
    }
data_frame_datasource0 =
 glueContext.write_dynamic_frame.from_options(connection_type="kafka",
 connection_options=kafka_options)
```

## Indicazioni di riferimento alle opzioni di connessione a Kafka

Per la lettura, utilizzare le seguenti opzioni di connessione con `"connectionType": "kafka"`:

- `"bootstrap.servers"`(Obbligatorio) Un elenco di server di bootstrap URLs, ad esempio, come`b-1.vpc-test-2.o4q88o.c6.kafka.us-east-1.amazonaws.com:9094`. Questa opzione deve essere specificata nella chiamata API o definita nei metadati della tabella in catalogo dati.
- `"security.protocol"`: (obbligatorio) Il protocollo utilizzato per comunicare con i broker. I valori possibili sono `"SSL"` o `"PLAINTEXT"`.

- `"topicName"`: (obbligatorio) un elenco separato da virgole di argomenti a cui iscriversi. Devi specificare solo uno tra `"topicName"`, `"assign"` o `"subscribePattern"`.
- `"assign"`: (obbligatorio) una stringa JSON che specifica il `TopicPartitions` specifico da utilizzare. Devi specificare solo uno tra `"topicName"`, `"assign"` o `"subscribePattern"`.

  Esempio: '{"topicA":[0,1],"topicB":[2,4]}'
- `"subscribePattern"`: (obbligatorio) una stringa regex Java che identifichi l'elenco degli argomenti a cui effettuare la sottoscrizione. Devi specificare solo uno tra `"topicName"`, `"assign"` o `"subscribePattern"`.

  Esempio: 'topic.*'
- `"classification"` (obbligatorio): il formato di file utilizzato dai dati nel record. Obbligatorio, a meno che non sia fornito tramite Catalogo dati.
- `"delimiter"` (opzionale): il separatore di valori utilizzato quando `classification` è CSV. Il valore predefinito è ",".
- `"startingOffsets"`: (opzionale) la posizione di partenza nell'argomento Kafka da cui leggere i dati. I valori possibili sono `"earliest"` o `"latest"`. Il valore predefinito è `"latest"`.
- `"startingTimestamp"`: (Facoltativo, supportato solo per AWS Glue versione 4.0 o successiva) Il timestamp del record nell'argomento Kafka da cui leggere i dati. Il valore possibile è una stringa timestamp in formato UTC nel modello `yyyy-mm-ddTHH:MM:SSZ`, dove `Z` rappresenta un offset del fuso orario UTC con un segno +/- (ad esempio: "2023-04-04T08:00:00-04:00").

  Nota: nell'elenco delle opzioni di connessione dello script di streaming AWS Glue può essere presente solo uno tra 'startingOffsets' o 'startingTimestamp', l'inclusione di entrambe queste proprietà comporterà un errore del lavoro.
- `"endingOffsets"`: (opzionale) il punto di fine di una query batch. I valori possibili sono `"latest"` o una stringa JSON che specifica un offset finale per ogni `TopicPartition`.

  Per la stringa JSON, il formato è `{"topicA":{"0":23,"1":-1},"topicB":{"0":-1}}`. Il valore `-1` come offset rappresenta `"latest"`.
- `"pollTimeoutMs"`: (opzionale) il timeout in millisecondi per il polling dei dati da Kafka negli executor del processo Spark. Il valore predefinito è `600000`.
- `"numRetries"`: (opzionale) i numero di tentativi prima di non riuscire a recuperare gli offset Kafka. Il valore predefinito è `3`.
- `"retryIntervalMs"`: (opzionale) il tempo di attesa in millisecondi prima di riprovare a recuperare gli offset Kafka. Il valore predefinito è `10`.

- `"maxOffsetsPerTrigger"`: (opzionale) il limite di velocità sul numero massimo di offset elaborati per intervallo di trigger. Il numero totale di offset specificato viene suddiviso proporzionalmente tra `topicPartitions` di diversi volumi. Il valore di default è null, il che significa che il consumer legge tutti gli offset fino all'ultimo offset noto.
- `"minPartitions"`: (opzionale) il numero minimo desiderato di partizioni da leggere da Kafka. Il valore di default è null, il che significa che il numero di partizioni Spark è uguale al numero di partizioni Kafka.
- `"includeHeaders"`: (opzionale) indica se includere le intestazioni Kafka. Quando l'opzione è impostata su "true", l'output dei dati conterrà una colonna aggiuntiva denominata "glue_streaming_kafka_headers" con tipo `Array[Struct(key: String, value: String)]`. Il valore di default è "false". Questa opzione è disponibile in AWS Glue versione 3.0 o successive.
- `"schema"`: (obbligatorio quando inferSchema è impostato su false) lo schema da utilizzare per elaborare il payload. Se la classificazione è `avro`, lo schema fornito dovrà essere nel formato dello schema Avro. Se la classificazione è `avro`, lo schema fornito dovrà essere nel formato dello schema DDL.

  Di seguito sono riportati alcuni esempi di schema.

  Example in DDL schema format

  ```
  'column1' INT, 'column2' STRING , 'column3' FLOAT
  ```

  Example in Avro schema format

  ```
  {
  "type":"array",
  "items":
  {
  "type":"record",
  "name":"test",
  "fields":
  [
    {
      "name":"_id",
      "type":"string"
    },
    {
      "name":"index",
      "type":
      [
  ```

```
      "int",
      "string",
      "float"
    ]
  }
]
}
}
```

- `"inferSchema"`: (facoltativo) il valore di default è "false". Se impostato su "true", lo schema verrà rilevato in fase di runtime dal payload all'interno di `foreachbatch`.
- `"avroSchema"`: (obsoleto) parametro utilizzato per specificare uno schema di dati Avro quando viene utilizzato il formato Avro. Questo parametro è obsoleto. Utilizzo del parametro `schema`.
- `"addRecordTimestamp"`: (opzionale) Quando questa opzione è impostata su "true", l'output dei dati conterrà una colonna aggiuntiva denominata "__src_timestamp" che indica l'ora in cui il record corrispondente è stato ricevuto dall'argomento. Il valore predefinito è "false". Questa opzione è supportata in AWS Glue versione 4.0 o successive.
- `"emitConsumerLagMetrics"`: (Facoltativo) Quando l'opzione è impostata su «true», per ogni batch, emette le metriche relative alla durata compresa tra il record più vecchio ricevuto dall'argomento e l'ora in cui arriva. AWS Glue CloudWatch Il nome della metrica è «glue.driver.streaming. maxConsumerLagInMs». Il valore predefinito è "false". Questa opzione è supportata in AWS Glue versione 4.0 o successive.

Per la scrittura, utilizzare le seguenti opzioni di connessione con `"connectionType": "kafka"`:

- `"connectionName"`(Obbligatorio) Nome della connessione AWS Glue utilizzata per connettersi al cluster Kafka (simile al codice sorgente Kafka).
- `"topic"` (obbligatorio) Se esiste una colonna di argomento, il relativo valore viene utilizzato come argomento quando si scrive la riga specifica in Kafka, a meno che non sia impostata l'opzione di configurazione dell'argomento. In altre parole, l'opzione di configurazione `topic` sovrascrive la colonna dell'argomento.
- `"partition"` (opzionale) Se viene specificato un numero di partizione valido, `partition` verrà utilizzato per l'invio del record.

  Se non viene specificata alcuna partizione ma è presente `key`, verrà scelta una partizione utilizzando un hash della chiave.

Se non sono presenti né `key` né `partition`, verrà scelta una partizione in base al partizionamento permanente delle modifiche quando nella partizione vengono prodotti almeno batch.size byte.

- `"key"` (opzionale) Utilizzato per il partizionamento se `partition` è null.
- `"classification"` (opzionale) Il formato di file utilizzato dai dati nel record. Supportiamo solo JSON, CSV e Avro.

  Con il formato Avro, possiamo fornire un AvroSchema personalizzato con cui serializzare, ma bisogna considerare che tale schema deve essere fornito anche nell'origine per la deserializzazione. Altrimenti, per impostazione predefinita utilizza Apache per la serializzazione. AvroSchema

Inoltre, è possibile eseguire il fine-tuning del sink Kafka in base alle esigenze aggiornando i parametri di configurazione del produttore di Kafka. Da notare che non esiste un elenco delle opzioni di connessione consentite, tutte le coppie chiave-valore vengono mantenute nel sink così come sono.

Tuttavia, esiste un piccolo elenco di opzioni di rifiuto che non avranno effetto. Per ulteriori informazioni, consulta le configurazioni specifiche di Kafka.

## Utilizzo delle connessioni Kinesis

È possibile usare una connessione Kinesis per leggere e scrivere su flussi di dati Amazon Kinesis utilizzando le informazioni archiviate in una tabella del Catalogo dati o fornendo informazioni per accedere direttamente al flusso di dati. Puoi leggere le informazioni da Kinesis in Spark DataFrame, quindi convertirle in un Glue. AWS DynamicFrame Puoi DynamicFrames scrivere su Kinesis in formato JSON. Se accedi direttamente al flusso di dati, utilizza queste opzioni per fornire le informazioni su come accedere al flusso di dati.

Se si utilizza `getCatalogSource` o `create_data_frame_from_catalog` per consumare i registri da una sorgente di streaming Kinesis, il processo avrà le informazioni sul database catalogo dati e sul nome della tabella, e potrà usarle per ottenere alcuni parametri di base per la lettura dalla sorgente di streaming Kinesis. Se si utilizza `getSource`, `getSourceWithFormat`, `createDataFrameFromOptions` o `create_data_frame_from_options`, dovrai specificare questi parametri di base utilizzando le opzioni di connessione descritte qui.

È possibile specificare le opzioni di connessione per Kinesis utilizzando i seguenti argomenti per i metodi specificati nella classe `GlueContext`.

- Scala
  - `connectionOptions`: utilizza con `getSource`, `createDataFrameFromOptions` e `getSink`
  - `additionalOptions`: utilizza con `getCatalogSource`, `getCatalogSink`
  - `options`: utilizza con `getSourceWithFormat`, `getSinkWithFormat`
- Python
  - `connection_options`: utilizza con `create_data_frame_from_options`, `write_dynamic_frame_from_options`
  - `additional_options`: utilizza con `create_data_frame_from_catalog`, `write_dynamic_frame_from_catalog`
  - `options`: utilizza con `getSource`, `getSink`

Per osservazioni e restrizioni sui processi ETL dei flussi di dati, consultare la pagina the section called "Streaming di note e restrizioni ETL".

## Configurazione di Kinesis

Per connetterti a un flusso di dati Kinesis in un job AWS Glue Spark, avrai bisogno di alcuni prerequisiti:

- In caso di lettura, il job AWS Glue deve disporre delle autorizzazioni IAM di livello di accesso Read per il flusso di dati Kinesis.
- In fase di scrittura, il job AWS Glue deve disporre delle autorizzazioni IAM di livello di accesso Write per il flusso di dati Kinesis.

In alcuni casi, è necessario configurare ulteriori prerequisiti:

- Se il tuo job AWS Glue è configurato con connessioni di rete aggiuntive (in genere per connettersi ad altri set di dati) e una di queste connessioni offre opzioni di rete Amazon VPC, questo indirizzerà il tuo lavoro a comunicare tramite Amazon VPC. In questo caso, per comunicare tramite Amazon VPC dovrai configurare anche il flusso di dati Kinesis. È possibile farlo creando un endpoint VPC di interfaccia tra l'Amazon VPC e il flusso di dati Kinesis. Per ulteriori informazioni, consulta la pagina Using Amazon Kinesis Data Streams with Interface VPC Endpoints.
- Quando si specifica un flusso di dati Amazon Kinesis in un altro account, è necessario impostare i ruoli e le policy per consentire l'accesso multi-account. Per ulteriori informazioni, consultare Esempio: lettura da un flusso Kinesis in un account diverso.

Per ulteriori informazioni sui prerequisiti dei processi ETL dei flussi di dati, consultare la pagina the section called "Aggiunta di processi di streaming ETL".

## Lettura da Kinesis

Esempio: lettura da flussi Kinesis

Usato in combinazione con the section called "forEachBatch".

Esempio per l'origine di streaming Amazon Kinesis:

```
kinesis_options =
   { "streamARN": "arn:aws:kinesis:us-east-2:777788889999:stream/fromOptionsStream",
     "startingPosition": "TRIM_HORIZON",
     "inferSchema": "true",
     "classification": "json"
   }
data_frame_datasource0 =
 glueContext.create_data_frame.from_options(connection_type="kinesis",
 connection_options=kinesis_options)
```

## Scrittura su Kinesis

Esempio: scrittura su flussi Kinesis

Usato in combinazione con the section called "forEachBatch". Il tuo DynamicFrame verrà scritto nello stream in formato JSON. Se il processo non riesce a scrivere dopo diversi tentativi, riporterà un errore. Per impostazione predefinita, ogni DynamicFrame record viene inviato allo stream Kinesis singolarmente. È possibile configurare questo comportamento utilizzando `aggregationEnabled` e i parametri associati.

Esempio di scrittura su Amazon Kinesis da un processo di streaming:

Python

```
glueContext.write_dynamic_frame.from_options(
    frame=frameToWrite
    connection_type="kinesis",
    connection_options={
        "partitionKey": "part1",
        "streamARN": "arn:aws:kinesis:us-east-1:111122223333:stream/streamName",
    }
```

```
)
```

Scala

```
glueContext.getSinkWithFormat(
                connectionType="kinesis",
                options=JsonOptions("""{
                    "streamARN": "arn:aws:kinesis:us-
east-1:111122223333:stream/streamName",
                    "partitionKey": "part1"
                }"""),
            )
            .writeDynamicFrame(frameToWrite)
```

## Parametri di connessione Kinesis

Indica le opzioni di connessione ad Amazon Kinesis Data Streams.

Utilizza le seguenti opzioni di connessione per le origini dati in streaming Kinesis:

- `"streamARN"`: (obbligatorio) utilizzato per la lettura/scrittura. L'ARN del flusso di dati di Kinesis.
- `"classification"`: (obbligatorio per la lettura) utilizzato per la lettura. Il formato di file utilizzato dai dati nel record. Obbligatorio, a meno che non sia fornito tramite Catalogo dati.
- `"streamName"`: (facoltativo) utilizzato per la lettura. Il nome di un flusso di dati Kinesis da cui leggere. Usato con `endpointUrl`.
- `"endpointUrl"`: (facoltativo) utilizzato per la lettura. Predefinito: "https://kinesis.us-east-1.amazonaws.com». L' AWS endpoint del flusso Kinesis. Non è necessario modificarlo a meno che non ci si stia connettendo a una regione speciale.
- `"partitionKey"`: (facoltativo) utilizzato per la scrittura. La chiave di partizione di Kinesis utilizzata per la produzione dei record.
- `"delimiter"`: (facoltativo) utilizzato per la lettura. Il separatore di valori utilizzato quando `classification` è CSV. Il valore predefinito è “`,`”.
- `"startingPosition"`: (opzionale) utilizzato per la lettura. La posizione di partenza nel flusso dei dati Kinesis da cui leggere i dati. I valori possibili sono `"latest"`, `"trim_horizon"`, `"earliest"` o una stringa di timestamp in formato UTC con il modello `yyyy-mm-ddTHH:MM:SSZ`, dove `Z` rappresenta uno scostamento del fuso orario UTC con un +/- (ad esempio: "2023-04-04T08:00:00-04:00"). Il valore predefinito è `"latest"`. Nota: la stringa

Timestamp in formato UTC per `"startingPosition"` è supportata solo per AWS Glue versione 4.0 o successiva.

- `"failOnDataLoss"`: (facoltativo) non è possibile eseguire il processo se una partizione attiva è mancante o scaduta. Il valore predefinito è `"false"`.
- `"awsSTSRoleARN"`: (facoltativo) utilizzato per la lettura/scrittura. L'Amazon Resource Name (ARN) del ruolo da assumere utilizzando AWS Security Token Service (AWS STS). Questo ruolo deve disporre delle autorizzazioni per descrivere o leggere le operazioni dei registri per il flusso di dati Kinesis. Quando si accede a un flusso di dati in un altro account, è necessario utilizzare questo parametro. Usato in combinazione con `"awsSTSSessionName"`.
- `"awsSTSSessionName"`: (facoltativo) utilizzato per la lettura/scrittura. Un identificatore della sessione che assume il ruolo usando AWS STS. Quando si accede a un flusso di dati in un altro account, è necessario utilizzare questo parametro. Usato in combinazione con `"awsSTSRoleARN"`.
- `"awsSTSEndpoint"`: (Facoltativo) L' AWS STS endpoint da utilizzare quando ci si connette a Kinesis con un ruolo presunto. Ciò consente di utilizzare l' AWS STS endpoint regionale in un VPC, cosa non possibile con l'endpoint globale predefinito.
- `"maxFetchTimeInMs"`: (opzionale) utilizzato per la lettura. Il tempo massimo impiegato affinché l'esecutore del processo legga i record della batch attuale dal flusso di dati Kinesis, specificato in millisecondi (ms). Entro questo periodo è possibile effettuate più chiamate API `GetRecords`. Il valore predefinito è `1000`.
- `"maxFetchRecordsPerShard"`: (opzionale) utilizzato per la lettura. Il numero massimo di record da recuperare per shard nel flusso di dati Kinesis per microbatch. Nota: il client può superare questo limite se il processo di streaming ha già letto i record aggiuntivi da Kinesis (nella stessa chiamata get-records). Se `maxFetchRecordsPerShard` deve essere rigoroso, allora deve essere un multiplo di `maxRecordPerRead`. Il valore predefinito è `100000`.
- `"maxRecordPerRead"`: (opzionale) utilizzato per la lettura. Il numero massimo di record da recuperare nel flusso di dati Kinesis in ciascuna operazione `getRecords`. Il valore predefinito è `10000`.
- `"addIdleTimeBetweenReads"`: (opzionale) utilizzato per la lettura. Aggiunge un ritardo tra due operazioni consecutive `getRecords`. Il valore predefinito è `"False"`. Questa opzione è configurabile solo per Glue versione 2.0 e successive.
- `"idleTimeBetweenReadsInMs"`: (opzionale) utilizzato per la lettura. Il ritardo minimo tra due operazioni consecutive `getRecords`, specificato in ms. Il valore predefinito è `1000`. Questa opzione è configurabile solo per Glue versione 2.0 e successive.

- `"describeShardInterval"`: (opzionale) utilizzato per la lettura. L'intervallo di tempo minimo tra due chiamate API `ListShards` affinché lo script consideri il resharding. Per ulteriori informazioni, consultare Strategie per il resharding nella Guida per gli sviluppatori di Amazon Kinesis Data Streams. Il valore predefinito è `1s`.
- `"numRetries"`: (opzionale) utilizzato per la lettura. Il numero massimo di tentativi per le richieste API Kinesis Data Streams. Il valore predefinito è `3`.
- `"retryIntervalMs"`: (opzionale) utilizzato per la lettura. Il periodo di raffreddamento (specificato in ms) prima di riprovare la chiamata API Kinesis Data Streams. Il valore predefinito è `1000`.
- `"maxRetryIntervalMs"`: (opzionale) utilizzato per la lettura. Il periodo di raffreddamento (specificato in ms) tra due tentativi di chiamata API Kinesis Data Streams. Il valore predefinito è `10000`.
- `"avoidEmptyBatches"`: (opzionale) utilizzato per la lettura. Impedisce la creazione di un processo microbatch vuoto controllando la presenza di dati non letti nel flusso dei dati Kinesis prima che il batch venga avviato. Il valore predefinito è `"False"`.
- `"schema"`: (obbligatorio quando inferSchema è impostato su falso) utilizzato per la lettura. Lo schema da utilizzare per elaborare il payload. Se la classificazione è `avro`, lo schema fornito dovrà essere nel formato dello schema Avro. Se la classificazione è `avro`, lo schema fornito dovrà essere nel formato dello schema DDL.

  Di seguito sono riportati alcuni esempi di schema.

  Example in DDL schema format

  ```
  `column1` INT, `column2` STRING , `column3` FLOAT
  ```

  Example in Avro schema format

  ```
  {
    "type":"array",
    "items":
    {
      "type":"record",
      "name":"test",
      "fields":
      [
        {
          "name":"_id",
          "type":"string"
        },
  ```

```
      {
        "name":"index",
        "type":
        [
          "int",
          "string",
          "float"
        ]
      }
    ]
  }
}
```

- `"inferSchema"`: (opzionale) utilizzato per la lettura. Il valore predefinito è "false". Se impostato su "true", lo schema verrà rilevato in fase di runtime dal payload all'interno di `foreachbatch`.
- `"avroSchema"`: (obsoleto) utilizzato per la lettura. Parametro utilizzato per specificare uno schema di dati Avro quando viene utilizzato il formato Avro. Questo parametro è obsoleto. Utilizzo del parametro `schema`.
- `"addRecordTimestamp"`: (opzionale) utilizzato per la lettura. Quando questa opzione è impostata su "true", l'output dei dati conterrà una colonna aggiuntiva denominata "__src_timestamp" che indica l'ora in cui il record corrispondente è stato ricevuto dal flusso. Il valore predefinito è "false". Questa opzione è supportata in AWS Glue versione 4.0 o successive.
- `"emitConsumerLagMetrics"`: (opzionale) utilizzato per la lettura. Quando l'opzione è impostata su «true», per ogni batch emette le metriche relative alla durata compresa tra il record più vecchio ricevuto dallo stream e il momento in cui arriva a. AWS Glue CloudWatch Il nome della metrica è «glue.driver.streaming. maxConsumerLagInMs». Il valore predefinito è "false". Questa opzione è supportata in AWS Glue versione 4.0 o successive.
- `"fanoutConsumerARN"`: (opzionale) utilizzato per la lettura. L'ARN di un consumatore di un flusso Kinesis per il flusso specificato in `streamARN`. Utilizzato per abilitare la modalità di fan-out avanzato per la connessione Kinesis. Per ulteriori informazioni sull'utilizzo di un flusso Kinesis con fan-out avanzato, consultare la pagina the section called "Utilizzo del fan-out avanzato nei processi di flussi di dati Kinesis".
- `"recordMaxBufferedTime"`: (opzionale) utilizzato per la scrittura. Predefinito: 1000 (ms). Tempo massimo di memorizzazione nel buffer di un record in attesa di essere scritto.
- `"aggregationEnabled"`: (opzionale) utilizzato per la scrittura. Default: true. Speciifica se i record devono essere aggregati prima di inviarli a Kinesis.

- `"aggregationMaxSize"`: (opzionale) utilizzato per la scrittura. Impostazione predefinita: 51200 (byte). Se un record è superiore a questo limite, ignorerà l'aggregatore. Ricorda che Kinesis impone un limite di 50 KB alla dimensione del record. Se imposti questo valore oltre i 50 KB, i record di grandi dimensioni verranno rifiutati da Kinesis.
- `"aggregationMaxCount"`: (facoltativo) utilizzato per la scrittura. Predefinito: 4294967295. Numero massimo di voci da inserire in un record aggregato.
- `"producerRateLimit"`: (facoltativo) utilizzato per la scrittura. Predefinito: 150 (%). Limita la velocità di trasmissione effettiva per partizione inviata da un singolo produttore (ad esempio, il tuo processo), come percentuale del limite di backend.
- `"collectionMaxCount"`: (facoltativo) utilizzato per la scrittura. Predefinito: 500. Numero massimo di articoli da imballare in una PutRecords richiesta.
- `"collectionMaxSize"`: (facoltativo) utilizzato per la scrittura. Impostazione predefinita: 5242880 (byte). Quantità massima di dati da inviare con una PutRecords richiesta.

# AWS Glue scalabilità automatica dello streaming

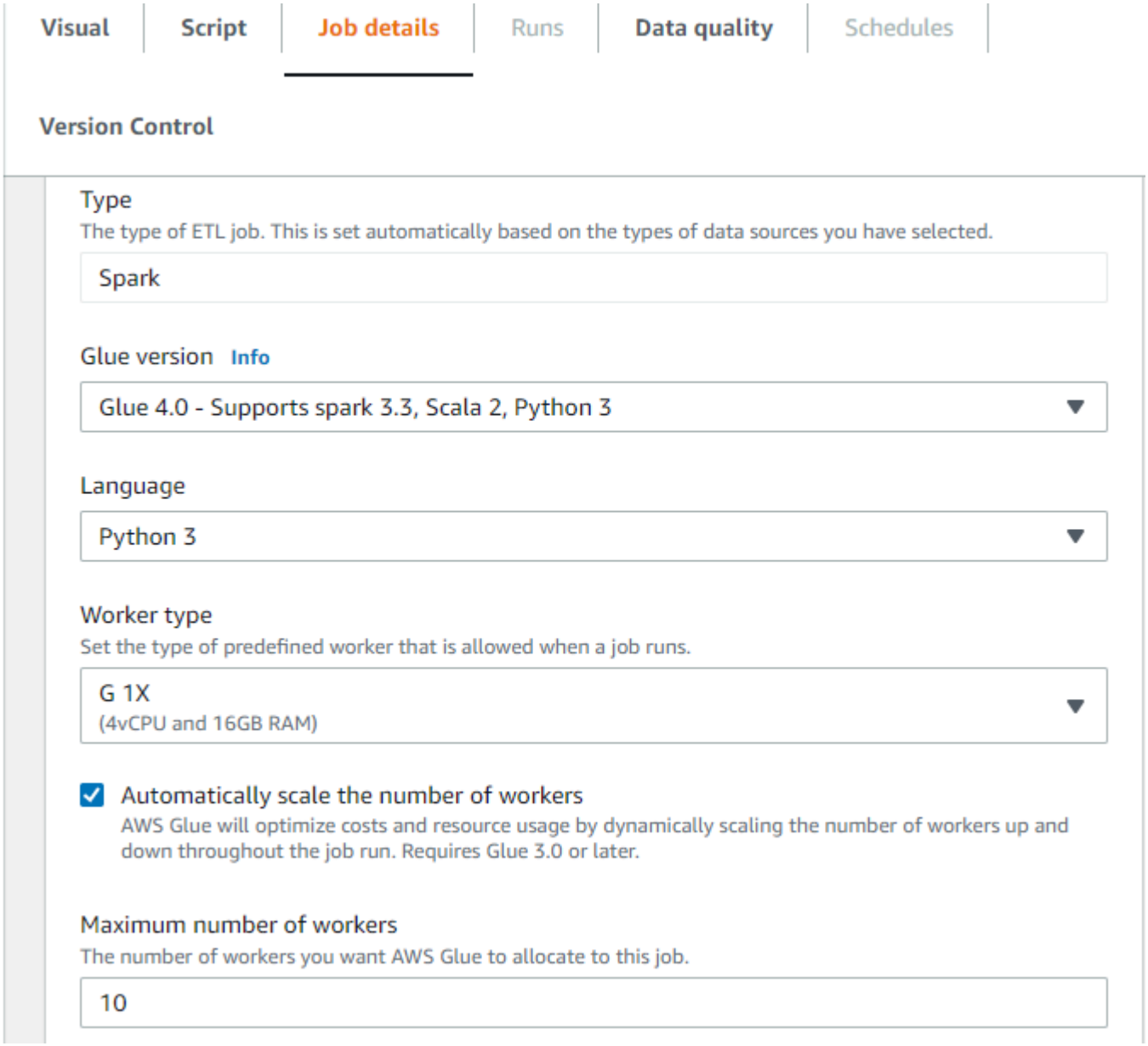
AWS Glue i lavori ETL in streaming consumano continuamente dati provenienti da fonti di streaming, puliscono e trasformano i dati in corso di elaborazione e li rendono disponibili per l'analisi. Monitorando ogni fase dell'esecuzione del lavoro, la AWS Glue scalabilità automatica può disattivare i lavoratori quando sono inattivi o aggiungere lavoratori se è possibile un'ulteriore elaborazione parallela.

Le seguenti sezioni forniscono informazioni sulla scalabilità automatica in streaming AWS Glue

## Attivazione dell'Auto Scaling in AWS Glue Studio

Nella scheda Dettagli del lavoro in AWS Glue Studio, scegli il tipo come Spark o Spark Streaming e la versione Glue come **`Glue 3.0`** o. **`Glue 4.0`** In seguito, una casella di controllo verrà visualizzata sotto Worker type (tipo di worker).

- Seleziona l'opzione Dimensiona automaticamente il numero di worker.
- Imposta la proprietà Numero massimo di dipendenti per definire il numero massimo di dipendenti che possono essere ceduti all'esecuzione del processo.




## Abilitazione dell'Auto Scaling con CLI o SDK AWS

Per abilitare Auto Scaling dalla AWS CLI per l'esecuzione del processo, esegui `start-job-run` con la seguente configurazione:

```
{
    "JobName": "<your job name>",
    "Arguments": {
        "--enable-auto-scaling": "true"
    },
```

```
    "WorkerType": "G.2X", // G.1X, G.2X, G.4X, G.8X, G.12X, G.16X, R.1X, R.2X, R.4X,
 and R.8X are supported for Auto Scaling Jobs
    "NumberOfWorkers": 20, // represents Maximum number of workers
    ...other job run configurations...
}
```

Una volta terminata l'esecuzione del processo ETL, puoi anche chiamare`get-job-run` per verificare l'effettivo utilizzo delle risorse del processo eseguito in secondi DPU. Nota: il nuovo campo DPUSecondsverrà visualizzato solo per i lavori in batch nella AWS Glue versione 3.0 o successiva abilitata con Auto Scaling. Questo campo non è supportato per i processi di streaming.

```
$ aws glue get-job-run --job-name your-job-name --run-id jr_xx --endpoint https://
glue.us-east-1.amazonaws.com --region us-east-1
{
    "JobRun": {
        ...
        "GlueVersion": "3.0",
        "DPUSeconds": 386.0
    }
}
```

È inoltre possibile configurare le esecuzioni dei processi con Auto Scaling utilizzando l'SDK AWS Glue con la stessa configurazione.

## Come funziona

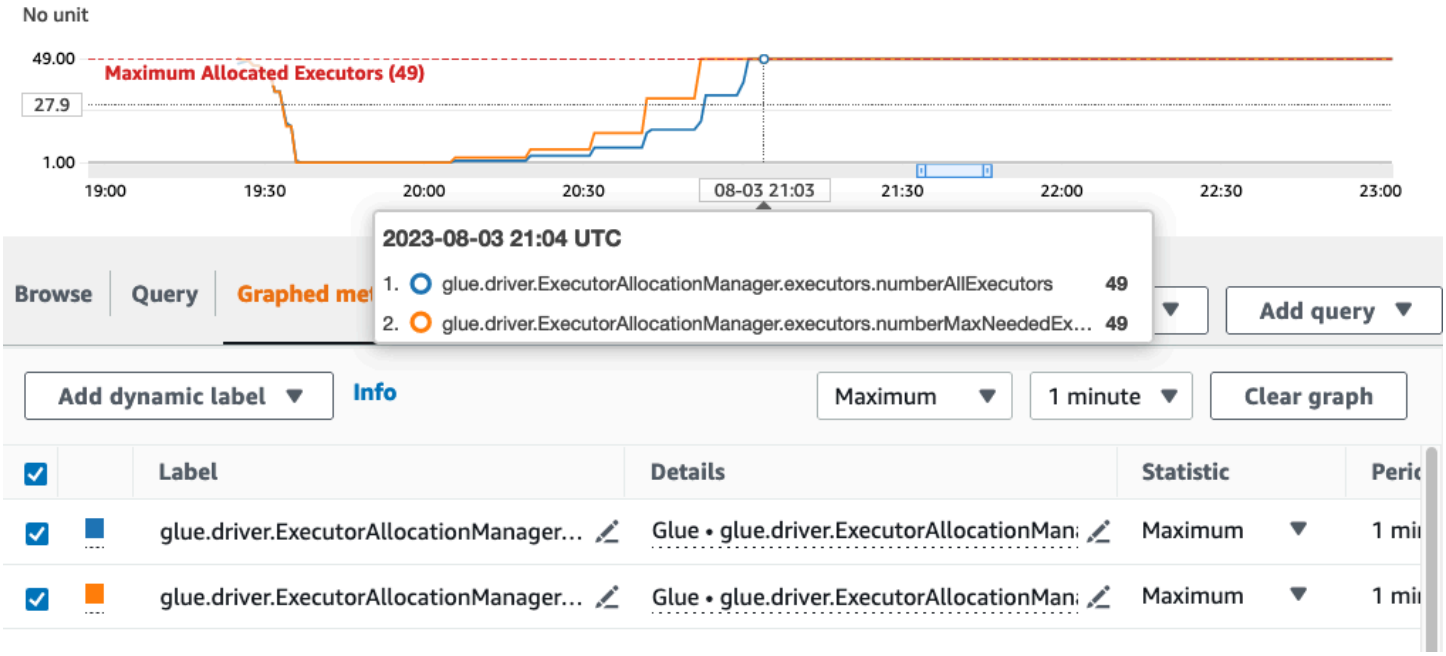
Dimensionamento tra microbatch

L'esempio seguente viene utilizzato per descrivere come funziona il dimensionamento automatico.

- Hai un AWS Glue lavoro che inizia con 50 DPUs.
- Il dimensionamento automatico è abilitato.

In questo esempio, AWS Glue esamina la metrica batchProcessingTime InMs ««per alcuni micro batch e determina se i lavori vengono completati entro le dimensioni della finestra stabilite. Se i tuoi processi vengono completati prima e a seconda della tempistica con cui vengono completati, AWS Glue potrebbe ridursi. Questa metrica, tracciata con» numberAllExecutors «, può essere monitorata Amazon CloudWatch per vedere come funziona la scalabilità automatica.

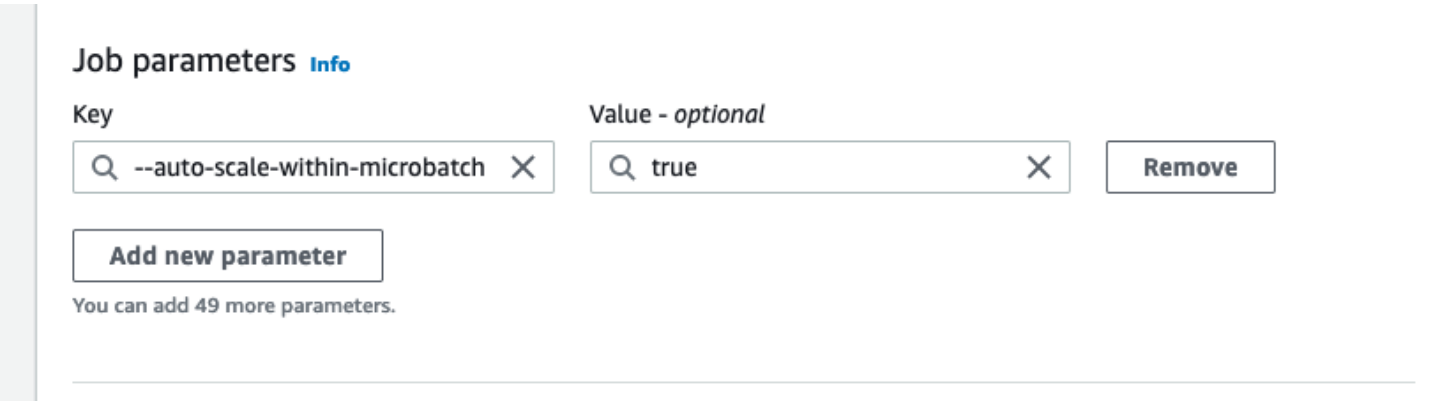
Il numero di esecutori aumenta o diminuisce in modo esponenziale solo dopo il completamento di ogni microbatch. Come si può vedere dal registro di Amazon CloudWatch monitoraggio, AWS Glue analizza il numero di executor necessari (linea arancione) e ridimensiona gli executor (linea blu) in modo che corrispondano automaticamente a tale numero.




Once AWS Glue riduce il numero di esecutori e osserva che il volume di dati aumenta, aumentando di conseguenza il tempo di elaborazione in microbatch, AWS Glue aumenterà fino a 50 DPUs, che è il limite massimo specificato.

Dimensionamento all'interno di un microbatch

Nell'esempio precedente, il sistema monitora alcuni microbatch completati per decidere se eseguire un aumento o una riduzione. Per le finestre più lunghe, il dimensionamento automatico è utile per rispondere più rapidamente all'interno del microbatch, anziché attendere pochi microbatch. In questi casi, è possibile utilizzare la configurazione aggiuntiva `--auto-scaIe-within-microbatch` per `true`. È possibile aggiungerlo alle proprietà del AWS Glue lavoro AWS Glue Studio come illustrato di seguito.




# Finestre di manutenzione per AWS Glue lo streaming

AWS Glue esegue periodicamente attività di manutenzione. Durante queste finestre di manutenzione, AWS Glue sarà necessario riavviare i processi di streaming. Puoi controllare quando i lavori vengono riavviati specificando le finestre di manutenzione. In questa sezione, descriviamo dove è possibile configurare la finestra di manutenzione e i comportamenti specifici da prendere in considerazione.

Argomenti

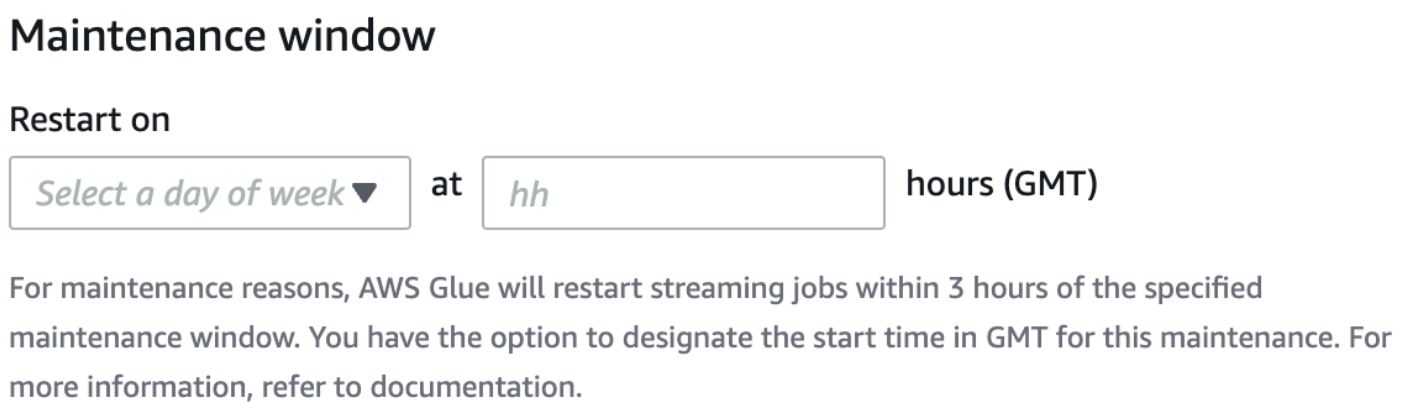

- Impostazione della finestra di manutenzione
- Comportamento della finestra di manutenzione
- Monitoraggio del lavoro
- Gestione della perdita di dati


## Impostazione della finestra di manutenzione

È possibile impostare una finestra di manutenzione utilizzando AWS Glue Studio o APIs.

### Configurazione di una finestra di manutenzione in AWS Glue Studio

È possibile specificare una finestra di manutenzione nella pagina Job Details del job AWS Glue Streaming. Puoi specificare il giorno e l'ora in GMT. AWS Glue riavvierà il lavoro entro la finestra temporale specificata.




### Configurazione di una finestra di manutenzione nell'API

In alternativa, puoi configurare la finestra di manutenzione nell'API Create Job. Ecco un esempio di configurazione di una finestra di manutenzione tramite l'API.

```
aws glue create-job –name jobName –role roleArnForTheJob –command
 Name=gluestreaming,ScriptLocation=s3-path-to-the-script --maintenance-window="Sun:10"
```

Di seguito è riportato un comando di esempio:

```
aws glue create-job –name testMaintenance –role arn:aws:iam::012345678901:role/
Glue_DefaultRole –command Name=gluestreaming,ScriptLocation=s3://glue-example-test/
example.py –maintenance-window="Sun:10
```

## Comportamento della finestra di manutenzione

AWS Glue esegue una serie di passaggi per decidere quando riavviare un lavoro:

1. Quando viene avviato un nuovo processo di streaming, verifica AWS Glue innanzitutto se è presente un timeout associato all'esecuzione del lavoro. Un timeout consente di configurare l'ora di fine del processo. Se il timeout è inferiore a 7 giorni, il processo non verrà riavviato.
2. Se il timeout è superiore a 7 giorni, AWS Glue controlla se la finestra di manutenzione è configurata per il lavoro. In tal caso, quella finestra viene selezionata e la finestra viene assegnata all'esecuzione del lavoro. AWS Glue riavvierà il lavoro entro 3 ore dalla finestra di manutenzione specificata. Ad esempio, se la finestra di manutenzione viene impostata per lunedì alle 10:00 GMT, i processi verranno riavviati tra le 10:00 GMT e le 13:00 GMT.

3. Se la finestra di manutenzione non è configurata, imposta AWS Glue automaticamente il tempo di riavvio su 7 giorni dopo l'inizio dell'esecuzione del processo. Ad esempio, se hai avviato il lavoro il giorno 1/7/2024 alle 12:00 GMT e non hai specificato gli intervalli di manutenzione, il lavoro verrà impostato per il riavvio il giorno 8/7/2024 alle 12:00 GMT.

> **Note**
>
> Se stai già eseguendo lavori di streaming, questa modifica avrà effetto su di te a partire dal 1° luglio 2024. Hai tempo fino al 30 giugno per configurare le finestre di manutenzione. Dopo il 1° luglio, tutti i processi di streaming avviati verranno riavviati in base a questa documentazione. Se hai bisogno di ulteriore assistenza, puoi contattare l' AWS assistenza.

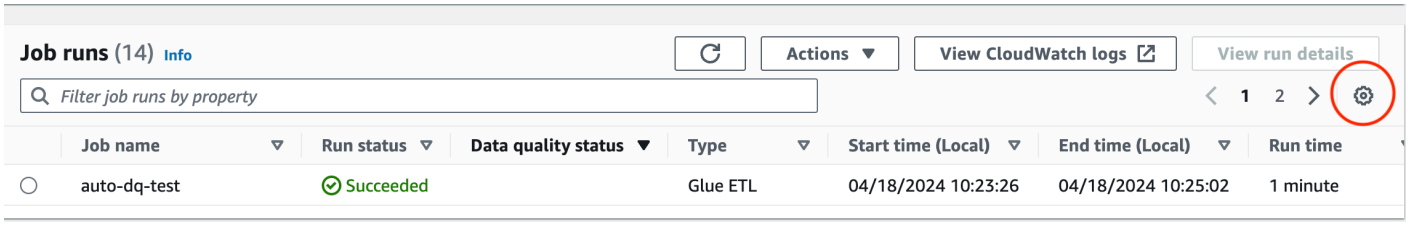
4. A volte, AWS Glue potrebbe non essere possibile riavviare il processo, specialmente quando il microbatch in corso non viene elaborato. In questi casi, il lavoro non verrà interrotto. In questi casi, AWS Glue riavvierà il processo dopo 14 giorni e, in questo caso, la finestra di manutenzione non verrà rispettata.

## Monitoraggio del lavoro

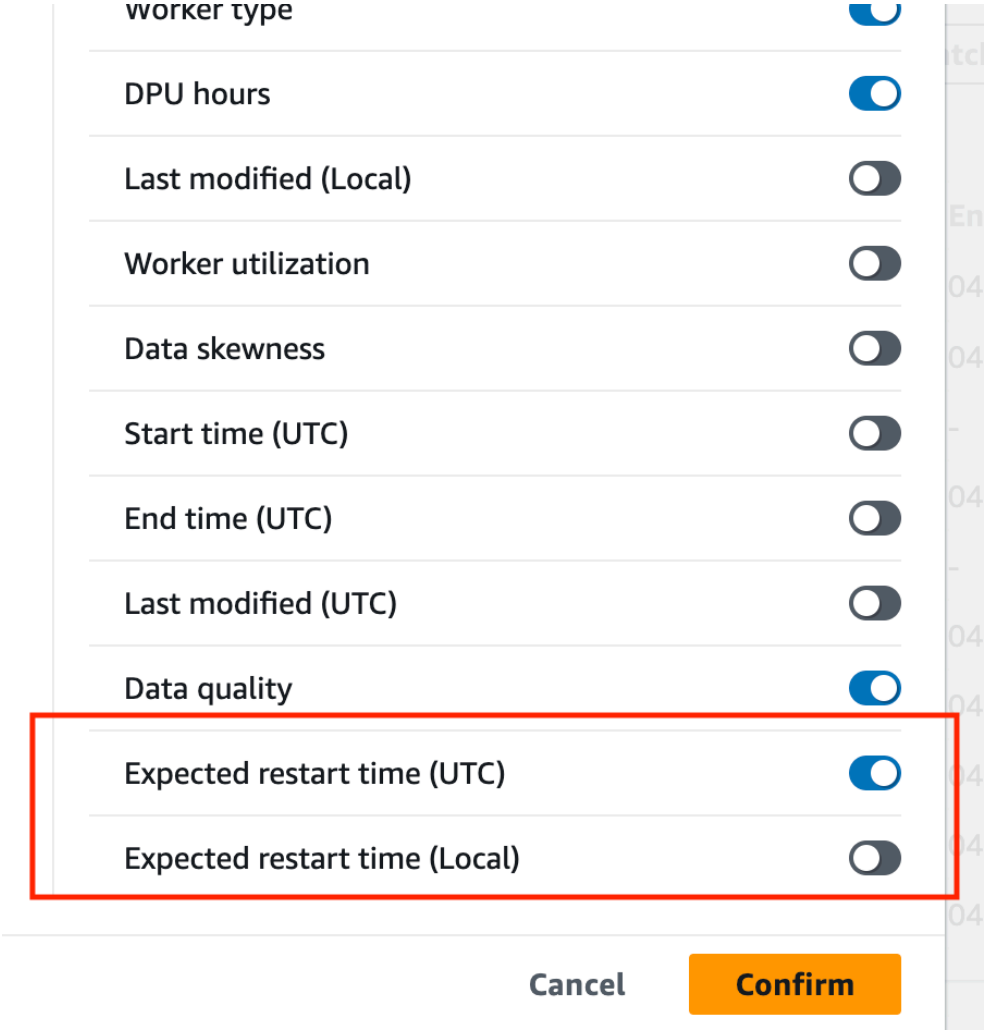
È possibile monitorare i lavori nella pagina AWS Glue Studio Monitoring.

Per vedere il prossimo orario di riavvio previsto dei lavori di streaming, mostra la colonna nella tabella Esecuzione lavoro nella pagina Monitoraggio.

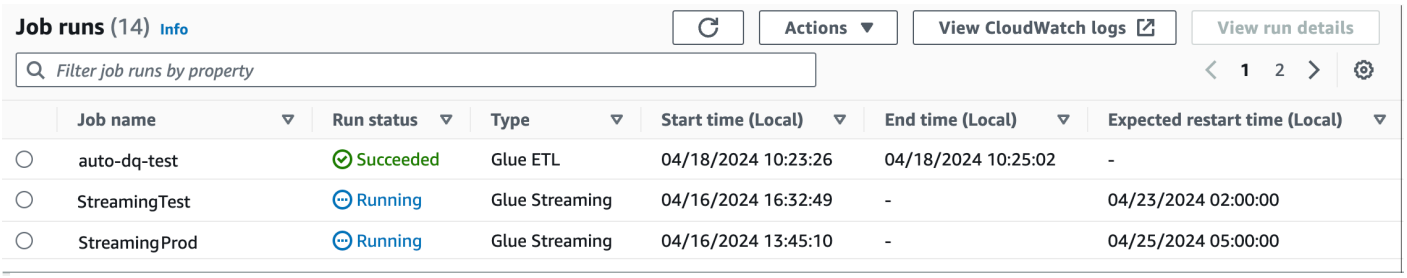
1. Fai clic sull'icona a forma di ingranaggio in alto a destra nella tabella.




2. Scorri verso il basso e attiva la colonna Tempo di riavvio previsto. Sono disponibili sia l'ora UTC che l'ora locale.




3. Puoi quindi visualizzare le colonne nella tabella.




Il lavoro originale avrà lo stato “SCADUTO” e la nuova istanza di lavoro avrà lo stato “IN ESECUZIONE”. La nuova esecuzione del lavoro che è stato riavviato avrà un ID di esecuzione del lavoro come concatenazione dell'ID iniziale dell'esecuzione del processo più il prefisso “restart_”, che rappresenta il conteggio dei riavvii. Ad esempio, se l'ID iniziale dell'esecuzione del lavoro è `jr_1234`, l'esecuzione del lavoro riavviato avrà l'ID `jr1234_restart_1` del primo riavvio. Il secondo riavvio sarà `jr1234_restart_2` per il secondo riavvio e così via.

I riavvii non influiscono sui tentativi. Se un'esecuzione fallisce viene avviata una nuova esecuzione di un nuovo tentativo automatico, il contatore di riavvio ricomincerà da 1. Ad esempio, se un'esecuzione ha esito negativo a `jr_1234_attempt_3_restart_5`, un nuovo tentativo automatico avvierà una nuova esecuzione con ID: `jr_id1_attempt_4`. Quando questo tentativo verrà riavviato dopo 7 giorni, il nuovo ID di esecuzione sarà: `jr_id1_attempt_4_restart_1`.

## Gestione della perdita di dati

Durante i riavvii di manutenzione, AWS Glue Streaming segue un processo che garantisce l'integrità e la coerenza dei dati tra l'esecuzione del processo precedente e l'esecuzione del processo riavviato. Tieni presente che AWS Glue non garantisce l'integrità e la coerenza dei dati tra i riavvii dei processi e consigliamo di prendere in considerazione l'architettura per gestire i dati duplicati all'interno dei processi di streaming.

1. Rilevamento delle condizioni di riavvio per manutenzione: AWS Glue lo streaming monitora le condizioni che indicano quando deve essere attivato un riavvio di manutenzione, ad esempio quando viene raggiunta una finestra di manutenzione dopo 7 giorni o è necessario un riavvio forzato dopo 14 giorni.
2. Richiamo di un'interruzione graduale: quando vengono soddisfatte le condizioni di riavvio della manutenzione, AWS Glue Streaming avvia un processo di interruzione graduale per il processo attualmente in esecuzione. Il processo prevede le seguenti fasi:
   a. Interruzione dell'inserimento di nuovi dati: il lavoro di streaming smette di consumare nuovi dati dalle fonti di input (ad esempio, argomenti Kafka, dirette Kinesis o file).
   b. Elaborazione dei dati in sospeso: il processo continua a elaborare tutti i dati già presenti nei buffer o nelle code interni.
   c. Immissione di offset e checkpoint: il lavoro trasferisce gli offset o i checkpoint più recenti su sistemi esterni (ad esempio, Kafka, Kinesis o Amazon S3) per garantire che il lavoro riavviato possa riprendere da dove è stato interrotto il lavoro precedente.
3. Riavvio del lavoro: una volta completato il processo di interruzione graduale, Streaming riavvia il lavoro utilizzando lo stato e i checkpoint preservati. AWS Glue Il lavoro riavviato riprende l'elaborazione dall'ultimo offset o checkpoint eseguito, assicurando che nessun dato venga perso o duplicato.
4. Ripresa dell'elaborazione dei dati: il lavoro riavviato riprende l'elaborazione dei dati dal punto in cui era stato interrotto il lavoro precedente. Continua a importare nuovi dati dalle fonti di input, a partire dall'ultimo offset o checkpoint impegnato ed elabora i dati secondo la logica ETL definita.

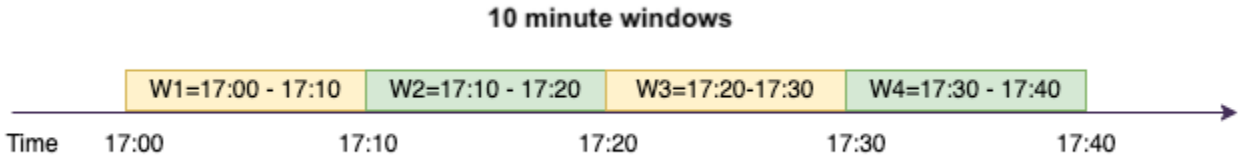
# Concetti avanzati relativi allo streaming in AWS Glue

Nelle odierne applicazioni basate sui dati, l'importanza dei dati diminuisce nel tempo e il loro valore predittivo si trasforma nella possibilità di reagire. Di conseguenza, i clienti vogliono elaborare i dati in tempo reale per prendere decisioni più rapide. Quando si gestiscono feed di dati in tempo reale, ad esempio dai sensori IoT, i dati possono arrivare non ordinati o subire ritardi nell'elaborazione dovuti alla latenza della rete e ad altri errori legati all'origine durante l'importazione. Come parte della piattaforma AWS Glue, AWS Glue Streaming si basa su queste funzionalità per fornire capacità di ETL in streaming scalabili e serverless. Inoltre, si basa sullo streaming strutturato di Apache Spark, che consente agli utenti di elaborare i dati in tempo reale.

In questo argomento, esploreremo i concetti e le funzionalità di streaming avanzati di AWS Glue Streaming.

## Considerazioni di carattere temporale relative all'elaborazione dei flussi

Esistono quattro nozioni di tempo relative all'elaborazione dei flussi:




- Ora dell'evento: l'ora in cui si è verificato l'evento. Nella maggior parte dei casi, questo campo è incorporato nei dati degli eventi stessi all'origine.
- E vent-time-window — L'intervallo di tempo tra due momenti dell'evento. Come mostrato nel diagramma precedente, W1 è compreso tra le 17:00 e le 17:10. event-time-window Ciascuno event-time-window è un raggruppamento di più eventi.
- Tempo di attivazione: il tempo di attivazione controlla la frequenza con cui si verificano l'elaborazione dei dati e l'aggiornamento dei risultati. Si tratta dell'ora in cui è iniziata l'elaborazione del microbatch.
- Ora di importazione: l'ora in cui i dati del flusso sono stati importati nel servizio di streaming. Se l'ora dell'evento non è incorporata nell'evento stesso, in alcuni casi può essere utilizzata per la creazione di finestre.

# Raggruppamenti in finestre

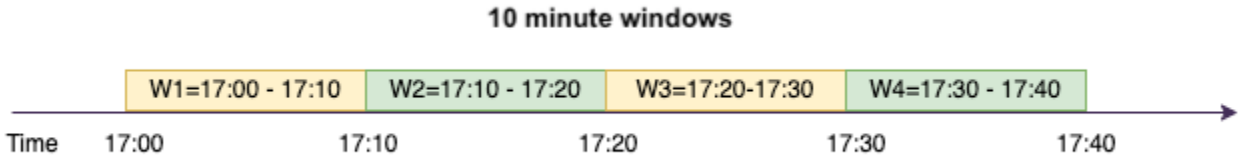
Il windowing è una tecnica che consente di raggruppare e aggregare più eventi in base a. event-time-window Esploreremo i vantaggi del windowing e le possibilità di utilizzarlo nei seguenti esempi.

A seconda del caso d'uso aziendale, Spark supporta tre tipi di finestre temporali.

- Tumbling window: una serie di dimensioni fisse non sovrapposte su cui aggregare. event-time-windows
- Finestra scorrevole: come le finestre a cascata ha dimensioni fisse, ma a differenza di esse può sovrapporsi o scorrere, a condizione che la durata dello scorrimento sia inferiore alla durata della finestra stessa.
- Finestra di sessione: inizia con un evento relativo ai dati di input e continua a espandersi fintantoché riceve input entro un intervallo di tempo o un periodo di inattività. Una finestra di sessione può avere una lunghezza fissa o dinamica a seconda degli input.

## Finestra a cascata

La finestra tumbling è una serie di dimensioni fisse non sovrapposte su cui si aggregano. event-time-windows Cerchiamo di capirlo con un esempio tratto dalla realtà.




La società ABC Auto vuole lanciare una campagna di marketing per un nuovo marchio di auto sportive. Vuole scegliere la città con il maggior numero di appassionati di auto sportive. Per raggiungere questo obiettivo, pubblica sul suo sito web un breve annuncio pubblicitario di 15 secondi di presentazione dell'auto. Tutti i «clic» e la «città» corrispondente vengono registrati e trasmessi in streaming. Amazon Kinesis Data Streams Vogliamo contare il numero di clic in una finestra di 10 minuti e raggrupparlo per città per vedere quale città registra la domanda maggiore. Di seguito è riportato l'output dell'aggregazione.

| ora_inizio_finestra | ora_fine_finestra | città | clic_totali |
|---|---|---|---|
| 2023-07-10 17:00:00 | 2023-07-10 17:10:00 | Dallas | 75 |

| ora_inizio_finestra | ora_fine_finestra | città | clic_totali |
|---|---|---|---|
| 2023-07-10 17:00:00 | 2023-07-10 17:10:00 | Chicago | 10 |
| 2023-07-10 17:20:00 | 2023-07-10 17:30:00 | Dallas | 20 |
| 2023-07-10 17:20:00 | 2023-07-10 17:30:00 | Chicago | 50 |

Come spiegato sopra, questi event-time-windows sono diversi dagli intervalli di tempo di attivazione. Ad esempio, anche se l'intervallo di attivazione è ogni minuto, i risultati di output mostreranno solo finestre di aggregazione di 10 minuti non sovrapposte. Per l'ottimizzazione, è preferibile che l'intervallo di attivazione sia allineato con. event-time-window

Nella tabella precedente, nella finestra 17:00-17:10 Dallas ha registrato 75 clic mentre Chicago ha registrato 10 clic. Inoltre, per nessuna città sono presenti dati per la finestra 17:10-17:20, quindi questa finestra viene omessa.

Ora puoi eseguire ulteriori analisi su questi dati nell'applicazione di analisi a valle per determinare la città più indicata per la conduzione della campagna di marketing.

Utilizzo delle finestre a cascata in AWS Glue

1. Crea un file Amazon Kinesis Data Streams DataFrame e leggi da esso. Esempio:

```
parsed_df = kinesis_raw_df \
                .selectExpr('CAST(data AS STRING)') \
                .select(from_json("data", ticker_schema).alias("data")) \
                .select('data.event_time','data.ticker','data.trade','data.volume',
 'data.price')
```

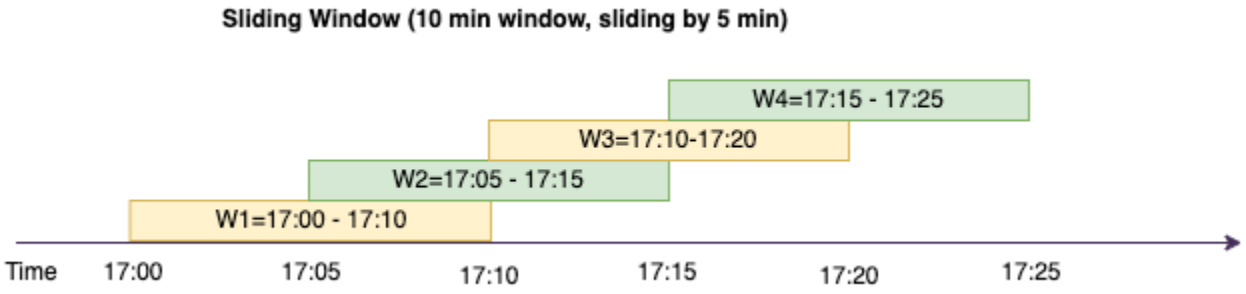
2. Elabora i dati in una finestra a cascata. Nell'esempio seguente, i dati vengono raggruppati in base al campo di input "ora_evento" in finestre a cascata di 10 minuti e l'output viene scritto in un data lake Amazon S3.

```
grouped_df = parsed_df \
                .groupBy(window("event_time", "10 minutes"), "city") \
                .agg(sum("clicks").alias("total_clicks"))

summary_df = grouped_df \
```

```
                .withColumn("window_start_time", col("window.start")) \
                .withColumn("window_end_time", col("window.end")) \
                .withColumn("year", year("window_start_time")) \
                .withColumn("month", month("window_start_time")) \
                .withColumn("day", dayofmonth("window_start_time")) \
                .withColumn("hour", hour("window_start_time")) \
                .withColumn("minute", minute("window_start_time")) \
                .drop("window")

write_result = summary_df \
                .writeStream \
                .format("parquet") \
                .trigger(processingTime="10 seconds") \
                .option("checkpointLocation", "s3a://bucket-stock-stream/stock-
stream-catalog-job/checkpoint/") \
                .option("path", "s3a://bucket-stock-stream/stock-stream-catalog-
job/summary_output/") \
                .partitionBy("year", "month", "day") \
                .start()
```

## Finestra scorrevole

Come le finestre a cascata, le finestre scorrevoli hanno dimensioni fisse, ma a differenza di esse possono sovrapporsi o scorrere, a condizione che la durata dello scorrimento sia inferiore alla durata della finestra stessa. In virtù della natura dello scorrimento, uno stesso input può essere associato a più finestre.




Per comprendere meglio, prendiamo in considerazione l'esempio di una banca che desidera rilevare potenziali frodi relative alle carte di credito. Un'applicazione di streaming potrebbe monitorare un flusso continuo delle transazioni con carta di credito. Queste transazioni potrebbero essere aggregate in finestre della durata di 10 minuti e ogni 5 minuti la finestra scorrerebbe in avanti, eliminando i 5

minuti di dati più vecchi e aggiungendo gli ultimi 5 minuti di dati più recenti. All'interno di ciascuna finestra, le transazioni potrebbero essere raggruppate per paese, verificando la presenza di schemi sospetti, ad esempio una transazione negli Stati Uniti seguita immediatamente da un'altra in Australia. Per semplicità, tali transazioni vengono classificate come frodi quando l'importo totale delle transazioni è superiore a 100 USD. Se viene rilevato uno schema di questo tipo, viene segnalata una frode potenziale e la carta potrebbe essere bloccata.

Il sistema di elaborazione delle carte di credito sta inviando a Kinesis una serie di transazioni con i dati relativi agli ID delle carte di credito e al paese. Un AWS Glue job esegue l'analisi e produce il seguente output aggregato.

| ora_inizio_finestra | ora_fine_finestra | ultime_quattro_cifre_carta | paese | importo_totale |
|---|---|---|---|---|
| 2023-07-10 17:00:00 | 2023-07-10 17:10:00 | 6544 | US | 85 |
| 2023-07-10 17:00:00 | 2023-07-10 17:10:00 | 6544 | Australia | 10 |
| 2023-07-10 17:05:45 | 2023-07-10 17:15:45 | 6544 | US | 50 |
| 2023-07-10 17:10:45 | 2023-07-10 17:20:45 | 6544 | US | 50 |
| 2023-07-10 17:10:45 | 2023-07-10 17:20:45 | 6544 | Australia | 150 |

In base all'aggregazione di cui sopra, si può osservare come la finestra di 10 minuti scorra ogni 5 minuti, sommata per importo della transazione. L'anomalia viene rilevata nella finestra 17:10-17:20 dove compare un valore anomalo, ossia una transazione del valore di 150 USD in Australia. AWS Glue è in grado di rilevare questa anomalia e inviare un evento di allarme con la chiave incriminata a un argomento SNS utilizzando boto3. Inoltre, una funzione Lambda può iscriversi a questo argomento e agire di conseguenza.

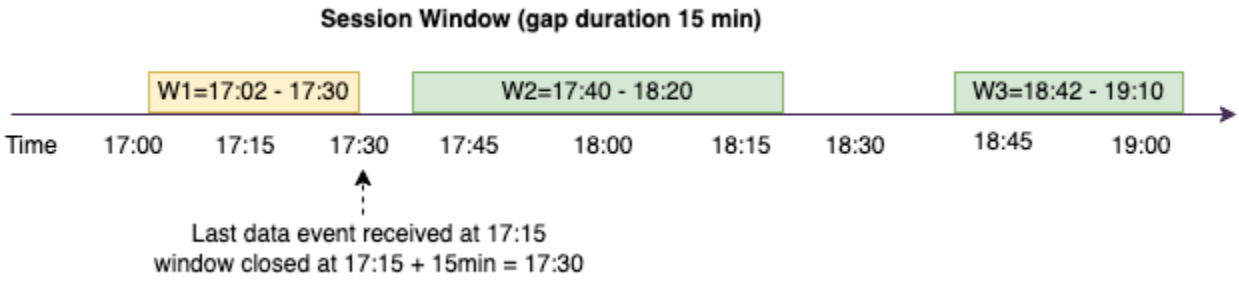
Elaborazione dei dati in una finestra scorrevole

Per implementare la finestra scorrevole vengono utilizzate la clausola `group-by` e la funzione finestra, come mostrato di seguito.

```
grouped_df = parsed_df \
            .groupBy(window(col("event_time"), "10 minute", "5 min"), "country",
 "card_last_four") \
            .agg(sum("tx_amount").alias("total_amount"))
```

## Finestra di sessione

A differenza delle due finestre precedenti, che hanno una dimensione fissa, la finestra di sessione può avere una lunghezza fissa o dinamica a seconda degli input. Una finestra di sessione inizia con un evento di dati di input e continua a espandersi finché riceve input entro un intervallo di tempo o un periodo di inattività.




Facciamo un esempio. L'hotel ABC vuole scoprire qual è il periodo più trafficato della settimana e proporre agli ospiti offerte più allettanti. Non appena un ospite effettua il check-in, viene avviata una finestra di sessione e Spark mantiene uno stato con relativa aggregazione. event-time-window Ogni volta che un ospite effettua il check-in, viene generato un evento a cui viene inviato. Amazon Kinesis Data Streams L'hotel decide che se non ci sono check-in per un periodo di 15 minuti, event-time-window può essere chiuso. Il prossimo event-time-window ricomincerà quando ci sarà un nuovo check-in. L'output è simile al seguente.

| ora_inizio_finestra | ora_fine_finestra | città | checkin_totali |
|---|---|---|---|
| 2023-07-10 17:02:00 | 2023-07-10 17:30:00 | Dallas | 50 |
| 2023-07-10 17:02:00 | 2023-07-10 17:30:00 | Chicago | 25 |

| ora_inizio_finestra | ora_fine_finestra | città | checkin_totali |
|---|---|---|---|
| 2023-07-10 17:40:00 | 2023-07-10 18:20:00 | Dallas | 75 |
| 2023-07-10 18:50:45 | 2023-07-10 19:15:45 | Dallas | 20 |

Il primo check-in è avvenuto all'ora_evento=17:02. L'aggregazione avrà inizio alle 17:02. event-time-window L'aggregazione continuerà fintantoché verranno ricevuti eventi nell'arco di 15 minuti. Nell'esempio precedente, l'ultimo evento è stato ricevuto alle 17:15, poi per i successivi 15 minuti non si sono verificati eventi. Di conseguenza, Spark lo ha chiuso alle 17:15 +15min = event-time-window 17:30 e lo ha impostato come 17:02 - 17:30. È iniziato un nuovo evento alle 17:47 quando ha ricevuto un nuovo event-time-window evento relativo ai dati del check-in.

Elaborazione dei dati in una finestra di sessione

Per implementare la finestra scorrevole vengono utilizzate la clausola `group-by` e la funzione finestra.

```
grouped_df = parsed_df \
            .groupBy(session_window(col("event_time"), "10 minute"), "city") \
            .agg(count("check_in").alias("total_checkins"))
```

## Modalità di output

La modalità di output è la modalità in cui i risultati della tabella illimitata vengono scritti nel sink esterno. Sono disponibili tre modalità. Nell'esempio seguente si contano le occorrenze di una parola mentre le righe di dati vengono trasmesse ed elaborate in ogni microbatch.

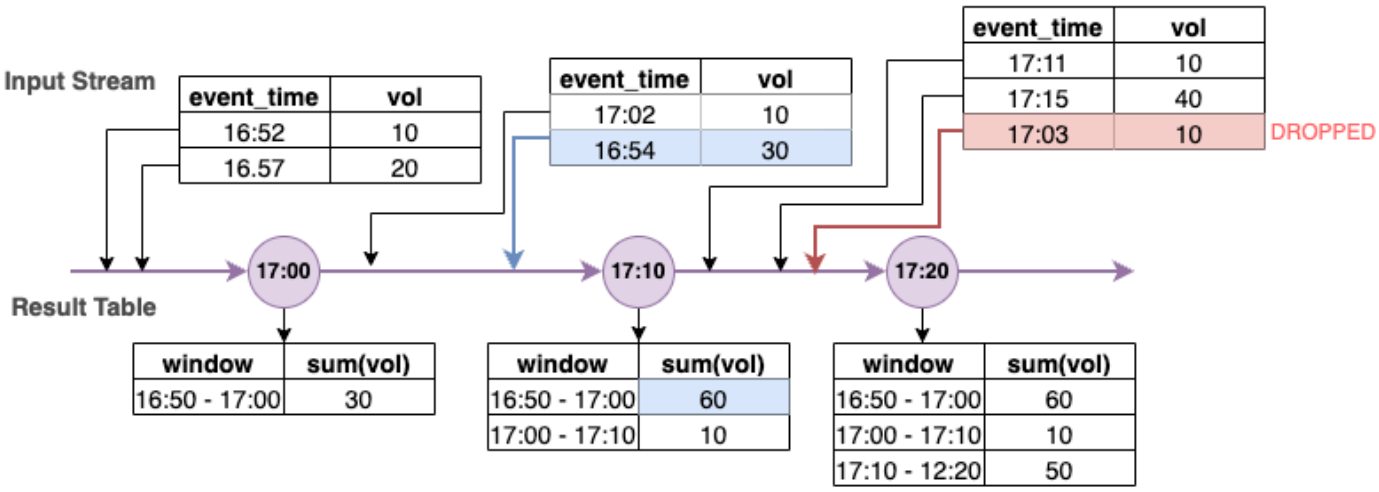
- Modalità completa: l'intera tabella dei risultati verrà scritta nel sink dopo ogni elaborazione in microbatch, anche se il conteggio delle parole non è stato aggiornato nella versione corrente event-time-window.
- Modalità di aggiunta: questa è la modalità predefinita, in cui solo le nuove parole e/o righe aggiunte alla tabella dei risultati dall'ultima attivazione vengono scritte nel sink. Questa modalità è utile per lo streaming stateless per query come map, flatMap, filter, ecc.
- Modalità di aggiornamento: nel sink vengono scritte solo le parole e/o le righe che sono state aggiornate o aggiunte nella tabella dei risultati dall'ultima attivazione.

> **Note**
>
> La modalità di output "aggiornamento" non è supportata per le finestre di sessione.

## Gestione di dati in ritardo e filigrane

Quando si lavora con dati in tempo reale, potrebbero verificarsi ritardi nell'arrivo dei dati a causa della latenza della rete e di guasti a monte e abbiamo bisogno di un meccanismo per eseguire nuovamente l'aggregazione dei dati persi. event-time-window Tuttavia, a tale scopo, è necessario mantenere lo stato. Allo stesso tempo, per limitare le dimensioni dello stato, è necessario rimuovere i dati più vecchi. La versione 2.1 di Spark ha aggiunto il supporto per una funzionalità chiamata "watermarking", ossia "applicazione della filigrana", che mantiene lo stato e consente all'utente di specificare la soglia per i dati in ritardo.

Facendo riferimento all'esempio sul simbolo azionario riportato sopra, poniamo che i dati in ritardo non possano superare la soglia dei 10 minuti. Per semplificare, supponiamo di utilizzare le finestre a cascata, il simbolo AMZ e l'opzione ACQUISTA.




Nel diagramma precedente, calcoliamo il volume totale su una finestra a cascata di 10 minuti. L'attivazione è impostata alle ore 17:00, 17:10 e 17:20. Sopra la freccia della linea temporale si trova il flusso di dati di input, mentre sotto si trova la tabella dei risultati illimitata.

Nella prima finestra a cascata di 10 minuti i dati sono stati aggregati in base a ora_evento e il volume_totale calcolato è stato 30. Nel secondo event-time-window, spark ha ricevuto il primo evento di dati con event_time= 17:02. Poiché questo è il valore massimo di ora_evento visto finora da Spark,

la soglia della filigrana viene riportata indietro di 10 minuti (ossia, ora_evento_filigrana=16:52). Qualsiasi evento di dati con un valore di ora_evento successivo alle 16:52 verrà preso in considerazione per l'aggregazione entro i limiti temporali, mentre gli eventi di dati precedenti verranno eliminati. Ciò consente a Spark di mantenere uno stato intermedio per altri 10 minuti per accogliere i dati in ritardo. Intorno alle 17:08, Spark ha ricevuto un evento con un valore di ora_evento=16:54 che rientrava nella soglia. Quindi spark ha ricalcolato «16:50 - 17:00 « event-time-windowe il volume totale è stato aggiornato da 30 a 60.

Tuttavia, all'ora di attivazione 17:20, quando Spark ha ricevuto un evento con ora_evento=17:15, ha impostato ora_evento_filigrana=17:05. Pertanto, l'evento di dati in ritardo con il valore di ora_evento=17:03 è stato considerato successivo alla soglia di tolleranza e quindi ignorato.

```
Watermark Boundary = Max(Event Time) - Watermark Threshold
```

### Utilizzo delle filigrane in AWS Glue

Spark non emette né scrive i dati nel sink esterno finché non viene superato il limite della filigrana. Per implementare una filigrana in AWS Glue, consulta l'esempio seguente.

```
grouped_df = parsed_df \
                .withWatermark("event_time", "10 minutes") \
                .groupBy(window("event_time", "5 minutes"), "ticker") \
                .agg(sum("volume").alias("total_volume"))
```

## Informazioni sul monitoraggio dei lavori di AWS Glue streaming

Il monitoraggio dei processi di streaming è una parte fondamentale della creazione delle pipeline di ETL. Oltre a utilizzare l'interfaccia utente Spark, puoi anche utilizzare Amazon CloudWatch per monitorare le metriche. Di seguito è riportato un elenco delle metriche di streaming emesse dal framework. AWS Glue Per un elenco completo di tutte le AWS Glue metriche, consulta Monitoraggio AWS Glue tramite i CloudWatch parametri Amazon.

AWS Glue utilizza un framework di streaming strutturato per elaborare gli eventi di input. Puoi utilizzare l'API Spark direttamente nel tuo codice o sfruttare il valore `ForEachBatch` fornito da `GlueContext`, che pubblica questi parametri. Per comprendere questi parametri, dobbiamo prima capire `windowSize`.

windowSize: `windowSize` è l'intervallo di microbatch fornito. Se si specifica una dimensione della finestra di 60 secondi, il processo di AWS Glue streaming aspetterà 60 secondi (o più se il batch precedente non è stato completato entro tale data) prima di leggere i dati in un batch dalla sorgente di streaming e applicare le trasformazioni fornite in`ForEachBatch`. Questo valore viene chiamato anche intervallo di attivazione.

Esaminiamo i parametri in modo più dettagliato per comprendere le caratteristiche di integrità e prestazioni.

> **Note**
>
> Questi parametri vengono emessi ogni 30 secondi. Se il valore `windowSize` fornito è inferiore a 30 secondi, i parametri riportati sono un'aggregazione. Ad esempio, supponiamo che il valore `windowSize` fornito sia di 10 secondi e che si stiano elaborando costantemente 20 record per microbatch. In questo scenario, il valore del parametro emesso per numRecords sarebbe 60.
> Un parametro non viene emesso se per esso non sono disponibili dati. Inoltre, nel caso del parametro del ritardo del consumatore, è necessario abilitare la funzionalità per ottenere i parametri associati.

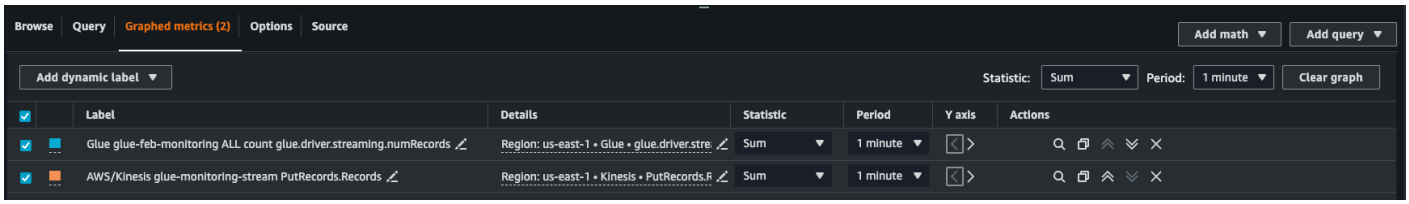
## Visualizzazione delle metriche di streaming AWS Glue

Per tracciare i parametri visivamente:

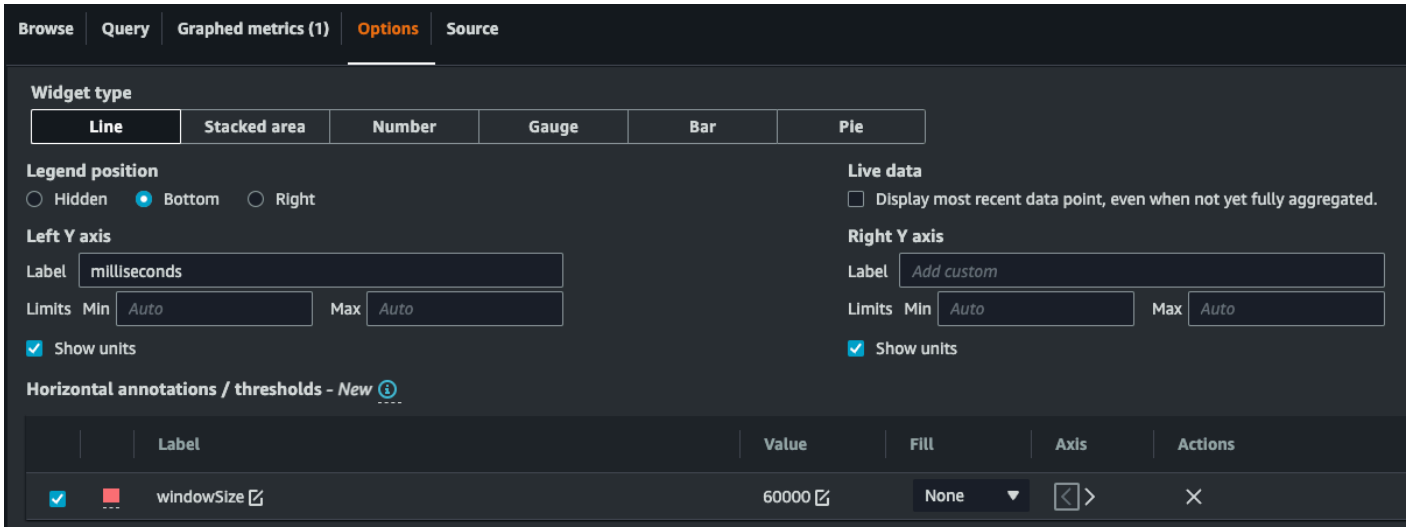
1. Vai a Metrics nella CloudWatch console Amazon, quindi seleziona la scheda Sfoglia. In "Spazi dei nomi personalizzati", scegli Glue.

2. Scegli Parametri processo per visualizzare i parametri di tutti i processi.
3. Filtra le metriche in base a JobName = glue-feb-monitoring e poi a =ALL. JobRunId Puoi fare clic sul segno "+", come mostrato nella figura seguente, per aggiungerlo al filtro di ricerca.

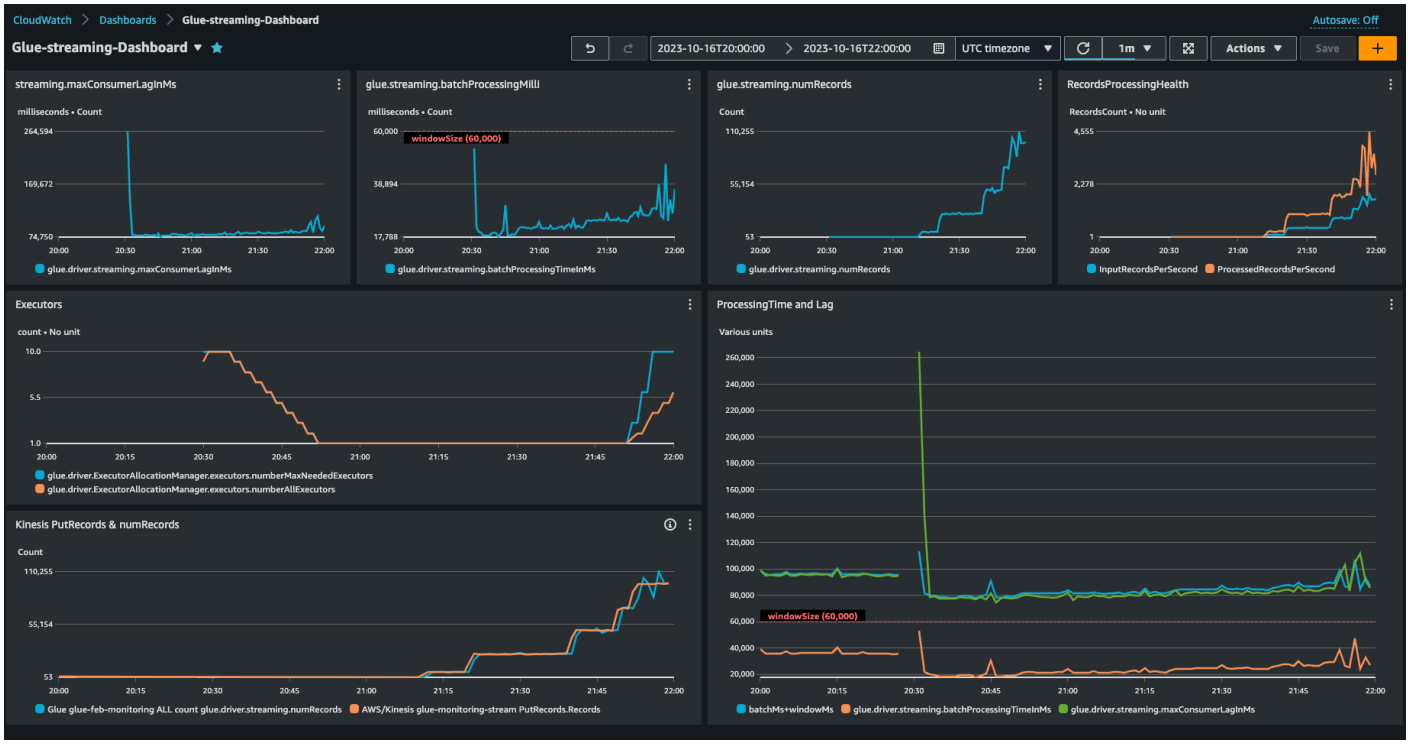
4. Seleziona la casella di controllo relativa ai parametri che ti interessano. Nella figura seguente abbiamo selezionato `numberAllExecutors` e `numberMaxNeededExecutors`.




5. Dopo aver selezionato questi parametri, puoi andare alla scheda Parametri definiti e applicare le tue statistiche.
6. Poiché i parametri vengono emesse ogni minuto, puoi applicare la "media" su un minuto per `batchProcessingTimeInMs` e `maxConsumerLagInMs`. Per `numRecords`, puoi applicare la "somma" per ogni minuto.
7. È possibile aggiungere un'annotazione `windowSize` orizzontale al grafico tramite la scheda Opzioni.




8. Dopo aver selezionato i parametri, crea un pannello di controllo e aggiungilo. Di seguito è mostrato un pannello di controllo di esempio.

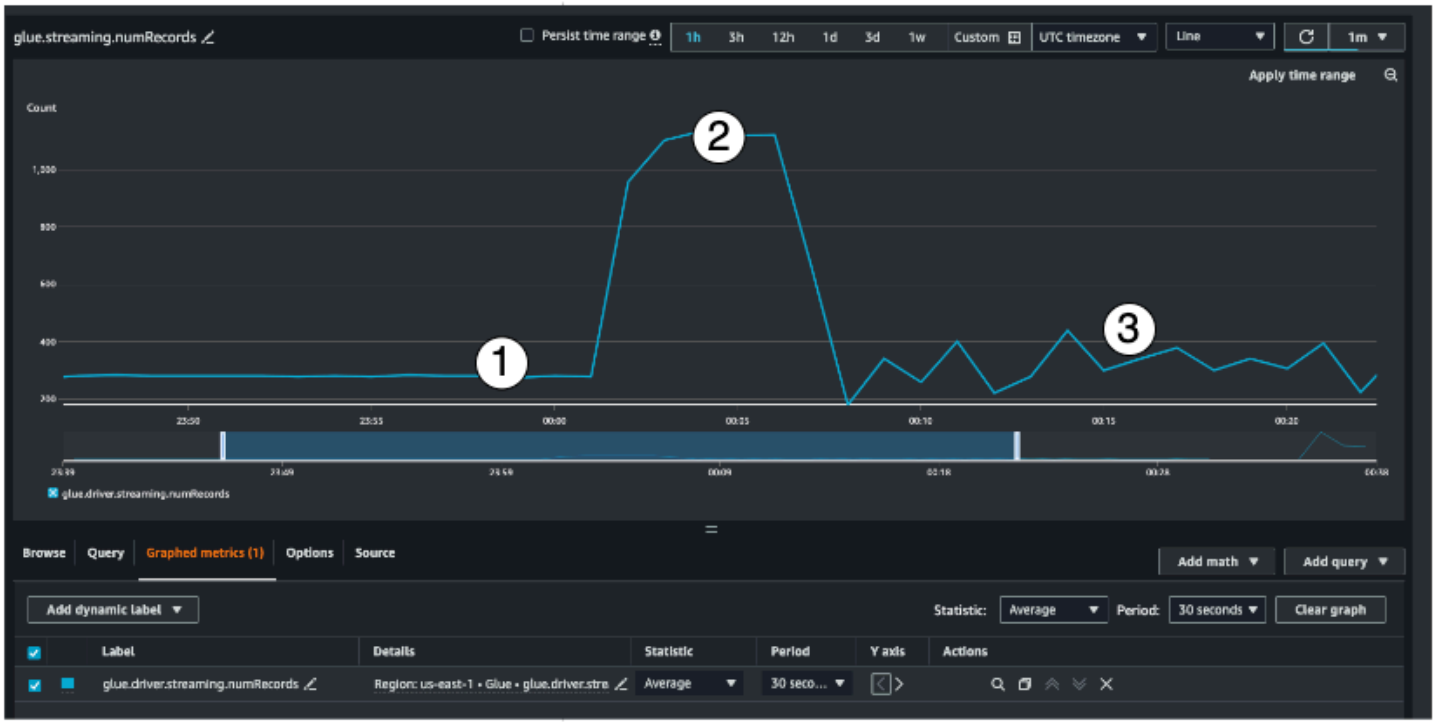



# Utilizzo delle metriche di AWS Glue streaming

Questa sezione descrive ciascuno dei parametri e il modo in cui sono correlati tra loro.

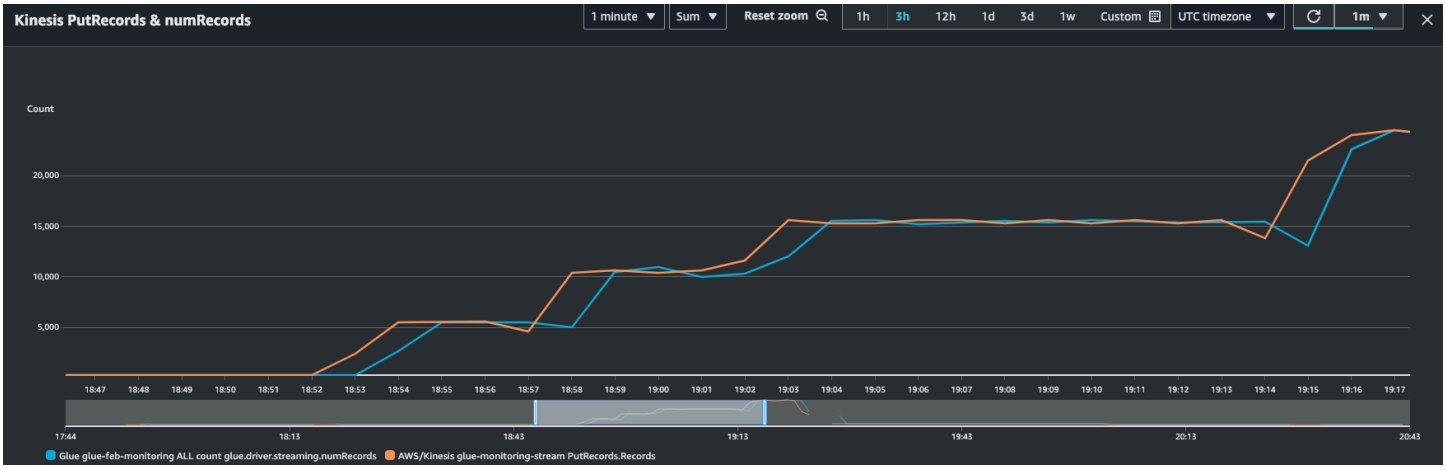
## Numero di record (parametro: streaming.numRecords)

Questo parametro indica quanti record sono in fase di elaborazione.




Questo parametro di streaming consente di visualizzare il numero di record in fase di elaborazione in una finestra. Oltre a specificare il numero di record in fase di elaborazione, agevola la comprensione del comportamento del traffico di input.

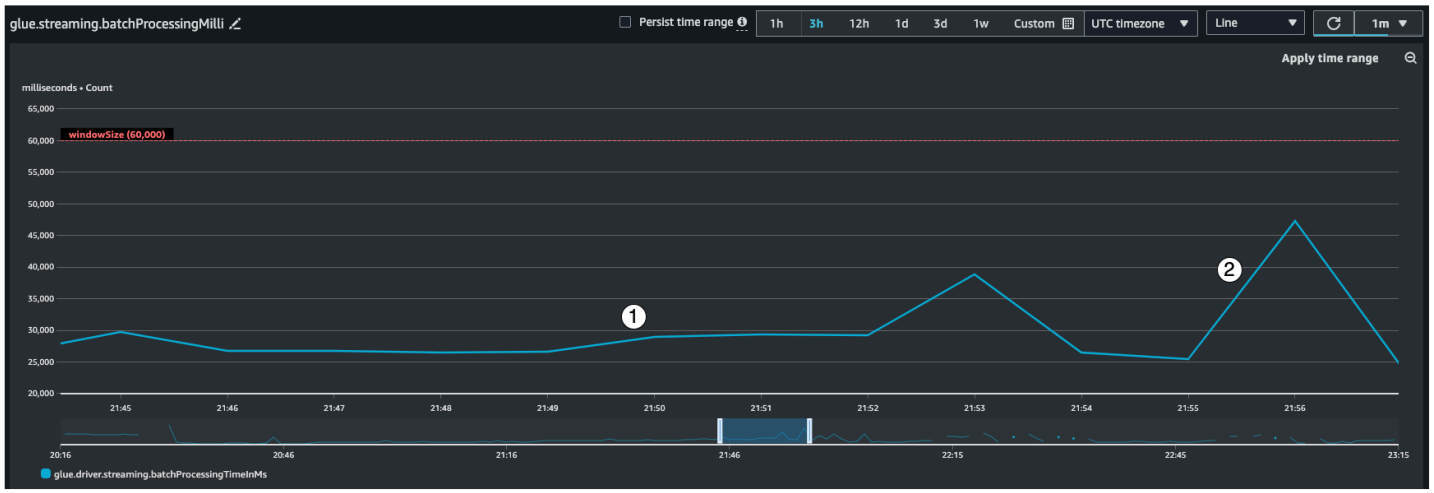
- L'indicatore n. 1 mostra un esempio di traffico stabile senza picchi. In genere si tratta di applicazioni come i sensori IoT che raccolgono dati a intervalli regolari e li inviano all'origine di streaming.
- L'indicatore n. 2 mostra un esempio di improvviso aumento del traffico su un carico altrimenti stabile. In un'applicazione clickstream, ciò può accadere in concomitanza con un evento di marketing come il Black Friday, quando si verifica un aumento esponenziale del numero di clic.
- L'indicatore n. 3 mostra un esempio di traffico imprevedibile. L'imprevedibilità del traffico non indica un problema, ma è una caratteristica intrinseca dei dati di input. Tornando all'esempio del sensore IoT, immaginiamo centinaia di sensori che inviano eventi di cambiamenti meteorologici all'origine di streaming. Poiché i cambiamenti meteorologici non sono prevedibili, non lo sono nemmeno i dati. Comprendere l'andamento del traffico è fondamentale per dimensionare gli esecutori. Se l'input presenta molti picchi, potresti prendere in considerazione l'utilizzo del dimensionamento automatico, di cui parleremo più avanti.




Puoi combinare questa metrica con la metrica PutRecords Kinesis per assicurarti che il numero di eventi da inserire e il numero di record letti siano quasi gli stessi. Questo è particolarmente utile quando si cerca di comprendere il ritardo. All'aumentare del tasso di ingestione, aumenta anche il readby. `numRecords` AWS Glue

## Tempo di elaborazione in batch (metrica: streaming). batchProcessingTimeInMs)

Il parametro del tempo di elaborazione del batch consente di determinare se il provisioning del cluster è insufficiente o eccessivo.

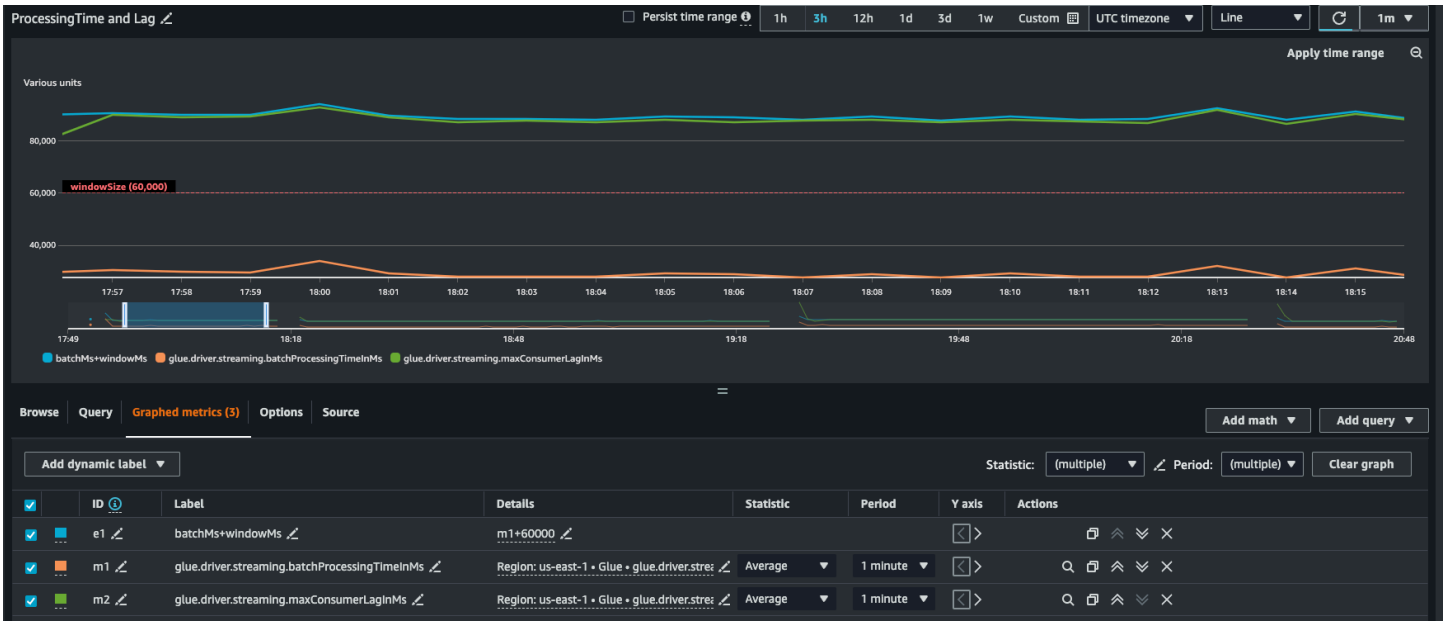



Questo parametro indica il numero di millisecondi necessari per elaborare ogni microbatch di record. L'obiettivo principale in questo caso è monitorare questo periodo per assicurarsi che sia inferiore all'intervallo `windowSize`. È accettabile che `batchProcessingTimeInMs` salga temporaneamente, purché torni alla normalità nell'intervallo di finestra successivo. L'indicatore n. 1 mostra un tempo più o meno stabile richiesto per elaborare il processo. Tuttavia, se il numero di record di input aumenta, il tempo necessario per elaborare il processo aumenta di conseguenza,

come segnalato dall'indicatore n. 2. Se `numRecords` non aumenta ma il tempo di elaborazione sì, è necessario esaminare più a fondo l'elaborazione del processo sugli esecutori. È buona norma impostare una soglia e un allarme per assicurarsi che `batchProcessingTimeInMs` non superi il 120% per più di 10 minuti. Per ulteriori informazioni sull'impostazione degli allarmi, consulta Using Amazon CloudWatch alarms.

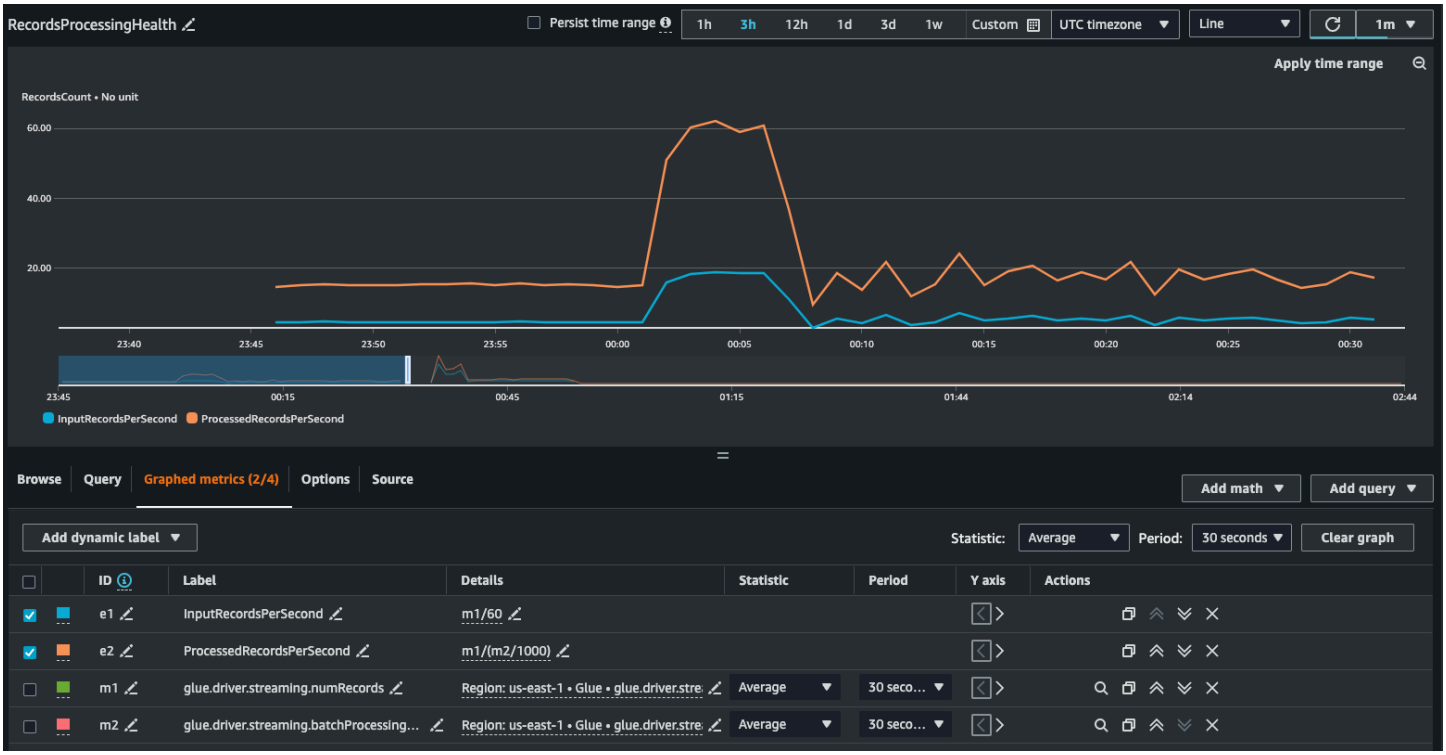
## Ritardo dei consumatori (metrica: streaming). maxConsumerLagInMs)

Il parametro del ritardo dei consumatori aiuta a comprendere se sussiste un ritardo nell'elaborazione degli eventi. Se il ritardo è troppo elevato, potresti non rispettare lo SLA di elaborazione sottoscritto dalla tua azienda, anche se disponi di un windowSize corretto. È necessario abilitare esplicitamente questi parametri utilizzando l'opzione di connessione `emitConsumerLagMetrics`. Per ulteriori informazioni, consulta KinesisStreamingSourceOptions.




## Parametri derivati

Per ottenere informazioni più approfondite, puoi creare metriche derivate per saperne di più sui tuoi lavori di streaming in Amazon CloudWatch.




Puoi creare un grafico con metriche derivate per decidere se è necessario utilizzarne di più. DPUs Sebbene il dimensionamento automatico ti aiuti a farlo automaticamente, puoi utilizzare parametri derivati per stabilire se il dimensionamento automatico funziona in modo efficace.

- InputRecordsPerSecondindica la frequenza con cui vengono ricevuti i record di input. È derivato come segue: numero di record di input (glue.driver.streaming.numRecords)/. WindowSize
- ProcessingRecordsPerSecondindica la velocità con cui vengono elaborati i record. È derivato come segue: numero di record di input (glue.driver.streaming.numRecords)/. batchProcessingTime InMs

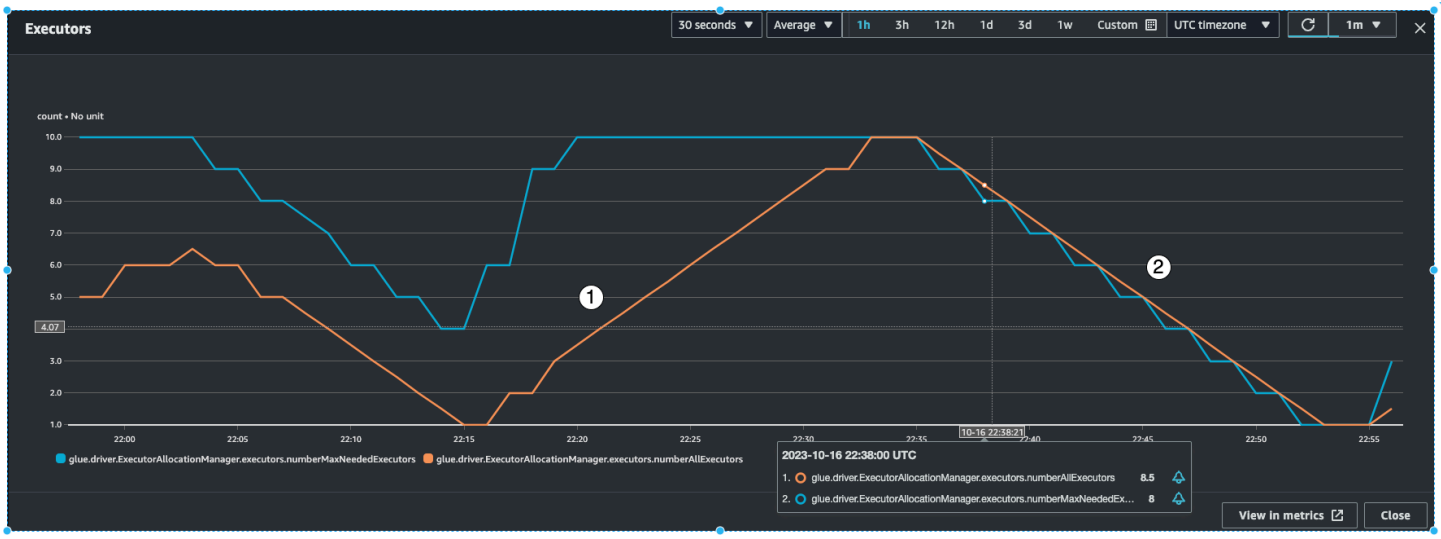
Se la velocità di input è superiore a quella di elaborazione, potrebbe essere necessario incrementare la capacità per elaborare il processo oppure aumentare il parallelismo.

## Parametri di dimensionamento automatico

Se il traffico in entrata ha molti picchi, dovresti valutare l'abilitazione del dimensionamento automatico e specificare il numero massimo di worker. In tal caso ottieni due parametri aggiuntivi, `numberAllExecutors` e `numberMaxNeededExecutors`.

- numberAllExecutorsè il numero di esecutori di lavori che eseguono attivamente
- numberMaxNeededExecutor è il numero massimo di job executor (in esecuzione attiva e in sospeso) necessari per soddisfare il carico corrente.

Questi due parametri ti aiuteranno a capire se il dimensionamento automatico funziona correttamente.




AWS Glue monitorerà la `batchProcessingTimeInMs` metrica su alcuni microbatch e farà una delle due cose. Aumenterà gli esecutori, se `batchProcessingTimeInMs` è più vicino a `windowSize`, oppure ridurrà gli esecutori, se `batchProcessingTimeInMs` è relativamente più basso di `windowSize`. Inoltre, utilizzerà un algoritmo per dimensionare gradualmente gli esecutori.

- L'indicatore n. 1 mostra come gli esecutori attivi sono aumentati fino a raggiungere il numero massimo di esecutori necessari per elaborare il carico.
- L'indicatore n. 2 mostra che gli esecutori attivi sono diminuiti rispetto a quando `batchProcessingTimeInMs` era basso.

È possibile utilizzare questi parametri per monitorare l'attuale parallelismo a livello di esecutore e regolare di conseguenza il numero massimo di worker nella configurazione di dimensionamento automatico.

## Come ottenere le prestazioni migliori

Spark cercherà di creare per ogni shard un'attività da cui leggere nel flusso Amazon Kinesis. I dati in ogni shard diventano una partizione. Quindi distribuirà queste attività tra gli esecutori/worker, a seconda del numero di core di ciascun worker (il numero di core per worker dipende dal tipo di worker selezionato, come `G.025X`, `G.1X` e così via). Tuttavia, il modo in cui le attività vengono distribuite non è deterministico. Tutte le attività vengono eseguite in parallelo sui rispettivi core. Se sono presenti più shard rispetto al numero di core esecutori disponibili, le attività vengono messe in coda.

È possibile utilizzare una combinazione dei parametri precedenti e del numero di shard per fornire agli esecutori un carico stabile con un certo margine per eventuali picchi. Si consiglia di eseguire alcune iterazioni del processo per determinare il numero approssimativo di worker. Per un unstable/spiky carico di lavoro puoi fare lo stesso impostando la scalabilità automatica e il numero massimo di lavoratori.

Imposta il valore di `windowSize` in base ai requisiti SLA della tua azienda. Ad esempio, se la tua azienda richiede che i dati elaborati non possano essere più vecchi di 120 secondi, imposta un valore di `windowSize` di almeno 60 secondi, in modo che il ritardo medio dei consumatori sia inferiore a 120 secondi (consulta la sezione precedente sul ritardo dei consumatori). Da lì, a seconda del numero `numRecords` e del numero di frammenti, pianifica la capacità in modo da DPUs assicurarti di occupare meno del 70% del tuo `batchProcessingTimeInMs` `windowSize` tempo per la maggior parte del tempo.

> **Note**
>
> Gli hot shard possono causare la distorsione dei dati, il che significa che alcuni shards/partitions sono molto più grandi degli altri. Ciò può far sì che alcune attività eseguite in parallelo richiedano più tempo, cosicché alcune attività restano indietro. Di conseguenza, il batch successivo non può iniziare fino al completamento di tutte le attività del precedente, il che influirà sul valore di `batchProcessingTimeInMillis` e sul ritardo massimo.

# Zero-ETL integrazioni

Zero-ETLè un set di integrazioni completamente gestite AWS che riduce al minimo la necessità di creare pipeline di dati ETL per casi d'uso comuni di acquisizione e replica. Rende disponibili i dati nell'architettura Lakehouse di Amazon SageMaker e Amazon Redshift da più fonti operative, transazionali e applicative. Attualmente AWS Glue Zero-ETL supporta come sorgenti DynamoDB, Oracle Database@AWS e SaaS come Salesforce, SAP, Zendesk. Con l'integrazione zero-ETL, hai a disposizione dati più aggiornati per l'analisi e il reporting. AI/ML Si ottengono informazioni più accurate e tempestive per casi d'uso come dashboard aziendali, esperienza di gioco ottimizzata, monitoraggio della qualità dei dati e analisi del comportamento dei clienti. È possibile fare previsioni basate sui dati con maggiore sicurezza, migliorare le esperienze dei clienti e promuovere approfondimenti basati sui dati in tutta l'azienda.

Amazon Redshift è un servizio di data warehouse veloce, completamente gestito e della dimensione di petabyte che semplifica ed economica l'analisi efficiente di tutti i dati utilizzando gli strumenti di business intelligence esistenti.

L'architettura Lakehouse di Amazon SageMaker unifica tutti i tuoi dati nei data lake e nei data warehouse di Amazon Simple Storage Service (Amazon S3), aiutandoti a creare analisi Amazon Redshift AI/ML e applicazioni potenti su un'unica copia dei dati. L'architettura Lakehouse di Amazon ti SageMaker offre la flessibilità di accedere e interrogare i tuoi dati sul posto con tutti gli strumenti e i motori compatibili con Apache Iceberg. Con SageMaker Lakehouse, hai anche la flessibilità di accedere e interrogare i tuoi dati sul posto con strumenti e motori compatibili con Apache Iceberg. Inoltre, è possibile proteggere i propri dati con controlli di accesso integrati e granulari, che vengono applicati a tutti i dati in tutti gli strumenti e i motori di analisi. È possibile definire le autorizzazioni una sola volta e condividere con sicurezza i dati all'interno dell'organizzazione.

## Zero-ETL funzionalità in AWS Glue

Zero-ETL integrazioni per AWS Glue semplificare l'inserimento e la replica dei dati dai servizi di AWS dati e dalle applicazioni di terze parti verso le destinazioni. AWS

AWS i servizi supportati come fonti zero-ETL includono: AWS Glue

- Amazon DynamoDB
- Oracle at, ODB AWS

Third-party le applicazioni supportate come sorgenti zero-ETL includono: AWS Glue

- Annunci su Facebook
- Annunci Instagram
- Salesforce
- Salesforce Marketing Cloud Account Engagement
- SAP OData
- ServiceNow
- Zendesk
- Zoho CRM

Self-managed i database supportati come sorgenti zero-ETL includono: AWS Glue

- Oracle
- SQL Server
- MySQL
- PostgreSQL

Per ulteriori informazioni sulle integrazioni zero-ETL da fonti autogestite, consulta Integrazione zero-ETL per fonti di database autogestite.AWS

AWS i AWS Glue servizi supportati dagli obiettivi includono: Zero-ETL

- Bucket Amazon S3 per uso generico tramite l'architettura Lakehouse di Amazon SageMaker
- Tabelle Amazon S3 tramite l'architettura Lakehouse di Amazon SageMaker
- Redshift Managed Storage tramite l'architettura Lakehouse di Amazon SageMaker
- Amazon Redshift Magazzino dati

# Configurazione di un'origine per un'integrazione Zero-ETL

La configurazione di un'integrazione richiede alcuni prerequisiti sull'origine, come la configurazione dei ruoli o delle politiche IAM che AWS Glue utilizza per accedere ai dati dall'origine (per scrivere sulla destinazione), l'uso di chiavi KMS per crittografare i dati in una posizione intermedia, ecc.

Argomenti


- Configurazione del codice sorgente Amazon DynamoDB
- Configurazione dell'origine della connessione (SaaS)
- Configurazione del database Oracle AWS come fonte


## Configurazione del codice sorgente Amazon DynamoDB

Per accedere ai dati dalla tabella Amazon DynamoDB di origine AWS Glue , è necessario l'accesso per descrivere la tabella ed esportare i dati da essa. Amazon DynamoDB ha introdotto di recente una funzionalità che consente di configurare una policy Resource Based Access (RBAC).

Il seguente esempio di policy Resource Based Access (RBAC) utilizza una wild card (*) per l'integrazione:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "1111",
      "Effect": "Allow",
      "Principal": {
        "Service": "glue.amazonaws.com"
      },
      "Resource": "*",
      "Action": [
        "dynamodb:ExportTableToPointInTime",
        "dynamodb:DescribeTable",
        "dynamodb:DescribeExport"
      ],
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "111122223333"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:glue:us-east-1:111122223333:integration:*"
        }
      }
    }
  ]
}
```

- Per la tabella Amazon DynamoDB che desideri replicare, incolla il modello di policy RBAC sopra riportato nella policy for table basata sulle risorse e compila i campi.
- Se si desidera rendere la policy restrittiva è necessario aggiornarla dopo aver creato l'integrazione e specificare l'`integrationArn` completa e utilizzare la condizione `StringEquals` invece di `StringLike`.
- Assicurati che Point-in-time il ripristino (PITR) sia abilitato per la tabella Amazon DynamoDB.
- Assicurarsi di aggiungere `Describe Export` alla policy Resource Based Access (RBAC).

È possibile aggiungere la policy RBAC alla tabella anche utilizzando il comando seguente:

```
aws dynamodb put-resource-policy \
--resource-arn arn:aws:dynamodb:region:account-id:table/ddb-table-name \
--policy file://resource-policy-with-condition.json \
--region region
```

Per verificare che la policy sia applicata correttamente, utilizzare il comando seguente per ottenere la policy delle risorse per una tabella:

```
aws dynamodb get-resource-policy \
--resource-arn arn:aws:dynamodb:region:account-id:table/ddb-table-name \
--region region
```

## Configurazione dell'origine della connessione (SaaS)

### Prerequisiti per la configurazione di un'integrazione

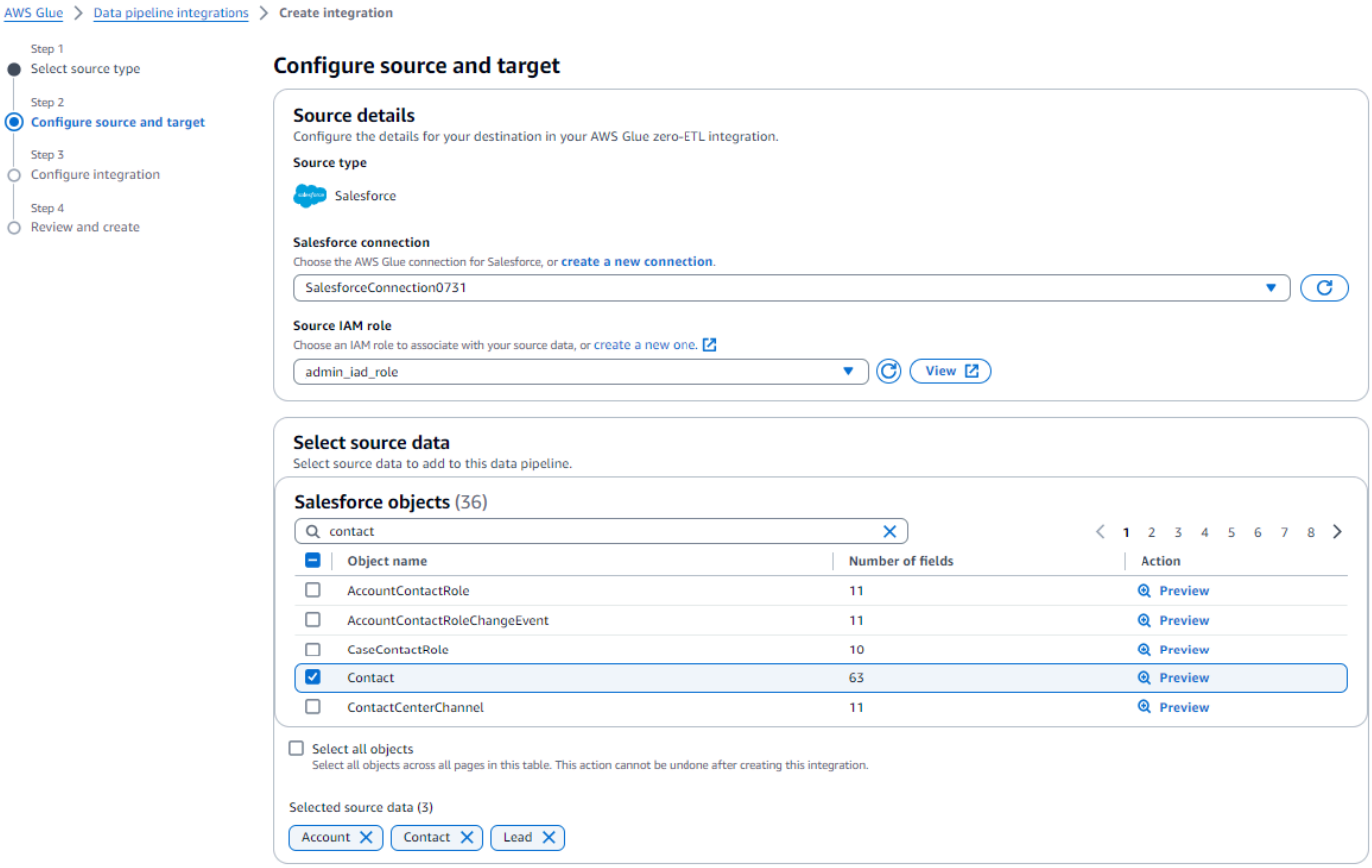
Prima di creare un'integrazione zero-ETL da fonti SaaS, è necessario completare le seguenti attività di configurazione:

- Crea una AWS Glue connessione con la fonte SaaS.
- Crea il ruolo di origine per fornire l'accesso alla connessione.
- Associa il ruolo alla connessione.

### Configurazione del codice sorgente di Salesforce

Per creare una connessione per una fonte Salesforce, vedi Connessione a Salesforce.

Dopo aver creato la connessione è possibile specificare i dati di origine da replicare.




Utilizzando l'integrazione Zero-ETL è possibile eseguire operazioni DDL per le entità supportate. Per un elenco delle entità che non sono supportate, consultare Entità e campi non supportati per Salesforce.

Configurazione Salesforce aggiuntiva

Salesforce Zero-ETL necessita dell'autorizzazione Lake Formation AWS Glue sul database, altrimenti `IngestionFailed` riceverà dal Log il seguente errore:

```
"errorMessage": "Insufficient lake formation permissions on Target Glue database."
```

Entità e campi non supportati per Salesforce

L'utilizzo delle entità o dei campi Salesforce seguenti non è supportato in un'integrazione Zero-ETL con un'origine Salesforce.

```
AccountChangeEvent, AccountContactRoleChangeEvent, AccountHistory, AccountShare,
 ActiveFeatureLicenseMetric, ActivePermSetLicenseMetric, ActiveProfileMetric,
 ActivityFieldHistory, amzsec__asi_Telemetry_Data_Store__ChangeEvent,
 amzsec__asi_Telemetry_Data_Store__History, amzsec__asi_Telemetry_Data_Store__Share,
```

```
amzsec__asi_Telemetry_Job_Log__ChangeEvent, amzsec__asi_Telemetry_Job_Log__History,
amzsec__asi_Telemetry_Job_Log__Share, amzsec__asi_Telemetry_Requirement__ChangeEvent,
amzsec__asi_Telemetry_Requirement__History, amzsec__asi_Telemetry_Requirement__Share,
ApexClass, ApexComponent, ApexLog, ApexPage, ApexTestQueueItem, ApexTestResult,
ApexTrigger, AssetChangeEvent, AssetHistory, AssetRelationshipHistory,
AssetShare, AssignmentRule, AssociatedLocationHistory, AsyncApexJob,
AuditTrailFileExportShare, AuthorizationFormConsentChangeEvent,
AuthorizationFormConsentHistory, AuthorizationFormConsentShare,
AuthorizationFormDataUseHistory, AuthorizationFormDataUseShare,
AuthorizationFormHistory, AuthorizationFormShare, AuthorizationFormTextHistory,
AuthProvider, AuthSession, BatchJobHistory, BatchJobPartFailedRecordHistory,
BatchJobPartHistory, BatchJobShare, BrandTemplate, BriefcaseAssignmentChangeEvent,
BriefcaseDefinitionChangeEvent, BusinessBrandShare, BusinessHours,
BusinessProcess, CalcMatrixColumnRangeHistory, CalcProcStepRelationshipHistory,
CalculationMatrixColumnHistory, CalculationMatrixHistory, CalculationMatrixRowHistory,
CalculationMatrixShare, CalculationMatrixVersionHistory, CalculationProcedureHistory,
CalculationProcedureShare, CalculationProcedureStepHistory,
CalculationProcedureVariableHistory, CalculationProcedureVersionHistory, Calendar,
CalendarViewShare, CallCenter, CallCoachConfigModifyEvent, CampaignChangeEvent,
CampaignHistory, CampaignMemberChangeEvent, CampaignMemberStatusChangeEvent,
CampaignShare, CaseChangeEvent, CaseHistory, CaseHistory2 CaseHistory2ChangeEvent,
CaseRelatedIssueChangeEvent, CaseRelatedIssueHistory, CaseShare, CaseStatus,
CaseTeamMember, CaseTeamRole, CaseTeamTemplate, CaseTeamTemplateMember,
CaseTeamTemplateRecord, CategoryNode, ChangeRequestChangeEvent, ChangeRequestHistory,
ChangeRequestRelatedIssueChangeEvent, ChangeRequestRelatedIssueHistory,
ChangeRequestRelatedItemChangeEvent, ChangeRequestRelatedItemHistory,
ChangeRequestShare, ChatRetirementRdyMetrics, ChatterActivity, ClientBrowser,
CollaborationGroup, CollaborationGroupMember, CollaborationGroupMemberRequest,
CollaborationInvitation, CommSubscriptionChannelTypeHistory,
CommSubscriptionChannelTypeShare, CommSubscriptionConsentChangeEvent,
CommSubscriptionConsentHistory, CommSubscriptionConsentShare, CommSubscriptionHistory,
CommSubscriptionShare, CommSubscriptionTimingHistory, Community, ConnectedApplication,
ContactChangeEvent, ContactHistory, ContactPointAddressChangeEvent,
ContactPointAddressHistory, ContactPointAddressShare, ContactPointConsentChangeEvent,
ContactPointConsentHistory, ContactPointConsentShare, ContactPointEmailChangeEvent,
ContactPointEmailHistory, ContactPointEmailShare, ContactPointPhoneChangeEvent,
ContactPointPhoneHistory, ContactPointPhoneShare, ContactPointTypeConsentChangeEvent,
ContactPointTypeConsentHistory, ContactPointTypeConsentShare, ContactRequestShare,
ContactShare, ContentDocumentChangeEvent, ContentDocumentHistory,
ContentDocumentLink, ContentDocumentLinkChangeEvent, ContentDocumentSubscription,
ContentFolderItem, ContentFolderLink, ContentFolderMember, ContentNote,
ContentNotification, ContentTagSubscription, ContentUserSubscription,
ContentVersionChangeEvent, ContentVersionComment, ContentVersionHistory,
ContentVersionRating, ContentWorkspace, ContentWorkspaceMember,
```

```
ContentWorkspacePermission, ContentWorkspaceSubscription, ContractChangeEvent,
ContractHistory, ContractLineItemChangeEvent, ContractLineItemHistory,
ContractStatus, Conversation, ConversationParticipant, CronJobDetail,
CronTrigger, CustomBrand, CustomBrandAsset, CustomerShare, CustomHttpHeader,
DashboardComponent, DataUseLegalBasisHistory, DataUseLegalBasisShare,
DataUsePurposeHistory, DataUsePurposeShare, DecisionTableRecordset, DeleteEvent,
DocumentAttachmentMap, Domain, DomainSite, DTRecordsetReplicaShare, EmailBounceEvent,
EmailMessageChangeEvent, EmailServicesAddress, EmailServicesFunction,
EmailTemplate, EmailTemplateChangeEvent, EngagementAttendeeChangeEvent,
EngagementAttendeeHistory, EngagementChannelTypeHistory, EngagementChannelTypeShare,
EngagementInteractionChangeEvent, EngagementInteractionHistory,
EngagementInteractionShare, EngagementInterface, EngagementTopicChangeEvent,
EngagementTopicHistory, EntitlementChangeEvent, EntitlementHistory,
EntitlementTemplate, EntityMilestoneHistory, EntitySubscription, EventChangeEvent,
EventRelationChangeEvent, EventRelayConfigChangeEvent, ExpressionSetHistory,
ExpressionSetShare, ExpressionSetVersionHistory, ExternalEventMappingShare,
FeedAttachment, FeedLike, FeedPollChoice, FeedPollVote, FeedSignal,
FieldPermissions, FieldSecurityClassification, FiscalYearSettings,
FlowInterviewLogShare, FlowInterviewShare, FlowOrchestrationEvent,
FlowOrchestrationInstanceShare, FlowOrchestrationStageInstanceShare,
FlowOrchestrationStepInstanceShare, FlowOrchestrationWorkItemShare,
FlowRecordShare, FlowRecordVersionChangeEvent, FlowTestResultShare, Folder,
Group, GroupMember, Holiday, IdeaComment, IdpEventLog, ImageHistory, ImageShare,
IncidentChangeEvent, IncidentHistory, IncidentRelatedItemChangeEvent,
IncidentRelatedItemHistory, IncidentShare, IndividualChangeEvent,
IndividualHistory, IndividualShare, KnowledgeableUser, LeadChangeEvent, LeadHistory,
LeadShare, LeadStatus, LightningExitByPageMetrics, LightningToggleMetrics,
LightningUsageByAppTypeMetrics, LightningUsageByBrowserMetrics,
LightningUsageByFlexiPageMetrics, LightningUsageByPageMetrics, ListEmailChangeEvent,
ListEmailShare, ListView, LocationChangeEvent, LocationHistory, LocationShare,
LocationTrustMeasureShare, LoginHistory, LoginIp, MacroChangeEvent,
MacroHistory, MacroInstructionChangeEvent, MacroShare, MacroUsageShare,
ManagedContentVariantChangeEvent, MessagingEndUserHistory, MessagingEndUserShare,
MessagingSessionHistory, MessagingSessionShare, MilestoneType, MLEngagementEvent,
ObjectPermissions, OpportunityChangeEvent, OpportunityContactRoleChangeEvent,
OpportunityFieldHistory, OpportunityLineItemChangeEvent, OpportunityShare,
OpportunityStage, OrderChangeEvent, OrderHistory, OrderItemChangeEvent,
OrderItemHistory, OrderShare, OrderStatus, Organization, OrgEmailAddressSecurity,
OrgWideEmailAddress, OutgoingEmail, OutgoingEmailRelation, PackageLicense,
PartnerRole, PartyConsentChangeEvent, PartyConsentHistory, PartyConsentShare,
Period, PermissionSet, PermissionSetAssignment, PermissionSetTabSetting,
Pricebook2ChangeEvent, Pricebook2History, PricebookEntryChangeEvent,
PricebookEntryHistory, PrivacyJobSessionShare, PrivacyObjectSessionShare,
PrivacyRTBFRequestHistory, PrivacyRTBFRequestShare, PrivacySessionRecordFailureShare,
```

```
ProblemChangeEvent, ProblemHistory, ProblemIncidentChangeEvent,
ProblemIncidentHistory, ProblemRelatedItemChangeEvent, ProblemRelatedItemHistory,
ProblemShare, ProcessDefinition, ProcessExceptionEvent, ProcessExceptionShare,
ProcessInstanceChangeEvent, ProcessInstanceStep, ProcessInstanceStepChangeEvent,
ProcessNode, Product2ChangeEvent, Product2History, ProductEntitlementTemplate,
Profile, ProfileSkillEndorsementHistory, ProfileSkillHistory,
ProfileSkillShare, ProfileSkillUserHistory, PromptActionShare, PromptErrorShare,
QueueSobject, QuickTextChangeEvent, QuickTextHistory, QuickTextShare,
QuickTextUsageShare, RecentlyViewed, RecommendationChangeEvent,
RecordActionHistory, RecordAlertHistory, RecordAlertShare, RecordType,
ScorecardShare, SellerHistory, SellerShare, ServiceContractChangeEvent,
ServiceContractHistory, ServiceContractShare, SetupAuditTrail, SetupEntityAccess,
SharingRecordCollectionShare, Site, SiteHistory, SiteRedirectMapping,
SocialPersonaHistory, SocialPostChangeEvent, SocialPostHistory, SocialPostShare,
SolutionHistory, SolutionStatus, StaticResource, StreamingChannelShare,
TableauHostMappingShare, TaskChangeEvent, TaskPriority, TaskStatus,
ThreatDetectionFeedback, TimelineObjectDefinitionChangeEvent, TodayGoalShare,
Topic, TopicUserEvent, Translation, User, UserAppMenuCustomizationShare,
UserChangeEvent, UserDefinedLabelAssignmentShare, UserDefinedLabelShare,
UserEmailPreferredPersonShare, UserLicense, UserLogin, UserPackageLicense,
UserPreference, UserPrioritizedRecordShare, UserProvisioningRequestShare,
UserRole, UserShare, VideoCallChangeEvent, VideoCallParticipantChangeEvent,
VideoCallRecordingChangeEvent, VideoCallShare, VisualforceAccessMetrics,
VoiceCallChangeEvent, VoiceCallRecordingChangeEvent, VoiceCallShare, Vote,
WebLink, WorkAccessShare, WorkBadgeDefinitionHistory, WorkBadgeDefinitionShare,
WorkOrderChangeEvent, WorkOrderHistory, WorkOrderLineItemChangeEvent,
WorkOrderLineItemHistory, WorkOrderLineItemStatus, WorkOrderShare, WorkOrderStatus,
WorkPlanChangeEvent, WorkPlanHistory, WorkPlanShare, WorkPlanTemplateChangeEvent,
WorkPlanTemplateEntryChangeEvent, WorkPlanTemplateEntryHistory,
WorkPlanTemplateHistory, WorkPlanTemplateShare, WorkStepChangeEvent, WorkStepHistory,
WorkStepStatus, WorkStepTemplateChangeEvent, WorkStepTemplateHistory,
WorkStepTemplateShare, WorkThanksShare
```

## Configurazione di un'origine Salesforce Marketing Cloud Account Engagement

Per creare una connessione per una fonte Salesforce Marketing Cloud Account Engagement, vedi Connessione a Salesforce Marketing Cloud Account Engagement.

Utilizzando l'integrazione Zero-ETL è possibile eseguire operazioni DDL per le seguenti entità supportate:

| Etichetta dell'entità | Nome dell'entità |
|---|---|
| Campagna | campaign |
| List | list |
| Contenuti dinamici | dynamic-content |
| Elenco dei membri | list-membership |
| Prospettiva | prospect |
| Utente | user |
| EmailTemplate | email-template |
| EngagementStudioProgram | engagement-studio-program |
| Pagina di destinazione | landing-page |
| Elenco di e-mail | list-email |

## Configurazione di una fonte SAP OData

Per creare una connessione per una OData fonte SAP, vedi Connessione a SAP. OData

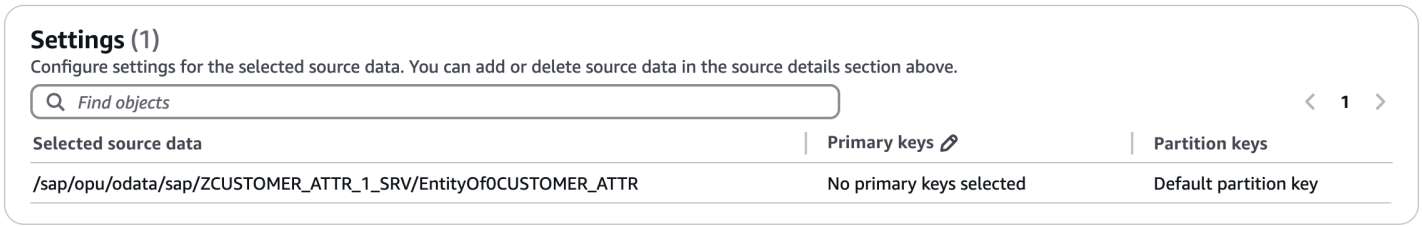
Le integrazioni zero-ETL con una fonte SAP OData ora supportano entità a partire da. `Entity0f` La possibilità di sovrascrivere la chiave primaria è attualmente supportata solo per gli oggetti. SAPOData `Entity0f` Una volta impostata, questa proprietà non può essere modificata.

Support per entità SAP speciali

AWS Glue Zero-ETL supporta OData le entità SAP che utilizzano il framework ODP (Operational Data Provisioning) di SAP e quelle che non utilizzano il framework ODP (entità non ODP). L'elenco delle entità supportate include: ODP_SAP (Business Warehouse o BW extractors), ODP_CDS (Core Data Services o CDS Views) e servizi non basati su ODP per SAP. OData APIs AWS Glue Zero-ETL supporta l'acquisizione completa di istantanee e modifiche incrementali dei dati per entità SAP ODP e non ODP. Per le entità ODP, le modifiche incrementali vengono acquisite utilizzando collegamenti delta. Per le entità non ODP, se viene selezionato un campo interrogabile che può essere utilizzato per l'ingestione basata su timestamp, Zero-ETL utilizzerà quel campo per l'ingestione incrementale.

Durante l'acquisizione di dati da entità SAP utilizzando zero-ETL, è necessario tenere presente quanto segue: AWS Glue

- Zero-ETL può importare solo entità SAP che sono state configurate per il metodo GET_ENTITYSET in SAP.
- Per le entità SAP non ODP, se non viene selezionato un campo timestamp per gli aggiornamenti incrementali, Zero-ETL supporta l'estrazione e la replica complete dei dati solo con upsert (nessuna eliminazione). AWS Glue
- Per le entità di estrazione ODP, determiniamo i set di chiavi primarie validi durante l'elaborazione dei dati. Altre entità SAP richiedono un passaggio aggiuntivo per fornire il set di chiavi primarie valido come input, in particolare le entità SAP che iniziano con. `EntityOf` Quando viene selezionata un'entità `EntityOf`, verrà richiesto di fornire il set di chiavi primarie.




## Configurazione di una fonte ServiceNow

Per creare una connessione per una ServiceNow fonte, vedi Connessione a ServiceNow.

Entità e campi non supportati per ServiceNow

L'uso ServiceNow delle entità o dei campi seguenti non è supportato in un'integrazione zero-ETL con un codice sorgente. ServiceNow

```
ais_acl_overrides, ais_async_genius_result, ais_async_request, ais_connection,
 ais_genius_result_configuration_parameters, ais_partition_health,
 ais_partition_health_response, ais_publish_history, ais_relevancy_training_execution,
 ais_relevancy_training_staging, ais_search_profile_relevancy_model,
 catalog_draft_entities, clone_log, clone_log0000, clone_log0001, clone_log0002,
 clone_log0003, clone_log0004, clone_log0005, clone_log0006, clone_log0007,
 cmdb_ie_context, cmdb_ie_log, cmdb_ie_run, cmdb_ire_partial_payloads_index,
 cmdb_qb_result_base$par1, cmdb$par1, discovery_log, discovery_log0000,
 discovery_log0001, discovery_log0002, discovery_log0003, discovery_log0004,
 discovery_log0005, discovery_log0006, discovery_log0007, ecc_agent_log,
 entitlement_data, entl_subscription_map, gs_entitlement_plugin_mapping,
```

```
ih_transaction_exclusion, import_log, import_log0000, import_log0001, import_log0002,
import_log0003, import_log0004, import_log0005, import_log0006, import_log0007,
jrobin_archive, jrobin_database, jrobin_datasource, jrobin_definition, jrobin_graph,
jrobin_graph_line, jrobin_graph_set, jrobin_graph_set_member, jrobin_shard,
jrobin_shard_location, license_role_discovery_run, logger_configuration_validation,
m2m_analytics_event_logger, m2m_user_consent_info, ml_artifact_object_store,
np$sys_gen_ai_filter_sample, np$sys_ui_element, np$sys_ui_list_element,
one_api_service_plan_feature_invocation, one_api_service_plan_invocation,
open_nlu_predict_log, open_nlu_predict_log0000, open_nlu_predict_log0001,
open_nlu_predict_log0002, open_nlu_predict_log0003, open_nlu_predict_log0004,
open_nlu_predict_log0005, open_nlu_predict_log0006, open_nlu_predict_log0007,
pa_diagnostic_log, pa_diagnostic_log0000, pa_diagnostic_log0001,
pa_diagnostic_log0002, pa_diagnostic_log0003, pa_diagnostic_log0004,
pa_diagnostic_log0005, pa_diagnostic_log0006, pa_diagnostic_log0007, pa_favorites,
pa_job_log_rows, pa_job_log_rows0000, pa_job_log_rows0001, pa_job_log_rows0002,
pa_job_log_rows0003, pa_job_log_rows0004, pa_job_log_rows0005, pa_job_log_rows0006,
pa_job_log_rows0007, pa_migration_ignored_scores, pa_scores_l1, pa_scores_l2,
pa_scores_migration_groups, par_dashboard_conversion_backup, promin_log,
promin_log0000, promin_log0001, promin_log0002, promin_log0003, promin_log0004,
promin_log0005, promin_log0006, promin_log0007, promin_request_object,
proposed_change_verification_log, proposed_change_verification_log0000,
proposed_change_verification_log0001, proposed_change_verification_log0002,
proposed_change_verification_log0003, proposed_change_verification_log0004,
proposed_change_verification_log0005, proposed_change_verification_log0006,
proposed_change_verification_log0007, protected_table_log, protected_table_log0000,
protected_table_log0001, protected_table_log0002, protected_table_log0003,
protected_table_log0004, protected_table_log0005, protected_table_log0006,
protected_table_log0007, pwd_history, qb_query_results, scan_log,
scan_log0000, scan_log0001, scan_log0002, scan_log0003, scan_log0004,
scan_log0005, scan_log0006, scan_log0007, schema_validator_error,
sla_repair_log_entry, sla_repair_log_entry0000, sla_repair_log_entry0001,
sla_repair_log_entry0002, sla_repair_log_entry0003, sla_repair_log_entry0004,
sla_repair_log_entry0005, sla_repair_log_entry0006, sla_repair_log_entry0007,
sla_repair_log_message, sla_repair_log_message0000, sla_repair_log_message0001,
sla_repair_log_message0002, sla_repair_log_message0003, sla_repair_log_message0004,
sla_repair_log_message0005, sla_repair_log_message0006, sla_repair_log_message0007,
sn_bm_client_activity, sn_ci_analytics_st_actionable_notifs,
sn_ci_analytics_st_conv_completion_by_cat, sn_ci_analytics_st_conv_dynamic_property,
sn_ci_analytics_st_conversation, sn_ci_analytics_st_count_by_date,
sn_ci_analytics_st_event_occurrence, sn_ci_analytics_st_event_property_value_trend,
sn_ci_analytics_st_issue_auto_resolution, sn_ci_analytics_st_no_clicks,
sn_ci_analytics_st_no_results, sn_ci_analytics_st_property_summary_by_event,
sn_ci_analytics_st_session_count_per_locale, sn_ci_analytics_st_session_duration,
sn_ci_analytics_st_spokes_usage, sn_ci_analytics_st_topic_execution_stats,
```

```
sn_ci_analytics_st_topic_occurrence, sn_ci_analytics_st_trending_content,
sn_ci_analytics_st_trending_queries, sn_ci_analytics_st_users,
sn_cs_plugin_signatures, sn_cs_telemetry_log, sn_dfc_application, sn_dfc_product,
sn_employee_position, sn_entitlement_st_subscription_application_family,
sn_entitlement_st_subscription_application_users, sn_hr_sp_st_relevant_for_you,
sn_instance_clone_log, sn_instance_clone_log0000, sn_instance_clone_log0001,
sn_instance_clone_log0002, sn_instance_clone_log0003, sn_instance_clone_log0004,
sn_instance_clone_log0005, sn_instance_clone_log0006, sn_instance_clone_log0007,
sn_km_mr_st_kb_knowledge, sn_me_st_topic, sn_rf_conditional_definition,
sn_rf_evaluation_type, sn_rf_evaluation_type_input, sn_rf_recommendation_action,
sn_rf_recommendation_experience, sn_rf_recommendation_history,
sn_rf_recommendation_rule, sn_rf_record_display_configuration, sn_rf_trend_definition,
sn_sub_man_st_account_level_entitlement, sn_sub_man_st_gen_ai_metadata,
sn_sub_man_st_instance_used_assist_count, sn_sub_man_st_now_assist_creator_instances,
sn_sub_man_st_now_assists_aggregate, sn_sub_man_st_subscribed_groups,
sn_sub_man_st_subscription_insights, sn_sub_man_st_subscription_license_detail_metric,
sn_sub_man_st_unallocated_group_recommendation, sn_sub_man_st_unconfirmed_user_group,
sn_wn_user_app_activity, sn_wn_user_content_activity, snc_monitorable_item,
snpar_sched_export_v_scheduled_export_visualization, spotlight,
spotlight_audit, spotlight_copy_log_row, spotlight_copy_log_row0000,
spotlight_copy_log_row0001, spotlight_copy_log_row0002, spotlight_copy_log_row0003,
spotlight_copy_log_row0004, spotlight_copy_log_row0005, spotlight_copy_log_row0006,
spotlight_copy_log_row0007, spotlight_job_log_row, spotlight_job_log_row0000,
spotlight_job_log_row0001, spotlight_job_log_row0002, spotlight_job_log_row0003,
spotlight_job_log_row0004, spotlight_job_log_row0005, spotlight_job_log_row0006,
spotlight_job_log_row0007, st_dfc_performance_metric, st_license_detail_metric,
st_on_call_hour, st_sc_wizard_question, st_sys_catalog_items_and_variable_sets,
st_sys_design_system_icon, subscription_instance_stats, svc_container_config,
svc_environment_config, svc_layer_config, svc_model_assoc_ci,
svc_model_checkpoint_attr, svc_model_obj_cluster, svc_model_obj_constraint,
svc_model_obj_deployable, svc_model_obj_element, svc_model_obj_impact,
svc_model_obj_impactrule, svc_model_obj_package, svc_model_obj_path,
svc_model_obj_relation, svc_model_obj_service, sys_administrative_script_transaction,
sys_amb_message, sys_amb_processor, sys_analytics_batch_state,
sys_analytics_config, sys_analytics_data_points_error, sys_analytics_event,
sys_analytics_logger, sys_analytics_logger_field, sys_app_payload_loader_rule,
sys_app_payload_unloader_rule, sys_app_scan_payload, sys_app_scan_variable,
sys_app_scan_variable_type, sys_archive_destroy_log, sys_archive_destroy_run,
sys_archive_log, sys_archive_run, sys_atf_transaction_log, sys_attachment_doc,
sys_attachment_doc_v2, sys_attachment_soft_deleted, sys_audit, sys_audit_relation,
sys_auth_policy_api_allowed, sys_aw_registered_scripting_modal,
sys_cache_flush, sys_data_egress_source, sys_dm_delete_count,
sys_export_set_log, sys_export_set_log0000, sys_export_set_log0001,
sys_export_set_log0002, sys_export_set_log0003, sys_export_set_log0004,
```

```
sys_export_set_log0005, sys_export_set_log0006, sys_export_set_log0007,
sys_flow_compiled_flow, sys_flow_compiled_flow_chunk, sys_flow_context_chunk,
sys_flow_context_chunk_archive, sys_flow_context_inputs_chunk,
sys_flow_execution_history, sys_flow_log, sys_flow_log0000, sys_flow_log0001,
sys_flow_log0002, sys_flow_log0003, sys_flow_plan_context_binding,
sys_flow_report_doc, sys_flow_report_doc_chunk, sys_flow_report_doc_chunk_archive,
sys_flow_runtime_state_chunk, sys_flow_runtime_value_chunk,
sys_flow_subflow_plan_chunk, sys_flow_trigger_plan_chunk, sys_flow_val_listener,
sys_flow_value, sys_flow_value_chunk, sys_gen_ai_config_example,
sys_gen_ai_feature_mapping, sys_gen_ai_strategy_mapping, sys_gen_ai_usage_log,
sys_generative_ai_capability_definition, sys_generative_ai_log,
sys_generative_ai_response_validator, sys_generative_ai_validator,
sys_geo_routing, sys_geo_routing_config, sys_hop_token, sys_hub_action_plan_chunk,
sys_hub_snapshot_chunk, sys_journal_field, sys_journal_field_edit, sys_json_chunk,
sys_kaa_policy, sys_kaa_subidentity_assertion, sys_kaa_user_policy_mapping,
sys_mapplication, sys_mass_encryption_job, sys_notification_execution_log,
sys_notification_execution_log0000, sys_notification_execution_log0001,
sys_notification_execution_log0002, sys_notification_execution_log0003,
sys_notification_execution_log0004, sys_notification_execution_log0005,
sys_notification_execution_log0006, sys_notification_execution_log0007,
sys_nowmq_message, sys_nowmq_provider_param_definition, sys_orchestrator_action,
sys_pd_asset_configuration, sys_pd_context_chunk, sys_pd_context_log,
sys_pd_snapshot_chunk, sys_pd_trigger_license, sys_processing_framework_job,
sys_query_index_hint, sys_query_rewrite, sys_query_string_log, sys_replication_queue,
sys_replication_queue0, sys_replication_queue1, sys_replication_queue2,
sys_replication_queue3, sys_replication_queue4, sys_replication_queue5,
sys_replication_queue6, sys_replication_queue7, sys_request_performance,
sys_rollback_conflict, sys_rollback_incremental, sys_rollback_log,
sys_rollback_log0000, sys_rollback_log0001, sys_rollback_log0002,
sys_rollback_log0003, sys_rollback_log0004, sys_rollback_log0005,
sys_rollback_log0006, sys_rollback_log0007, sys_rollback_run,
sys_rollback_schema_change, sys_rollback_schema_conflict, sys_rollback_sequence,
sys_scheduler_assignment, sys_scheduler_memory_pressure_job_log,
sys_script_adapter_rule, sys_script_batch_adapter_rule, sys_search_source_filter,
sys_service_authentication, sys_signing_job, sys_suggestion_reader,
sys_sync_history_review, sys_trend, sys_unreferenced_preview,
sys_unreferenced_record_rule, sys_upgrade_manifest, sys_upgrade_state,
sys_ux_asset_cache_buster, sys_ux_lib_component_prop, sys_ux_lib_presource,
sys_ux_page_action, sys_ux_page_action_binding, sysevent_queue_runtime,
syslog, syslog_app_scope0000, syslog_app_scope0001, syslog_app_scope0002,
syslog_app_scope0003, syslog_app_scope0004, syslog_app_scope0005,
syslog_app_scope0006, syslog_app_scope0007, syslog_email0000, syslog_email0001,
syslog_email0002, syslog_email0003, syslog_email0004, syslog_email0005,
syslog_email0006, syslog_email0007, syslog_transaction, syslog_transaction0000,
```

```
syslog_transaction0001, syslog_transaction0002, syslog_transaction0003,
syslog_transaction0004, syslog_transaction0005, syslog_transaction0006,
syslog_transaction0007, syslog0000, syslog0001, syslog0002, syslog0003, syslog0004,
syslog0005, syslog0006, syslog0007, ts_attachment, ts_chain, ts_deleted_doc,
ts_document, ts_field, ts_index_stats, ts_phrase, ts_search_stats, ts_v4_attachment,
ts_word, ua_app_metadata, ua_audit_stats, ua_extra_page, ua_monitor_property,
ua_monitor_property_audit, ua_shared_service, ua_sn_table_inventory, ua_sp_known_bot,
ua_upload_log, spotlight_criteria, pa_snapshots, pa_widget_indicators, pa_widgets,
sn_cim_register, global, gsw_change_log, gsw_content, gsw_content_group,
gsw_content_information, gsw_status_of_content, multi_factor_browser_fingerprint,
multi_factor_criteria, pa_filters, password_policy, plan_execution,
plan_mysql, plan_oracle, plan_postgres, sc_rest_api_without_access_policy,
sn_actsub_activity, sn_actsub_activity_fanout, sn_actsub_activity_stream,
sn_actsub_activity_type, sn_actsub_atype_attributes, sn_actsub_atype_notif_pref,
sn_actsub_module, sn_actsub_notif_object, sn_actsub_subobject_stream,
sn_actsub_subscribable_object, sn_actsub_subscription_notif_pref,
sn_actsub_user_stream, sn_appclient_store_outbound_http_quota,
sn_appcreator_app_template, sn_critical_update, sn_docker_spoke_images,
sn_employee_app, sn_employee_app_access, sn_employee_app_access_criteria,
sn_entitlement_genai_assist_counts, sn_entitlement_genai_creator_user_counts,
sn_entitlement_genai_creator_users, sn_mif_instance, sn_mif_sync_data,
sn_mif_sync_status, sn_mif_table_registration, sn_mif_trust_config,
sn_vsc_best_practice_configurations, sn_vsc_best_practice_goals,
sn_vsc_changed_hardening_settings, sn_vsc_changed_scan_findings,
sn_vsc_check_security_area, sn_vsc_elevation_event, sn_vsc_event,
sn_vsc_export_event, sn_vsc_export_setting, sn_vsc_harc_compliance_status_lookup,
sn_vsc_hardening_compliance_scores, sn_vsc_impersonation_event,
sn_vsc_instance_hardening_settings, sn_vsc_login_event,
sn_vsc_scan_comparisons, sn_vsc_scan_summary, sn_vsc_security_check_categories,
sn_vsc_security_check_configurations, sn_vsc_security_configuration_groups,
sn_vsc_security_privacy_capabilities, sn_vsc_updated_settings,
sn_vsc_user_comparisons, sys_app_hash_inventory, sys_coalesce_strategy_deferred,
sys_flow_secure_data, sys_formula_function, sys_geocoding_request,
sys_global_file_hash, sys_import_set_row, sys_index, sys_index_explain,
sys_installation_schedule, sys_installation_schedule_item, sys_offline_app,
sys_package, sys_package_dependency_item, sys_package_dependency_m2m,
sys_plugins, sys_querystat, sys_reap_package, sys_scoped_plugin,
sys_stage_storage_alias, sys_storage_alias, sys_storage_table_alias, sys_store_app,
sys_table_partition, sys_upgrade_history_log, sys_user_public_credential,
sys_webauthn_authentication_request, sys_webauthn_registration_request,
syslog_app_scope, syslog_email, syslog_page_timing, ua_instance_state_config,
v_expression_cache, v_private_cache, v_shared_cache, sys_amb_message0002,
sys_amb_message0004, sys_amb_message0005, sys_metadata, v_par_unified_report_viz
```

## Configurazione di un'origine Zendesk

Per creare una connessione per una fonte Zendesk, vedi Connessione a Zendesk.

Utilizzando l'integrazione Zero-ETL è possibile eseguire le seguenti operazioni DDL per le entità supportate:

| Etichetta dell'entità | Nome dell'entità | Crea supportato | Aggiorna supportato | Elimina supportato |
|---|---|---|---|---|
| Ticket | biglietti | Y | Y | Y |
| Utente | utenti | Y | Y | Y |
| Indice di soddisfazione | satisfaction-rating | Y | Y | N |
| Articoli | articoli | Y | Y | N |
| Organizzazione | organizations | Y | Y | Y |
| Chiamate | chiamate | Y | Y | N |
| Leg chiamata | leg | Y | Y | N |

## Configurazione di un'origine Zoho CRM

Per creare una connessione per una fonte Zoho CRM, consulta Connessione a Zoho CRM.

Utilizzando l'integrazione Zero-ETL è possibile eseguire le seguenti operazioni DDL per le entità supportate:

| Etichetta dell'entità | Nome dell'entità | Inserimento DML | Modifica DML | Eliminazione DML | Inserimento DDL-D | Modifica DDL | DDL-Elimina |
|---|---|---|---|---|---|---|---|
| Lead | lead | Y | Y | Y | Y | Y | Y |
| Account | account | Y | Y | Y | Y | Y | Y |

| Etichetta dell'entità | Nome dell'entità | Inserimento DML | Modifica DML | Eliminazione DML | Inserimento DDL-D | Modifica DDL | DDL-Elimina |
|---|---|---|---|---|---|---|---|
| Contatti | contact | Y | Y | Y | Y | Y | Y |
| Campagne | campaign | Y | Y | Y | Y | Y | Y |
| Processi | task | Y | Y | Y | Y | Y | Y |
| Eventi | evento | Y | Y | Y | Y | Y | Y |
| Chiamate | call | Y | Y | Y | Y | Y | Y |
| Soluzioni | soluzione | Y | Y | Y | Y | Y | Y |
| Prodotti | prodotto | Y | Y | Y | Y | Y | Y |
| Fornitori | fornitore | Y | Y | Y | Y | Y | Y |
| Preventivi | preventivi | Y | Y | Y | Y | Y | Y |
| Ordini di vendita | sales-order | Y | Y | Y | Y | Y | Y |
| Ordini di acquisto | purchase-order | Y | Y | Y | Y | Y | Y |
| Fatture | fattura | Y | Y | Y | Y | Y | Y |
| Casi | caso | Y | Y | Y | Y | Y | Y |
| Prezzari | price-book | Y | Y | Y | Y | Y | Y |

## Configurazione di un'origine Facebook Ads

Per creare una connessione per una fonte di annunci di Facebook, consulta Connessione a Facebook Ads.

Utilizzando l'integrazione Zero-ETL è possibile eseguire le seguenti operazioni DDL per le entità supportate:

| Etichetta dell'entità | Nome dell'entità | Crea supportato | Aggiorna supportato | Elimina supportato |
|---|---|---|---|---|
| Risorsa | */adsets | Y | Y | Y |
| Campagna | */campaigns | Y | Y | Y |
| Inserzioni | */ads | Y | Y | Y |

## Configurazione di un'origine Instagram Ads

Per creare una connessione per una fonte di annunci Instagram, consulta Connessione a Instagram Ads.

Utilizzando l'integrazione Zero-ETL è possibile eseguire le seguenti operazioni DDL per le entità supportate:

| Nome dell'entità | Crea supportato | Aggiorna supportato | Elimina supportato |
|---|---|---|---|
| */adsets | Y | Y | Y |
| */campaigns | Y | Y | Y |
| */ads | Y | Y | Y |

## Impostazione del ruolo di origine per l'integrazione

Crea un ruolo di origine per consentire all'integrazione zero-ETL di accedere alla tua connessione. Questo è applicabile solo per le fonti SaaS. Questo è un prerequisito per creare integrazioni con sorgenti SaaS.

> **Note**
>
> Per limitare l'accesso solo a poche connessioni, è possibile creare prima la connessione per ottenere l'ARN della connessione. Per informazioni, consulta Configurazione di un'origine per un'integrazione Zero-ETL.

Creare un ruolo con le autorizzazioni per l'integrazione per accedere alla connessione:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "GlueConnections",
            "Effect": "Allow",
            "Action": [
                "glue:GetConnections",
                "glue:GetConnection"
            ],
            "Resource": [
                "arn:aws:glue:*:111122223333:catalog",
                "arn:aws:glue:us-east-1:111122223333:connection/*"
            ]
        },
        {
            "Sid": "GlueActionBasedPermissions",
            "Effect": "Allow",
            "Action": [
                "glue:ListEntities",
                "glue:RefreshOAuth2Tokens"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "CloudWatchLogging",
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Policy di trust:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "glue.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Associa il ruolo alla connessione utilizzando la AWS Glue CLI o l'API:

```
aws glue create-integration-resource-property \
--resource-arn arn:aws:glue:us-east-1:123456789012:connection/connectionName \
--source-processing-properties "{\"RoleArn\" :
 \"arn:aws:iam::123456789012:role/rolename\"}" \
--region us-east-1
```

## Configurazione del database Oracle AWS come fonte

Attualmente AWS Glue supporta Oracle Databases AWS come sorgente e Redshift come destinazione. I dettagli della configurazione sono disponibili in Configurazione di Zero-ETL per Oracle Database all'indirizzo. AWS

# Configurazione di un target per un'integrazione zero-ETL

Esistono diverse opzioni offerte AWS Glue durante la configurazione di un target per un'integrazione zero-ETL. L'obiettivo può essere un Amazon Redshift data warehouse crittografato o un'architettura Lakehouse di Amazon. SageMaker

Prima di selezionare la destinazione per l'integrazione Zero-ETL, è necessario configurare una delle seguenti risorse di destinazione. Le opzioni di configurazione per una destinazione in un'integrazione Zero-ETL includono:

- Un bucket Amazon S3 per uso generico che utilizza l'architettura Lakehouse di Amazon. SageMaker Per informazioni, consulta Configurazione di un target bucket S3 per uso generico.
- Un bucket Amazon S3 Tables che utilizza l'architettura Lakehouse di Amazon. SageMaker Per informazioni, consulta Configurazione di un target del bucket Amazon S3 Tables.
- Uno storage Amazon Redshift gestito che utilizza l'architettura Lakehouse di Amazon. SageMaker Per informazioni, consulta Configurazione di un target di storage gestito Amazon Redshift.
- Un Amazon Redshift data warehouse identificato da un namespace Redshift. Per informazioni, consulta Configurazione di un obiettivo di data warehouse Amazon Redshift.

> **Note**
>
> Non è possibile modificare la destinazione di un'integrazione Zero-ETL dopo la creazione.

## Configurazione di un target bucket S3 per uso generico

Questa sezione descrive i prerequisiti e i passaggi di configurazione per configurare un bucket S3 generico come storage per il tuo target in un'integrazione zero-ETL, utilizzando l'architettura Lakehouse di Amazon. SageMaker

Prima di creare un'integrazione zero-ETL con l'architettura Lakehouse di SageMaker Amazon utilizzando lo storage S3 generico, devi completare le seguenti attività di configurazione:

- Configura un AWS Glue database
- Fornire la policy RBAC del catalogo
- Creare un ruolo IAM
- Associa il ruolo di destinazione, KMS (opzionale) e la connessione (opzionale) alla risorsa di destinazione
- (Facoltativo) Configura le proprietà della tabella di destinazione

### Configurazione di un AWS Glue database

Per configurare un database di destinazione nel Data Catalog con una posizione del bucket per uso generico Amazon S3:

1. Nella home page della AWS Glue console, seleziona Database in Data Catalog.

2. Scegliere Aggiungi database in alto a destra. Se è stato già creato un database, assicurarsi che sia impostata la posizione con l'URI di Amazon S3 per il database.
3. Inserire un nome e una Posizione (URI Amazon S3). Si prega di notare che la posizione è necessaria per l'integrazione Zero-ETL. Al termine, fare clic su Crea database.

> **Note**
>
> Il bucket Amazon S3 per uso generico deve trovarsi nella stessa regione del database. AWS Glue

Per informazioni sulla creazione di un nuovo database in AWS Glue, consulta Guida introduttiva al Data Catalog.

È possibile anche usare la CLI `create-database` per creare il database in AWS Glue. Tenere presente che `LocationUri` in `--database-input` è obbligatorio.

Ottimizzazione delle tabelle Iceberg

Una volta creata una tabella AWS Glue nel database di destinazione, puoi abilitare la compattazione per velocizzare le query in Amazon Athena. Per informazioni sulla configurazione delle risorse (Ruolo IAM) per la compattazione, consultare Prerequisiti per l'ottimizzazione delle tabelle.

Per ulteriori informazioni sulla configurazione della compattazione sulla AWS Glue tabella creata dall'integrazione, consulta Optimizing Iceberg tables.

## Fornire una policy RBAC (Resource Based Access) per il catalogo

Per le integrazioni che utilizzano un AWS Glue database, aggiungi le seguenti autorizzazioni alla politica RBAC del catalogo per consentire le integrazioni tra origine e destinazione.

> **Note**
>
> Per le integrazioni tra account, sia l'utente che crea la politica dei ruoli di integrazione che la politica delle risorse del catalogo devono consentire l'accesso alla risorsa. `glue:CreateInboundIntegration` Per lo stesso account è sufficiente una policy in materia di risorse o una policy di ruolo che consenta a `glue:CreateInboundIntegration` l'utilizzo della risorsa.

Entrambi gli scenari devono comunque consentire a `glue.amazonaws.com` per `glue:AuthorizeInboundIntegration`.

È possibile accedere alle Impostazioni del catalogo in Data Catalog. Fornire quindi le seguenti autorizzazioni e inserire le informazioni mancanti.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Principal": {
        "AWS": [
            "arn:aws:iam::123456789012:user/Alice"
        ]
      },
      "Effect": "Allow",
      "Action": [
        "glue:CreateInboundIntegration"
      ],
      "Resource": [
          "arn:aws:glue:us-east-1:111122223333:catalog",
          "arn:aws:glue:us-east-1:111122223333:database/database-name"
      ],
      "Condition": {
        "StringLike": {
        "aws:SourceArn": "arn:aws:dynamodb:us-east-1:444455556666:table/table-name"
        }
      }
    },
    {
      "Principal": {
        "Service": [
          "glue.amazonaws.com"
        ]
      },
      "Effect": "Allow",
      "Action": [
        "glue:AuthorizeInboundIntegration"
      ],
      "Resource": [
          "arn:aws:glue:us-east-1:111122223333:catalog",
          "arn:aws:glue:us-east-1:111122223333:database/database-name"
```

```
      ],
      "Condition": {
        "StringEquals": {
        "aws:SourceArn": "arn:aws:dynamodb:us-east-1:444455556666:table/table-name"
        }
      }
    }
  ]
}
```

## Creazione di un ruolo IAM di destinazione

Creare un ruolo IAM con le seguenti autorizzazioni e relazioni di fiducia:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": "s3:ListBucket",
            "Resource": "arn:aws:s3:::amzn-s3-bucket",
            "Effect": "Allow"
        },
        {
            "Action": [
                "s3:GetObject",
                "s3:PutObject",
                "s3:DeleteObject"
            ],
            "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/prefix/*",
            "Effect": "Allow"
        },
        {
            "Action": [
                "glue:GetDatabase"
            ],
            "Resource": [
                "arn:aws:glue:us-east-1:111122223333:catalog",
                "arn:aws:glue:us-east-1:111122223333:database/database-name"
            ],
            "Effect": "Allow"
        },
        {
            "Action": [
```

```
                "glue:CreateTable",
                "glue:GetTable",
                "glue:GetTables",
                "glue:DeleteTable",
                "glue:UpdateTable",
                "glue:GetTableVersion",
                "glue:GetTableVersions",
                "glue:GetResourcePolicy"
            ],
            "Resource": [
                "arn:aws:glue:us-east-1:111122223333:catalog",
                "arn:aws:glue:us-east-1:111122223333:database/database-name",
                "arn:aws:glue:us-east-1:111122223333:table/database-name/*"
            ],
            "Effect": "Allow"
        },
        {
            "Action": [
                "cloudwatch:PutMetricData"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "cloudwatch:namespace": "AWS/Glue/ZeroETL"
                }
            },
            "Effect": "Allow"
        },
        {
            "Action": [
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

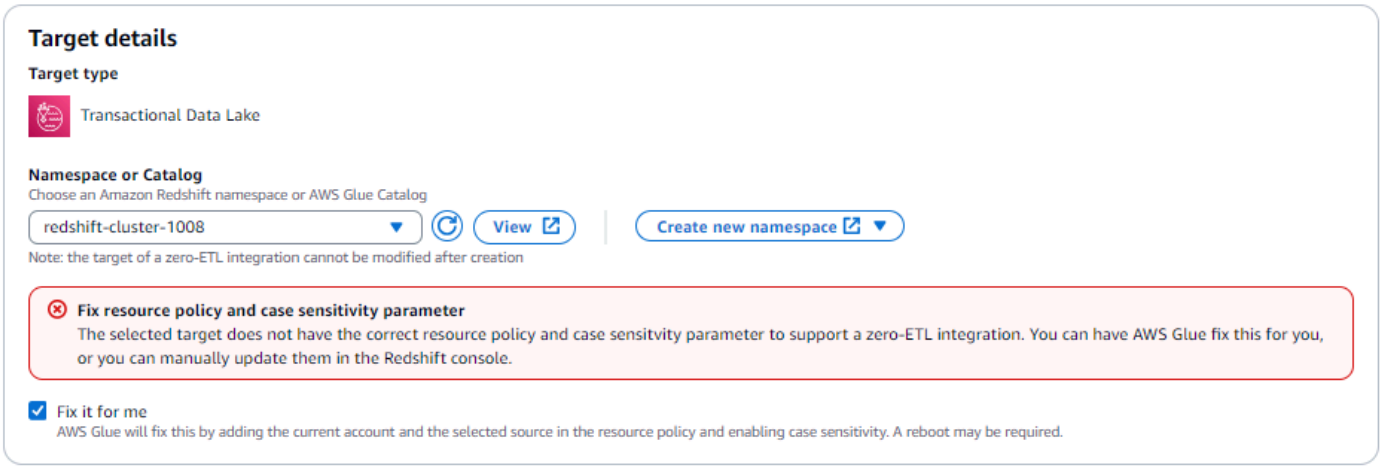
Aggiungi la seguente politica di attendibilità per consentire al AWS Glue servizio di assumere il ruolo:

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "glue.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

## Associa il ruolo di destinazione, KMS (opzionale) e Connection (opzionale) alla risorsa di destinazione

Associa il ruolo di destinazione sopra riportato alla risorsa di destinazione, ad esempio il AWS Glue database. Facoltativamente, per il database di destinazione è possibile configurare KMS per crittografare i dati prima dell'archiviazione nella tabella iceberg di destinazione e Connection ARN per accedere al bucket S3. AWS Glue Ciò consentirà di accedere AWS Glue ai dati sulla posizione S3 di destinazione utilizzando il ruolo fornito e, facoltativamente, di crittografarli utilizzando la chiave KMS fornita. Se il bucket S3 di destinazione è configurato per essere accessibile utilizzando un determinato VPC, è possibile associare l'ARN di connessione per consentire l'esecuzione dell'elaborazione AWS Glue all'interno di quel VPC. Per ulteriori informazioni sulla configurazione di un VPC, consulta Creare un VPC.




Oppure utilizzando la AWS Glue CLI/API:

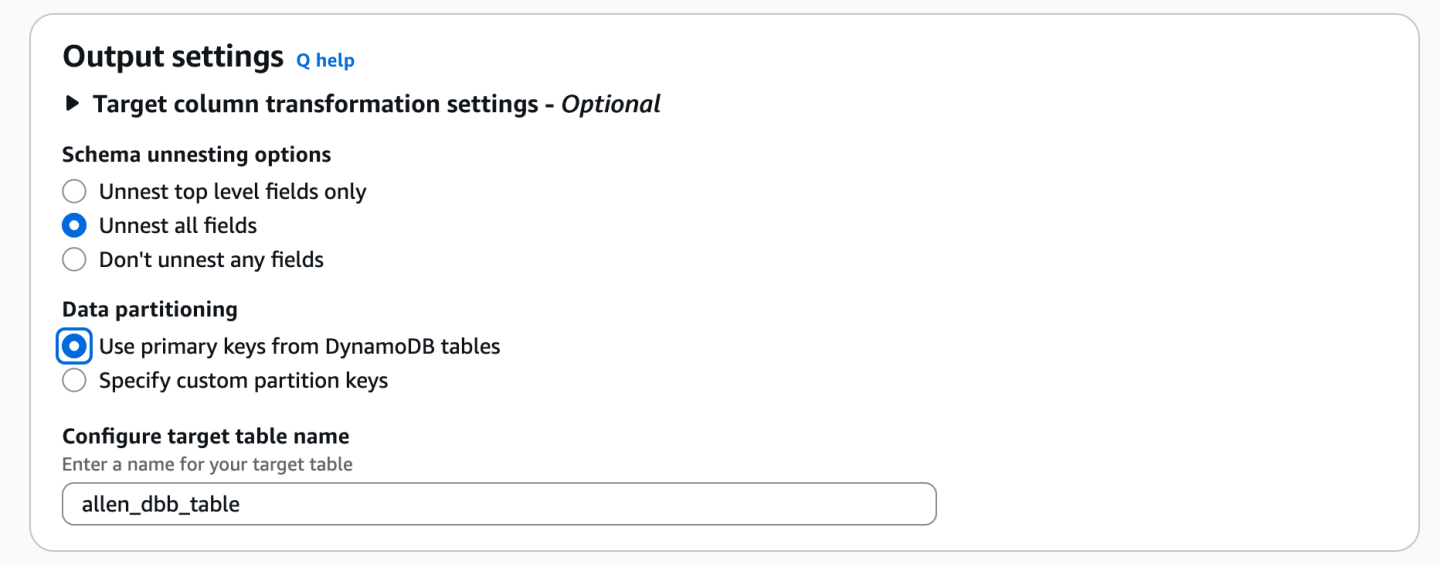
```
aws glue create-integration-resource-property \
--resource-arn arn:aws:glue:us-east-1:123456789012:database/database-name \
```

```
--target-processing-properties '{"RoleArn":
 "arn:aws:iam::123456789012:role/gmi_target_role"}' \
--region us-east-1
```

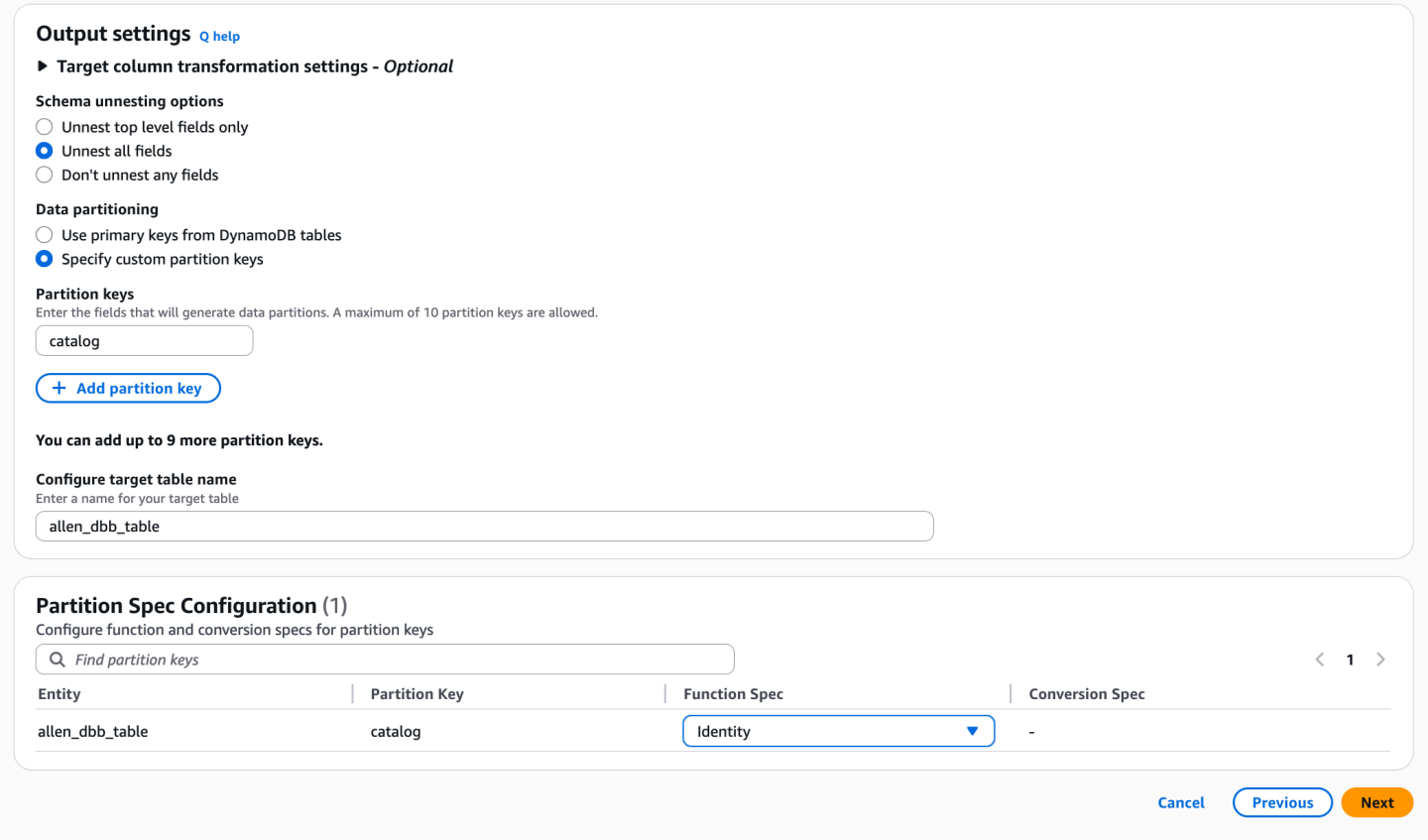
## (Facoltativo) Configura le proprietà della tabella di destinazione

Facoltativamente, è possibile configurare le proprietà della tabella di destinazione per le tabelle di destinazione che verranno sincronizzate con la destinazione.

Puoi configurare queste impostazioni nella sezione Impostazioni di output del flusso di lavoro per la creazione dell'integrazione nella AWS Glue console:




Quando si seleziona Specificare chiavi di partizione personalizzate, è possibile configurare le chiavi di partizione e le relative specifiche di funzione e conversione:




Se l'origine e la destinazione si trovano nello stesso account, questa configurazione può essere eseguita come parte del flusso di lavoro di creazione dell'integrazione dall'interfaccia utente della console. AWS Glue Ma se la destinazione si trova in un account diverso, è necessario completare questa configurazione prima di creare l'integrazione. Quando si utilizza la CLI o l'API, è necessario eseguire questa operazione prima di richiamare l'API Create-Integration anche quando sia l'origine che la destinazione si trovano nello stesso account. AWS Glue l'interfaccia utente della console incapsula semplicemente questa chiamata API per lo stesso scenario di account.

Se non è configurato, verranno utilizzati i valori predefiniti durante la sincronizzazione della tabella. Questa configurazione può anche essere modificata in qualsiasi momento dopo la creazione dell'integrazione.

> **Note**
>
> Se questa proprietà viene aggiornata dopo la creazione dell'integrazione, potrebbe attivare una risincronizzazione completa della tabella quando la configurazione aggiornata è in conflitto con la configurazione esistente. Ad esempio, aggiornando la tabella «un-nesting» da 'No-Unnest' a 'Full-Unnest' o cambiando la colonna della partizione.

Utilizzo della CLI o dell'API:

```
aws glue create-integration-table-properties \
--resource-arn arn:aws:glue:us-east-1:123456789012:database/database-name \
--table-name table-name \
--target-table-config '{
        "UnnestSpec":"TOPLEVEL"|"FULL"|"NOUNNEST",
        "PartitionSpec":
            [
                {
                    "FieldName":"string",
                    "FunctionSpec":"string",
                    "ConversionSpec":"string"}
                    ...
             ],
         "TargetTableName":"string"
     }' \
--region us-east-1
```

Dopo aver configurato l'architettura Lakehouse di Amazon SageMaker con lo storage bucket Amazon S3 per uso generico, puoi procedere con il completamento della configurazione dell'integrazione. Configurazione dell'integrazione con la destinazione

## Configurazione di un target del bucket Amazon S3 Tables

Questa sezione descrive i prerequisiti e i passaggi di configurazione per configurare Amazon S3 Tables come destinazione per l'integrazione zero-ETL, utilizzando l'architettura lakehouse di Amazon. SageMaker

Prima di creare un'integrazione Zero-ETL con Tabelle Amazon S3 come destinazione, è necessario completare le seguenti attività di configurazione:

- Configurazione del bucket di tabelle Amazon S3 (e integrazione dei servizi di analisi)
- Fornire la policy RBAC del catalogo
- Creare un ruolo IAM
- Associa il ruolo di destinazione, KMS (opzionale) e la connessione (opzionale) alla risorsa di destinazione
- (Facoltativo) Configura le proprietà della tabella di destinazione

## Configurazione del bucket di tabelle Amazon S3 (con integrazione dei servizi di analisi)

1. Creare un bucket Tabelle S3 nel proprio account seguendo le istruzioni in Nozioni di base su Tabelle Amazon S3.
2. Abilita le integrazioni di Analytics con il tuo bucket S3-Table seguendo queste istruzioni: Integrazione dei servizi con AWS Amazon S3 Tables.
3. Questo creerà un nuovo catalogo S3-Table in. AWS Lake Formation

## Fornire la policy RBAC del catalogo

È necessario aggiungere le seguenti autorizzazioni alla Policy RBAC del catalogo per consentire le integrazioni tra l'origine e la destinazione del catalogo Tabelle Amazon S3.

La politica delle risorse di Target AWS Glue Catalog deve includere le autorizzazioni AWS Glue di servizio per. `AuthorizeInboundIntegration` Inoltre, è richiesta l'`CreateInboundIntegration`autorizzazione sul principale di origine che crea l'integrazione o nella politica AWS Glue delle risorse di destinazione.

> **Note**
>
> In uno scenario con più account, sia la politica delle risorse principale di origine che quella delle risorse AWS Glue del catalogo di destinazione devono includere `glue:CreateInboundIntegration` le autorizzazioni sulla risorsa.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Principal": {
        "AWS": [
            "arn:aws:iam::123456789012:user/Alice"
        ]
      },
      "Effect": "Allow",
      "Action": [
        "glue:CreateInboundIntegration"
      ],
      "Resource": [
```

```
          "arn:aws:glue:us-east-1:111122223333:catalog/s3tablescatalog/*"
      ],
      "Condition": {
        "StringLike": {
        "aws:SourceArn": "arn:aws:dynamodb:us-east-1:444455556666:table/table-name"
        }
      }
    },
    {
      "Principal": {
        "Service": [
          "glue.amazonaws.com"
        ]
      },
      "Effect": "Allow",
      "Action": [
        "glue:AuthorizeInboundIntegration"
      ],
      "Resource": [
      "arn:aws:glue:us-east-1:111122223333:catalog/s3tablescatalog/*"
      ],
      "Condition": {
        "StringEquals": {
        "aws:SourceArn": "arn:aws:dynamodb:us-east-1:444455556666:table/table-name"
        }
      }
    }
  ]
}
```

> **Note**
>
> Sostituiscila *s3tablescatalog* con il nome del catalogo principale delle tue tabelle S3 (se diverse). Il valore predefinito (quando si ospita S3-Table Catalog nello stesso account) è. `s3tablescatalog`

## Creare un ruolo IAM

Creare un ruolo IAM con le seguenti autorizzazioni e relazioni di fiducia:

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3tables:ListTableBuckets",
        "s3tables:GetTableBucket",
        "s3tables:GetTableBucketEncryption",
        "s3tables:GetNamespace",
        "s3tables:CreateNamespace",
        "s3tables:ListNamespaces",
        "s3tables:CreateTable",
        "s3tables:DeleteTable",
        "s3tables:GetTable",
        "s3tables:GetTableEncryption",
        "s3tables:ListTables",
        "s3tables:GetTableMetadataLocation",
        "s3tables:UpdateTableMetadataLocation",
        "s3tables:GetTableData",
        "s3tables:PutTableData"
      ],
      "Resource": "arn:aws:s3tables:us-east-1:111122223333:bucket/s3-table-bucket",
      "Effect": "Allow"
    },
    {
      "Action": [
        "cloudwatch:PutMetricData"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "cloudwatch:namespace": "AWS/Glue/ZeroETL"
        }
      },
      "Effect": "Allow"
    },
    {
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "*",
      "Effect": "Allow"
    }
```

```
  ]
}
```

Aggiungi la seguente policy di fiducia nel ruolo IAM di destinazione per consentire a AWS Glue Service di assumerla:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "glue.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

> **Note**
>
> Assicurarsi che non sia presente un'istruzione DENY esplicita per questo ruolo IAM di destinazione nella policy delle risorse del bucket S3-Tables. Un DENY esplicito sovrascriverebbe qualsiasi autorizzazione ALLOW e impedirebbe il corretto funzionamento dell'integrazione.

## Associa il ruolo di destinazione, KMS (opzionale) e Connection (opzionale) alla risorsa di destinazione

Associa il ruolo target di cui sopra alla risorsa di destinazione. Facoltativamente, è possibile configurare KMS per crittografare i dati prima dell'archiviazione nella tabella iceberg di destinazione e Connection ARN per accedere al bucket S3 di destinazione. Se il bucket S3 di destinazione è configurato per essere accessibile utilizzando un determinato VPC, è possibile associare l'ARN di connessione per consentire l'esecuzione dell'elaborazione AWS Glue all'interno di quel VPC. Per ulteriori informazioni sulla configurazione di un VPC, consulta Creare un VPC.

Utilizzo della AWS Glue CLI/API:

```
aws glue create-integration-resource-property \
```

```
--resource-arn arn:aws:glue:us-east-1:123456789012:catalog/s3tablescatalog/S3 table
 bucket name \
--target-processing-properties '{
                    "RoleArn": "arn:aws:iam::123456789012:role/target_role"
                }' \
--region us-east-1
```

### (Facoltativo) Configura le proprietà della tabella di destinazione

Facoltativamente, è possibile configurare le proprietà della tabella di destinazione per le tabelle di destinazione che verranno sincronizzate con la destinazione. Si applicano le stesse regole descritte nella sezione sugli obiettivi S3 per uso generico.

Utilizzo della CLI o dell'API:

```
aws glue create-integration-table-properties \
--resource-arn arn:aws:glue:us-east-1:123456789012:catalog/s3tablescatalog/S3 table
 bucket name \
--table-name table-name \
--target-table-config '' \
--region us-east-1
```

Dopo aver configurato lo storage Amazon S3-Tables utilizzando l'architettura Lakehouse di SageMaker Amazon, puoi procedere Configurazione dell'integrazione con la destinazione al completamento della configurazione dell'integrazione.

## Configurazione di un target di storage gestito Amazon Redshift

Questa sezione descrive i prerequisiti e i passaggi di configurazione per configurare uno storage Amazon Redshift gestito (RMS) come destinazione per l'integrazione zero-ETL, utilizzando l'architettura lakehouse di Amazon. SageMaker

Prima di creare un'integrazione zero-ETL con un'architettura Lakehouse di Amazon SageMaker utilizzando lo storage gestito Redshift, devi completare le seguenti attività di configurazione:

- Configura un cluster o un gruppo di lavoro Serverless Amazon Redshift
- Registra l' Amazon Redshift integrazione con Lake Formation
- Creazione di un catalogo gestito in Lake Formation
- Configurazione delle autorizzazioni IAM

### Configurazione dello storage Amazon Redshift gestito

Per configurare lo storage Amazon Redshift gestito per l'integrazione Zero-ETL:

- Crea o utilizza un Amazon Redshift cluster o un gruppo di lavoro Serverless esistente. Assicurati che il `enable_case_sensitive_identifier` parametro sia attivato per il Amazon Redshift gruppo di lavoro o il cluster di destinazione affinché l'integrazione abbia successo. Per ulteriori informazioni sull'attivazione della distinzione tra maiuscole e minuscole, consulta Attivare la distinzione tra maiuscole e minuscole per il data warehouse nella guida alla Amazon Redshift gestione.
- Registrare un'integrazione da Redshift nel catalogo in AWS Lake Formation. Vedi Registrazione di Amazon Redshift cluster e namespace nel Data Catalog.
- Crea un catalogo federato o gestito in. AWS Lake Formation Per ulteriori informazioni, consulta:
  - Inserimento Amazon Redshift dei dati nel Data Catalog
  - Creazione di un catalogo Amazon Redshift gestito nel Data Catalog
- Configurare le autorizzazioni IAM per il ruolo di destinazione. Il ruolo richiede le autorizzazioni per accedere alle risorse di Redshift e Lake Formation. Il ruolo dovrebbe avere almeno:
  - Autorizzazioni di accesso al cluster o gruppo di lavoro Redshift
  - Autorizzazioni per accedere al catalogo Lake Formation
  - Autorizzazioni per creare e gestire tabelle nel catalogo
  - CloudWatch e CloudWatch registra le autorizzazioni per il monitoraggio

Dopo aver configurato il catalogo Amazon SageMaker Lakehouse con lo storage Amazon Redshift gestito, puoi procedere con il Configurazione dell'integrazione con la destinazione completamento della configurazione dell'integrazione.

## Configurazione di un obiettivo di data warehouse Amazon Redshift

Questa sezione descrive i prerequisiti e i passaggi di configurazione per configurare un Amazon Redshift data warehouse come destinazione per l'integrazione zero-ETL.

Prima di creare un'integrazione zero-ETL con un obiettivo di Amazon Redshift data warehouse, è necessario completare le seguenti attività di configurazione:

- Configurare un Amazon Redshift cluster o un gruppo di lavoro Serverless
- Configurazione della distinzione tra maiuscole e minuscole

- Configurazione delle autorizzazioni IAM

## Configurazione del Amazon Redshift data warehouse

Per configurare un Amazon Redshift data warehouse per l'integrazione Zero-ETL:

1. Andare alla console Amazon Redshift e fare clic su Crea cluster o utilizzare un cluster esistente. Per creare un Amazon Redshift cluster, vedi Creazione di un cluster. Per un gruppo di lavoro Amazon Redshift serverless, fare clic su Crea gruppo di lavoro. Per creare un gruppo di lavoro Amazon Redshift serverless, consulta Creazione di un gruppo di lavoro con uno spazio dei nomi.
2. Se crei un nuovo cluster, scegli una dimensione del cluster appropriata e assicurarsi che il cluster sia crittografato. Per Serverless, configurare le impostazioni del gruppo di lavoro in base alle proprie esigenze.
3. Assicurati che il `enable_case_sensitive_identifier` parametro sia attivato per il Amazon Redshift gruppo di lavoro o il cluster di destinazione affinché l'integrazione abbia successo. Per ulteriori informazioni sull'attivazione della distinzione tra maiuscole e minuscole, consulta Turn on case sensitivity for your data warehouse nella Guida alla gestione di Amazon Redshift.
4. Configura le autorizzazioni IAM per consentire all'integrazione Zero-ETL di accedere al tuo data warehouse. Amazon Redshift Sarà necessario creare un ruolo IAM con le seguenti autorizzazioni:
   - Autorizzazioni per accedere al cluster o al gruppo di lavoro Amazon Redshift
   - Autorizzazioni per creare e gestire database e tabelle in Amazon Redshift
   - CloudWatch e CloudWatch registra le autorizzazioni per il monitoraggio
5. Una volta completata la configurazione del Amazon Redshift gruppo di lavoro o del cluster, è necessario configurare il data warehouse per le integrazioni zero-ETL. Per ulteriori informazioni consultare Nozioni base sulle integrazioni Zero-ETL nella Guida alla gestione di Amazon Redshift.

> **Note**
>
> Quando si utilizza un Amazon Redshift data warehouse come destinazione, l'integrazione crea uno schema nel database specificato per archiviare i dati replicati. Il nome dello schema deriva dal nome dell'integrazione.

> **Note**
>
> Il Amazon Redshift gruppo di lavoro o il cluster di destinazione deve avere il `enable_case_sensitive_identifier` parametro attivato affinché l'integrazione abbia successo.

Dopo aver configurato il Amazon Redshift data warehouse, puoi procedere con il Configurazione dell'integrazione con la destinazione completamento della configurazione dell'integrazione.

## Configurazione dell'integrazione con la destinazione

Dopo aver configurato le risorse di origine e di destinazione, segui questi passaggi per completare la configurazione dell'integrazione:

1. Vai alla pagina «Integrazioni zero-ETL» e avvia il flusso di lavoro per la creazione dell'integrazione.
2. Seleziona la risorsa di origine configurata nei passaggi precedenti.
3. Seleziona o specifica la risorsa di destinazione (stesso account o account multiplo) configurata nei passaggi precedenti.
4. Seleziona il ruolo IAM di destinazione configurato in precedenza.
5. Seleziona l'opzione Fix it for me (disponibile solo quando il target si trova nello stesso account).
   - Per il normale target Amazon S3 (AWS Glue Database) e S3-Table (Catalog), questo sarà:
     - Applica un servizio principale autorizzato alla politica delle risorse del catalogo di destinazione.
     - Applica un ARN principale di AWS Glue origine autorizzato alla politica delle risorse del catalogo di destinazione.
   - Per l' Amazon Redshift obiettivo, ciò consentirà di:
     - Applicare un servizio principale autorizzato sul Amazon Redshift cluster o sul gruppo di lavoro Serverless.
     - Applica un ARN AWS Glue di origine autorizzato al Amazon Redshift cluster o al gruppo di lavoro Serverless.
     - Associare un nuovo set di parametri con `enable_case_sensitive_identifier = true`.

Utilizza quanto segue per creare l'integrazione tramite API o CLI: CreateIntegration API.

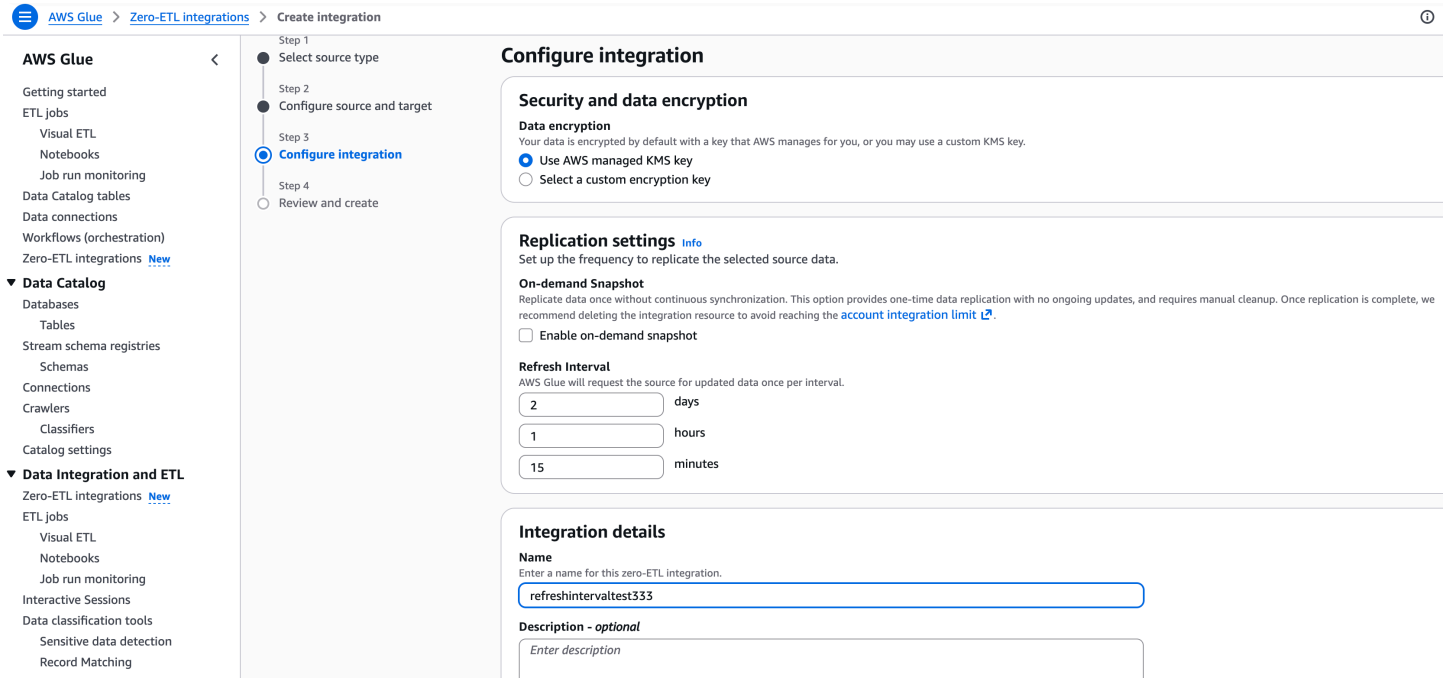
# Configurazione di un'integrazione

Quando si configura un'integrazione zero-ETL, è possibile configurare vari parametri per controllare la modalità di sincronizzazione dei dati tra i sistemi di origine e di destinazione. Le seguenti impostazioni sono attualmente disponibili solo per le sorgenti SaaS.

## Configurazione dell'intervallo di aggiornamento

È possibile configurare l'intervallo di aggiornamento per l'integrazione per le sorgenti SaaS al momento della creazione dell'integrazione. Il valore predefinito è 1 ora. È possibile configurare la frequenza con cui devono verificarsi le sollecitazioni o i carichi incrementali del CDC (Change Data Capture). Ciò offre la flessibilità necessaria per allineare la frequenza di aggiornamento ai modelli di aggiornamento dei dati specifici, alle considerazioni sul carico del sistema e agli obiettivi di ottimizzazione delle prestazioni. L'incremento di tempo può essere impostato su un valore da 15 minuti a 8640 minuti (sei giorni). L'intervallo di aggiornamento non può essere modificato dopo la creazione dell'integrazione quando la destinazione è Redshift. Per altri target, l'intervallo di aggiornamento può essere modificato dopo la creazione dell'integrazione. Per le sorgenti DynamoDB con intervalli di aggiornamento di 24 ore o più, Batch giornalieri sequenziali per sorgenti DynamoDB vedi per dettagli sull'elaborazione sequenziale giornaliera in batch.

Questa operazione può essere eseguita tramite console, aggiornando l'intervallo di aggiornamento all'interno delle impostazioni di replica.



L'incremento di tempo può essere impostato da 15 minuti a 8640 minuti (sei giorni), in modo da bilanciare l'aggiornamento dei dati e l'utilizzo delle risorse di sistema. Attualmente, l'intervallo di aggiornamento è personalizzabile sia per le fonti DynamoDB che per quelle SaaS:

- Intervallo minimo: 15 minuti
- Intervallo massimo: 8640 minuti (6 giorni)
- Valore predefinito: 15 minuti per il sorgente DynamoDB e 60 minuti per l'origine SaaS

Fattori da considerare nella scelta di un intervallo di aggiornamento:

- Volatilità dei dati: con quale frequenza vengono modificati i dati di origine
- Requisiti aziendali: quanto devono essere aggiornati i dati di analisi
- Considerazioni sui costi: aggiornamenti più frequenti possono comportare costi di elaborazione e archiviazione più elevati

> **Note**
>
> RefreshInterval il parametro definisce la frequenza di attivazione del CDC. La frequenza di aggiornamento effettiva può essere influenzata dal volume delle modifiche nei dati di origine e dalla capacità di elaborazione del sistema di destinazione. Monitora le prestazioni di integrazione e regola l'intervallo di aggiornamento secondo necessità per ottimizzarlo in base al caso d'uso specifico.

Oppure tramite l'API passando l'`RefreshInterval`interno IntegrationConfigcome parte di CreateIntegration Request. Per modificare l'intervallo di aggiornamento a livello di codice, puoi utilizzare l'ModifyIntegration API con il parametro. IntegrationConfig

## Batch giornalieri sequenziali per sorgenti DynamoDB

Per le integrazioni zero-ETL con una fonte Amazon DynamoDB, quando si configura un intervallo di aggiornamento di 1440 minuti (24 ore) o superiore, l'integrazione utilizza l'elaborazione sequenziale giornaliera in batch anziché una singola operazione di esportazione. Questo comportamento è dovuto alla limitazione della finestra di esportazione di DynamoDB, che ha un periodo massimo di esportazione di 24 ore.

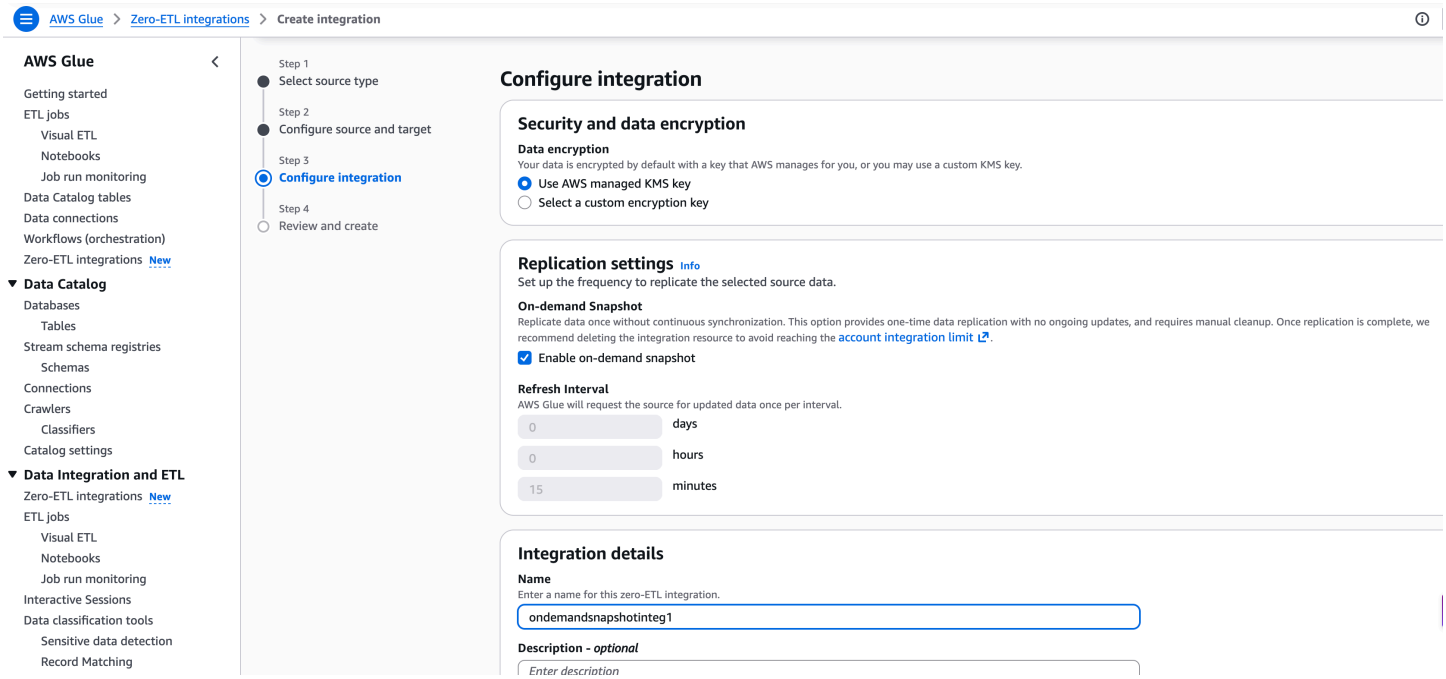
Quando l'intervallo di aggiornamento supera le 24 ore, l'integrazione funziona come segue:

1. Il processo CDC attende la durata completa dell'intervallo di aggiornamento (ad esempio, 6 giorni per un intervallo di 8640 minuti).
2. Trascorso l'intervallo di aggiornamento, l'integrazione esegue più esportazioni sequenziali di DynamoDB, ciascuna delle quali copre fino a una finestra di 24 ore.
3. I job CDC elaborano ogni batch in sequenza per acquisire tutte le modifiche avvenute durante il periodo di intervallo di aggiornamento.

Ad esempio, se imposti un intervallo di aggiornamento di 8640 minuti (6 giorni), l'integrazione aspetterà 6 giorni e quindi eseguirà 6 o 7 esportazioni sequenziali (1 esportazione finale che copre il tempo aggiuntivo dedicato alle operazioni di esportazione) e job CDC per sincronizzare tutte le modifiche di quel periodo.

## Istantanea su richiesta

Zero-ETL per impostazione predefinita include l'acquisizione continua dei dati (CDC), ma se hai casi d'uso per replicare tutti i dati una volta, puoi farlo utilizzando la funzione On-demand Snapshot. La funzionalità attualmente supportata solo per le fonti SaaS può essere utilizzata per replicare i dati una sola volta senza sincronizzazione continua. Questa opzione offre la replica dei dati una tantum senza aggiornamenti continui e richiede la pulizia manuale. Una volta completata la replica, consigliamo di eliminare la risorsa di integrazione per evitare di raggiungere il limite di integrazione dell'account.



Oppure tramite API impostando il `ContinuousSync` parametro su `false` within IntegrationConfigcome parte di Request. CreateIntegration

> **Note**
>
> L'impostazione On-demand Snapshot non può essere modificata dopo la creazione dell'integrazione. Scegliete attentamente questa opzione in base ai requisiti di sincronizzazione dei dati.

## Modifica dell'intervallo di aggiornamento

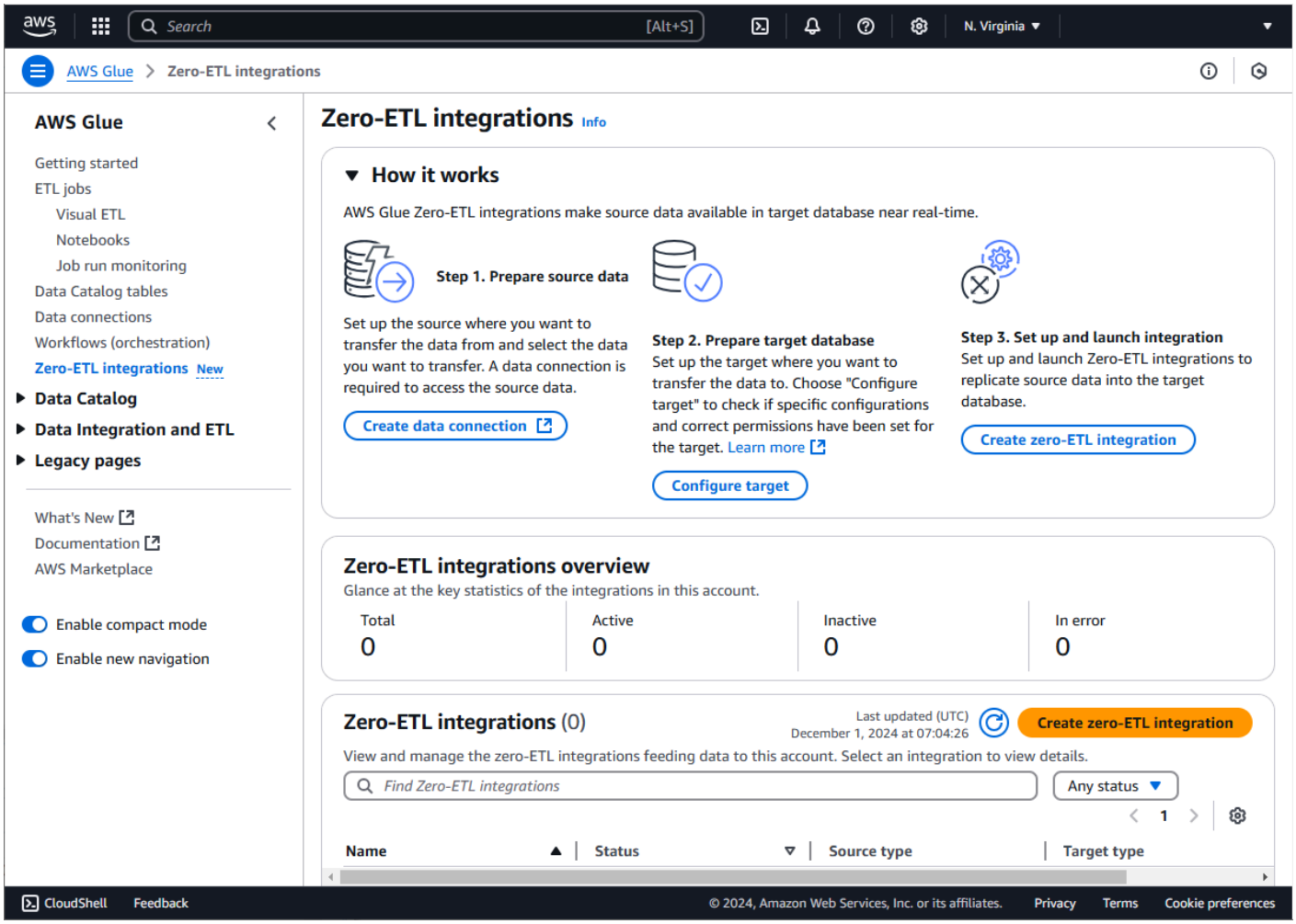
Questa funzionalità è attualmente disponibile solo per le AWS Glue destinazioni e consente di aggiornare l'intervallo di aggiornamento per un'integrazione esistente.

# Creazione e gestione di integrazioni

## Creazione di un'integrazione

Questa sezione descrive i passaggi generali per creare un'integrazione. Questo esempio utilizza Amazon DynamoDB come origine.

1. Nella home page della AWS Glue console, seleziona Integrazioni zero-ETL.
2. Puoi visualizzare tutte le tue integrazioni nella home page dell'integrazione Zero ETL. Per creare una nuova integrazione selezionare Crea Integrazione Zero-ETL.




3. Verrà richiesto di selezionare un Tipo di origine. Selezionare l'origine e fare clic su Avanti. Fai riferimento alle sezioni sulla configurazione del codice sorgente per le fonti di integrazione SaaS.
4. Nella pagina Configura origine e destinazione selezionare le tabelle o le entità da replicare. Per Amazon DynamoDB, assicurarsi che la policy PITR e RBAC sia configurata.
5. Specificare la propria integrazione di destinazione:
   - Per una destinazione AWS Glue Data Catalog, seleziona il AWS Glue database in cui vuoi replicare i dati.
   - Per una destinazione di Amazon Redshift data warehouse, seleziona lo spazio dei nomi del cluster Redshift o lo spazio dei nomi del gruppo di lavoro Redshift Serverless.

   Per ulteriori informazioni, consulta Configurazione dell'integrazione con la destinazione.
6. Fornire il Ruolo IAM di destinazione che è stato creato nei prerequisiti.

7. Se si desidera configurare una Chiave KMS di destinazione opzionale per l'archiviazione dei dati nella destinazione, fornire una chiave KMS abilitata. Allo stesso modo, se desideri configurare una connessione di rete di destinazione, seleziona una connessione. AWS Glue
8. Il pulsante Ripara Destinazione configura alcuni passaggi nella sezione Prerequisiti di questa documentazione. In particolare, 1) fornirà una policy RBAC del catalogo e 2) se non viene fornito alcun URI Amazon S3, ne genererà uno per l'utente, altrimenti utilizzerà l'URI fornito.
9. Nella sezione Impostazioni di output della pagina Configura origine e destinazione selezionare l'opzione di annidamento dello schema che si desidera per i propri dati nella destinazione. Se si desidera utilizzare le chiavi di partizione del cliente per i propri dati, selezionare Specifica chiavi di partizione personalizzate e fornire fino a 10 chiavi. Altrimenti, è possibile utilizzare semplicemente le chiavi di partizione assegnate alla tabella DynamoDB da replicare.
10. Nella sezione Sicurezza e crittografia dei dati è possibile fornire una chiave KMS che verrà utilizzata nel processo intermedio di replica dei dati sulla destinazione. In caso contrario, verrà utilizzata una chiave KMS AWS gestita. Inserire un nome per l'integrazione Zero-ETL nei Dettagli dell'integrazione.
11. Controllare e assicurarsi che tutti i dettagli forniti siano corretti. Fare clic su Crea e avvia l'integrazione una volta che tutto è stato confermato.
12. Nella home page di Zero ETL è possibile selezionare l'integrazione che è stata creata e verranno visualizzati i dettagli delle proprie integrazioni. Lo “Stato” indica lo stato dell'integrazione.

## Modifica di un'integrazione

È possibile modificare un'integrazione esistente.

1. Selezionare Modifica nell'angolo in alto a destra della pagina dei dettagli dell'integrazione.
2. Nella pagina Modifica origine e destinazione è possibile modificare il ruolo IAM di destinazione e la connessione di rete di destinazione. Gli altri campi non sono modificabili dopo la creazione dell'integrazione. Fare clic su Avanti.
3. È possibile anche modificare il nome e la descrizione dell'integrazione nella pagina Modifica integrazione e configurazione. Fare clic su Avanti.
4. Controllare le modifiche e, una volta confermate, fare clic su Aggiorna integrazione.

## Eliminazione di un'integrazione

L'eliminazione è uno stato terminale per un'integrazione. Una volta eliminata, l'integrazione non può essere ripristinata. L'eliminazione di un'integrazione cancella tutti i metadati interni e tutti i dati intermedi memorizzati.

Durante questo processo, tutte le attività in esecuzione che consistono nella scrittura di dati su una tabella di destinazione vengono interrotte. AWS Glue non eliminerà o pulirà il AWS Glue database di destinazione (nel Data Catalog) e i dati associati nel bucket Amazon S3 del tuo account. È necessario ripulirli esplicitamente, se necessario.

Per eliminare un'integrazione:

1. Nella pagina dei dettagli dell'integrazione fare clic su Elimina.
2. Inserire “Elimina” e fare clic su Elimina. Nota: si tratta di un'operazione irreversibile.
3. Nella pagina dei dettagli dell'integrazione lo stato è “Eliminazione in corso”. Una volta eliminata, l'integrazione non verrà più visualizzata nella home page dell'integrazione Zero ETL.

## Stati di un'integrazione

L'integrazione passa attraverso vari stati, dalla creazione all'eliminazione:

- `CREATING`- Questo è il primo stato in cui viene avviata la creazione dell'integrazione. In questo stato, AWS Glue esegue le inizializzazioni. Questo stato dovrebbe passare rapidamente allo stato CREATED a meno che non manchino alcune configurazioni.
- `ACTIVE`- Una volta che l'integrazione raggiunge questo stato, AWS Glue inizierà il trasferimento dei dati (caricamento iniziale a pieno carico). A meno che non vi siano problemi di autorizzazione, una volta completato il caricamento completo iniziale, seguirà l'acquisizione periodica dei dati relativi alle modifiche.
- `MODIFYING`- Una volta apportata la modifica all'integrazione, l'integrazione passa allo stato di modifica. Una volta applicata la modifica, l'integrazione passa a verificare `ACTIVE` se l'integrazione ha avuto successo dopo la modifica o se verrà completata `NEEDS_ATTENTION` o `FAILED` se ci sono stati problemi.
- `NEEDS_ATTENTION`- L'integrazione passerà a questo stato in caso di errore dell'utente o del sistema. L'errore dell'utente include autorizzazioni mancanti, risorse di origine o destinazione mancanti, errori nei dati non supportati. L'errore di sistema include errori interni del sistema. Per

entrambi i tipi di errore, AWS Glue Zero ETL continuerà a riprovare la sincronizzazione dei dati per 7 giorni prima di contrassegnare l'integrazione come FALLITA. Se risolvi il problema prima, l'integrazione tornerà ATTIVA e inizierà a trasferire i dati.

- `SYNCING`- L'integrazione passerà a questo stato se AWS Glue Zero ETL rileva eventuali modifiche al tipo di dati relative allo schema in entrata per le colonne all'interno di tabella/tabelle. In questi casi AWS Glue Zero ETL richiederà un nuovo set di istantanee per tutte queste tabelle. Durante questo periodo l'integrazione sarà in stato di SINCRONIZZAZIONE e alla fine passerà allo stato ATTIVO una volta che le nuove istantanee richieste saranno disponibili per l'ingestione.
- `FAILED`- Si tratta di uno stato non recuperabile. Una volta che l'integrazione passa a questo stato, non può essere ripristinata. L'unico modo per riavviare il trasferimento dei dati dall'origine alla destinazione è eliminare e ricreare l'integrazione. Se AWS Glue Zero ETL rileva che l'errore dell'utente o l'errore di sistema non è stato corretto per un periodo di 7 giorni e tutti i tentativi sono esauriti, AWS Glue Zero ETL contrassegnerà l'integrazione come FALLITA.
- `DELETING`- Quando richiamate l'API delete-integration, per prima cosa sposta l'integrazione nello stato DELETING AWS Glue . Dopo che tutti i metadati sono stati cancellati e le elaborazioni interne sono terminate, AWS Glue l'integrazione passerà allo stato DELETED.
- `DELETED`- Questo è lo stato del terminale per l'integrazione. L'integrazione non può essere spostata da questo stato a nessun altro stato. Se è necessario il trasferimento dei dati dalla stessa origine alla destinazione, è necessario creare nuovamente l'integrazione.

# Monitoraggio di un'integrazione

## Visualizzazione dei CloudWatch log di Amazon per un'integrazione

AWS Glue Le integrazioni zero-ETL generano CloudWatch log per la visibilità del movimento dei dati. Gli eventi di registro relativi a ciascun inserimento riuscito o a eventuali errori riscontrati a causa di record di dati problematici all'origine o errori di scrittura dei dati dovuti a modifiche dello schema o autorizzazioni insufficienti vengono emessi a un gruppo di log predefinito creato in un account cliente.

Per ogni integrazione creata, gli eventi di registro relativi a tale integrazione verranno raccolti in. `/aws-glue/zeroETL-integrations/logs/` CloudWatch All'interno del gruppo di log, i messaggi di log verranno suddivisi in flussi di log. Ciascuna integrazione creata ha un flusso di log dedicato in cui vengono scritti tutti i log relativi a tale integrazione.

> **Note**
>
> In uno scenario con più account, i log di elaborazione di origine vengono emessi nell'account di origine in cui esiste l'integrazione e i log di elaborazione di destinazione vengono emessi nell'account di destinazione in cui esiste il database di destinazione.

## Autorizzazioni IAM necessarie per abilitare la registrazione

Quando si crea l'integrazione, i ruoli di origine e destinazione hanno bisogno delle seguenti autorizzazioni IAM per abilitare la registrazione per un'integrazione. CloudWatch AWS Glue Le integrazioni zero-ETL utilizzano queste autorizzazioni fornite nei ruoli di origine e destinazione per inviare log agli account dei clienti. CloudWatch

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

## Messaggi di log

Formato di log: le integrazioni Zero-ETL emettono quattro tipi di messaggi di log:

```
// Ingestion started
{
```

```
"integrationArn": "arn:aws:glue:us-east-2:123456789012:integration/1a012bba-123a-1bba-
ab1c-173de3b12345",
...
    "messageType": "IngestionStarted",
    "details": {
        "tableName": "testDDBTable",
        "message": "Ingestion Job started"
    }
}
// Data processing stats on successful table ingestion
{
...
    "messageType": "IngestionProcessingStats",
    "details": {
        "tableName": "testDDBTable",
        "insert_count": 100,
        "update_count": 10,
        "delete_count": 10
    }
}
// Ingestion failure logs for failed table-processing
{
...
    "messageType": "IngestionFailed",
    "details": {
        "tableName": "testDDBTable",
        "errorMessage": "Failed to ingest data with error: Target Glue database not
 found.",
        "error_code" : "client_error"
    }
}
// Ingestion completed notification with lastSyncedTimestamp
{
...
    "messageType": "IngestionCompleted",
    "details": {
        "tableName": "testDDBTable",
        "message": "Ingestion Job completed"
        "lastSyncedTimestamp": "1132344255745"
    }
}
```

## Visualizzazione dei CloudWatch parametri di Amazon per un'integrazione

Una volta completata l'integrazione, puoi vedere queste CloudWatch metriche generate nel tuo account per ogni AWS Glue processo eseguito:

CloudWatch namespace delle metriche: "/glue/ZeroEtl»AWS

Dimensioni dei parametri:

- `integrationArn`
- `loadType`
- `tableName`

Nomi dei parametri:

- `InsertCount`: numero di record inseriti nella tabella Iceberg di destinazione.
- `UpdateCount`: numero di record aggiornati nella tabella Iceberg di destinazione.
- `DeleteCount`: numero di record eliminati dalla tabella Iceberg di destinazione.
- `IngestionSucceeded`: conta 1, se l'ingestione è riuscita per l'integrazione.
- `IngestionFailed`: conta 1, se l'ingestione non è riuscita per l'integrazione.
- `LastSyncTimestamp`: data e ora fino a quando la sorgente è stata sincronizzata con la destinazione.

## Gestione delle notifiche di eventi con Amazon EventBridge

Le integrazioni zero-ETL utilizzano EventBridge Amazon per gestire le notifiche degli eventi e up-to-date tenerti aggiornato sulle modifiche apportate alle integrazioni. Amazon EventBridge è un servizio di bus eventi senza server che puoi utilizzare per connettere le tue applicazioni con dati provenienti da una varietà di fonti. In questo caso, l'origine dell'evento è AWS Glue. Gli eventi, che sono modifiche monitorate in un ambiente, vengono inviati AWS Glue automaticamente EventBridge da Gli eventi vengono distribuiti pressoché in tempo reale.

Per acquisire tutte le notifiche zero-ETL, crea una EventBridge regola che corrisponda alla seguente:

```
{
```

```
  "source": [{
    "prefix": "aws.glue-zero-etl"
  }],
  "detail-type": [{
    "prefix": "Glue Zero ETL"
  }]
}
```

La tabella seguente include gli eventi di integrazione zero-ETL:

| Tipo di dettaglio | Spiegazione |
| --- | --- |
| AWS Glue Ingestione Zero ETL completata | L'esecuzione individuale per un'entità è stata completata correttamente. |
| AWS Glue Ingestione Zero ETL non riuscita | L'esecuzione individuale per un'entità non è stata completata correttamente (con un errore del client o del sistema). |
| AWS Glue Integrazione Zero ETL risincron izzata | L'integrazione è stata RISINCRONIZZATA. |
| AWS Glue Integrazione Zero ETL non riuscita | Lo stato dell'integrazione è stato modificato in Non riuscita a causa di un errore. |
| AWS Glue L'integrazione Zero ETL richiede attenzione | Lo stato dell'integrazione è stato modificato in NEEDS_ATTENTION a causa di un errore. |
| AWS Glue Zero ingestione di ETL in corso | L'esecuzione individuale per un'entità ha compiuto progressi parziali verso il completam ento. |

# Annidificazione dello schema e partizionamento dei dati

Quando si lavora con origini dati NoSQL come applicazioni DynamoDB e SaaS, i dati spesso presentano sfide uniche per l'analisi:

1. I record all'interno della stessa tabella possono avere uno schema diverso
2. I record annidati all'interno della stessa tabella possono essere rappresentati in modo diverso

3. Strutture nidificate complesse come mappe e array richiedono una trasformazione per un'esecuzione efficiente delle interrogazioni
4. È necessaria un'organizzazione ottimale dei dati per garantire prestazioni di query su larga scala

AWS Le integrazioni Glue Zero-ETL risolvono queste sfide attraverso due potenti funzionalità:

- Disnidificazione dello schema: appiattisce automaticamente complesse strutture di dati annidate in formati compatibili con l'analisi, con livelli configurabili di disnidificazione per bilanciare la conservazione della struttura dei dati e l'ottimizzazione della semplicità delle query.
- Partizionamento dei dati: organizza i dati in partizioni logiche basate su colonne o dimensioni temporali specifiche, migliorando le prestazioni delle query e riducendo i costi abilitando l'eliminazione delle partizioni durante l'esecuzione delle query.

Per interrogare efficacemente tali fonti di dati, AWS Glue Zero-ETL fornisce schemi di gestione out-of-the-box dello schema e di partizionamento per i dati di origine replicati nel database Glue di destinazione. AWS È possibile configurare le impostazioni di unnesting e partizionamento dello schema per ogni tabella tramite l' CreateIntegrationTableProperty API, consentendo un controllo preciso su come i dati sono strutturati e organizzati per i carichi di lavoro di analisi.

## Comportamento predefinito di disnidificazione e partizionamento

1. AWS L'impostazione predefinita di Glue zero-ETL è FULL Unnest quando non sono fornite opzioni Unnesting per la tabella di destinazione
2. AWS Glue Zero-ETL utilizza come impostazione predefinita il partizionamento Bucket quando non viene fornito alcun elemento per la tabella di destinazione PartitionSpec

## Disnidificazione dello schema

Quando ti integri con i servizi di analisi tramite Zero-ETL, puoi scegliere in che modo le strutture annidate vengono rappresentate nelle tabelle di destinazione. AWS Glue Zero-ETL offre opzioni di unnesting dello schema per appiattire strutture di dati complesse in formati più intuitivi per l'analisi.

### Opzioni di disnidificazione

Quando si crea un'integrazione Zero-ETL con un'origine, è possibile scegliere tra le seguenti opzioni di disnidificazione. Queste opzioni corrispondono a valori di enumerazione specifici che utilizzerai per chiamare l'API. CreateIntegrationTableProperty Per tutte le opzioni non necessarie, passeremmo

alla maggior parte del livello interno e mapperemmo il tipo DDB per indirizzarlo al tipo primitivo con il massimo sforzo. spark/iceberg La mappatura dei tipi tra il DDB di origine e la tabella di destinazione è la seguente:

| Tipo di dati di origine DDB | Tipo di dati della tabella di destinazione |
|---|---|
| "S" | StringType |
| "B" | BinaryType |
| "N" | DoubleType |
| "BOOL" | BooleanType |
| "SS" | ArrayType(StringType) |
| "NS" | ArrayType(DoubleType) |
| "BS" | ArrayType(BinaryType) |
| "L" | ArrayType(StringType) |
| "NULL" | Ignorare |
| "M" | StructType (TOP/SOSTANTIVO) |

Nessun annidamento - NO_UNNEST

Valore API: **`NO_UNNEST`**

Conserva la struttura nidificata originale degli elementi di Amazon DynamoDB. Le mappe e gli elenchi vengono memorizzati come colonne strutturate nella destinazione.

Ideale per: preservare l'esatta struttura dei dati di Amazon DynamoDB quando gli strumenti di analisi possono funzionare con dati annidati.

Livello superiore: TOP_LEVEL

Valore API: **`TOP_LEVEL`**

Appiattisce il livello superiore delle mappe nidificate in singole colonne. Le strutture degli elenchi rimangono annidate.

Conserva l'esatta struttura dei dati di Amazon DynamoDB quando gli strumenti di analisi possono funzionare con dati annidati con tutte le informazioni di tipo DDB rimosse.

Ideale per: bilanciare tra la conservazione della struttura dei dati e la semplicità delle query quando gli elementi della tabella Amazon DynamoDB hanno uno schema coerente.

Unnest tutti i livelli - FULL (impostazione predefinita)

Valore API: **`FULL`**

Appiattisce in modo ricorsivo tutte le strutture annidate (mappe ed elenchi) in singole colonne con notazione a punti per la denominazione.

Ideale per: massimizzare la semplicità delle query quando si lavora con strutture e strumenti di analisi profondamente annidati che preferiscono schemi piatti.

> **Note**
>
> La disnidificazione completa può portare a tabelle molto ampie con molte colonne se i dati DynamoDB hanno strutture variabili o profondamente annidate.

Example Utilizzo delle opzioni di disnidificazione nell'API

Quando configuri il unnesting dello schema tramite l' CreateIntegrationTableProperty API, specifica l'opzione di unnesting nel parametro: UnnestSpec

```
aws glue create-integration-table-property
  --resource-arn "arn:aws:glue:us-east-1:123456789012:database/my_db"
  --table-name "my-table"
  --cli-input-json '{
      "TargetTableConfig": {
          "UnnestSpec": "FULL",
          "TargetTableName": "my-target-table",
      }
  }'
```

## Esempi di disnidificazione

Si consideri un elemento di DynamoDB con la seguente struttura:

```
// Input DynamoDB Record
```

```
{
  "Item": {
    "col_1": {
      "S": "value_1"
    },
    "col_2": {
      "M": {
        "col_3": {
          "M": {
            "id": {
              "S": "value_3"
            }
          }
        },
        "col_4": {
          "BOOL": true
        }
      }
    }
  }
}
```

Esempio NO_UNNEST

Con NO_UNNEST, l'intera riga viene memorizzata all'interno di una colonna più la chiave primaria. Le informazioni sul tipo DynamoDB vengono conservate. Ciò mantiene la compatibilità con i pattern di interrogazione di Redshift.

Tabella Iceberg risultante (supponendo che col_1 sia la chiave primaria):

| col_1 (stringa) | valore (struttura) |
| --- | --- |
| valore_1 | <pre>{<br>  "col_2": {<br>    "M": {<br>      "col_3": {<br>        "M": {<br>          "id": {<br>            "S": "value_3"<br>          }<br>        }<br>      },</pre> |

| col_1 (stringa) | valore (struttura) |
| --- | --- |
| | <pre>"col_4": {<br>  "BOOL": true<br>}<br>}<br>}<br>}</pre> |

Le query dovrebbero utilizzare modelli di accesso a strutture e array:

```
SELECT
  value.col_1,
  value.col_2.M.col_3.M.id.S,
  value.col_2.M.col_4.BOOL
FROM product_table;
```

Esempio TOP_LEVEL

Con TOP_LEVEL, solo i campi di primo livello non vengono annidati, mantenendo intatti i campi nidificati come strutture. Le informazioni sul tipo DynamoDB vengono rimosse e la digitazione viene mantenuta. Converte in tipo stringa quando si verificano conflitti di schema.

Tabella Glue risultante dopo la replica:

| col_1 (stringa) | col_2 (struttura) |
| --- | --- |
| valore_1 | <pre>{<br>  "col_3": {<br>  "id": "value_3"<br>  },<br>  "col_4": true<br>}</pre> |

Le query sarebbero semplificate per il primo livello:

```
SELECT
  col_1,
  col_2.col_3.id,
```

```
  col_2.col_4
FROM product_table;
```

Esempio COMPLETO

Con l'unnesting FULL, sia i campi di primo livello che i campi nidificati vengono appiattiti. struct/map La notazione a punti viene utilizzata per i campi annidati (ad esempio, «col_2.col_3.id»). Gli elementi dell'array rimangono non annidati. Ogni nodo foglia diventa una colonna di primo livello.

Tabella Glue risultante dopo la replica:

| col_1 (stringa) | col_2.col_3.id (stringa) | col_2.col_4 (booleano) |
|---|---|---|
| valore_1 | valore_3 | TRUE |

Le query verrebbero completamente appiattite:

```
SELECT
  col_1,
  "col_2.col_3.id",
  "col_2.col_4"
FROM product_table;
```

# Partizionamento dei dati

## Cos'è il partizionamento?

Il partizionamento dei dati è una tecnica che divide set di dati di grandi dimensioni in segmenti più piccoli e più gestibili chiamati partizioni. Nel contesto delle integrazioni AWS Glue Zero-ETL, il partizionamento organizza i dati nella posizione di destinazione in base a valori di colonna specifici o trasformazioni di tali valori.

Vantaggi del partizionamento dei dati

Un partizionamento efficace dei dati offre diversi vantaggi chiave per i carichi di lavoro di analisi:

- Prestazioni di query migliorate: le query possono saltare le partizioni irrilevanti (potatura delle partizioni), riducendo la quantità di dati da scansionare.
- Costi ridotti: scansionando meno dati, puoi ridurre i costi di calcolo e i costi per le tue query di analisi. I/O

- Migliore scalabilità: il partizionamento consente l'elaborazione parallela dei segmenti di dati, consentendo una scalabilità più efficiente dei carichi di lavoro di analisi.
- Gestione semplificata del ciclo di vita dei dati: è possibile gestire le policy di conservazione a livello di partizione, semplificando l'archiviazione o l'eliminazione dei dati più vecchi.

### Concetti chiave del partizionamento

Colonne di partizione

Colonne dei dati utilizzate per determinare come i record sono organizzati in partizioni. Delle colonne di partizione efficaci devono essere allineate ai modelli di query più comuni e avere una cardinalità appropriata.

Funzioni di partizionamento

Trasformazioni applicate ai valori delle colonne di partizione per creare i limiti effettivi delle partizioni. Gli esempi includono l'identità (utilizzando il valore grezzo) e le funzioni basate sul tempo (anno, mese, giorno, ora).

Potatura delle partizioni

Il processo in cui il motore di query identifica e salta le partizioni che non contengono dati pertinenti per una query, migliorando in modo significativo le prestazioni.

Granularità delle partizioni

Il livello di dettaglio con cui i dati vengono partizionati. Una granularità più precisa (più partizioni) può migliorare le prestazioni delle query ma può aumentare il sovraccarico dei metadati. Una granularità più grossolana (meno partizioni) riduce il sovraccarico dei metadati, ma può comportare la scansione di più dati del necessario.

### Partizionamento nelle integrazioni AWS Glue Zero-ETL

AWS Le integrazioni Glue Zero-ETL utilizzano il formato di tabella Apache Iceberg, che fornisce funzionalità di partizionamento avanzate. Quando si crea un'integrazione Zero-ETL, è possibile:

- Utilizzare strategie di partizionamento predefinite ottimizzate per la propria origine dati
- Definire specifiche di partizionamento personalizzate in base ai propri modelli di query
- Applicare trasformazioni alle colonne delle partizioni (particolarmente utile per il partizionamento basato su timestamp)

- Combinare più strategie di partizione per un partizionamento a più livelli

Le configurazioni di partizionamento vengono specificate tramite l'API `CreateIntegrationTableProperty` durante la configurazione dell'integrazione Zero-ETL. Una volta configurato, AWS Glue applica automaticamente queste strategie di partizionamento per organizzare i dati nella posizione di destinazione.

## Riferimento all'API di specificazione delle partizioni

Utilizza i seguenti parametri nell' CreateIntegrationTableProperties API per configurare il partizionamento:

PartitionSpec

Un array di specifiche di partizione che definisce il modo in cui i dati vengono partizionati nella posizione di destinazione.

```
{
  "partitionSpec": [
    {
      "fieldName": "timestamp_col",
      "functionSpec": "month",
      "conversionSpec": "epoch_milli"
    },
    {
      "fieldName": "category",
      "functionSpec": "identity"
    }
  ]
}
```

FieldName

Una stringa UTF-8 (1-128 byte) che specifica il nome della colonna da utilizzare per il partizionamento.

FunctionSpec

Specifica la funzione di partizionamento. Valori validi:

- `identity` - Utilizza i valori di origine direttamente senza trasformazione
- `year`: estrae l'anno dai valori di timestamp (ad es., 2023)

- `month`: estrae il mese dai valori di timestamp (ad es., 2023-01)
- `day`: estrae il giorno dai valori di timestamp (ad es., 2023-01-15)
- `hour` - Estrae l'ora dai valori del timestamp (ad esempio, 2023-01-15-14)

> **Note**
>
> Le funzioni basate sul tempo (`year`, `month`, `day`, `hour`) richiedono il parametro `ConversionSpec` per specificare il formato del timestamp di origine.

ConversionSpec

Una stringa UTF-8 che specifica il formato del timestamp dei dati di origine. I valori validi sono:

- `epoch_sec`: timestamp epoch Unix in secondi
- `epoch_milli`: timestamp epoch Unix in millisecondi
- `iso` - Timestamp in formato ISO 8601

## Strategie di partizionamento

Partizionamento predefinito

Quando non viene specificata alcuna colonna di partizione, AWS Glue Zero-ETL applica strategie di partizionamento predefinite ottimizzate per la fonte di dati:

- Partizionamento basato sulla chiave primaria: per le orgini con chiavi primarie (come le tabelle DynamoDB), Glue Zero-ETL di AWS partiziona automaticamente i dati utilizzando la chiave primaria con bucketing per prevenire l'esplosione delle partizioni.

Il partizionamento predefinito è progettato per funzionare bene con i modelli di query più comuni senza richiedere una configurazione manuale. Tuttavia, per modelli di query o requisiti prestazionali specifici, è possibile definire strategie di partizionamento personalizzate.

Strategie di partizionamento definite dall'utente

AWS Glue zero-ETL consente di definire strategie di partizionamento personalizzate utilizzando il parametro. `PartitionSpec` È possibile specificare una o più colonne di partizione e applicare diverse funzioni di partizionamento a ciascuna colonna.

Il partizionamento delle identità utilizza i valori grezzi di una colonna per creare partizioni. Questa strategia è utile per le colonne con cardinalità da bassa a media, come i campi di categoria, regione o stato.

Example Esempio di partizionamento delle identità

```
{
  "partitionSpec": [
    {
      "fieldName": "category",
      "functionSpec": "identity"
    }
  ]
}
```

Questo crea partizioni separate per ogni valore univoco nella colonna "categoria".

> **Warning**
>
> Evitare di utilizzare il partizionamento delle identità con colonne ad alta cardinalità (come chiavi primarie o timestamp) poiché può portare a un'esplosione delle partizioni, che riduce le prestazioni e aumenta il sovraccarico dei metadati.

Il partizionamento basato sul tempo organizza i dati in base ai valori di timestamp con diverse granularità (anno, mese o giorno). Questa strategia è ideale per i dati di serie temporali e consente di eseguire query efficienti su intervalli di tempo.

Quando si utilizza il partizionamento basato sul tempo, AWS Glue Zero-ETL può convertire automaticamente vari formati di timestamp in un formato standardizzato prima di applicare la funzione di partizione. Questa conversione viene specificata utilizzando il parametro `ConversionSpec`.

Example Esempio di partizionamento basato sul tempo

```
{
  "partitionSpec": [
    {
      "fieldName": "created_at",
      "functionSpec": "month",
      "conversionSpec": "epoch_milli"
```

```
    }
  ]
}
```

Questo partiziona i dati per mese in base alla colonna "created_at", che contiene i timestamp epoch Unix in millisecondi.

AWS Glue Zero-ETL supporta le seguenti funzioni di partizione basate sul tempo:

- anno: partiziona i dati per anno (ad esempio: 2023, 2024)
- mese: partiziona i dati per mese (ad esempio: 2023-01, 2023-02)
- giorno: partiziona i dati per giorno (ad esempio: 2023-01-01, 2023-01-02)
- ora: partiziona i dati per ora (ad esempio: 2023-01-01-01, 2023-01-01-02)

AWS Glue zero-ETL supporta i seguenti formati di timestamp tramite il parametro: `ConversionSpec`

- epoch_sec: timestamp epoch Unix in secondi
- epoch_milli: timestamp epoch Unix in millisecondi
- iso: timestamp in formato ISO 8601

> **Note**
>
> I valori delle colonne originali rimangono invariati nei dati di origine. AWS Glue trasforma solo i valori delle colonne di partizione in Timestamp Type nella tabella del database di destinazione. Le trasformazioni si applicano solo al processo di partizionamento.

Il partizionamento a più livelli combina più strategie di partizione per creare uno schema di partizionamento gerarchico. Ciò è utile per ottimizzare diversi tipi di query sullo stesso set di dati.

Example Esempio di partizionamento a più livelli

```
{
  "partitionSpec": [
    {
      "fieldName": "created_at",
      "functionSpec": "month",
```

```
      "conversionSpec": "iso"
    },
    {
      "fieldName": "region",
      "functionSpec": "identity"
    }
  ]
}
```

Questo crea uno schema di partizionamento a due livelli: prima per mese (dalla colonna "created_at"), poi per regione. Ciò consente query efficienti che filtrano per intervalli di date, aree specifiche o una combinazione di queste dimensioni.

Quando si progettano schemi di partizionamento a più livelli, considerare:

- Posizionamento delle colonne con selettività più elevata al primo posto nella gerarchia delle partizioni
- Bilanciamento della granularità delle partizioni con il numero delle partizioni
- Allineamento dello schema di partizionamento con i modelli di query più comuni

## Best practice

Selezione delle colonne di partizione

- Non utilizzare colonne ad alta cardinalità con la funzione di partizione `identity`. L'utilizzo di colonne ad alta cardinalità con partizionamento delle identità crea molte partizioni di piccole dimensioni, che possono ridurre in modo significativo le prestazioni di inserimento. Le colonne ad alta cardinalità possono includere:
  - Chiavi primarie
  - Campi timestamp (come `LastModifiedTimestamp`, `CreatedDate`)
  - Timestamp generati dal sistema
- Non selezionare più partizioni di timestamp sulla stessa colonna. Esempio:

```
"partitionSpec": [
     {"fieldName": "col1", "functionSpec": "year", "conversionSpec" :
 "epoch_milli"},
     {"fieldName": "col1", "functionSpec": "month", "conversionSpec" :
 "epoch_milli"},
     {"fieldName": "col1", "functionSpec": "day", "conversionSpec" : "epoch_milli"},
```

```
      {"fieldName": "col1", "functionSpec": "hour", "conversionSpec" : "epoch_milli"}
]
```

FunctionSpec/ConversionSpec Selezione della partizione

- Specificate il formato corretto ConversionSpec (epoch_sec | epoch_milli | iso) che rappresenta il formato dei valori delle colonne scelti per il partizionamento basato sul timestamp quando si utilizzano le funzioni di partizione basate sul timestamp. AWS Glue Zero-ETL utilizza questo parametro per trasformare correttamente i dati di origine in formato timestamp prima del partizionamento.
- Utilizzate la granularità appropriata () in base al volume di dati. year/month/day/hour
- Considerate le implicazioni relative al fuso orario quando utilizzate i timestamp ISO. AWS Glue zero-ETL popola tutti i valori dei record della colonna timestamp scelta con il fuso orario UTC.

### Gestione degli errori

Stato NEEDS_ATTENTION

Un'integrazione entra nello stato NEEDS_ATTENTION quando:

- Le colonne di partizione specificate non esistono nell'origine
- La conversione del timestamp non riesce per le colonne di partizione

## Limitazioni

### Restrizioni di partizionamento

- Le specifiche delle partizioni non possono essere modificate dopo la creazione di un'integrazione. Per utilizzare una strategia di partizionamento diversa, è necessario creare una nuova integrazione.
- Il numero massimo di colonne di partizione è limitato a 10.

### Restrizioni sulle integrazioni tra account

- Quando si creano integrazioni tra account, AWS Glue Console presenta una limitazione in quanto non richiama l' CreateIntegrationTableProperty API per la configurazione e per UnnestSpec le

tabelle AWS Glue di PartitionSpec destinazione ospitate nell'account in cui l'integrazione non esiste.

Soluzione alternativa: l' CreateIntegrationTableProperty API deve essere richiamata da CX dall'account in cui esiste il database di destinazione.

### Limitazioni delle integrazioni multiple

- Se è necessario replicare la stessa fonte con unnest/partition configurazioni di schema diverse, è necessario creare un nuovo database AWS Glue per ogni integrazione separatamente. Successivamente richiama CreateIntegrationTableProperty per ogni tabella dal singolo database AWS Glue con le configurazioni di disnidificazione e partizionamento dello schema desiderate.

## Creazione di integrazioni utilizzando APIs

Puoi utilizzare quanto segue APIs per creare e gestire integrazioni zero-ETL in: AWS Glue

- CreateIntegration
- CreateIntegrationTableProperties
- CreateIntegrationResourceProperty
- UpdateIntegrationTableProperties
- UpdateIntegrationResourceProperty
- ModifyingIntegration
- DeleteIntegration
- DeleteIntegrationTableProperties
- DescribeIntegrations
- DescribeInboundIntegrations
- GetIntegrationTableProperties
- GetIntegrationResourceProperty

Per ulteriori informazioni, consulta Integrazione APIs in AWS Glue.

# Limitazioni

Di seguito sono riportate le limitazioni o le considerazioni generali sulle integrazioni Zero-ETL:

- Le proprietà delle risorse hanno una one-to-one relazione con la risorsa corrispondente. Di conseguenza, tutte le integrazioni create utilizzando tale risorsa devono rispettare la proprietà singolare della risorsa. La modifica della proprietà di una risorsa avrà quindi un impatto su tutte le integrazioni associate a tale risorsa.
- Le proprietà della tabella hanno una one-to-one relazione con la tabella o l'oggetto corrispondente all'interno di una risorsa. Di conseguenza, tutte le integrazioni che elaborano la stessa tabella da o verso la stessa risorsa devono rispettare la proprietà singolare della tabella.
- Non è possibile rinominare una colonna all'origine. Se una colonna viene rinominata, non vi è alcuna garanzia che il rilevamento dello schema venga eseguito con precisione AWS Glue e le ripercussioni sull'integrazione non sono definite.
- La seguente considerazione si applica al funzionamento dell'integrazione con le tabelle AWS Lake Formation gestite: per impostazione predefinita, si utilizza la AWS Glue politica IAM/ per gestire tabelle e database.
- Se desideri utilizzare AWS Lake Formation per gestire la creazione di tabelle in quel database, devi assicurarti che al ruolo siano concessi i permessi di Lake Formation sufficienti per creare, modificare ed eliminare la tabella e il database.
- Al momento, la pagina di riepilogo Zero-ETL non contiene alcuna metrica.

Di seguito sono riportate le limitazioni specifiche della sorgente delle integrazioni Zero-ETL:

- Le integrazioni zero-ETL con una OData fonte SAP ora supportano entità a partire da. `Entity0f` La possibilità di sovrascrivere la chiave primaria è attualmente supportata solo per gli oggetti. SAPOData `Entity0f` Una volta impostata, questa proprietà non può essere modificata.
- Le integrazioni zero-ETL da Amazon DynamoDB ad SageMaker Amazon Lakehouse (tramite S3) supportano una dimensione massima delle tabelle DynamoDB di 100 TB.
- La tabella DynamoDB di origine deve essere crittografata con una chiave KMS di proprietà di Amazon o gestita dal cliente. AWS La crittografia gestita da Amazon non è supportata per la tabella DynamoDB di origine.
- SAP OData funziona utilizzando un token delta, in cui la combinazione di un OAuth client più un'entità o l'autenticazione di base più un'entità può avere un solo token delta. Evitare di utilizzare la stessa entità in due diverse integrazioni con lo stesso client.

- L'utilizzo delle entità o dei campi Salesforce seguenti non è supportato in un'integrazione Zero-ETL con un'origine Salesforce. Per informazioni, consulta Entità e campi non supportati per Salesforce.
- L'uso ServiceNow delle seguenti entità o campi non è supportato in un'integrazione zero-ETL con un codice sorgente. ServiceNow Per informazioni, consulta Entità e campi non supportati per ServiceNow.

Di seguito sono riportate le limitazioni specifiche del target delle integrazioni zero-ETL:

- A causa delle limitazioni di Data Catalog, AWS Glue Zero ETL può supportare solo la replica di tabelle con nomi di tabella inferiori a 255 caratteri.
- A causa delle limitazioni di Athena, il nome del column/nested campo di origine non deve contenere caratteri speciali «:», il che causerà una visualizzazione errata dei metadati dello schema della tabella di destinazione. In questi casi, AWS Glue Zero ETL sposterà l'integrazione allo stato FAILED.
- A causa delle limitazioni di Iceberg, AWS Glue Zero ETL Target può supportare solo la replica di 1000 colonne per tabella. Se la tabella scelta in qualsiasi integrazione ha più di 1000 colonne, AWS Glue Zero ETL sposterà l'integrazione allo stato FAILED.
- A causa delle limitazioni di Data Catalog, AWS Glue Zero ETL può supportare solo la replica di tabelle con una dimensione dello schema di 10 MB. Se la tabella scelta in qualsiasi integrazione ha una dimensione dello schema superiore a 10 MB, AWS Glue Zero ETL sposterà l'integrazione allo stato FAILED.

## Service Quotas

AWS Glue Zero-ETL ha le seguenti quote di servizio:

- Per impostazione predefinita, ogni account può creare 25 integrazioni.
- Ogni fonte può essere associata a un massimo di 15 integrazioni.
- Ogni destinazione può essere associata a un massimo di 50 integrazioni.

## Regioni supportate

Le seguenti regioni sono disponibili con AWS Glue Zero-ETL:

| Region | AWS Glue Zero-ETL |
| --- | --- |
| Africa (Città del Capo) | Non disponibile |
| Asia Pacifico (Hong Kong) | Disponibilità |
| Asia Pacifico (Tokyo) | Disponibilità |
| Asia Pacifico (Seul) | Disponibilità |
| Asia Pacifico (Osaka) | Non disponibile |
| Asia Pacifico (Mumbai) | Non disponibile |
| Asia Pacifico (Hyderabad) | Non disponibile |
| Asia Pacifico (Singapore) | Disponibilità |
| Asia Pacifico (Sydney) | Disponibilità |
| Asia Pacifico (Giacarta) | Non disponibile |
| Asia Pacifico (Melbourne) | Non disponibile |
| Asia Pacifico (Malesia) | Non disponibile |
| Canada (Centrale) | Disponibilità |
| Canada occidentale (Calgary) | Non disponibile |
| Cina (Pechino) | Non disponibile |
| Cina (Ningxia) | Non disponibile |
| Europa (Francoforte) | Disponibilità |
| Europa (Zurigo) | Non disponibile |
| Europa (Stoccolma) | Disponibilità |
| Europa (Milano) | Non disponibile |

| Region | AWS Glue Zero-ETL |
| --- | --- |
| Europa (Spagna) | Non disponibile |
| Europa (Irlanda) | Disponibilità |
| Europa (Londra) | Disponibilità |
| Europa (Parigi) | Non disponibile |
| Israele (Tel Aviv) | Non disponibile |
| Medio Oriente (Emirati Arabi Uniti) | Non disponibile |
| Medio Oriente (Bahrein) | Non disponibile |
| Sud America (San Paolo) | Disponibilità |
| Stati Uniti orientali (Virginia settentrionale) | Disponibilità |
| Stati Uniti orientali (Ohio) | Disponibilità |
| Stati Uniti occidentali (California settentrionale) | Non disponibile |
| Stati Uniti occidentali (Oregon) | Disponibilità |
| AWS GovCloud (Stati Uniti orientali) | Non disponibile |
| AWS GovCloud (Stati Uniti occidentali) | Non disponibile |

# Risoluzione dei problemi delle integrazioni Zero-ETL

Utilizza le seguenti sezioni per risolvere i problemi relativi alle integrazioni AWS Glue zero-ETL.

# Risoluzione dei problemi delle integrazioni zero-ETL con il codice sorgente Amazon DynamoDB

## Policy RBAC o point-in-time ripristino mancanti sulla tabella DynamoDB di origine

Prima di creare l'integrazione, il codice sorgente deve essere configurato correttamente. Se nella tabella DynamoDB di origine manca la policy RBAC con le autorizzazioni appropriate o se il ripristino è disabilitato, l'integrazione passerà point-in-time allo stato Needs_Attention. Per risolvere questo problema, correggi le autorizzazioni e abilita il ripristino. and/or point-in-time L'integrazione dovrebbe ripristinarsi automaticamente dopo qualche tempo una volta corrette le configurazioni mancanti.

# Risoluzione dei problemi delle integrazioni zero-ETL con sorgenti SaaS (tramite connessione) AWS Glue

## Connessione non configurata correttamente

Se la AWS Glue connessione non è configurata correttamente, l'integrazione potrebbe non riuscire ad accedere alla fonte SaaS. Verifica che le credenziali di connessione siano valide e che il ruolo di origine disponga delle autorizzazioni appropriate per accedere alla connessione.

# Risoluzione dei problemi di integrazioni zero-ETL con target Amazon S3 generico

## A Target-role mancano le autorizzazioni

Se al ruolo di destinazione mancano le autorizzazioni appropriate o se la configurazione è errata, l'integrazione passerà allo stato NEEDS_ATTENTION. Per risolvere il problema, consulta la sezione relativa alla configurazione del ruolo di destinazione. L'integrazione dovrebbe essere ripristinata automaticamente dopo qualche tempo, una volta risolto il problema.

## La politica RBAC di Target Catalog non è configurata correttamente

Se la politica delle risorse del catalogo di destinazione non è configurata correttamente, l'integrazione passerà anche allo stato NEEDS_ATTENTION. Per risolvere il problema, consulta la sezione relativa alla configurazione del ruolo di destinazione. L'integrazione dovrebbe essere ripristinata automaticamente dopo qualche tempo, una volta risolto il problema.

# Risoluzione dei problemi delle integrazioni zero-ETL con il target Amazon S3-Table

## A Target-role mancano le autorizzazioni

Se al ruolo di destinazione mancano le autorizzazioni appropriate o se la configurazione è errata, l'integrazione passerà allo stato NEEDS_ATTENTION. Per risolvere il problema, consulta la sezione relativa alla configurazione del ruolo di destinazione. L'integrazione dovrebbe essere ripristinata automaticamente dopo qualche tempo, una volta risolto il problema.

## La politica RBAC di Target Catalog non è configurata correttamente

Se la politica delle risorse del catalogo di destinazione non è configurata correttamente, l'integrazione passerà anche allo stato NEEDS_ATTENTION. Per risolvere il problema, consulta la sezione relativa alla configurazione del ruolo di destinazione. L'integrazione dovrebbe essere ripristinata automaticamente dopo qualche tempo, una volta risolto il problema.

# Guida generale alla risoluzione dei problemi relativi agli errori di AWS Glue integrazione zero-ETL

Tutte le integrazioni emettono CloudWatch registri dopo il completamento di ogni processo, ad esempio il caricamento completo dei dati o l'acquisizione dei dati delle modifiche. È possibile fare riferimento a tali registri per determinare l'esatta causa principale dell'errore o dell'errore.

Inoltre, crea AWS Glue anche una tabella di sistema nel AWS Glue database di destinazione o in S3-Table. Durante il periodo in cui l'integrazione rimane operativa (cioè non in stato FAILED o DELETED), AWS Glue continuerà ad aggiungere gli stati per ogni singola operazione sulla destinazione, ad esempio il completamento del caricamento completo dei dati o l'acquisizione dei dati di modifica con statistiche come il numero di record, il numero di inserimenti, eliminazioni ecc.

# AWS Glue Qualità dei dati

AWS Glue La qualità dei dati consente di misurare e monitorare la qualità dei dati in modo da poter prendere buone decisioni aziendali. Basato su un DeeQu framework open source, AWS Glue Data Quality offre un'esperienza gestita e senza server. AWS Glue Data Quality funziona con Data Quality Definition Language (DQDL), un linguaggio specifico del dominio utilizzato per definire le regole di qualità dei dati. Per ulteriori informazioni su DQDL e sui tipi di regole supportati, consulta la pagina Riferimento a Data Quality Definition Language (DQDL).

Per informazioni aggiuntive sul prodotto e sui prezzi, consulta la pagina del servizio Qualità dei dati di AWS Glue.

## Vantaggi e funzionalità principali

I vantaggi e le caratteristiche principali di AWS Glue Data Quality includono:

- Serverless: non è necessaria alcuna installazione, applicazione di patch o manutenzione.
- Inizia subito: AWS Glue Data Quality analizza rapidamente i tuoi dati e crea regole di qualità dei dati per te. È possibile iniziare con due clic: "Crea regole sulla qualità dei dati → Regole suggerite".
- Rilevamento dei problemi di qualità dei dati: utilizza il machine learning (ML) per individuare anomalie e problemi di qualità dei dati difficili da rilevare.
- Improvvisa le regole: con più di 25 regole DQ pronte all'uso da cui partire, puoi creare regole adatte alle tue esigenze specifiche.
- Valuta la qualità e prendi decisioni aziendali con fiducia: una volta valutate le regole, ottieni un punteggio di qualità dei dati che fornisce una panoramica dello stato dei tuoi dati. Utilizza il punteggio di qualità dei dati per prendere decisioni aziendali con fiducia.
- Concentrati sui dati errati: AWS Glue Data Quality ti aiuta a identificare i record esatti che hanno causato la diminuzione dei punteggi di qualità. Identificali, mettili in quarantena e correggili facilmente.
- Pagamento in base al consumo: non sono necessarie licenze annuali per utilizzare AWS Glue Data Quality.
- Nessun vincolo: AWS Glue Data Quality è basato sull'open source DeeQu e ti consente di mantenere le regole che stai creando in un linguaggio aperto.
- Controlli della qualità dei dati: è possibile applicare i controlli di qualità dei dati su Data Catalog e sulle pipeline di ETL AWS Glue e gestire la qualità dei dati inattivi e in transito.

- ML-based rilevamento della qualità dei dati: utilizza l'apprendimento automatico (ML) per rilevare anomalie e problemi di qualità dei dati difficili da rilevare.
- Linguaggio aperto per esprimere regole: garantisce che le regole sulla qualità dei dati siano redatte in modo coerente e semplice. Gli utenti aziendali possono esprimere facilmente le regole sulla qualità dei dati in un linguaggio semplice e comprensibile. Per gli ingegneri, questo linguaggio offre la flessibilità necessaria per generare codice, implementare un controllo coerente delle versioni e automatizzare le implementazioni.

# Come funziona

Esistono due punti di accesso per la qualità AWS Glue dei dati: i job e ETL. AWS Glue Data Catalog AWS Glue Questa sezione fornisce una panoramica dei casi d'uso e delle AWS Glue funzionalità supportate da ciascun punto di ingresso.

## Qualità dei dati per AWS Glue Data Catalog

AWS Glue Data Quality valuta gli oggetti archiviati in. AWS Glue Data Catalog Offre ai non programmatori un modo semplice per impostare regole di qualità dei dati. Queste figure includono amministratori di dati e analisti aziendali.

È possibile scegliere questa opzione per i seguenti casi d'uso:

- Desideri eseguire attività relative alla qualità dei dati su set di dati che hai già catalogato in AWS Glue Data Catalog.
- Ti occupi di governance dei dati e devi identificare o valutare i problemi di qualità dei dati nel tuo data lake su base continuativa.

È possibile gestire la qualità dei dati per Catalogo dati utilizzando le seguenti interfacce:

- La AWS Glue console di gestione
- AWS Glue API

Per iniziare a usare AWS Glue Data Quality for the AWS Glue Data Catalog seeNozioni di base su AWS Glue Data Quality per Data Catalog.

# Qualità dei dati per AWS Glue lavori ETL

AWS Glue Data Quality for AWS Glue ETL Jobs consente di eseguire attività proattive sulla qualità dei dati. Le attività proattive ti aiutano a identificare e filtrare i dati errati prima di caricare un set di dati nel tuo data lake.

Video: Presentazione della qualità AWS Glue dei dati per le pipeline ETL

È possibile scegliere la qualità dei dati per i processi ETL per i seguenti casi d'uso:

- Desideri integrare attività relative alla qualità dei dati nei tuoi processi ETL
- Desideri scrivere codice che definisca le attività relative alla qualità dei dati negli script ETL
- Vuoi gestire la qualità dei dati che fluiscono nelle tue pipeline di dati visive

È possibile gestire la qualità dei dati per i processi ETL utilizzando le seguenti interfacce:

- AWS Glue Studio, AWS Glue Studio notebook e sessioni interattive AWS Glue
- AWS Glue librerie per lo scripting ETL
- AWS Glue API

Per iniziare a utilizzare la qualità dei dati per i processi ETL, consulta la pagina Tutorial: Getting started with Data Quality nella Guida per l'utente di AWS Glue Studio .

## Confronto della qualità dei dati per Catalogo dati con la qualità dei dati per i processi ETL

Questa tabella fornisce una panoramica delle funzionalità supportate da ogni punto di ingresso di AWS Glue Data Quality.

| Funzionalità | Qualità dei dati per Catalogo dati | Qualità dei dati per i processi ETL |
|---|---|---|
| Origini dati | Amazon S3 Amazon Redshift, sorgenti JDBC compatibili con Data Catalog e formati di data lake transazionali come Apache Iceberg, Apache | Tutte le fonti di dati supportat e da AWS Glue, inclusi connettori personalizzati e connettori di terze parti. |

| Funzionalità | Qualità dei dati per Catalogo dati | Qualità dei dati per i processi ETL |
|---|---|---|
| | Hudi e Delta Lake. AWS Lake Formation sono supportati anche i formati OTF gestiti con alcune limitazioni. Amazon Athena le viste catalogate in AWS Glue Data Catalog non sono supportate. Consulta Tipi di origini supportate. | |
| Suggerimenti di regole di Qualità dei dati | Supportata | Non supportata |
| Creazione ed esecuzione di regole DQDL | Supportata | Supportata |
| Dimensionamento automatico | Non supportata | Supportata |
| AWS Glue Supporto Flex | Non supportata | Supportata |
| Pianificazione | Supportato durante la valutazione delle regole di Qualità dei dati e tramite Step Functions. | Supportato durante l'utilizzo di Step Functions e flussi di lavoro. |
| Identificazione dei record che non hanno superato i controlli di qualità dei dati | Non supportata | Supportata |
| Integrazione con Amazon EventBridge | Supportata | Supportata |
| Integrazione con Cloudwatch AWS | Supportata | Supportata |
| Scrittura dei risultati di qualità dei dati in Amazon S3 | Supportata | Supportata |

| Funzionalità | Qualità dei dati per Catalogo dati | Qualità dei dati per i processi ETL |
|---|---|---|
| Qualità incrementale dei dati | Supportato tramite predicati pushdown | Supportato tramite segnalibri AWS Glue |
| AWS CloudFormation supporto | Supportata | Supportata |
| ML-based rilevamento di anomalie | Non supportata | Supportata |
| Regole dinamiche | Non supportata | Supportata |

## Considerazioni

Prendi in considerazione i seguenti elementi prima di utilizzare AWS Glue Data Quality:

- Le regole di qualità dei dati non possono valutare origini dati annidate o di tipo elenco. Per informazioni, consulta Appiattimento di strutture annidate.

## Terminologia

L'elenco seguente definisce i termini correlati alla qualità AWS Glue dei dati.

Data Quality Definition Language (DQDL)

Linguaggio specifico del dominio che è possibile utilizzare per scrivere regole di qualità AWS Glue dei dati.

Per ulteriori informazioni su DQDL, consulta la guida di Riferimento a Data Quality Definition Language (DQDL).

qualità dei dati

Descrive in che modo un set di dati soddisfa il suo scopo specifico. AWS Glue Data Quality valuta le regole rispetto a un set di dati per misurare la qualità dei dati. Ogni regola verifica caratteristiche particolari come la freschezza o l'integrità dei dati. Per quantificare la qualità dei dati, è possibile utilizzare un punteggio di qualità dei dati.

punteggio di qualità dei dati

La percentuale di regole sulla qualità dei dati che vengono rispettate (risultano vere) quando si valuta un set di regole con Data Quality. AWS Glue

regola

Un'espressione DQDL che controlla i dati per una caratteristica specifica e restituisce un valore booleano. Per ulteriori informazioni, consulta Struttura delle regole.

analyzer

Un'espressione DQDL che raccoglie statistiche sui dati. Un analizzatore raccoglie statistiche sui dati che possono essere utilizzate dagli algoritmi ML per individuare anomalie e problemi di qualità dei dati difficili da rilevare nel tempo.

set di regole

Una AWS Glue risorsa che comprende una serie di regole di qualità dei dati. Un set di regole deve essere associato a una tabella in AWS Glue Data Catalog. Quando salvi un set di regole, AWS Glue assegna un nome della risorsa Amazon (ARN) al set di regole.

punteggio di qualità dei dati

La percentuale di regole di qualità dei dati che vengono approvate (risultano vere) quando si valuta un set di regole con AWS Glue Data Quality.

osservazione

Informazioni non confermate generate da AWS Glue analizzando le statistiche sui dati raccolte da regole e analizzatori nel tempo.

# Limits

AWS Glue Limiti del servizio Data Quality:

- Puoi avere 2.000 regole in un set di regole. Se i tuoi set di regole sono maggiori, ti consigliamo di suddividerli in più set di regole.
- La dimensione del set di regole è di 65 KB. Se i tuoi set di regole sono maggiori, ti consigliamo di suddividerli in più set di regole.
- AWS Glue Data Quality raccoglie statistiche quando si crea una regola o un analizzatore. Non vi è alcun costo associato alla memorizzazione di queste statistiche. Tuttavia, esiste un limite di 100.000 statistiche per account e tali statistiche verranno conservate per un massimo di due anni.

# Note di rilascio per AWS Glue Data Quality

Questo argomento descrive le funzionalità introdotte in AWS Glue Data Quality.

## Disponibilità generale: nuove funzionalità

Le seguenti nuove funzionalità sono disponibili con la disponibilità generale di AWS Glue Data Quality:

- La capacità di identificare quali record non hanno superato i controlli di qualità dei dati è ora supportata in AWS Glue Studio
- Nuovi tipi di regole sulla qualità dei dati, come la convalida dell'integrità referenziale dei dati tra due set di dati, il confronto dei dati tra due set di dati e il controllo dei tipi di dati
- Esperienza utente migliorata in AWS Glue Data Catalog
- Supporto per Apache Iceberg, Apache Hudi e Delta Lake
- Supporto per Amazon Redshift
- Notifica semplificata con Amazon EventBridge
- AWS CloudFormation supporto per la creazione di set di regole
- Miglioramenti delle prestazioni: opzione di memorizzazione nella cache in ETL e AWS Glue Studio per prestazioni più rapide nella valutazione della qualità dei dati

## 27 novembre 2023 (anteprima)

- ML-powered le funzionalità di rilevamento delle anomalie sono ora disponibili in ETL e. AWS Glue AWS Glue Studio In questo modo, ora è possibile individuare anomalie e problemi di qualità dei dati difficili da rilevare.
- Dynamic Rules consente di fornire soglie dinamiche (ad esempio:) `RowCount> avg(last(10))`

## 12 marzo 2024

- Miglioramenti per DQDL
  - Supporto per parole chiave come NULL, BLANKS, WHITESPACES_ONLY
  - Opzioni per specificare in che modo AWS Glue Data Quality deve gestire le regole Composite
  - ColumnValues il tipo di regola non consentirà il passaggio di valori NULL durante i confronti

- Supporto per l'operatore NOT in DQDL

## 26 giugno 2024

- Miglioramenti per DQDL
  - DQDL ora supporta la clausola where in modo da poter filtrare i dati prima di applicare le regole DQ

## 7 agosto 2024

- Il Rilevamento delle anomalie e le Regole dinamiche sono ora disponibili a livello generale

## 22 novembre 2024

- Le regole composite complesse consentono di creare regole aziendali più complesse con supporto annidato
- Nuovi tipi di regole per la gestione della qualità dei dati per i file
  - FileFreshness
  - FileSize
  - FileUniqueness
  - FileMatch
- Controlli di qualità dei dati predefiniti nei processi dell'ETL visivo

## 6 dicembre 2024

- AWS Glue Data Quality ora supporta Amazon SageMaker AI LakeHouse tabelle e tabelle Iceberg, Delta e HUDI AWS Lake Formation gestite in ETL 5.0. AWS Glue

## 7 luglio 2025

- AWS Glue Qualità dei dati; ora supporta Amazon S3 Tables, RMS, Lakehouse e le tabelle Iceberg AWS Lake Formation gestite in Data Catalog. AWS Glue

## 21 novembre 2025

- AWS Glue Data Quality ora supporta l'etichettatura delle regole per una reportistica avanzata. Puoi organizzare e analizzare i risultati sulla qualità dei dati in modo più efficace interrogando i risultati mediante etichette specifiche per identificare le regole non valide all'interno di particolari categorie, contare i risultati delle regole per team o dominio e creare report mirati per le diverse parti interessate. Per ulteriori informazioni, consulta Etichette.
- AWS Glue Data Quality ora supporta Constants in DQDL, consentendoti di definire valori costanti e di farvi riferimento in tutto lo script. Questo aiuta a prevenire problemi relativi ai limiti delle dimensioni delle query quando si lavora con istruzioni SQL di grandi dimensioni. Per ulteriori informazioni, consulta Costanti.

# Rilevamento delle anomalie in Qualità dei dati di AWS Glue

Caso di studio sul machine learning in Qualità dei dati

Gli ingegneri gestiscono contemporaneamente centinaia di pipeline di dati. Ogni pipeline può estrarre dati da varie origini e caricarli nel data lake o in altri repository di dati. Per garantire che vengano forniti dati di alta qualità per il processo decisionale, vengono stabilite regole relative alla qualità dei dati. Queste regole valutano i dati sulla base di criteri fissi che rispecchiano lo stato attuale dell'azienda. Tuttavia, quando l'ambiente aziendale cambia, cambiano anche le proprietà dei dati, rendendo obsoleti questi criteri fissi e causando una scarsa qualità dei dati.

Ad esempio, un ingegnere dei dati presso un'azienda di vendita al dettaglio ha stabilito una regola che convalida che le vendite giornaliere devono superare una one-million-dollar soglia. Dopo qualche mese, le vendite giornaliere hanno superato i due milioni di dollari, rendendo obsoleta la soglia. L'ingegnere dei dati non ha potuto aggiornare le regole in modo che riflettessero le soglie più recenti a causa della mancanza di notifiche e dello sforzo richiesto per analizzare e aggiornare manualmente la regola. Più avanti nel mese, gli utenti aziendali hanno notato un calo del 25% nelle vendite. Dopo ore di indagini, i tecnici dei dati hanno scoperto che una pipeline ETL responsabile dell'estrazione dei dati da alcuni negozi aveva smesso di funzionare senza generare errori. La regola con soglie obsolete ha continuato a funzionare senza rilevare il problema.

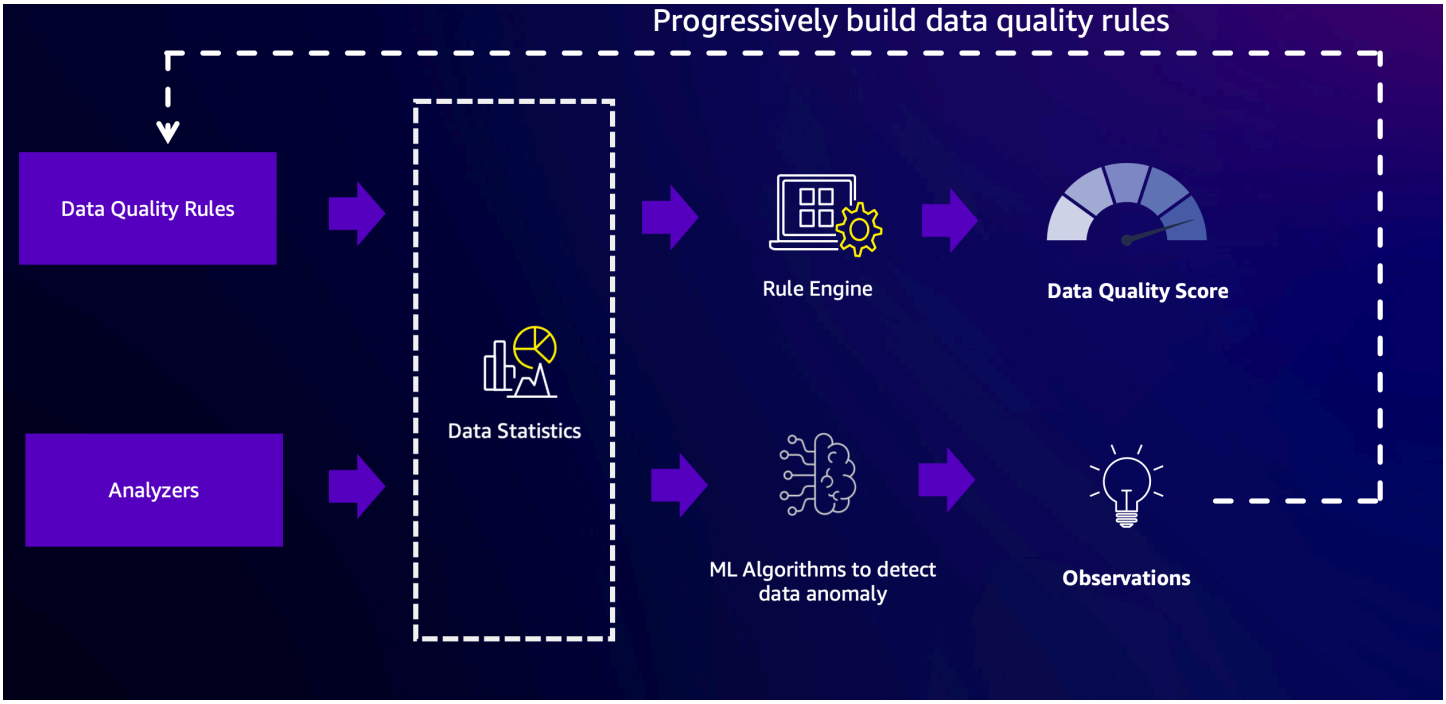
In alternativa, avvisi proattivi in grado di rilevare queste anomalie avrebbero potuto permettere agli utenti di individuare il problema. Inoltre, il monitoraggio della stagionalità nell'azienda può evidenziare problemi significativi relativi alla qualità dei dati. Ad esempio, le vendite al dettaglio possono essere più elevate nei fine settimana e nei periodi festivi, mentre sono relativamente basse nei giorni feriali.

Una divergenza da questo modello può indicare problemi nella qualità dei dati o cambiamenti nella situazione aziendale. Le regole di qualità dei dati non sono in grado di rilevare i modelli stagionali, in quanto ciò richiede algoritmi avanzati in grado di apprendere dai modelli passati che catturano la stagionalità per rilevare le deviazioni.

Infine, gli utenti trovano difficile creare e gestire le regole a causa della natura tecnica del processo di creazione delle regole e del tempo necessario per redigerle. Di conseguenza, prima di definire le regole preferiscono esplorare le informazioni ricavate dai dati. I clienti devono essere in grado di individuare facilmente le anomalie, in modo da poter rilevare in modo proattivo i problemi relativi alla qualità dei dati e prendere decisioni aziendali più consapevoli.

## Come funziona

> **Note**
>
> Il rilevamento delle anomalie è supportato solo in AWS Glue ETL. Non è supportato nella qualità dei dati basata sul Catalogo dati.




AWS Glue Data Quality combina la potenza della qualità dei dati basata su regole e le funzionalità di rilevamento delle anomalie per fornire dati di alta qualità. Per iniziare, prima di tutto è necessario configurare regole e analizzatori, quindi abilitare il rilevamento delle anomalie.

## Regole

Regole: le regole esprimono le aspettative relative ai dati in un linguaggio aperto denominato Data Quality Definition Language (DQDL). Di seguito è riportato un esempio di regola. Questa regola avrà esito positivo quando non ci saranno valori vuoti o NULL nella colonna 'passenger_count':

```
Rules = [
    IsComplete "passenger_count"
]
```

## Analizzatori

Nelle situazioni in cui si conoscono le colonne critiche ma non si dispone di informazioni sufficienti sui dati per scrivere regole specifiche, è possibile monitorare tali colonne utilizzando gli analizzatori. Gli analizzatori rappresentano un modo per raccogliere statistiche sui dati senza definire regole esplicite. Di seguito è riportato un esempio di configurazione degli analizzatori:

```
Analyzers = [
AllStatistics "fare_amount",
DistinctValuesCount "pulocationid",
RowCount
]
```

In questo esempio sono configurati tre analizzatori:

1. Il primo analizzatore, `AllStatistics «fare_amount"`, acquisirà tutte le statistiche disponibili per il campo `fare_amount`.
2. Il secondo analizzatore, `«DistinctValuesCount pulocationid"`, acquisirà il conteggio dei valori distinti nella colonna `pulocationid`.
3. Il terzo analizzatore, `RowCount`, acquisirà il numero totale di record nel set di dati.

Gli analizzatori rappresentano un modo semplice per raccogliere statistiche rilevanti sui dati senza specificare regole complesse. Monitorando queste statistiche, è possibile ottenere informazioni dettagliate sulla qualità dei dati e identificare potenziali problemi o anomalie che potrebbero richiedere ulteriori indagini o la creazione di regole specifiche.

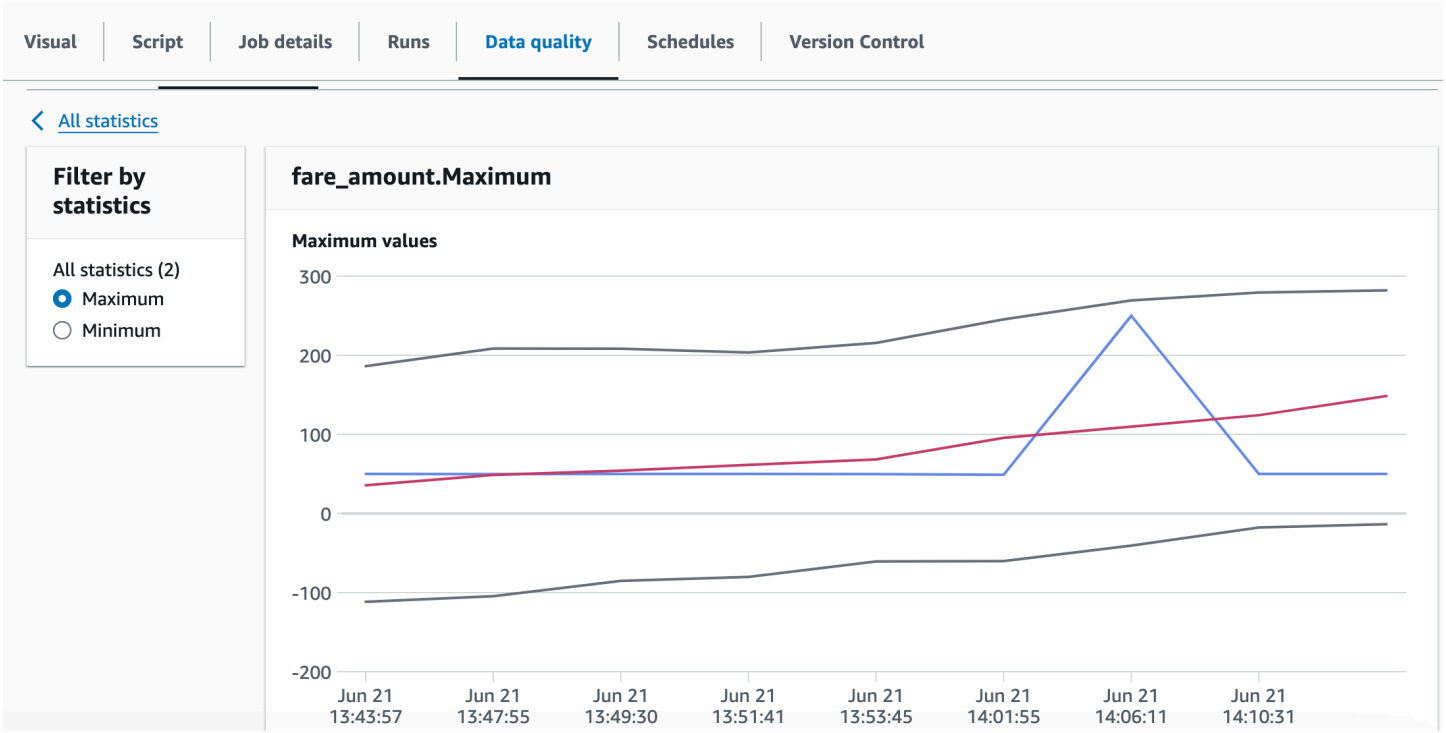
## Statistiche dei dati

Sia Analyzer che Rules in AWS Glue Data Quality raccolgono statistiche sui dati, note anche come profili di dati. Queste statistiche forniscono informazioni dettagliate sulle caratteristiche e sulla qualità dei dati. Le statistiche raccolte vengono archiviate nel tempo all'interno del servizio AWS Glue, che consente di tracciare e analizzare le modifiche nei profili di dati.

Puoi recuperare facilmente queste statistiche e scriverle su Amazon S3 per ulteriori analisi o per lo storage a lungo termine richiamando le informazioni appropriate. APIs Questa funzionalità permette di integrare la profilazione dei dati nei flussi di lavoro di elaborazione dei dati e di sfruttare le statistiche raccolte per diversi scopi, come il monitoraggio della qualità dei dati e il rilevamento delle anomalie.

Archiviando i profili di dati in Amazon S3, è possibile sfruttare la scalabilità, la durabilità e l'economicità del servizio di archiviazione di oggetti di Amazon. Inoltre, puoi sfruttare altri AWS servizi o strumenti di terze parti per analizzare e visualizzare i profili di dati, consentendoti di ottenere informazioni più approfondite sulla qualità dei dati e prendere decisioni informate sulla gestione e la governance dei dati.

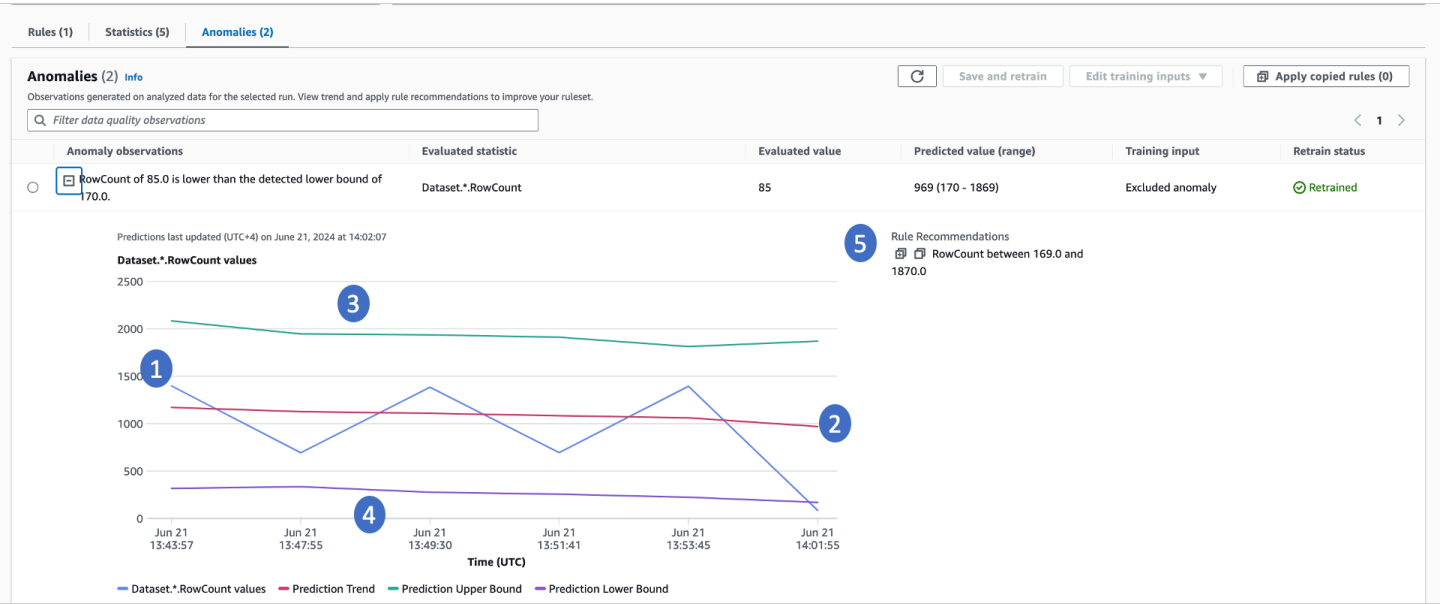
Di seguito è riportato un esempio di statistiche sui dati memorizzate nel tempo.



> **Note**
>
> AWS Glue Data Quality raccoglierà le statistiche una sola volta, anche se hai sia Rule che Analyzer per le stesse colonne, rendendo efficiente il processo di generazione delle statistiche.

## Rilevamento di anomalie

AWS Glue Data Quality richiede un minimo di tre punti dati per rilevare le anomalie. Sfrutta un algoritmo di machine learning per imparare dalle tendenze passate e quindi prevedere i valori futuri. Quando il valore effettivo non rientra nell'intervallo previsto, AWS Glue Data Quality crea un'osservazione delle anomalie. Offre una rappresentazione visiva del valore effettivo e delle tendenze. Nel seguente grafico vengono mostrati quattro valori.




1. La statistica attuale e la sua tendenza nel tempo.
2. Una tendenza derivata imparando dalla tendenza attuale. È utile per comprendere la direzione della tendenza.
3. Il possibile limite superiore per la statistica.
4. Il possibile limite inferiore per la statistica.
5. Regole per la qualità dei dati raccomandate in grado di rilevare questi problemi in futuro.

Ci sono alcune cose importanti da tenere presente riguardo alle anomalie:

- Quando si generano anomalie, i punteggi relativi alla qualità dei dati non subiscono alcuna variazione.
- Quando viene rilevata un'anomalia, questa viene considerata normale per le esecuzioni successive. A meno che questo valore anomalo non venga esplicitamente escluso, l'algoritmo di machine learning lo considererà come input.

### Riaddestramento

Il riaddestramento del modello di rilevamento delle anomalie è fondamentale per rilevare le anomalie corrette. Quando vengono rilevate anomalie, AWS Glue Data Quality include l'anomalia nel modello come valore normale. Per garantire che il rilevamento delle anomalie funzioni correttamente, è importante fornire un feedback riconoscendo o rifiutando l'anomalia. AWS Glue Data Quality fornisce meccanismi sia in AWS Glue Studio che APIs per fornire feedback al modello. Per ulteriori informazioni, consulta la documentazione sulla configurazione del rilevamento delle anomalie nelle pipeline AWS Glue ETL.

## Dettagli dell'algoritmo di rilevamento delle anomalie

- L'algoritmo di rilevamento delle anomalie esamina le statistiche dei dati nel tempo. L'algoritmo considera tutti i punti dati disponibili e ignora tutte le statistiche esplicitamente escluse.
- Queste statistiche sui dati vengono archiviate nel servizio AWS Glue e puoi fornire AWS KMS chiavi per crittografarle. Consulta la Guida alla sicurezza su come fornire AWS KMS le chiavi per crittografare le statistiche sulla qualità dei dati di AWS Glue.
- La componente del tempo è fondamentale per l'algoritmo di rilevamento delle anomalie. Sulla base dei valori passati, AWS Glue Data Quality determina i limiti superiore e inferiore. Nel determinarli, tiene in considerazione la componente del tempo. I limiti saranno diversi per gli stessi valori su un intervallo di un minuto, su un intervallo di un'ora o su un intervallo di un giorno.

### Acquisire la stagionalità

AWS L'algoritmo di rilevamento delle anomalie di Glue Data Quality può catturare modelli stagionali. Ad esempio, è in grado di comprendere che i modelli dei giorni feriali sono diversi da quelli dei fine settimana. Questo può essere visto nell'esempio seguente, in cui AWS Glue Data Quality rileva una tendenza stagionale nei valori dei dati. Non è necessario eseguire alcuna operazione specifica per abilitare questa funzionalità. Nel tempo, AWS Glue Data Quality apprende le tendenze stagionali e rileva le anomalie quando questi schemi si interrompono.

## Costo

Il costo sarà calcolato in base al tempo necessario per rilevare le anomalie. Ogni statistica comporta un costo di 1 DPU per il tempo necessario a rilevare le anomalie. Per esempi dettagliati, consultare i prezzi di AWS Glue.

## Considerazioni chiave

L'archiviazione delle statistiche è gratuita. Tuttavia, c'è un limite di 100.000 statistiche per account. Le statistiche saranno archiviate per un massimo di due anni.

# Configurazione delle autorizzazioni IAM per AWS Glue Data Quality

Questo argomento fornisce informazioni per aiutarti a comprendere le azioni e le risorse che un amministratore IAM può utilizzare in una policy AWS Identity and Access Management (IAM) per AWS Glue Data Quality. Include anche esempi di policy IAM con le autorizzazioni minime necessarie per utilizzare AWS Glue Data Quality con AWS Glue Data Catalog.

Per ulteriori informazioni sulla sicurezza in AWS Glue, vedereSicurezza in AWS Glue.

# Autorizzazioni IAM per AWS Glue Data Quality

La tabella seguente elenca le autorizzazioni richieste a un utente per eseguire operazioni di Qualità dei dati di AWS Glue specifiche. Per impostare un'autorizzazione granulare per AWS Glue Data Quality, puoi specificare queste azioni nell'`Action`elemento di una dichiarazione di policy IAM.

AWS Azioni di Glue Data Quality

| Azione | Description | Tipi di risorse |
| --- | --- | --- |
| `glue:CreateDataQualityRuleset` | Concede l'autorizzazione per creare un set di regole di qualità dei dati. | `::dataQualityRuleset/<name>` |
| `glue:DeleteDataQualityRuleset` | Concede l'autorizzazione per eliminare un set di regole di qualità dei dati. | `::dataQualityRuleset/<name>` |
| `glue:GetDataQualityRuleset` | Concede l'autorizzazione per recuperare un set di regole di qualità dei dati. | `::dataQualityRuleset/<name>` |
| `glue:ListDataQualityRulesets` | Concede l'autorizzazione per recuperare tutti i set di regole di qualità dei dati. | `::dataQualityRuleset/*` |
| `glue:UpdateDataQualityRuleset` | Concede l'autorizzazione per aggiornare un set di regole di qualità dei dati. | `::dataQualityRuleset/<name>` |
| `glue:GetDataQualityResult` | Concede l'autorizzazione per recuperare un risultato di esecuzione dell'attività di qualità dei dati. Questa azione IAM fornisce anche le autorizzazioni per le seguenti API:<br>• `BatchGetDataQualityQualityResult` | `::dataQualityRuleset/<name>` |

| Azione | Description | Tipi di risorse |
| --- | --- | --- |
| | • ListDataQualityStatistics<br>• ListDataQualityStatisticAnnotations | |
| glue:ListDataQualityResults | Concede l'autorizzazione per recuperare i risultati di tutte le esecuzioni delle attività di qualità dei dati. | ::dataQualityRuleset/* |
| glue:CancelDataQualityRuleRecommendationRun | Concede l'autorizzazione per interrompere l'esecuzione dell'attività di raccomandazione di qualità dei dati in corso. | ::dataQualityRuleset/* |
| glue:GetDataQualityRuleRecommendationRun | Concede l'autorizzazione per recuperare l'esecuzione di un'attività di raccomandazione di qualità dei dati. | ::dataQualityRuleset/* |
| glue:ListDataQualityRuleRecommendationRuns | Concede l'autorizzazione per recuperare tutte le esecuzioni dell'attività di raccomandazione di qualità dei dati. | ::dataQualityRuleset/* |
| glue:StartDataQualityRuleRecommendationRun | Concede l'autorizzazione per iniziare l'esecuzione di un'attività di raccomandazione di qualità dei dati. | ::dataQualityRuleset/* |
| glue:CancelDataQualityRulesetEvaluationRun | Concede l'autorizzazione per interrompere l'esecuzione dell'attività di qualità dei dati in corso. | ::dataQualityRuleset/* |

| Azione | Description | Tipi di risorse |
| --- | --- | --- |
| glue:GetDataQualityRulesetEvaluationRun | Concede l'autorizzazione per recuperare una esecuzione dell'attività di qualità dei dati. | ::dataQualityRuleset/* |
| glue:ListDataQualityRulesetEvaluationRuns | Concede l'autorizzazione per recuperare tutte le esecuzioni dell'attività di qualità dei dati. | ::dataQualityRuleset/* |
| glue:StartDataQualityRulesetEvaluationRun | Concede l'autorizzazione per iniziare l'esecuzione di un'attività di qualità dei dati. | ::dataQualityRuleset/<name> |
| glue:PublishDataQuality | Concede l'autorizzazione per pubblicare i risultati di qualità dei dati. | ::dataQualityRuleset/<name> |
| glue:GetDataQualityModel | Concede l'autorizzazione per recuperare il modello di qualità dei dati. | ::dataQualityRuleset/<name>, ::job/<name> |
| glue:GetDataQualityModelResult | Concede l'autorizzazione per recuperare i risultati del modello di qualità dei dati. | ::dataQualityRuleset/<name>, ::job/<name> |
| glue:PutDataQualityStatisticAnnotation | Concede l'autorizzazione per aggiungere annotazioni alle statistiche. Questa azione IAM fornisce anche le autorizzazioni per le seguenti API:<br>• BatchPutDataQualityStatisticAnnotation | ::dataQualityRuleset/<name>, ::job/<name> |

| Azione | Description | Tipi di risorse |
| --- | --- | --- |
| `glue:PutDataQualityProfileAnnotation` | Concede l'autorizzazione per aggiungere annotazioni a tutte le statistiche di un profilo. | `::dataQualityRuleset/<name>, ::job/<name>` |

# Configurazione IAM richiesta per la pianificazione delle esecuzioni di valutazione

## autorizzazioni IAM

Per eseguire esecuzioni di valutazione pianificate della qualità dei dati, devi aggiungere l'operazione `IAM:PassRole` alla policy delle autorizzazioni.

AWS EventBridge Autorizzazioni richieste da Scheduler

| Azione | Description | Tipi di risorse |
| --- | --- | --- |
| `iam:PassRole` | Concede a IAM l'autorizzazione per consentire all'utente di trasmettere i ruoli approvati. | ARN del ruolo utilizzato per chiamare `StartDataQualityRulesetEvaluationRun` |

Senza queste autorizzazioni si verifica il seguente errore:

```
"errorCode": "AccessDenied"
"errorMessage": "User: arn:aws:sts::account_id:assumed-role/AWSGlueServiceRole is not
authorized to perform: iam:PassRole on resource: arn:aws:iam::account_id:role/service-
role/AWSGlueServiceRole
because no identity-based policy allows the iam:PassRole action"
```

## Entità attendibili di IAM

I servizi AWS Glue and AWS EventBridge Scheduler devono essere elencati nelle entità attendibili per creare ed eseguire una pianificazione. `StartDataQualityEvaluationRun`

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "glue.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "scheduler.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

## Policy IAM di esempio

Un ruolo IAM per AWS Glue Data Quality richiede i seguenti tipi di autorizzazioni:

- Autorizzazioni per le operazioni di AWS Glue Data Quality in modo da poter ottenere le regole di qualità dei dati consigliate ed eseguire un'attività di qualità dei dati su una tabella nel AWS Glue Data Catalog. Le politiche IAM di esempio in questa sezione includono le autorizzazioni minime richieste per le operazioni di AWS Glue Data Quality.
- Autorizzazioni che concedono l'accesso alla tabella del catalogo dati e ai dati sottostanti. Queste autorizzazioni sono diverse a seconda del caso d'uso. Ad esempio, per i dati che cataloghi in Amazon S3, le autorizzazioni devono includere l'accesso ad Amazon S3.

> **Note**
>
> È necessario configurare le autorizzazioni Amazon S3 oltre alle autorizzazioni descritte in questa sezione.

## Autorizzazioni minime per ottenere le regole di qualità dei dati consigliate

Questa policy di esempio include le autorizzazioni necessarie per generare regole di qualità dei dati consigliate.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "AllowGlueRuleRecommendationRunActions",
      "Effect": "Allow",
      "Action": [
        "glue:GetDataQualityRuleRecommendationRun",
        "glue:PublishDataQuality",
        "glue:CreateDataQualityRuleset"
      ],
      "Resource": "arn:aws:glue:us-east-1:111122223333:dataQualityRuleset/*"
    },
 {
     "Sid": "AllowCatalogPermissions",
     "Effect": "Allow",
     "Action": [
        "glue:GetPartitions",
        "glue:GetTable"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Sid": "AllowS3GetObjectToRunRuleRecommendationTask",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": "arn:aws:s3:::aws-glue-*"
    },
  {
   "Sid": "AllowPublishingCloudwatchLogs",
   "Effect": "Allow",
   "Action": [
```

```
      "logs:CreateLogStream",
      "logs:CreateLogGroup",
      "logs:PutLogEvents"
    ],
    "Resource": "*"
     }
  ]
}
```

## Autorizzazioni minime per l'esecuzione di un'attività di qualità dei dati

Questa policy di esempio include le autorizzazioni necessarie per eseguire un'attività di valutazione della qualità dei dati.

Le seguenti dichiarazioni di policy sono facoltative, a seconda del caso d'uso:

- `AllowCloudWatchPutMetricDataToPublishTaskMetrics` - Obbligatorio se desideri pubblicare i parametri di esecuzione della qualità dei dati in Amazon CloudWatch.
- `AllowS3PutObjectToWriteTaskResults` - Obbligatorio se desideri scrivere i risultati di esecuzione della qualità dei dati su Amazon S3.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "AllowGlueGetDataQualityRuleset",
      "Effect": "Allow",
      "Action": [
        "glue:GetDataQualityRuleset"
      ],
      "Resource": "arn:aws:glue:us-east-1:111122223333:dataQualityRuleset/YOUR-
RULESET-NAME"
    },
    {
      "Sid": "AllowGlueRulesetEvaluationRunActions",
      "Effect": "Allow",
      "Action": [
        "glue:GetDataQualityRulesetEvaluationRun",
```

```
        "glue:PublishDataQuality"
      ],
      "Resource": "arn:aws:glue:us-east-1:111122223333:dataQualityRuleset/*"
    },
    {
      "Sid": "AllowCatalogPermissions",
      "Effect": "Allow",
      "Action": [
        "glue:GetPartitions",
        "glue:GetTable"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Sid": "AllowS3GetObjectForRulesetEvaluationRun",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": "arn:aws:s3:::aws-glue-*"
    },
    {
      "Sid": "AllowCloudWatchPutMetricDataToPublishTaskMetrics",
      "Effect": "Allow",
      "Action": [
        "cloudwatch:PutMetricData"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "cloudwatch:namespace": "Glue Data Quality"
        }
      }
    },
    {
      "Sid": "AllowS3PutObjectToWriteTaskResults",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject*"
      ],
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*"
    }
```

```
  ]
}
```

## Autorizzazioni minime per eseguire un processo ETL relativo alla qualità dei dati

Questa policy di esempio include le autorizzazioni necessarie per eseguire un processo ETL relativo alla qualità dei dati.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "AllowGluePublishDataQualityResult",
      "Effect": "Allow",
      "Action": [
        "glue:PublishDataQuality"
      ],
      "Resource": "arn:aws:glue:us-east-1:111122223333:dataQualityRuleset/*"
    },
    {
      "Sid": "AllowGlueGetDataQualityResult",
      "Effect": "Allow",
      "Action": [
        "glue:GetDataQualityResult"
      ],
      "Resource": "arn:aws:glue:us-east-1:111122223333:dataQualityRuleset/*"
    },
    {
      "Sid": "AllowGlueDataQualityStatisticAnnotation",
      "Effect": "Allow",
      "Action": [
        "glue:PutDataQualityStatisticAnnotation"
      ],
      "Resource": [
        "arn:aws:glue:us-east-1:111122223333:dataQualityRuleset/*",
        "arn:aws:glue:us-east-1:111122223333::job/{JobName}"
      ]
    },
    {
```

```
      "Sid": "AllowGlueDataQualityProfileAnnotation",
      "Effect": "Allow",
      "Action": [
        "glue:PutDataQualityProfileAnnotation"
      ],
      "Resource": [
        "arn:aws:glue:us-east-1:111122223333:dataQualityRuleset/*",
        "arn:aws:glue:us-east-1:111122223333::job/{JobName}"
      ]
    }
 ]
}
```

# Nozioni di base su AWS Glue Data Quality per Data Catalog

Questa sezione introduttiva fornisce istruzioni per aiutarti a iniziare a utilizzare AWS Glue Data Quality sulla console AWS Glue. Imparerai come completare attività essenziali come la generazione di raccomandazioni di regole di qualità dei dati e la valutazione di un set di regole rispetto ai propri dati.

Argomenti


- Prerequisiti
- Step-by-step esempio
- Generazione di raccomandazioni di regole
- Monitoraggio dei suggerimenti di regole
- Modifica dei set di regole suggeriti
- Creazione di un nuovo set di regole
- Esecuzione di un set di regole per valutare la qualità dei dati
- Visualizzazione del punteggio e dei risultati della qualità dei dati
- Utilizzo di interrogazioni di preelaborazione
- Tipi di origini supportate
- Argomenti correlati

## Prerequisiti

Prima di utilizzare AWS Glue Data Quality, è necessario conoscere l'utilizzo di Data Catalog e dei crawler in AWS Glue. Con AWS Glue Data Quality, è possibile valutare la qualità delle tabelle in un database Data Catalog. Devi disporre anche dei seguenti elementi:

- Una tabella nel Data Catalog rispetto alla quale valutare il set di regole di qualità dei dati.
- Un ruolo IAM per AWS Glue fornito quando si generano i suggerimenti di regole o se si esegue un'attività di qualità dei dati. Questo ruolo deve disporre dell'autorizzazione per l'accesso alle risorse che vari processi AWS Glue Data Quality richiedono per l'esecuzione per tuo conto. Queste risorse includono AWS Glue Amazon S3 e. CloudWatch Per visualizzare policy di esempio che includono le autorizzazioni minime per AWS Glue Data Quality, consulta la pagina Policy IAM di esempio.

  Per ulteriori informazioni sui ruoli IAM per AWS Glue, consulta le pagine Create an IAM policy for the AWS Glue service e Create an IAM role for the AWS Glue service. È inoltre possibile consultare un elenco di tutte le autorizzazioni AWS Glue specifiche per la qualità dei dati nella pagina Authorization for AWS Glue Data Quality actions.
- Un database con almeno una tabella che contiene una varietà di dati. La tabella utilizzata in questo tutorial è denominata `yyz-tickets`, con la tabella `tickets`. Questi dati sono una raccolta di informazioni disponibili al pubblico dalla città di Toronto per le violazioni in materia di sosta. Se crei la tua tabella, assicurati che sia compilata con una serie di dati validi per ottenere il miglior set di regole suggerite.

## Step-by-step esempio

Per un step-by-step esempio con set di dati di esempio, consulta il post sul blog AWS Glue Data Quality.

## Generazione di raccomandazioni di regole

I suggerimenti di regole consentono di iniziare a utilizzare facilmente la qualità dei dati senza scrivere codice. Con Qualità dei dati di AWS Glue è possibile analizzare i dati, identificare le regole e creare un set di regole che possono essere valutate in un'attività di qualità dei dati. Le esecuzioni di consigli vengono eliminate automaticamente dopo 90 giorni.

Generazione di raccomandazioni di regole di qualità dei dati

1. Apri la console AWS Glue all'indirizzo https://console.aws.amazon.com/glue/.
2. Nel pannello di navigazione, seleziona Tables (Tabelle). Scegliere quindi la tabella per la quale si desidera generare le raccomandazioni di regole di qualità dei dati.
3. Nella pagina dei dettagli della tabella, seleziona la scheda Qualità dei dati per accedere alle regole e alle impostazioni di AWS Glue Data Quality per la tua tabella.
4. Nella scheda Qualità dei dati, scegli Aggiungi regole e monitora la qualità dei dati.
5. Nella pagina Generatore set di regole, un avviso nella parte superiore della pagina ti chiederà di avviare un'attività di suggerimento se non sono presenti esecuzioni di suggerimenti di regole.
6. Scegli Regole suggerite per aprire il modale e inserisci i parametri per l'attività di suggerimento.
7. Scegli un ruolo IAM con accesso a AWS Glue. Questo ruolo deve avere l'autorizzazione ad accedere alle risorse che i vari processi di AWS Glue Data Quality richiedono per eseguire per tuo conto.
8. Dopo aver completato i campi in base alle tue preferenze, scegli Suggerisci regole per avviare l'esecuzione dell'attività di suggerimento. Se le esecuzioni di suggerimento sono in corso o completate, puoi gestirle in questo avviso. Potrebbe essere necessario aggiornare l'avviso per visualizzare la modifica dello stato. Le esecuzioni delle attività di suggerimento completate e in corso vengono visualizzate nella pagina Cronologia delle esecuzioni, che elenca tutte le esecuzioni di suggerimento effettuate negli ultimi 90 giorni.

## Cosa significano le regole suggerite

AWS Glue Data Quality genera regole basate sui dati di ogni colonna della tabella di input. Utilizza le regole per identificare i potenziali limiti entro i quali i dati possono essere filtrati per mantenere i requisiti di qualità. Il seguente elenco di regole generate include esempi utili per comprendere il significato delle regole e gli effetti che potrebbero avere se applicate ai dati.

Per un elenco completo dei tipi di regole Data Quality Definition Language (DQDL) generati, consulta la pagina DQDL rule type reference.

- `IsComplete "SET_FINE_AMOUNT"`: la regola `IsComplete` verifica che la colonna sia compilata in ogni riga specificata. Utilizza questa regola per contrassegnare le colonne come non facoltative nei dati.
- `Uniqueness "TICKET_NUMBER" > 0.95`: la regola `Uniqueness` verifica che i dati all'interno della colonna soddisfino una certa soglia di unicità. In questo esempio, è stato determinato che i

dati che compongono una determinata riga per `"TICKET_NUMBER"` sono identici al massimo al 95% nel contenuto a tutte le altre righe, il che suggerisce questa regola.

- `ColumnValues "PROVINCE" in ["ON", "QC", "AB", "NY",...]`: la regola `ColumnValues` definisce valori validi per la colonna in base al contenuto della colonna esistente. In questo esempio, i dati per ogni riga sono una targa di 2 lettere per uno stato o una provincia.
- `ColumnLength "INFRACTION_DESCRIPTION" between 15 and 31`: la regola `ColumnLength` impone una limitazione di lunghezza sui dati di una colonna. Questa regola viene generata dai dati di esempio in base alle lunghezze minime e massime registrate per una colonna di stringhe.

## Monitoraggio dei suggerimenti di regole

Quando sono in esecuzione i suggerimenti sulle regole di qualità dei dati, la pagina Aggiungi regole e monitora la qualità dei dati visualizza informazioni e operazioni aggiuntive che è possibile intraprendere nella barra superiore.

Quando sono in corso le esecuzioni dei suggerimenti sulle regole, puoi scegliere Interrompi esecuzione prima del completamento dell'attività di suggerimento. Mentre l'attività è in corso, vedrai lo stato In corso e la data e l'ora di inizio dell'esecuzione.

Una volta completati i suggerimenti di regole, la barra di suggerimento mostra il numero di regole suggerite, lo stato dell'ultima esecuzione di suggerimento e la data e l'ora del termine.

È possibile aggiungere le regole suggerite scegliendo Inserisci suggerimento di regola. Per visualizzare le regole precedentemente suggerite, seleziona una data specifica. Per eseguire un nuovo suggerimento, scegli Altre operazioni, quindi scegli Regole suggerite.

Configura le impostazioni predefinite scegliendo Gestisci impostazioni utente. È possibile impostare il percorso predefinito in cui Amazon S3 può archiviare i set di regole o configurare un ruolo predefinito per eseguire Catalogo dati.

## Modifica dei set di regole suggeriti

Poiché Qualità dei dati di AWS Glue genera regole basate sui dati esistenti che hai a disposizione, potresti vedere alcune regole impreviste o indesiderate nei suggerimenti automatici. Per ottenere il massimo dai set di regole suggeriti, è necessario valutarli e modificarli. In questo passaggio del tutorial, prendi le regole generate nel passaggio precedente e le modifichi per applicare qualità più

restrittive su alcuni dati. Inoltre, allenterai altre regole per garantire che dati corretti e univoci possano essere aggiunti in un secondo momento.

Modifica un set di regole suggerito

1. Nella console AWS Glue, scegli Data Catalog, quindi scegli Tabelle Database nel riquadro di navigazione. Seleziona la tabella `tickets`.
2. Nella pagina dei dettagli della tabella, scegli la scheda Qualità dei dati per accedere alle regole e alle impostazioni di AWS Glue Data Quality per la tabella.
3. Nella sezione Set di regole, seleziona il set di regole generato in Generazione di raccomandazioni di regole.
4. Scegli Operazioni, quindi scegli Modifica nella finestra della console. L'editor del set di regole viene caricato nella console. Include un riquadro di modifica delle regole e un riferimento rapido per DQDL.
5. Rimuovi la riga 2 dello script. Ciò allenta il requisito che prevede che la dimensione del database sia limitata entro un certo numero di righe. Dopo la modifica, il file dovrebbe contenere quanto segue nelle righe 1-3:

```
Rules = [
    IsComplete "TAG_NUMBER_MASKED",
    ColumnLength "TAG_NUMBER_MASKED" between 6 and 9,
```

6. Rimuovi la riga 25 dello script. Ciò allenta il requisito che prevede che il 96% delle province registrate sia 0N. Dopo la modifica, il file dovrebbe contenere quanto segue dalla riga 24 alla fine del set di regole:

```
ColumnValues "PROVINCE" in ["ON", "QC", "AB", "NY", "AZ", "NS", "BC", "MI", "PQ",
 "MB", "PA", "FL", "SK", "NJ", "OH", "NB", "IL", "MA", "CA",
    "VA", "TX", "NF", "MD", "PE", "CT", "NC", "GA", "IN", "OR", "MN", "TN", "WI",
 "KY", "MO", "WA", "NH", "SC", "CO", "OK", "VT", "RI", "ME", "AL",
    "YT", "IA", "DE", "AR", "LA", "XX", "WV", "MT", "KS", "NT", "DC", "NV", "NE",
 "UT", "MS", "NM", "ID", "SD", "ND", "AK", "NU", "GO", "WY", "HI"],
ColumnLength "PROVINCE" = 2
]
```

7. Modifica la riga 14 come segue:

```
IsComplete "TIME_OF_INFRACTION",
```

Ciò rafforza il requisito relativo alla colonna limitando il database ai soli ticket che contengono un orario di infrazione registrato. Nel contesto di questo set di dati, è importante considerare i ticket senza un orario di infrazione registrato come dati non validi. In alcune situazioni, potrebbe essere più appropriato considerare il partizionamento o la trasformazione dei dati al fine di consentire un ulteriore utilizzo o ispezione dei dati per determinare una regola di qualità.

8. Scegli Aggiorna set di regole nella parte inferiore della pagina della console.

## Creazione di un nuovo set di regole

Un set di regole è un gruppo di regole di qualità dei dati che vengono valutate in base ai tuoi dati. Nella console AWS Glue, puoi creare set di regole personalizzati utilizzando Data Quality Definition Language (DQDL).

Creazione di un set di regole di qualità dei dati

1. Nella console AWS Glue, scegli Data Catalog, scegli Database, quindi scegli Tabelle nel riquadro di navigazione. Seleziona la tabella `tickets`.
2. Apri la scheda Data quality (Qualità dei dati).
3. Nella sezione Set di regole, scegli Crea set di regole. L'editor DQDL viene avviato nella console. Dispone di un'area di testo per la modifica diretta e di un riferimento rapido alle regole DQDL e allo schema delle tabelle.
4. Inizia ad aggiungere regole all'area di testo dell'editor DQDL. Puoi scrivere le regole direttamente da questo tutorial o utilizzare la funzionalità Generatore di regole DQDL dell'editor delle regole sulla qualità dei dati.

> **Note**
>
> Come utilizzare il generatore di regole DQDL
>
> 1. Seleziona un tipo di regola dall'elenco e scegli il segno più per inserire la sintassi di esempio nel riquadro dell'editor.
> 2. Cambia i nomi delle colonne segnaposto con i nomi delle tue colonne. I nomi delle colonne della tabella sono disponibili nella scheda Schema.
> 3. Aggiorna il parametro dell'espressione per adattarlo al tuo caso. Per un elenco completo delle espressioni supportate da DQDL, consulta Espressioni.

Ad esempio, le seguenti regole sono vincoli per la convalida dei dati della colonna `ticket_number` nella tabella `tickets`. Per aggiungere le seguenti regole, utilizza il generatore di regole DQDL o modifica direttamente il tuo set di regole:

```
IsComplete "ticket_number",
IsUnique "ticket_number",
ColumnValues "ticket_number" > 9000000000
```

5. Fornisci un nome per il tuo nuovo set di regole nel campo Nome del set di regole.
6. Scegli Salva set di regole.

### Valutazione della qualità dei dati su più set di dati

Puoi impostare regole di qualità dei dati su più set di dati utilizzando set di DatasetMatch regole ReferentialIntegrity e. ReferentialIntegrityverifica se i dati del set di dati primario sono presenti in altri set di dati.

Per aggiungere un set di dati di riferimento, scegli la scheda Schema, quindi scegli Aggiorna tabelle di riferimento. Ti verrà richiesto di selezionare un database e una tabella. È possibile aggiungere la tabella e quindi impostare le regole di qualità dei dati. Tipi di regole come AggregateMatch,, RowCountMatch ReferentialIntegrity SchemaMatch, e DatasetMatch supportano la possibilità di eseguire controlli di qualità dei dati su più set di dati.

## Esecuzione di un set di regole per valutare la qualità dei dati

Quando esegui un'attività di qualità dei dati, AWS Glue Data Quality valuta un set di regole rispetto ai tuoi dati e calcola un punteggio di qualità dei dati. Questo punteggio rappresenta la percentuale di regole di qualità dei dati soddisfatte per l'input.

Esecuzione di un'attività di qualità dei dati

1. Nella console AWS Glue, scegli Data Catalog, scegli Database, quindi scegli Tabelle nel riquadro di navigazione. Seleziona la tabella `tickets`.
2. Scegli la scheda Qualità dei dati.
3. Nell'elenco Set di regole, scegli il set di regole rispetto al quale desideri valutare la tabella. Per questo passaggio, ti consigliamo di utilizzare un set di regole che hai già scritto o modificato anziché regole generate. Scegli Esegui.

4. Nel modale, scegli il tuo ruolo IAM. Questo ruolo deve avere l'autorizzazione ad accedere alle risorse che i vari processi di AWS Glue Data Quality richiedono per eseguire per tuo conto. È possibile salvare il ruolo IAM come predefinito o modificarlo accedendo alla pagina Impostazioni predefinite.
5. In Azioni sulla qualità dei dati, scegli se pubblicare le metriche su Amazon CloudWatch. Quando questa opzione è selezionata, AWS Glue Data Quality pubblica metriche che indicano il numero di regole passate e il numero di regole non riuscite. Per intervenire sulle metriche memorizzate in questo modo, puoi utilizzare gli allarmi. CloudWatch Le metriche chiave vengono inoltre pubblicate Amazon EventBridge per consentirti di impostare gli avvisi. Per ulteriori informazioni, consulta la pagina Setting up alerts, deployments, and scheduling.
6. In Frequenza di esecuzione, scegli l'esecuzione on demand oppure pianifica il set di regole. Quando pianifichi un set di regole, ti viene richiesto un nome per l'attività. La pianificazione verrà creata in Amazon EventBridge. Puoi modificare la tua pianificazione in Amazon EventBridge.
7. Per salvare i risultati della qualità dei dati in Amazon S3, scegli una Posizione per i risultati della qualità dei dati. Il ruolo IAM selezionato in precedenza per questa attività deve avere accesso di scrittura a questa posizione.
8. In Configurazioni aggiuntive, inserisci il numero richiesto di lavoratori che AWS Glue deve assegnare alla tua attività di qualità dei dati.
9. Facoltativamente, puoi impostare un filtro nell'origine dati. Questo contribuisce a ridurre i dati in fase di lettura. È inoltre possibile utilizzare un filtro per eseguire convalide incrementali selezionando le informazioni sulle partizioni e trasmettendole come parametri tramite chiamate API. Per migliorare le prestazioni, puoi fornire un predicato di partizione.
10. Scegli Esegui. La nuova attività dovrebbe essere riportata nell'elenco di esecuzioni delle attività relative alla qualità dei dati. Quando la colonna Stato di esecuzione dell'attività è visualizzata come Completata, è possibile visualizzare i risultati del punteggio di qualità. Potrebbe essere necessario aggiornare la finestra della console per visualizzare correttamente lo stato.
11. Per visualizzare la colonna con i dettagli dei risultati sulla qualità dei dati, scegli l'icona "+" per espandere il set di regole. I risultati mostrano le regole che hanno superato e quelle che non hanno superato la valutazione e cosa ha causato l'errore della regola.

## Visualizzazione del punteggio e dei risultati della qualità dei dati

Consultazione dell'ultima esecuzione su tutti i set di regole creati

1. Nella console AWS Glue, scegli Tabelle nel riquadro di navigazione. Scegliere quindi la tabella per la quale si desidera eseguire un'attività di qualità dei dati.
2. Scegli la scheda Qualità dei dati.
3. Snapshot della qualità dei dati mostra una tendenza generale delle esecuzioni nel tempo. Per impostazione predefinita, vengono visualizzate le ultime 10 esecuzioni di tutti i set di regole. Per filtrare per set di regole, seleziona quello desiderato dall'elenco a discesa. Se ci sono meno di 10 esecuzioni, vengono visualizzate tutte le esecuzioni completate disponibili.
4. Nella tabella Qualità dei dati, viene mostrato ogni set di regole con l'ultima esecuzione, se presente, insieme al relativo punteggio. L'espansione del set di regole mostra le regole presenti in quel set di regole insieme ai risultati delle regole per tale esecuzione.

Consultazione dell'ultima esecuzione su un particolare set di regole

1. Nella console AWS Glue, scegli Tabelle nel riquadro di navigazione. Scegliere quindi la tabella per la quale si desidera eseguire un'attività di qualità dei dati.
2. Scegli la scheda Qualità dei dati.
3. Nella tabella Qualità dei dati, scegli un set di regole specifico.
4. Nella pagina Dettagli del set di regole, scegli la scheda Cronologia di esecuzione.

   Tutte le esecuzioni di valutazione per questo particolare set di regole sono elencate nella tabella all'interno di questa scheda. È possibile visualizzare la cronologia dei punteggi e lo stato delle esecuzioni.
5. Per visualizzare ulteriori informazioni su una determinata esecuzione, scegli ID esecuzione per accedere alla pagina Dettagli dell'esecuzione di valutazione. In questa pagina, puoi visualizzare informazioni specifiche sull'esecuzione e ulteriori dettagli sullo stato dei risultati delle singole regole.

## Utilizzo di interrogazioni di preelaborazione

AWS Glue Data Quality supporta le query di preelaborazione che consentono di trasformare i dati prima di eseguire i controlli di qualità dei dati. Questa funzionalità consente di:

- Creare colonne derivate per la convalida della qualità dei dati.
- Filtra i dati in base a condizioni specifiche.
- Esegui calcoli o trasformazioni per i controlli di qualità.
- Convalida le relazioni tra le colonne.

> **Note**
>
> Questa funzionalità è supportata solo APIs e non è supportata tramite la console.

## Utilizzo di query di preelaborazione con CLI e SDK

La raccomandazione viene eseguita

Gli esempi seguenti mostrano come utilizzare le interrogazioni di preelaborazione con esecuzioni di raccomandazione.

AWS CLI:

```
aws glue start-data-quality-rule-recommendation-run \
  --data-source '{"DataQualityGlueTable": { \
    "DatabaseName": "mydatabase", \
    "TableName": "mytable", \
    "PreProcessingQuery": "SELECT sepal_length, sepal_width, petal_length, petal_width,
 class, (sepal_length + sepal_width) as sepal_total FROM `mydatabase.mytable`" \
  }}' \
  --role "arn:aws:iam::123456789012:role/GlueDataQualityRole" \
  --created-ruleset-name "my-ruleset-with-preprocessing"
```

SDK Java:

```
StartDataQualityRuleRecommendationRunRequest request = new
 StartDataQualityRuleRecommendationRunRequest()
    .withDataSource(new DataSource()
        .withDataQualityGlueTable(new DataQualityGlueTable()
            .withDatabaseName("mydatabase")
            .withTableName("mytable")
            .withPreProcessingQuery("SELECT sepal_length, sepal_width, " +
                "(sepal_length + sepal_width) as sepal_total " +
```

```
                "FROM `mydatabase.mytable`")))
    .withRole("arn:aws:iam::123456789012:role/GlueDataQualityRole")
    .withCreatedRulesetName("my-ruleset-with-preprocessing");

glueClient.startDataQualityRuleRecommendationRun(request);
```

La valutazione del set di regole viene eseguita

Gli esempi seguenti mostrano come utilizzare le query di preelaborazione con le esecuzioni di valutazione del set di regole.

AWS CLI:

```
aws glue start-data-quality-ruleset-evaluation-run \
  --data-source '{"DataQualityGlueTable": { \
    "DatabaseName": "mydatabase", \
    "TableName": "mytable", \
    "PreProcessingQuery": "SELECT order_id, amount, (tax + shipping) as total_fees FROM
 `mydatabase.mytable`" \
  }}' \
  --role "arn:aws:iam::123456789012:role/GlueDataQualityRole" \
  --ruleset-names '["my-ruleset"]'
```

SDK Java:

```
StartDataQualityRulesetEvaluationRunRequest request = new
 StartDataQualityRulesetEvaluationRunRequest()
    .withDataSource(new DataSource()
        .withDataQualityGlueTable(new DataQualityGlueTable()
            .withDatabaseName("mydatabase")
            .withTableName("mytable")
            .withPreProcessingQuery("SELECT order_id, amount, " +
                "(tax + shipping) as total_fees " +
                "FROM `mydatabase.mytable`")))
    .withRole("arn:aws:iam::123456789012:role/GlueDataQualityRole")
    .withRulesetNames(Arrays.asList("my-ruleset"));

glueClient.startDataQualityRulesetEvaluationRun(request);
```

## Considerazioni sulla creazione di query di preelaborazione

Quando si scrive una query di preelaborazione:

- Il riferimento alla tabella deve essere formattato `` `databaseName.tableName` `` utilizzando i backtick.
- La query deve essere un'istruzione SELECT valida.
- I nomi delle colonne nell'output della query verranno utilizzati per le regole di qualità dei dati.

L'esempio seguente mostra una query di preelaborazione:

```
SELECT
    sepal_length,
    sepal_width,
    petal_length,
    petal_width,
    class,
    (sepal_length + sepal_width) as sepal_total
FROM `mydatabase.mytable`
```

## Limitazioni

- La query deve fare riferimento alla tabella utilizzando i backtick e il formato completo. `` `databaseName.tableName` ``
- La lunghezza massima della query è di 51.200 caratteri.
- La query deve restituire almeno una riga di dati.
- Tutte le colonne a cui si fa riferimento nel set di regole devono essere presenti nell'output della query.

# Tipi di origini supportate

Supporto dei tipi di tabella in base alla configurazione AWS Lake Formation

| Tipo tabella | AWS Lake Formation - Accesso a tutti i tavoli | AWS Lake Formation Abilitato con colonne | AWS Lake Formation Abilitato con filtri di dati | AWS Lake Formation Supporto multiaccount - Accesso a tutti i tavoli | AWS Lake Formation Disabilitato |
|---|---|---|---|---|---|
| Parquet | Supportata | Non supportato | Non supportato | Supportato | Supportata |
| ORC | Supportata | Non supportato | Non supportato | Supportato | Supportata |
| CSV, JSON, TSV | Supportata | Non supportato | Non supportato | Supportato | Supportata |
| Avro | Supportata | Non supportato | Non supportato | Supportato | Supportata |
| JSON | Supportata | Non supportato | Non supportato | Supportato | Supportata |
| Iceberg | Supportata | Non supportato | Non supportato | Supportato | Supportata |
| HUDI | Non supportato | Non supportato | Non supportato | Non supportato | Supportata |
| Delta | Non supportato | Non supportato | Non supportato | Non supportato | Supportata |
| RMS | Supportato* | Supportato* | Supportato* | Non supportato | Non supportato |
| Tabelle di Amazon S3 | Supportato* | Non supportato | Non supportato | Non applicabile | Supportata |

| Tipo tabella | AWS Lake Formation - Accesso a tutti i tavoli | AWS Lake Formation Abilitato con colonne | AWS Lake Formation Abilitato con filtri di dati | AWS Lake Formation Supporto multiaccount - Accesso a tutti i tavoli | AWS Lake Formation Disabilitato |
|---|---|---|---|---|---|
| Amazon RDS e Aurora | Non applicabile | Non applicabile | Non applicabile | Non applicabile | Non supportato |
| JDBC | Non applicabile | Non applicabile | Non applicabile | Non applicabile | Supportata |

* Le tabelle Amazon S3 e il SageMaker supporto di Lakehouse in AWS Glue Console non sono supportati. Attualmente, le esecuzioni di raccomandazioni di Amazon S3 Table e SageMaker Lakehouse Data Catalog e le esecuzioni di Data Catalog Data Quality Evaluation sono supportate solo tramite la CLI.

### Altri limiti noti

- Tabelle Delta Lake Symlink: non supportate per le esecuzioni di raccomandazione di AWS Glue Data Quality o per le esecuzioni di valutazione della qualità dei dati di Data Catalog.
- Pubblicazione di Amazon S3 Table Asset in SageMaker Unified Studio: attualmente, la pubblicazione delle tabelle Amazon S3 come asset in SageMaker Unified Studio non è disponibile; di conseguenza, la visualizzazione delle esecuzioni di Amazon S3 Table Data Quality non è disponibile da Unified Studio. SageMaker

## Argomenti correlati

- Documentazione di riferimento del tipo di regola DQDL
- Riferimento a Data Quality Definition Language (DQDL)

# Valutazione della qualità dei dati con AWS Glue Studio

AWS Glue Data Quality consente di valutare e monitorare la qualità dei dati in base alle regole definite. In questo modo è facile identificare i dati che richiedono un'azione. In AWS Glue Studio, puoi

aggiungere nodi di qualità dei dati al tuo processo visivo per creare regole di qualità dei dati sulle tabelle del catalogo dati. Potrai quindi monitorare e valutare le modifiche ai set di dati nel corso del tempo. Per una panoramica su come utilizzare Qualità dei dati di AWS Glue in AWS Glue Studio, guarda il seguente video.

Di seguito sono riportati i passaggi di livello superiore per lavorare con Qualità dei dati di AWS Glue:

1. Create data quality rules (Crea regole di qualità dei dati): crea un set di regole di qualità dei dati utilizzando il generatore DQDL scegliendo i set di regole incorporati configurati.
2. Configure a data quality job  (Configura un processo di qualità dei dati): definisci le azioni in base ai risultati della qualità dei dati e alle opzioni di output.
3. Salva ed esegui un processo con la qualità dei dati: crea ed esegui un processo. Il salvataggio del processo salverà i set di regole creati per il processo.
4. Monitor and review the data quality results (Monitora ed esamina i risultati della qualità dei dati): esamina i risultati della qualità dei dati al termine dell'esecuzione del processo. Facoltativamente, pianifica il processo per una data futura.

## Vantaggi

Data analyst, data engineer e data scientist possono utilizzare il nodo di valutazione della qualità dei dati in AWS Glue Studio per analizzare, configurare, monitorare e migliorare la qualità dei dati dall'editor di processi visivi. I vantaggi dell'utilizzo del nodo di qualità dei dati includono i seguenti:

- È possibile rilevare problemi di qualità dei dati: puoi verificare la presenza di problemi creando regole che controllano le funzionalità dei set di dati.
- Iniziare è facile: puoi iniziare utilizzando regole e operazioni predefinite.
- Integrazione perfetta: è possibile utilizzare i nodi di qualità dei dati in AWS Glue Studio perché Qualità dei dati di AWS Glue viene eseguito su Catalogo dati AWS Glue.

## Valutazione della qualità dei dati per i processi ETL in AWS Glue Studio

In questo tutorial, si inizierà a usare Qualità dei dati di AWS Glue in AWS Glue Studio. Si imparerà come:

- Creare regole utilizzando il generatore di regole Data Quality Definition Language (DQDL).

- Specificare le azioni di qualità dei dati, i dati da emettere e la posizione di output dei risultati della qualità dei dati.
- Esaminare i risultati della qualità dei dati.

Per fare pratica con un esempio, consulta il post sul blog Getting started with AWS Glue Data Quality for ETL pipelines.

## Passaggio 1: aggiunta del nodo Valuta la qualità dei dati al processo visivo

In questo passaggio, verrà aggiunto il nodo di valutazione della qualità dei dati all'editor del processo visivo.

Aggiunta del nodo di qualità dei dati

1. Nella console AWS Glue Studio, scegli Visual con un'origine e una destinazione dalla sezione Crea lavoro, quindi scegli Crea.
2. Scegliere un nodo al quale desideri applicare la trasformazione della qualità dei dati. In genere, si tratta di un nodo di trasformazione o di un'origine dati.
3. Aprire il pannello delle risorse a sinistra scegliendo l'icona "+". È inoltre possibile digitare Valuta la qualità dei dati nella barra di ricerca e quindi scegliere Valuta la qualità dei dati dai risultati della ricerca.
4. L'editor del processo visivo mostrerà il nodo di trasformazione Valuta la qualità dei dati che si dirama dal nodo selezionato. Sul lato destro della console, la scheda Transform (Trasforma) è aperta automaticamente. Se è necessario modificare il nodo padre, scegliere la scheda Proprietà del nodo, quindi scegliere il nodo padre dal menu a discesa.

   Quando si sceglie un nuovo nodo principale, viene stabilita una nuova connessione tra il nodo principale e il nodo Evaluate Data Quality (Valuta la qualità dei dati). Rimuovere tutti i nodi principali indesiderati. È possibile collegare un solo nodo principale a un nodo Evaluate Data Quality (Valuta la qualità dei dati).
5. La trasformazione Valuta la qualità dei dati supporta più padri per consentire di convalidare le regole di qualità dei dati su più set di dati. Le regole che supportano più set di dati includono ReferentialIntegrity DatasetMatch,, SchemaMatch RowCountMatch, e AggregateMatch.

   Se si aggiungono più input alla trasformazione Valuta la qualità dei dati, è necessario selezionare l'input "primario". L'input primario è il set di dati del quale desideri convalidare la qualità dei dati. Tutti gli altri nodi o input vengono trattati come riferimenti.

È possibile utilizzare la trasformazione Valuta la qualità dei dati per identificare record specifici che non hanno superato i controlli di qualità dei dati. Ti consigliamo di scegliere il set di dati primario perché le nuove colonne che segnalano i record non validi vengono aggiunte a tale set di dati.

6. È possibile specificare degli alias per le origini dati di input. Gli alias forniscono un altro modo per fare riferimento alla sorgente di input quando si utilizza la ReferentialIntegrity regola. Poiché è possibile designare una sola origine dati come origine principale, ogni ulteriore origine dati che aggiungi richiederà un alias.

   Nell'esempio seguente, la ReferentialIntegrity regola specifica l'origine dati di input tramite il nome dell'alias ed esegue un one-to-one confronto con l'origine dati principale.

   ```
   Rules = [
    ReferentialIntegrity “Aliasname.name” = 1
   ]
   ```

## Fase 2: creazione di una regola con DQDL

In questa fase, viene creata una regola tramite DQDL. Per questo tutorial, verrà creata una singola regola utilizzando il tipo di regola Completezza. Questo tipo di regola verifica la percentuale di valori completi (non nulli) in una colonna rispetto a una determinata espressione. Per ulteriori informazioni sull'utilizzo di DQDL, consultare la pagina DQDL.

1. Nella scheda Trasforma, aggiungere un Tipo di regola facendo clic sul pulsante Inserisci. Questa operazione aggiunge il tipo di regola all'editor di regole, nel quale è possibile inserire i parametri per la regola.

   > **Note**
   >
   > Quando modifichi le regole, assicurarsi che le regole siano racchiuse tra parentesi e che siano separate da virgole. Ad esempio, un'espressione di regola completa avrà il seguente aspetto:
   >
   > ```
   > Rules= [
   >     Completeness "year">0.8, Completeness "month">0.8
   > ]
   > ```

Questo esempio specifica il parametro di completezza per le colonne denominate "anno" e "mese". Affinché la regola venga soddisfatta, queste colonne devono essere complete per più dell'80% o devono contenere dati in oltre l'80% delle istanze per ogni rispettiva colonna.

In questo esempio, cercare e inserire il tipo di regola Completezza. Questa operazione aggiunge il tipo di regola all'editor di regole. Questo tipo di regola ha la seguente sintassi: `Completeness <COL_NAME> <EXPRESSION>`.

La maggior parte dei tipi di regole richiede la specifica di un'espressione come parametro al fine di creare una risposta booleana. Per ulteriori informazioni sulle espressioni DQDL supportate, consultare la pagina DQDL expressions. Successivamente, aggiungere il nome della colonna.

2. Nel generatore di regole DQDL, selezionare la scheda Schema. Usare la barra di ricerca per individuare il nome della colonna nello schema di input. Lo schema di input visualizza il nome della colonna e il tipo di dati.
3. Nell'editor di regole, fare clic sulla destra del tipo di regola per inserire il cursore nel punto in cui verrà inserita la colonna. In alternativa, è possibile digitare il nome della colonna nella regola.

   Ad esempio, dall'elenco di colonne nell'elenco dello schema di input, fai clic sul pulsante Inserisci accanto alla colonna (in questo esempio, anno). Questa operazione aggiunge la colonna alla regola.
4. Quindi, nell'editor di regole, aggiungere un'espressione per valutare la regola. Poiché il tipo Completezza verifica la percentuale di valori completi (non nulli) in una colonna rispetto a una determinata espressione, immettere un'espressione come `> 0.8`. Questa regola controlla se la colonna contiene almeno l'80% di valori completi (non nulli).

## Passaggio 3: configurazione degli output di qualità dei dati

Dopo aver creato le regole di qualità dei dati, è possibile selezionare opzioni aggiuntive per specificare l'output del nodo della qualità dei dati.

1. In Output della trasformazione della qualità dei dati, scegliere tra le seguenti opzioni:

- Dati originali: scegliere di emettere i dati di input originali. Quando si sceglie questa opzione, al job viene aggiunto un nuovo nodo figlio «rowLevelOutcomes». Lo schema corrisponde allo schema del set di dati primario trasmesso come input alla trasformazione. Questa opzione è utile se si desidera soltanto trasmettere i dati e far sì che il processo abbia esito negativo se si verificano problemi di qualità.

  Un altro caso d'uso è quando si desidera rilevare record non validi che non hanno superato i controlli di qualità dei dati. Per rilevare i record non validi, scegliere l'opzione Aggiungi nuove colonne per indicare gli errori di qualità dei dati. Questa azione aggiunge quattro nuove colonne allo schema della trasformazione «rowLevelOutcomes».

  - DataQualityRulesPass(array di stringhe): fornisce una serie di regole che hanno superato i controlli di qualità dei dati.
  - DataQualityRulesFail(array di stringhe) — Fornisce una serie di regole che non hanno superato i controlli di qualità dei dati.
  - DataQualityRulesSkip(array di stringhe) — Fornisce una serie di regole che sono state ignorate. Le seguenti regole non possono identificare i record di errore perché vengono applicate a livello di set di dati.
    - AggregateMatch
    - ColumnCount
    - ColumnExists
    - ColumnNamesMatchPattern
    - CustomSql
    - RowCount
    - RowCountMatch
    - StandardDeviation
    - Media
    - ColumnCorrelation
  - DataQualityEvaluationResult— Fornisce lo stato «Passato» o «Non riuscito» a livello di riga. Tenere presente che i risultati complessivi possono essere non riusciti, ma un determinato record potrebbe essere riuscito. Ad esempio, la RowCount regola potrebbe non essere riuscita, ma tutte le altre regole potrebbero aver avuto successo. In questi casi, lo stato di questo campo è “Riuscito”.

2. Risultati della qualità dei dati: scegliere di visualizzare le regole configurate e il loro stato di riuscita o non riuscita. Questa opzione è utile se desideri scrivere i risultati su Amazon S3 o altri database.
3. Impostazioni di output della qualità dei dati (opzionale): scegliere Impostazioni di output della qualità dei dati per visualizzare il campo Posizione dei risultati della qualità dei dati. Quindi, fare clic su Sfoglia per cercare una posizione Amazon S3 da impostare come destinazione dell'output della qualità dei dati.

## Passaggio 4. Configurazione delle operazioni di qualità dei dati

Puoi utilizzare le azioni per pubblicare metriche CloudWatch o interrompere i lavori in base a criteri specifici. Le operazioni sono disponibili solo dopo aver creato una regola. Se si sceglie questa opzione, gli stessi parametri vengono pubblicati anche su Amazon EventBridge. È possibile utilizzare queste opzioni per creare avvisi di notifica.

- In caso di errore del set di regole: è possibile scegliere cosa fare se un set di regole ha esito negativo mentre il processo è in esecuzione. Se si desidera che il processo abbia esito negativo se la qualità dei dati non va a buon fine, è possibile scegliere quando far fallire il processo selezionando una delle seguenti opzioni. Per impostazione predefinita, questa operazione non è selezionata e l'esecuzione del processo sarà completata anche se le regole di qualità dei dati hanno esito negativo.
  - Nessuno: se si sceglie Nessuno (impostazione predefinita), il processo non ha esito negativo e continua a essere eseguito nonostante gli errori del set di regole.
  - Abbandonare il processo dopo il caricamento dei dati sulla destinazione: il processo ha esito negativo e non viene salvato alcun dato. Per salvare i risultati, scegliere una posizione Amazon S3 in cui salvare i risultati sulla qualità dei dati.
  - Abbandonare il processo senza caricare i dati sulla destinazione: questa opzione determina immediatamente l'esito negativo del processo quando si verifica un errore di qualità dei dati. Non carica alcuna destinazione dati, inclusi i risultati della trasformazione di qualità dei dati.

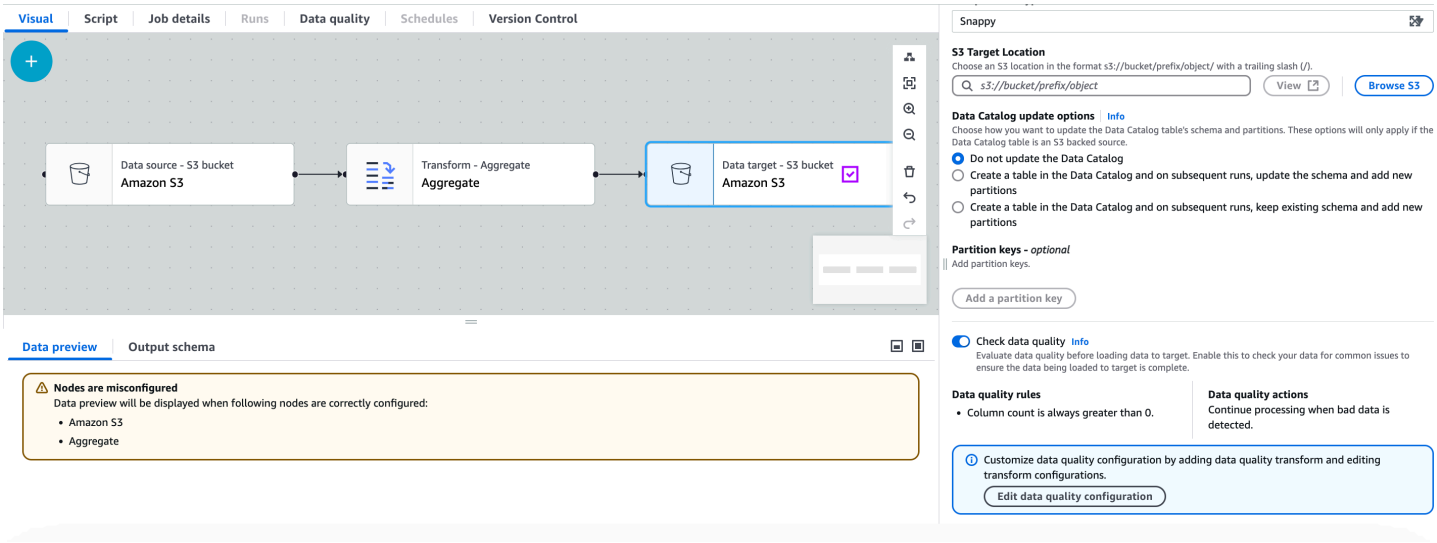
## Passaggio 5: visualizzazione dei risultati della qualità dei dati

Dopo aver eseguito il processo, visualizzare i risultati relativi alla qualità dei dati facendo clic sulla scheda Qualità dei dati.

1. Per ogni esecuzione di processo, visualizzare i risultati della qualità dei dati. Ogni nodo mostra lo stato della qualità dei dati e i dettagli dello stato. Scegliere un nodo per visualizzare tutte le regole e lo stato di ciascuna regola.
2. Fare clic su Scarica risultati per scaricare un file CSV contenente informazioni sull'esecuzione del processo e sui risultati relativi alla qualità dei dati.
3. Se si hanno più di una esecuzione di processo con risultati di qualità dei dati, è possibile filtrare i risultati per intervallo di data e ora. Scegliere Filtra per intervallo di data e ora per espandere la finestra del filtro.
4. È possibile scegliere un intervallo relativo o un intervallo assoluto. Per gli intervalli assoluti, utilizzare il calendario per selezionare i valori di data e ora per l'ora di inizio e l'ora di fine. Al termine, scegliere Applica.

## Qualità dei dati automatica

Quando crei un lavoro AWS Glue ETL con Amazon S3 come destinazione, AWS Glue ETL abilita automaticamente una regola di qualità dei dati che verifica se i dati caricati hanno almeno una colonna. Questa regola è progettata per garantire che i dati caricati non siano vuoti o danneggiati. Tuttavia, se questa regola ha esito negativo, il processo non avrà esito negativo; si noterà invece una riduzione del punteggio della qualità dei dati. Inoltre, per impostazione predefinita, è abilitato il rilevamento delle anomalie, che monitora il numero di colonne nei dati. Se ci sono variazioni o anomalie nel conteggio delle colonne, AWS Glue ETL ti informerà di queste anomalie. Questa funzione aiuta a identificare potenziali problemi con i dati e ad adottare le azioni appropriate. Per visualizzare la regola Data Quality e la relativa configurazione, puoi fare clic sulla destinazione Amazon S3 nel job AWS Glue ETL. La configurazione della regola verrà visualizzata, come mostrato nella seguente schermata.




È possibile aggiungere ulteriori regole sulla qualità dei dati selezionando Modifica configurazione della qualità dei dati.

## Parametri aggregati

Per creare dashboard, potrebbero essere necessari parametri aggregati come il numero di record passati, non riusciti, ignorati a livello di regola o a livello di set di regole. Per ottenere i parametri aggregati e i parametri delle regole per ciascuna regola, abilitare innanzitutto i Parametri aggregati aggiungendo l'opzione `publishAggregatedMetrics` alla funzione `EvaluateDataQuality`.

Le opzioni possibili per `additional_options` `publishAggregatedMetrics` sono `ENABLED` e `DISABLED`. Ad esempio:

```
EvaluateDataQualityMultiframe = EvaluateDataQuality().process_rows(
    frame=medicare_dyf,
    ruleset=EvaluateDataQuality_ruleset,
    publishing_options={
        "dataQualityEvaluationContext": "EvaluateDataQualityMultiframe",
        "enableDataQualityCloudWatchMetrics": False,
        "enableDataQualityResultsPublishing": False,
    },
    additional_options={"publishAggregatedMetrics.status": "ENABLED"},
)
```

Se non specificato, il `publishAggregatedMetrics.status` è `DISABLED` per impostazione predefinita e i parametri delle regole e i parametri aggregati verranno ora calcolati. Questa

funzionalità è attualmente supportata nelle AWS Glue Interactive Sessions e nei job Glue ETL. Questa funzionalità non è supportata in Glue Catalog Data Quality APIs.

Recupero dei risultati dei parametri aggregati

Quando `additionalOptions` `"publishAggregatedMetrics.status": "ENABLED"`, è possibile ottenere i risultati in due punti:

1. `AggregatedMetrics` e `RuleMetrics` vengono restituiti tramite `GetDataQualityResult()` quando si fornisce il `resultId` dove `AggregatedMetrics` e `RuleMetrics` includono:

   Parametri aggregati:

   - Righe totali elaborate
   - Righe totali riuscite
   - Righe totali non riuscite
   - Totale regole elaborate
   - Totale regole riuscite
   - Totale regole non riuscite

```
[redacted] % aws --region us-east-1 glue get-data-quality-result --result-id "dqresult-09655f0234b3e3d129381bd162d0347ed25d1c05"
{
    "ResultId": "dqresult-09655f0234b3e3d129381bd162d0347ed25d1c05",
    "ProfileId": "dqprofile-[redacted]",
    "Score": 0.6666666666666666,
    "RulesetName": "EvaluateDataQuality_node1741974822533",
    "EvaluationContext": "EvaluateDataQuality_node1741974822533",
    "StartedOn": 1750687515.374,
    "CompletedOn": 1750687537.485,
    "JobName": "[redacted]_aggregatedmetrics_test",
    "JobRunId": "jr_[redacted]",
    "RuleResults": [
        {
            "Name": "Rule_1",
            "Description": "IsUnique \"customer_identifier\"",
            "Result": "PASS",
            "EvaluatedMetrics": {
                "Column.customer_identifier.Uniqueness": 1.0
            },
            "EvaluatedRule": "IsUnique \"customer_identifier\"",
            "RuleMetrics": {
                "rows_passed": 10.0,
                "rows_failed": 0.0,
                "rows_skipped": 0.0,
                "total_rows": 10.0
            }
        },
        {
            "Name": "Rule_2",
            "Description": "RowCount > 10",
            "EvaluationMessage": "Value: 10 does not meet the constraint requirement!",
            "Result": "FAIL",
            "EvaluatedMetrics": {
                "Dataset.*.RowCount": 10.0
            },
            "EvaluatedRule": "RowCount > 10",
            "RuleMetrics": {
                "rows_passed": 0.0,
                "rows_failed": 0.0,
                "rows_skipped": 10.0,
                "total_rows": 10.0
            }
        },
        {
            "Name": "Rule_3",
            "Description": "Completeness \"customer_identifier\" > 0.5",
            "Result": "PASS",
            "EvaluatedMetrics": {
                "Column.customer_identifier.Completeness": 1.0
            },
            "EvaluatedRule": "Completeness \"customer_identifier\" > 0.5",
            "RuleMetrics": {
                "rows_passed": 10.0,
                "rows_failed": 0.0,
                "rows_skipped": 0.0,
                "total_rows": 10.0
            }
        }
    ],
    "AggregatedMetrics": {
        "TotalRowsProcessed": 10.0,
        "TotalRowsPassed": 10.0,
        "TotalRowsFailed": 0.0,
        "TotalRulesProcessed": 3.0,
        "TotalRulesPassed": 2.0,
        "TotalRulesFailed": 1.0
    }
}
```

Inoltre, a livello di regola, vengono forniti seguenti parametri:

Parametri delle regole:

- Righe riuscite
- Righe non riuscite
- Righe ignorate
- Righe totali elaborate

2. `AggregatedMetrics` viene restituito come frame di dati aggiuntivo e il frame di dati `RuleOutcomes` viene aumentato per includere `RuleMetrics`.

### Implementazione esemplificativa

L'esempio seguente mostra come implementare i parametri aggregati in Scala:

```
// Script generated for node Evaluate Data Quality
val EvaluateDataQuality_node1741974822533_ruleset = """
  # Example rules: Completeness "colA" between 0.4 and 0.8, ColumnCount > 10
  Rules = [
      IsUnique "customer_identifier",
      RowCount > 10,
      Completeness "customer_identifier" > 0.5
  ]
"""

val EvaluateDataQuality_node1741974822533 =
 EvaluateDataQuality.processRows(frame=ChangeSchema_node1742850392012,
 ruleset=EvaluateDataQuality_node1741974822533_ruleset,
 publishingOptions=JsonOptions("""{"dataQualityEvaluationContext":
 "EvaluateDataQuality_node1741974822533", "enableDataQualityCloudWatchMetrics":
 "true", "enableDataQualityResultsPublishing": "true"}"""),
 additionalOptions=JsonOptions("""{"compositeRuleEvaluation.method":"ROW","observations.scope":
 "publishAggregatedMetrics.status": "ENABLED"}""" ))

println("--------------------------------ROW LEVEL
 OUTCOMES--------------------------------")
val rowLevelOutcomes_node = EvaluateDataQuality_node1741974822533("rowLevelOutcomes")

rowLevelOutcomes_node.show(10)

 println("--------------------------------RULE LEVEL
 OUTCOMES--------------------------------")

val ruleOutcomes_node = EvaluateDataQuality_node1741974822533("ruleOutcomes")

ruleOutcomes_node.show()

 println("--------------------------------AGGREGATED
 METRICS--------------------------------")

val aggregatedMetrics_node = EvaluateDataQuality_node1741974822533("aggregatedMetrics")
```

```
aggregatedMetrics_node.show()
```

Risultati di esempio

I risultati vengono restituiti come segue:


```
{
    "Rule": "IsUnique \"customer_identifier\"",
    "Outcome": "Passed",
    "FailureReason": null,
    "EvaluatedMetrics": {
        "Column.customer_identifier.Uniqueness": 1
    },
    "EvaluatedRule": "IsUnique \"customer_identifier\"",
    "PassedCount": 10,
    "FailedCount": 0,
    "SkippedCount": 0,
    "TotalCount": 10
}
{
    "Rule": "RowCount > 10",
    "Outcome": "Failed",
    "FailureReason": "Value: 10 does not meet the constraint requirement!",
    "EvaluatedMetrics": {
        "Dataset.*.RowCount": 10
    },
    "EvaluatedRule": "RowCount > 10",
    "PassedCount": 0,
    "FailedCount": 0,
    "SkippedCount": 10,
    "TotalCount": 10
}
{
    "Rule": "Completeness \"customer_identifier\" > 0.5",
    "Outcome": "Passed",
    "FailureReason": null,
    "EvaluatedMetrics": {
        "Column.customer_identifier.Completeness": 1
    },
    "EvaluatedRule": "Completeness \"customer_identifier\" > 0.5",
    "PassedCount": 10,
    "FailedCount": 0,
    "SkippedCount": 0,
```

```
    "TotalCount": 10
}
```

I parametri aggregati sono i seguenti:

```
{ "TotalRowsProcessed": 10, "PassedRows": 10, "FailedRows": 0, "TotalRulesProcessed":
 3, "RulesPassed": 2, "RulesFailed": 1 }
```

## Generatore di regole di qualità dei dati

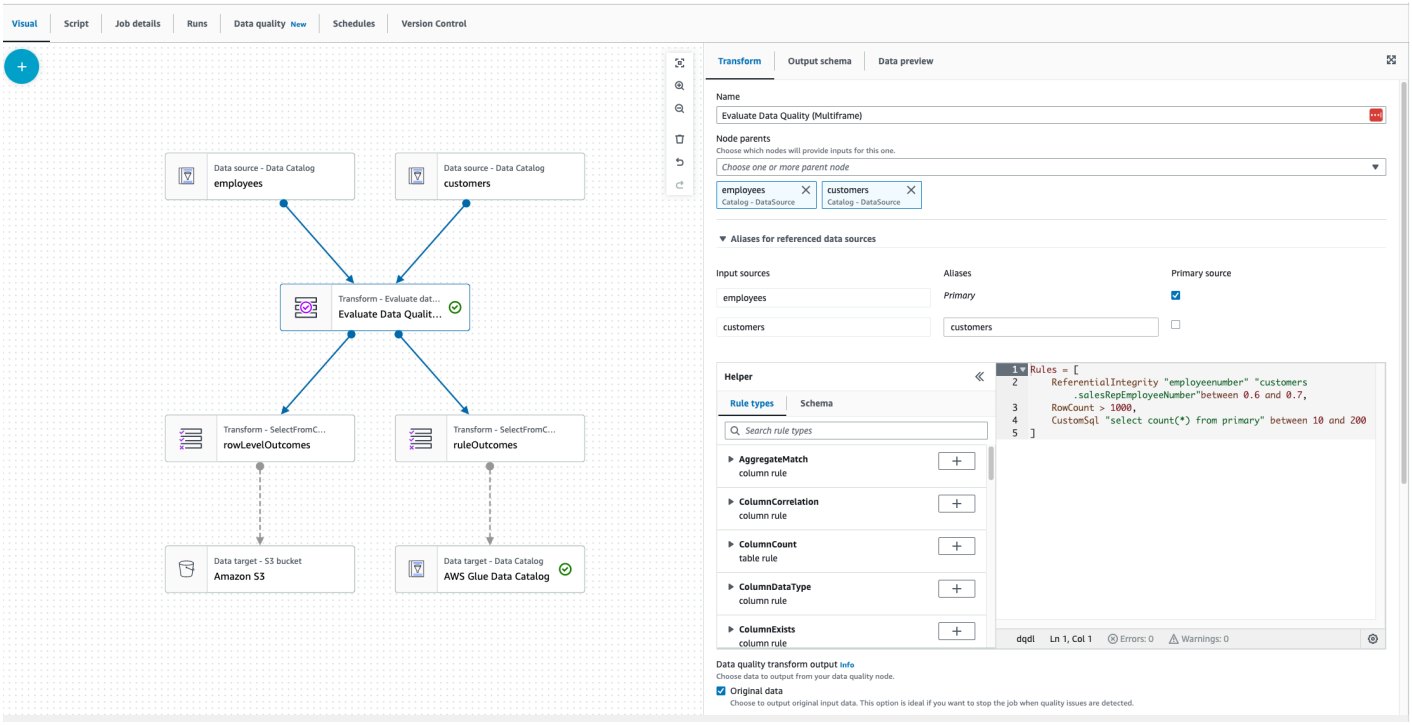
Con il generatore di regole Data Quality Definition Language (DQDL), puoi creare regole di qualità dei dati per valutare i tuoi dati. Inizia selezionando un tipo di regola, quindi specifica i parametri nell'editor delle regole. Durante il processo di creazione, l'editor delle regole mostra anche eventuali errori e avvisi.

La Guida di DQDL fornisce una documentazione completa su come costruire regole utilizzando la sintassi DQDL, i tipi di regole integrati e gli esempi.

### Nodo Evaluate Data Quality (Valuta la qualità dei dati)

Quando si lavora con il nodo di trasformazione Valuta la qualità dei dati e il generatore di regole DQDL, è possibile espandere lo spazio di lavoro.

- Per espandere la scheda Trasforma fino a riempire l'intero schermo, fai clic sull'icona di espansione nell'angolo in alto a destra del pannello dei dettagli del nodo.
- Per espandere l'editor delle regole DQDL, fai clic sull'icona << per espandere l'editor delle regole e comprimere le schede Tipi di regole e Schema.




## Componenti

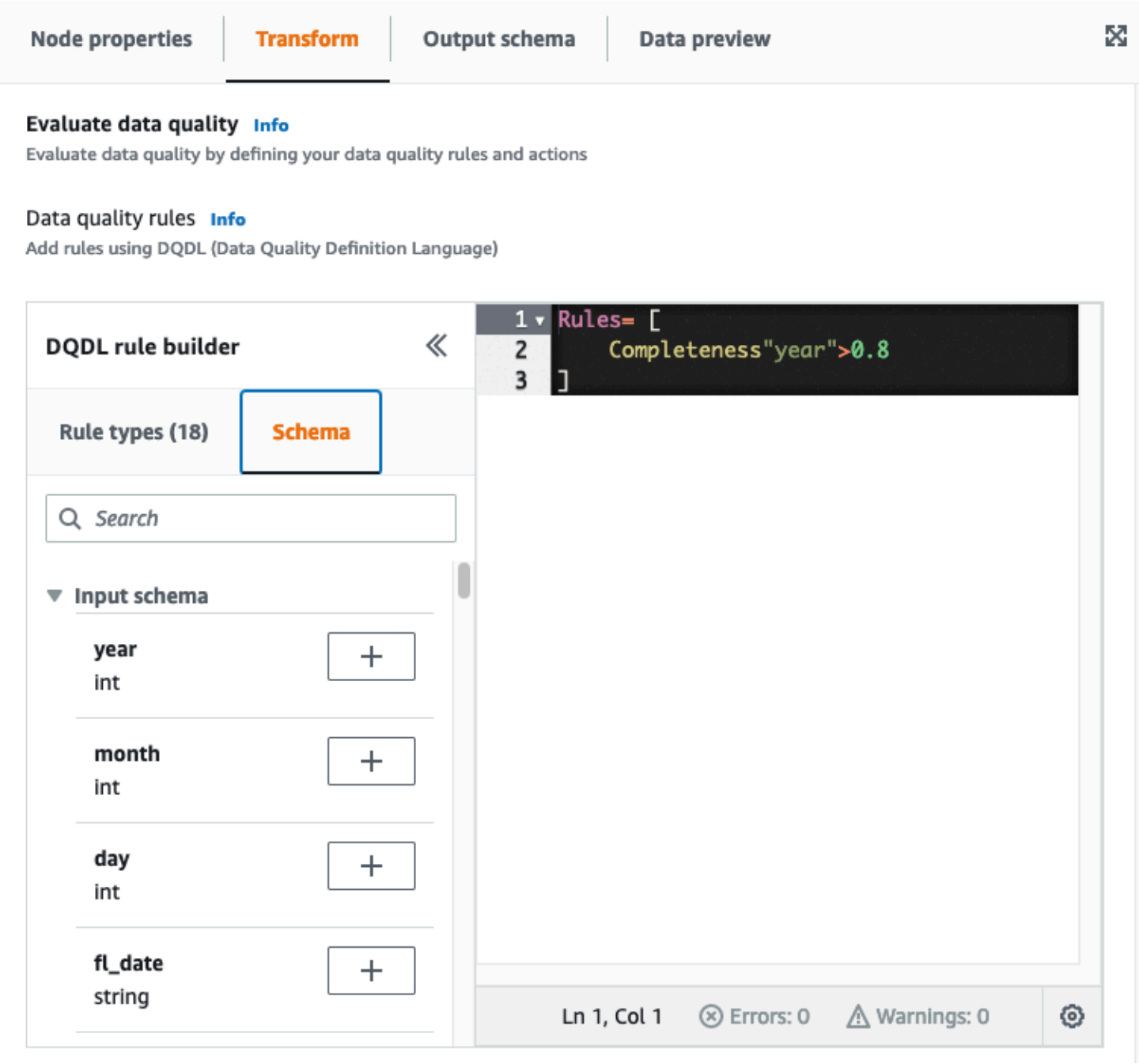
Esistono 26 tipi di regole incorporati in AWS Glue Studio. Ogni tipo di regola riporta una descrizione e vari esempi di come possono essere utilizzati.

### Tipi di regole di qualità dei dati

AWS Glue Studio fornisce tipi di regole integrati per semplificare la creazione di una regola. Per ulteriori informazioni sui tipi di regole, consulta Riferimento ai tipi di regole DQDL.

### Schema

La scheda Schema mostra i nomi delle colonne e il tipo di dati del nodo principale. Vengono visualizzati gli schemi di più nodi. È possibile visualizzare lo schema di input, effettuare una ricerca per nome della colonna e inserire la colonna nell'editor delle regole.



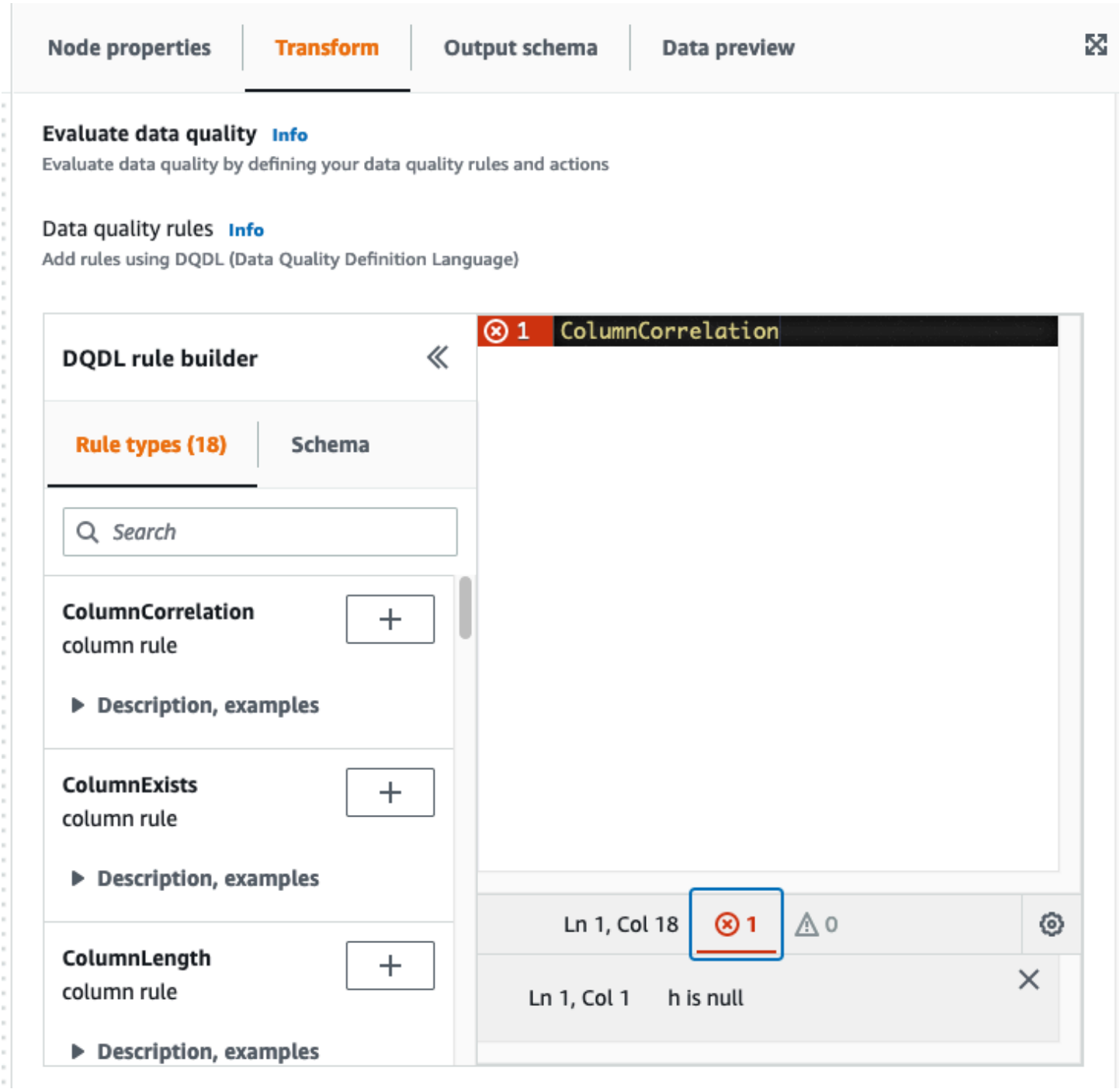

Editor delle regole

L'editor delle regole è un editor di testo in cui è possibile scrivere e modificare le regole. Se si seleziona un tipo di regola dal generatore di regole DQDL, questo verrà aggiunto all'editor delle regole. È quindi possibile specificare parametri, aggiungere e modificare le regole in base alle esigenze modificando il testo. AWS Glue Studio convalida le regole nell'editor delle regole e mostra errori e avvisi, se presenti.

## Errori e avvertenze

Se una regola non segue la sintassi delle regole DQDL, l'editor delle regole mostra diversi indicatori visivi che segnalano la presenza di un errore:

- L'editor delle regole mostra un'icona di errore e un colore rosso sulla riga con l'errore.
- Il numero di errori viene mostrato accanto all'icona rossa di errore.
- Quando scegli la riga con l'errore, nella parte inferiore dell'editor delle regole vengono mostrate una descrizione e la sua posizione (riga e colonna).




Operazioni di qualità dei dati

Per impostazione predefinita, questa operazione non è selezionata e l'esecuzione del processo sarà completata anche se le regole di qualità dei dati hanno esito negativo.

Scegli tra le seguenti operazioni. È possibile utilizzare le azioni per pubblicare risultati CloudWatch o interrompere i lavori in base a criteri specifici. Le operazioni sono disponibili solo dopo aver creato una regola.

- Pubblica risultati su CloudWatch: quando esegui un lavoro, aggiungi i risultati a CloudWatch.
- Processo fallito quando la qualità dei dati fallisce: se le regole sulla qualità dei dati falliscono, anche il processo fallisce di conseguenza.

Output di trasformazione della qualità dei dati

- Dati originali: scegli di emettere i dati di input originali. Questa opzione è ideale se si desidera interrompere il processo quando vengono rilevati problemi di qualità.
- Risultati della qualità dei dati: scegli di visualizzare le regole configurate per l'output e il loro stato di riuscita o non riuscita. Questa opzione è utile se desideri eseguire un'operazione personalizzata.

Impostazioni di output della qualità dei dati

Imposta la posizione dei risultati della qualità dei dati specificando la posizione Amazon S3 come destinazione dell'output della qualità dei dati.

## Configurazione del rilevamento delle anomalie nei job AWS Glue ETL

Per iniziare con il rilevamento delle anomalie in AWS Glue Studio, apri un job di AWS Glue Studio e fai clic su Evaluate Data Quality Transform.

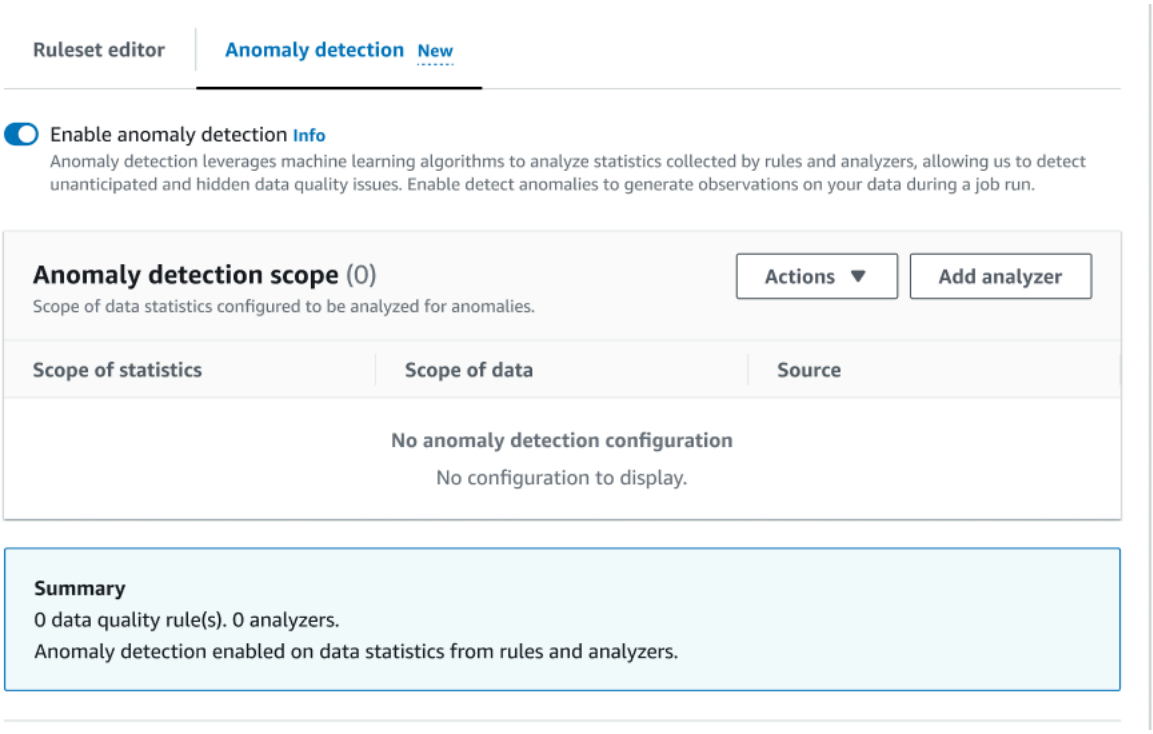
Abilitando questa funzionalità, AWS Glue Data Quality analizzerà i dati nel tempo per rilevare anomalie. Fornisce statistiche e osservazioni preziose sui dati, consentendo di intervenire sulle eventuali anomalie identificate.

Consulta la documentazione relativa al rilevamento delle anomalie per capire il funzionamento interno di questa funzionalità.

### Abilitazione del rilevamento delle anomalie

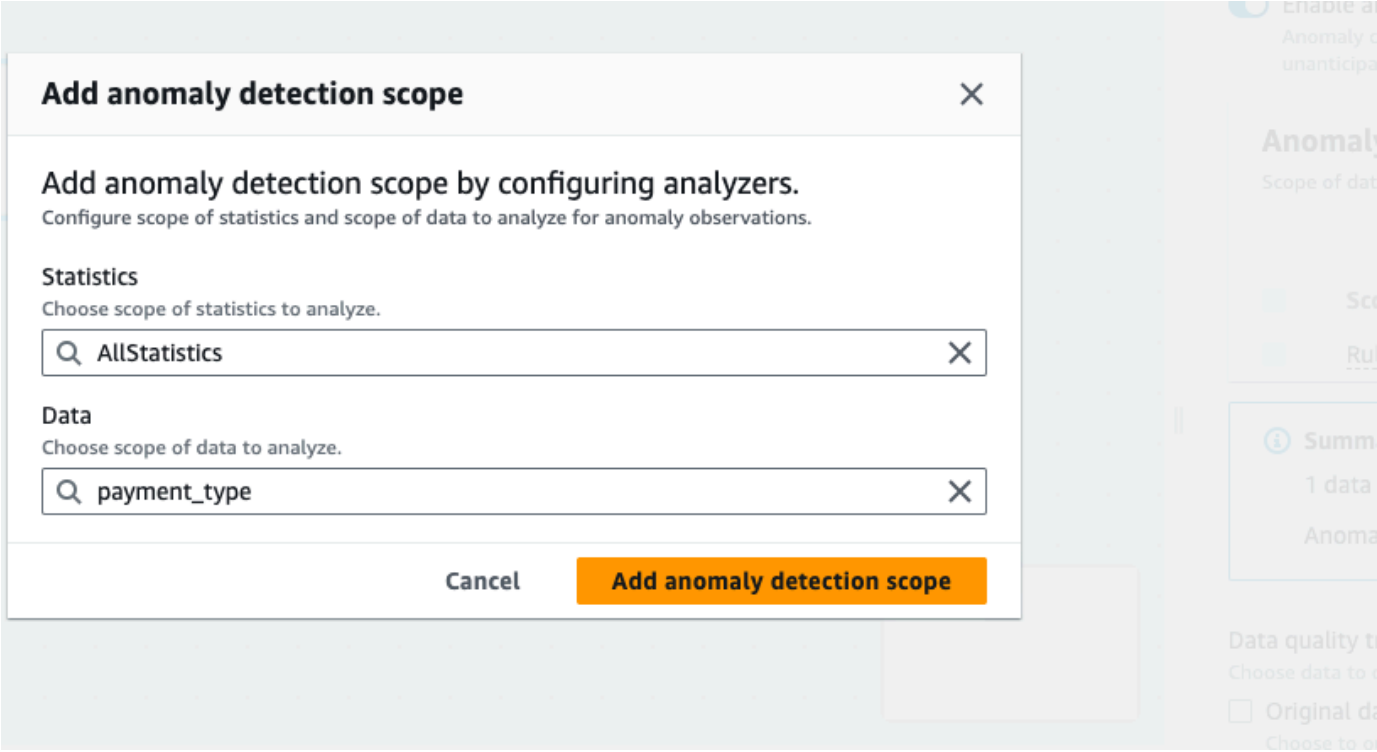
Per abilitare il rilevamento delle anomalie in AWS Glue Studio:

1. Scegli il nodo Qualità dei dati nel processo, quindi scegli la scheda Rilevamento delle anomalie. Attiva l'opzione Abilita rilevamento anomalie.




2. Definisci i dati da monitorare per rilevare eventuali anomalie scegliendo Aggiungi analizzatore. È possibile compilare due campi: Statistiche e Dati.

   - Le statistiche corrispondono alle informazioni sulla forma e su altre proprietà dei dati. È possibile scegliere una o più statistiche per volta oppure scegliere Tutte le statistiche. Le statistiche includono: completezza, unicità, media, somma StandardDeviation, entropia e altro. DistinctValuesCount UniqueValueRatio Per ulteriori dettagli, consulta la documentazione relativa agli analizzatori.
   - I dati corrispondono alle colonne del set di dati. Puoi scegliere colonne singole oppure sceglierle tutte.




3. Scegli Aggiungi ambito di rilevamento delle anomalie per salvare le modifiche. Dopo aver aggiunto gli analizzatori, puoi visualizzarli nella sezione Ambito di rilevamento delle anomalie.

   Puoi anche utilizzare il menu Operazioni per modificare gli analizzatori oppure scegliere la scheda Editor del set di regole e modificare l'analizzatore direttamente nel blocco note dell'editor del set di regole. Visualizzerai gli analizzatori che hai salvato sotto le regole che hai creato.

```
Rules = [

]

Analyzers = [
    Completeness “id”
]
```

Una volta configurati il set di regole e gli analizzatori aggiornati, AWS Glue Data Quality monitora continuamente i flussi di dati in entrata. Può segnalare potenziali anomalie tramite avvisi o interruzioni dei processi, a seconda delle impostazioni configurate. Questo monitoraggio proattivo contribuisce a garantire la qualità e l'integrità dei dati in tutte le pipeline di dati.

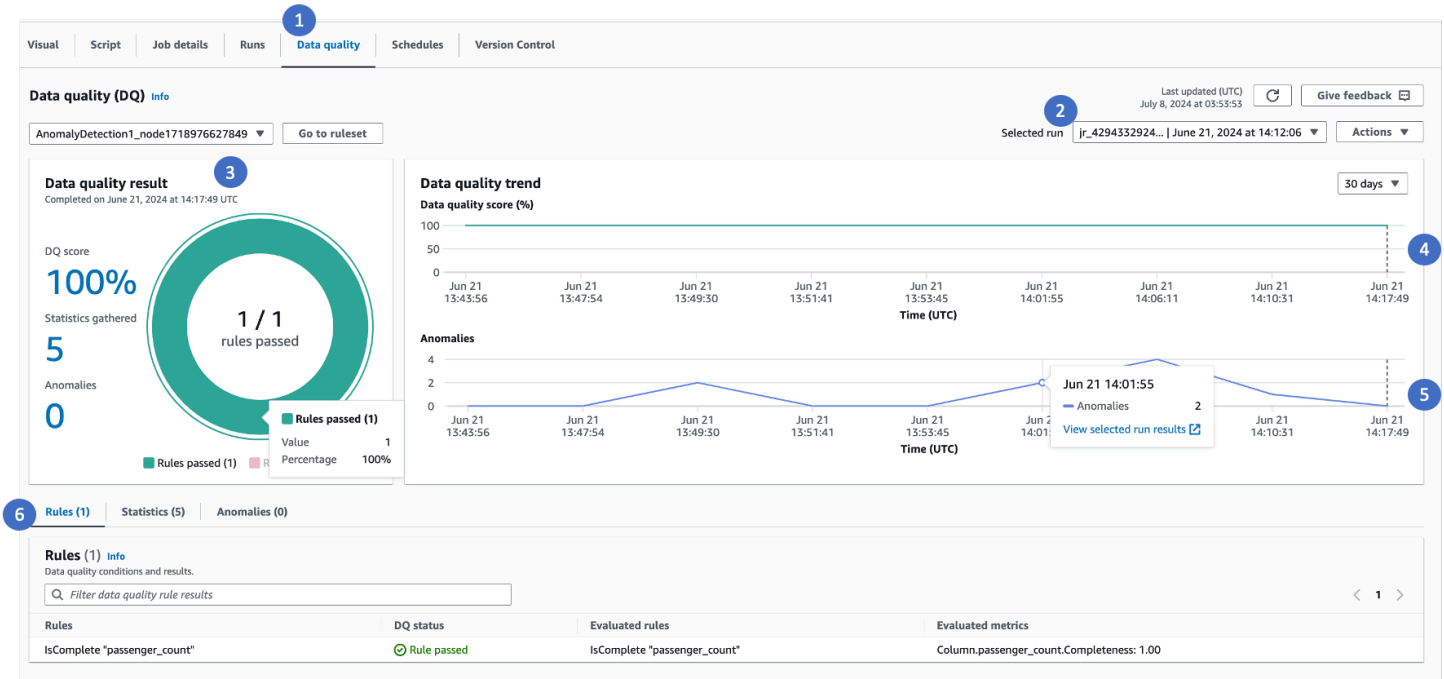
La prossima sezione mostra come monitorare in modo efficace le anomalie individuate dal sistema. Imparerai anche come visualizzare e analizzare le statistiche dei dati raccolte da AWS Glue Data Quality. Infine, mostra come fornire feedback al modello di machine learning che alimenta la funzionalità di rilevamento delle anomalie. Questo ciclo di feedback è essenziale per migliorare l'accuratezza del modello e garantire che sia in grado di rilevare efficacemente le anomalie in linea con i requisiti aziendali e i modelli di dati specifici.

## Visualizzazione dei punteggi e delle anomalie relativi alla qualità dei dati

Questa sezione illustra il pannello di controllo relativo alla qualità dei dati e le diverse funzionalità che offre.

### Visualizzare e comprendere metriche e tendenze di alto livello sulla qualità dei dati

Una volta completato il processo, seleziona la scheda Data Quality per visualizzare i punteggi relativi alla qualità dei dati e le anomalie.



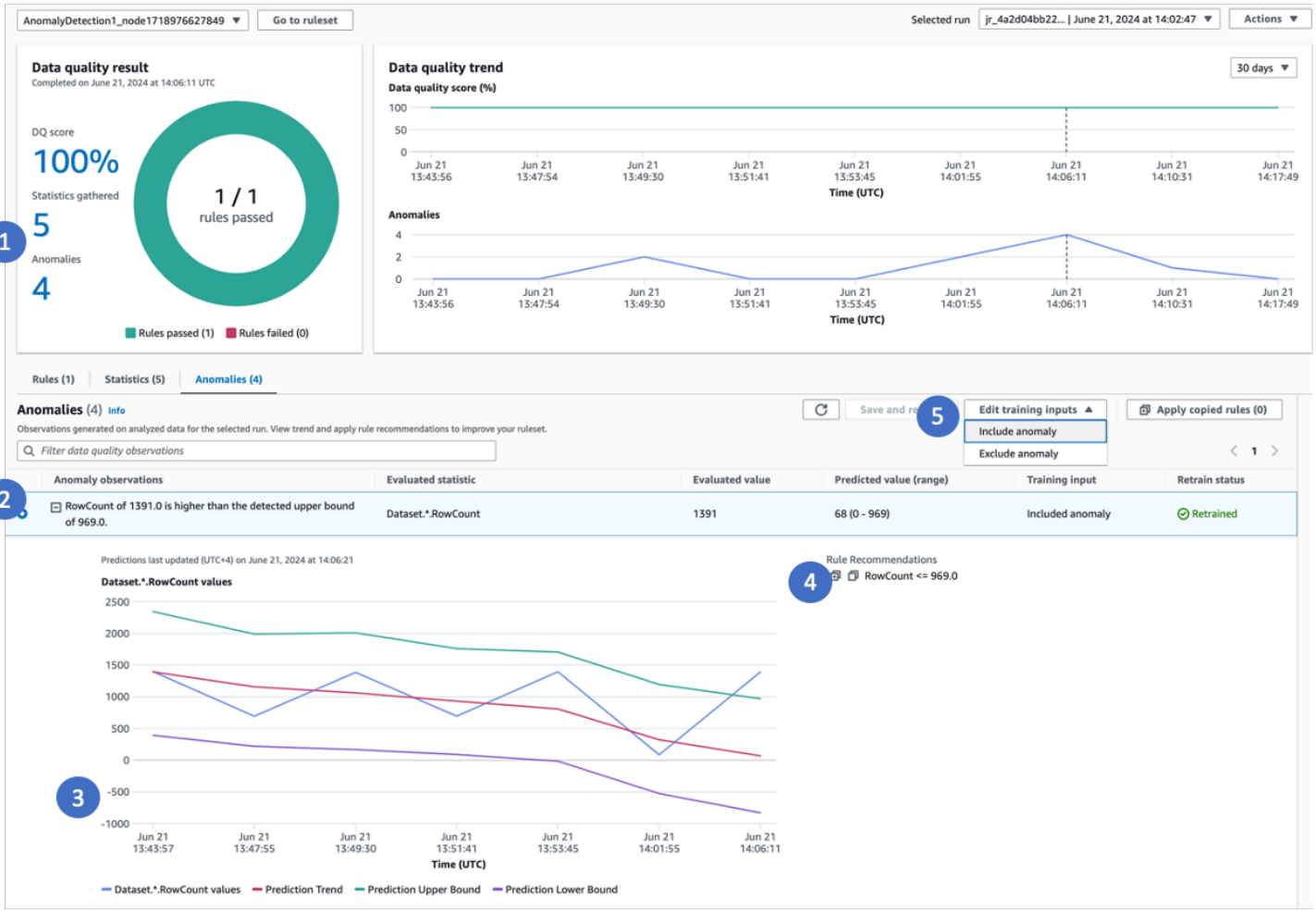

I seguenti componenti della scheda Data quality forniscono informazioni utili.

1. Seleziona la scheda Data quality per visualizzare le metriche relative alla qualità dei dati.
2. Seleziona un ID di esecuzione del processo specifico per visualizzare il punteggio di Data Quality.
3. Questo riquadro mostra tre informazioni importanti. Puoi selezionare ciascuna di esse per accedere a tabelle specifiche e visualizzare anomalie, statistiche sui dati o regole.

- Punteggio di Data Quality quando le regole sono configurate.
- Numero di statistiche raccolte da Regole e Analizzatori.
- Numero totale di anomalie rilevate.

4. Questo grafico delle tendenze mostra l'andamento della qualità dei dati nel tempo. Puoi passare il mouse sull'andamento e andare a un momento specifico in cui i punteggi di qualità dei dati sono peggiorati.
5. Le tendenze delle anomalie nel tempo mostrano il numero di anomalie rilevate nel corso del tempo.
6. Schede:
   - La scheda Regole è la scheda predefinita che mostra l'elenco di tutte le regole e il relativo stato. Regole valutate è utile nel caso di regole dinamiche, per visualizzare il valore effettivo che è stato dato alla regola.
   - La scheda Statistiche elenca tutte le statistiche, consentendo di visualizzare le metriche e le tendenze nel tempo.
   - La scheda Anomalie mostra l'elenco delle anomalie rilevate.

# Visualizzare le anomalie e addestrare l'algoritmo di rilevamento delle anomalie



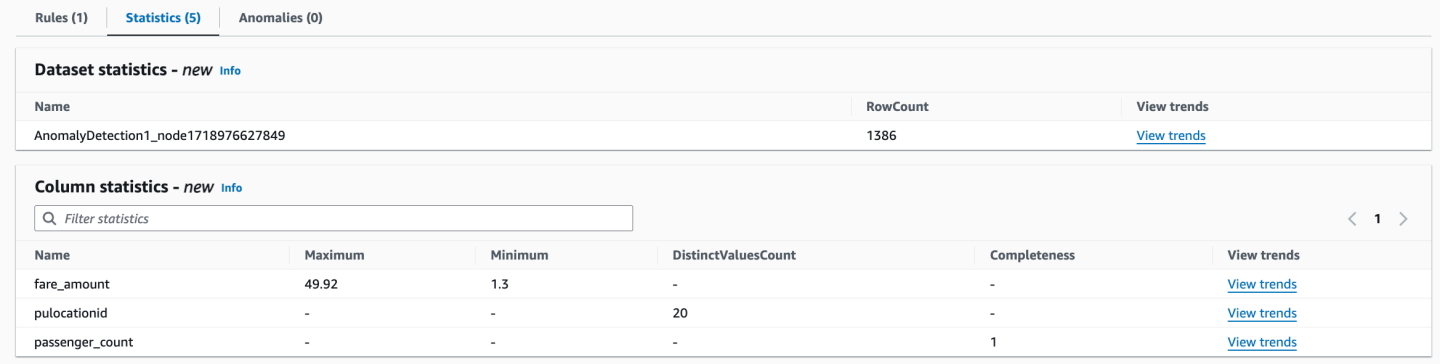

Didascalie per l'immagine sopra:

1. Quando vengono rilevate delle anomalie, fai clic sull'anomalia o seleziona la scheda Anomalie
2. AWS Glue Data Quality fornisce una spiegazione dettagliata dell'anomalia, del valore effettivo, dell'intervallo previsto
3. AWS Glue Data Quality mostra una linea di tendenza. Comprende il valore effettivo, una tendenza derivata basata sui valori effettivi (linea rossa), il limite superiore e il limite inferiore
4. AWS Glue Data Quality consiglia regole di qualità dei dati che possono essere utilizzate per catturare i modelli futuri. Puoi copiare tutte le regole consigliate e applicarle al nodo della qualità dei dati per acquisire questi modelli in modo efficace.
5. Puoi fornire input al modello di machine learning (ML) per escludere valori anomali, garantendo che le future esecuzioni rilevino le anomalie in modo accurato. Se non escludi esplicitamente le anomalie, AWS Glue Data Quality le considererà automaticamente come parte del modello per le

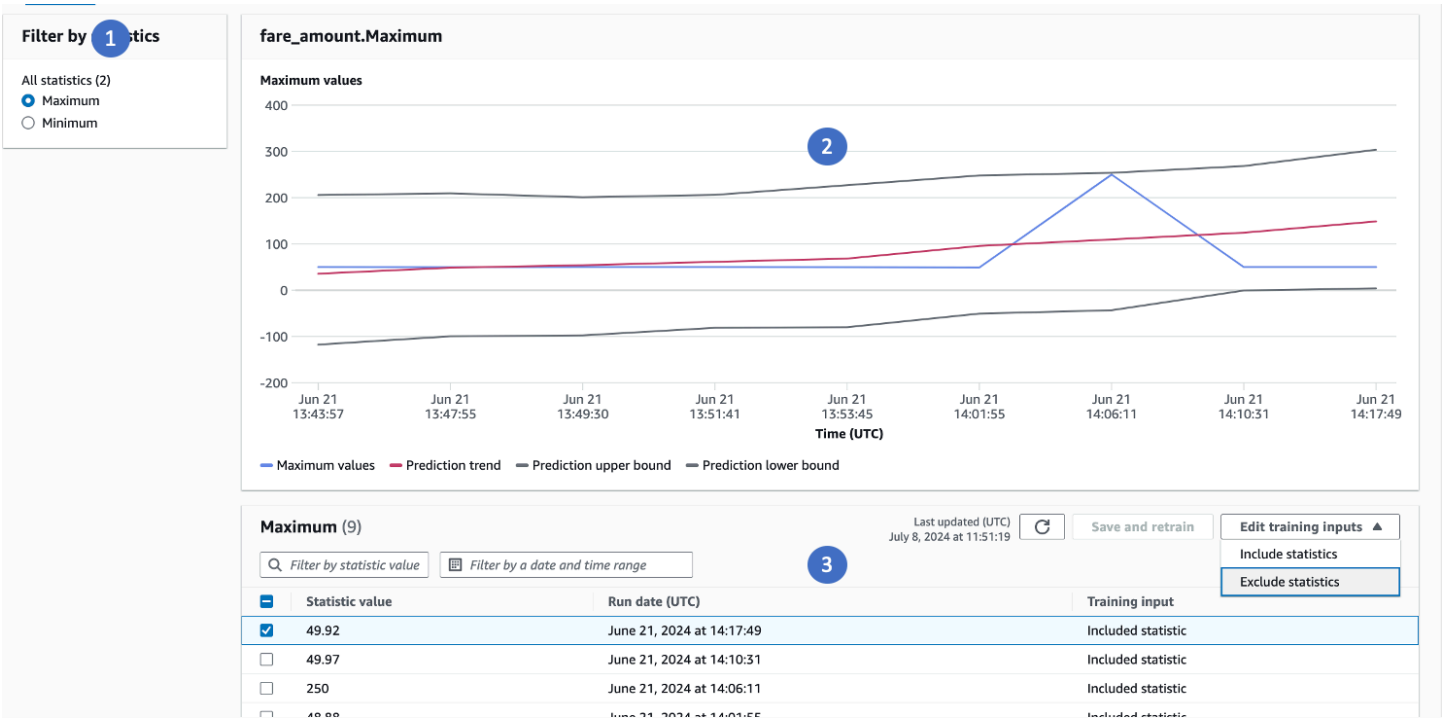
previsioni future. È importante osservare che solo l'ultima esecuzione rifletterà gli input del modello forniti. Ad esempio, se torni indietro ed escludi i punti anomali da alcune esecuzioni precedenti, il modello non rifletterà tali modifiche a meno che non visualizzi e aggiorni gli input del modello nell'ultima esecuzione. Il modello continuerà a usare gli input forniti in precedenza fino a quando non verranno apportate le modifiche necessarie nell'esecuzione più recente. Gestendo attivamente l'esclusione dei valori anomali, puoi affinare la comprensione del modello di ML relativamente a ciò che costituisce un'anomalia per i propri modelli di dati e requisiti specifici, ottenendo nel tempo un rilevamento delle anomalie più accurato.

## Visualizzare le statistiche dei dati nel tempo e fornire input per l'addestramento

A volte, potresti voler visualizzare statistiche o profili dei dati e vedere come evolvono nel tempo. Per farlo, seleziona Statistiche o apri la scheda Statistiche. È quindi possibile visualizzare le ultime statistiche sui dati raccolte da AWS Glue Data Quality.




Facendo clic su Visualizza tendenze viene mostrato l'andamento di ciascuna statistica nel tempo.




1. Puoi selezionare la statistica per una colonna specificata
2. Puoi visualizzare l'andamento delle tendenze
3. Puoi selezionare valori anomali e scegliere di escluderli o includerli. Fornendo questo feedback, l'algoritmo escluderà o includerà i punti dati anomali identificati e addestrerà nuovamente il modello. Questo processo di riaddestramento garantisce un rilevamento accurato delle anomalie in futuro, in quanto il modello apprende dal feedback fornito su quali valori debbano essere considerati anomali o meno.

   Grazie a questo ciclo di feedback, hai la possibilità di perfezionare la comprensione da parte dell'algoritmo di ciò che costituisce un'anomalia per i tuoi specifici modelli di dati e requisiti aziendali. Escludendo i valori che non dovrebbero essere contrassegnati come anomalie o includendo quelli che sono stati tralasciati, il modello addestrato sarà in grado di distinguere meglio tra i punti dati previsti e quelli realmente anomali.

# Qualità dei dati per lavori ETL nei notebook AWS Glue Studio

In questo tutorial, imparerai a utilizzare Qualità dei dati di AWS Glue per i processi di estrazione, trasformazione e caricamento (ETL) nei notebook AWS Glue Studio.

È possibile utilizzare i notebook AWS Glue Studio per modificare gli script di processo e visualizzare l'output senza dover eseguire un processo completo. È inoltre possibile aggiungere markdown e salvare i notebook come file `.ipynb` e script di processo. Nota che è possibile avviare un notebook senza installare software localmente o gestire server. Quando hai completato il lavoro sul codice, potrai utilizzare AWS Glue Studio per convertire facilmente il tuo notebook in un processo AWS Glue.

Il set di dati utilizzato in questo esempio è costituito da dati di pagamento di Medicare Provider scaricati da due set di dati Data.cms.gov: «Inpatient Prospective Payment System Provider Summary for the Top 100 Diagnosis-Related Groups» e «Inpatient Charge Data FY 2011". FY2011

Dopo aver scaricato i dati, abbiamo apportato delle modifiche al set di dati al fine di introdurre alcuni record errati nella parte finale del file. Questo file modificato si trova in un bucket pubblico Amazon S3 in `s3://awsglue-datasets/examples/medicare/Medicare_Hospital_Provider.csv`.

## Prerequisiti

- Ruolo AWS Glue con l'autorizzazione Amazon S3 per scrivere nel bucket Amazon S3 di destinazione
- Un nuovo notebook (consulta la pagina Getting started with notebooks in AWS Glue Studio)

## Creazione di un processo ETL in AWS Glue Studio

Creazione di un processo ETL

1. Modifica la versione della sessione in AWS Glue 3.0.

   Per fare ciò, rimuovi tutte le celle di codice boilerplate con il seguente magic ed esegui la cella. Nota che questo codice boilerplate viene fornito automaticamente nella prima cella quando viene creato un nuovo notebook.

   ```
   %glue_version 3.0
   ```

2. Copia il codice seguente e incollalo nella cella.

   ```
   import sys
   from awsglue.transforms import *
   from awsglue.utils import getResolvedOptions
   from pyspark.context import SparkContext
   from awsglue.context import GlueContext
   ```

```
from awsglue.job import Job

sc = SparkContext.getOrCreate()
glueContext = GlueContext(sc)
spark = glueContext.spark_session
job = Job(glueContext)
```

3. Nella cella successiva, importa la classe `EvaluateDataQuality` che valuta Qualità dei dati di AWS Glue.

```
from awsgluedq.transforms import EvaluateDataQuality
```

4. Nella cella successiva, leggi i dati di origine utilizzando il file .csv archiviato nel bucket pubblico Amazon S3.

```
medicare = spark.read.format(
"csv").option(
"header", "true").option(
"inferSchema", "true").load(
's3://awsglue-datasets/examples/medicare/Medicare_Hospital_Provider.csv')
medicare.printSchema()
```

5. Converti i dati AWS Glue DynamicFrame in un.

```
from awsglue.dynamicframe import DynamicFrame
medicare_dyf = DynamicFrame.fromDF(medicare,glueContext,"medicare_dyf")
```

6. Crea il set di regole utilizzando Data Quality Definition Language (DQDL).

```
EvaluateDataQuality_ruleset = """
    Rules = [
        ColumnExists "Provider Id",
        IsComplete "Provider Id",
        ColumnValues  " Total Discharges " > 15
]
    ]
"""
```

7. Convalida il set di dati rispetto al set di regole.

```
EvaluateDataQualityMultiframe = EvaluateDataQuality().process_rows(
    frame=medicare_dyf,
    ruleset=EvaluateDataQuality_ruleset,
    publishing_options={
        "dataQualityEvaluationContext": "EvaluateDataQualityMultiframe",
        "enableDataQualityCloudWatchMetrics": False,
        "enableDataQualityResultsPublishing": False,
    },
    additional_options={"performanceTuning.caching": "CACHE_NOTHING"},
)
```

8. Rivedi i risultati.

```
ruleOutcomes = SelectFromCollection.apply(
    dfc=EvaluateDataQualityMultiframe,
    key="ruleOutcomes",
    transformation_ctx="ruleOutcomes",
)

ruleOutcomes.toDF().show(truncate=False)
```

Output:

```
-------------------------------------+-------
+-----------------------------------------------------
+-----------------------------------------+
|Rule                                 |Outcome|FailureReason
                  |EvaluatedMetrics                           |
+-------------------------------------+-------
+-----------------------------------------------------
+-----------------------------------------+
|ColumnExists "Provider Id"           |Passed |null
                  |{}                                         |
|IsComplete "Provider Id"             |Passed |null
                  |{Column.Provider Id.Completeness -> 1.0}   |
|ColumnValues " Total Discharges " > 15|Failed |Value: 11.0 does not meet the
 constraint requirement!|{Column. Total Discharges .Minimum -> 11.0}|
```

```
+-------------------------------------+-------
+-------------------------------------------------
+-----------------------------------------+
```

9. Filtra le righe trasmesse e rivedi le righe con errori tra i risultati a livello di riga di qualità dei dati.

```
owLevelOutcomes = SelectFromCollection.apply(
dfc=EvaluateDataQualityMultiframe,
key="rowLevelOutcomes",
transformation_ctx="rowLevelOutcomes",
)

rowLevelOutcomes_df = rowLevelOutcomes.toDF() # Convert Glue DynamicFrame to
 SparkSQL DataFrame
rowLevelOutcomes_df_passed =
 rowLevelOutcomes_df.filter(rowLevelOutcomes_df.DataQualityEvaluationResult ==
 "Passed") # Filter only the Passed records.
rowLevelOutcomes_df.filter(rowLevelOutcomes_df.DataQualityEvaluationResult ==
 "Failed").show(5, truncate=False) # Review the Failed records
```

Output:

```
+----------------------------------------+-----------
+-------------------------------------+--------------------------+-------------
+--------------+-----------------+-----------------------------------
+------------------+-------------------------+-----------------------
+-------------------------+-------------------------
+-----------------------------------------+---------------------------
+---------------------------+
|DRG Definition                           |Provider Id|Provider Name
           |Provider Street Address   |Provider City|Provider State|Provider Zip
 Code|Hospital Referral Region Description| Total Discharges | Average Covered
 Charges | Average Total Payments |Average Medicare Payments|DataQualityRulesPass
      |DataQualityRulesFail                    |DataQualityRulesSkip        |
DataQualityEvaluationResult|
+----------------------------------------+-----------
+-------------------------------------+--------------------------+-------------
+--------------+-----------------+-----------------------------------
+------------------+-------------------------+-----------------------
+-------------------------+-------------------------
```

```
+----------------------------------------+---------------------------
+--------------------------+
|039 - EXTRACRANIAL PROCEDURES W/O CC/MCC|10005      |MARSHALL MEDICAL CENTER SOUTH
        |2505 U S HIGHWAY 431 NORTH|BOAZ         |AL            |35957
 |AL - Birmingham                       |14               |$15131.85
  |$5787.57               |$4976.71                |[IsComplete "Provider Id"]|
[ColumnValues " Total Discharges " > 15]|[ColumnExists "Provider Id"]|Failed
              |
|039 - EXTRACRANIAL PROCEDURES W/O CC/MCC|10046      |RIVERVIEW REGIONAL MEDICAL
 CENTER    |600 SOUTH THIRD STREET    |GADSDEN      |AL            |35901
   |AL - Birmingham                       |14               |$67327.92
   |$5461.57               |$4493.57                |[IsComplete "Provider Id"]|
[ColumnValues " Total Discharges " > 15]|[ColumnExists "Provider Id"]|Failed
              |
|039 - EXTRACRANIAL PROCEDURES W/O CC/MCC|10083      |SOUTH BALDWIN REGIONAL
 MEDICAL CENTER|1613 NORTH MCKENZIE STREET|FOLEY        |AL            |36535
        |AL - Mobile                           |15               |$25411.33
         |$5282.93               |$4383.73                |[IsComplete "Provider
 Id"]|[ColumnValues " Total Discharges " > 15]|[ColumnExists "Provider Id"]|Failed
                    |
|039 - EXTRACRANIAL PROCEDURES W/O CC/MCC|30002      |BANNER GOOD SAMARITAN MEDICAL
 CENTER |1111 EAST MCDOWELL ROAD   |PHOENIX      |AZ            |85006
 |AZ - Phoenix                          |11               |$34803.81
  |$7768.90               |$6951.45                |[IsComplete "Provider Id"]|
[ColumnValues " Total Discharges " > 15]|[ColumnExists "Provider Id"]|Failed
              |
|039 - EXTRACRANIAL PROCEDURES W/O CC/MCC|30010      |CARONDELET ST  MARYS HOSPITAL
        |1601 WEST ST MARY'S ROAD  |TUCSON       |AZ            |85745
 |AZ - Tucson                           |12               |$35968.50
  |$6506.50               |$5379.83                |[IsComplete "Provider Id"]|
[ColumnValues " Total Discharges " > 15]|[ColumnExists "Provider Id"]|Failed
              |
+----------------------------------------+-----------
+-----------------------------------+-------------------------+-------------
+--------------+-----------------+-------------------------------------
+------------------+-------------------------+----------------------
+-------------------------+-------------------------
+----------------------------------------+---------------------------
+--------------------------+
only showing top 5 rows
```

Nota che AWS Glue Data Quality ha aggiunto quattro nuove colonne (DataQualityRulesPass DataQualityRulesFail DataQualityRulesSkip,, e DataQualityEvaluationResult). Indica i record con esito positivo e quelli con esito negativo, le regole ignorate per la valutazione a livello di riga e i risultati complessivi a livello di riga.

10. Scrivi l'output in un bucket Amazon S3 per analizzare i dati e visualizzare i risultati.

```
#Write the Passed records to the destination.

glueContext.write_dynamic_frame.from_options(
       frame = rowLevelOutcomes_df_passed,
       connection_type = "s3",
       connection_options = {"path": "s3://glue-sample-target/output-dir/
medicare_parquet"},
       format = "parquet")
```

# Riferimento a Data Quality Definition Language (DQDL)

Il Data Quality Definition Language (DQDL) è un linguaggio specifico del dominio utilizzato per definire le regole per AWS Glue Data Quality.

Questa guida introduce i concetti chiave di DQDL per aiutarti a comprendere il linguaggio. Fornisce inoltre un riferimento per i tipi di regole DQDL con sintassi ed esempi. Prima di utilizzare questa guida, ti consigliamo di avere dimestichezza con AWS Glue Data Quality. Per ulteriori informazioni, consulta AWS Glue Qualità dei dati.

> **Note**
>
> DynamicRules sono supportati solo in AWS Glue ETL.

Indice


- Sintassi di DQDL
  - Struttura delle regole
  - Regole composite
    - Come funzionano le regole composite

- Espressioni
  - Parole chiave per NULL, EMPTY e WHITESPACES_ONLY
  - Filtraggio con la clausola Where
- Costanti
- Etichette
  - Sintassi per le etichette DQDL
    - Vincoli relativi alle etichette
  - Recupero delle etichette DQDL
    - Risultati delle regole
    - Risultati a livello di riga
    - Risposta API
- Regole dinamiche
- Analizzatori
- Commenti
- Documentazione di riferimento del tipo di regola DQDL
  - AggregateMatch
  - ColumnCorrelation
  - ColumnCount
  - ColumnDataType
  - ColumnExists
  - ColumnLength
  - ColumnNamesMatchPattern
  - ColumnValues
  - Completezza
  - CustomSQL
  - DataFreshness
  - DatasetMatch
  - DistinctValuesCount
  - Entropia
  - IsComplete

- IsPrimaryKey
- IsUnique
- Media
- ReferentialIntegrity
- RowCount
- RowCountMatch
- StandardDeviation
- Somma
- SchemaMatch
- Univocità
- UniqueValueRatio
- DetectAnomalies
- FileFreshness
- FileMatch
- FileUniqueness
- FileSize

## Sintassi di DQDL

Un documento DQDL fa distinzione tra maiuscole e minuscole e contiene un set di regole che raggruppa le singole regole di qualità dei dati. Per costruire un set di regole, è necessario creare un elenco denominato `Rules` (in maiuscolo), delimitato da una coppia di parentesi quadre. L'elenco deve contenere una o più regole DQDL separate da virgole come nell'esempio seguente.

```
Rules = [
   IsComplete "order-id",
   IsUnique "order-id"
]
```

### Struttura delle regole

La struttura di una regola DQDL dipende dal tipo di regola. Tuttavia, le regole DQDL generalmente si adattano al seguente formato.

```
<RuleType> <Parameter> <Parameter> <Expression>
```

`RuleType` è il nome (sensibile al maiuscolo/minuscolo) del tipo di regola che si desidera configurare. Ad esempio, `IsComplete`, `IsUnique` o `CustomSql`. I parametri delle regole sono diversi per ogni tipo di regola. Per un riferimento completo ai tipi di regole DQDL e ai relativi parametri, consulta Documentazione di riferimento del tipo di regola DQDL.

## Regole composite

DQDL supporta i seguenti operatori logici che possono essere utilizzati per combinare le regole. Queste regole vengono chiamate Regole composite.

and

L'operatore logico `and` restituisce `true` se e solo se le regole che connette sono `true`. Altrimenti, la regola combinata darà come risultato `false`. Ogni regola connessa all'operatore `and` deve essere racchiusa tra parentesi.

L'esempio seguente utilizza l'operatore `and` per combinare due regole DQDL.

```
(IsComplete "id") and (IsUnique "id")
```

or

L'operatore logico `or` restituisce `true` se e solo se una o più regole che connette sono `true`. Ogni regola connessa all'operatore `or` deve essere racchiusa tra parentesi.

L'esempio seguente utilizza l'operatore `or` per combinare due regole DQDL.

```
(RowCount "id" > 100) or (IsPrimaryKey "id")
```

È possibile utilizzare lo stesso operatore per connettere più regole, quindi la seguente combinazione di regole è consentita.

```
(Mean "Star_Rating" > 3) and (Mean "Order_Total" > 500) and (IsComplete "Order_Id")
```

È possibile combinare gli operatori logici in un'unica espressione. Esempio:

```
(Mean "Star_Rating" > 3) and ((Mean "Order_Total" > 500) or (IsComplete "Order_Id"))
```

È anche possibile creare regole più complesse e annidate.

```
(RowCount > 0) or ((IsComplete "colA") and (IsUnique "colA"))
```

Come funzionano le regole composite

Per impostazione predefinita, le regole composite vengono valutate come regole individuali sull'intero set di dati o tabella e i risultati vengono poi combinati. In altri termini, prima valuta l'intera colonna e poi applica l'operatore. Questo comportamento predefinito viene spiegato di seguito con un esempio:

```
# Dataset

+------+------+
|myCol1|myCol2|
+------+------+
|     2|     1|
|     0|     3|
+------+------+

# Overall outcome

+-------------------------------------------------------------+-------+
|Rule                                                         |Outcome|
+-------------------------------------------------------------+-------+
|(ColumnValues "myCol1" > 1) OR (ColumnValues "myCol2" > 2)|Failed |
+-------------------------------------------------------------+-------+
```

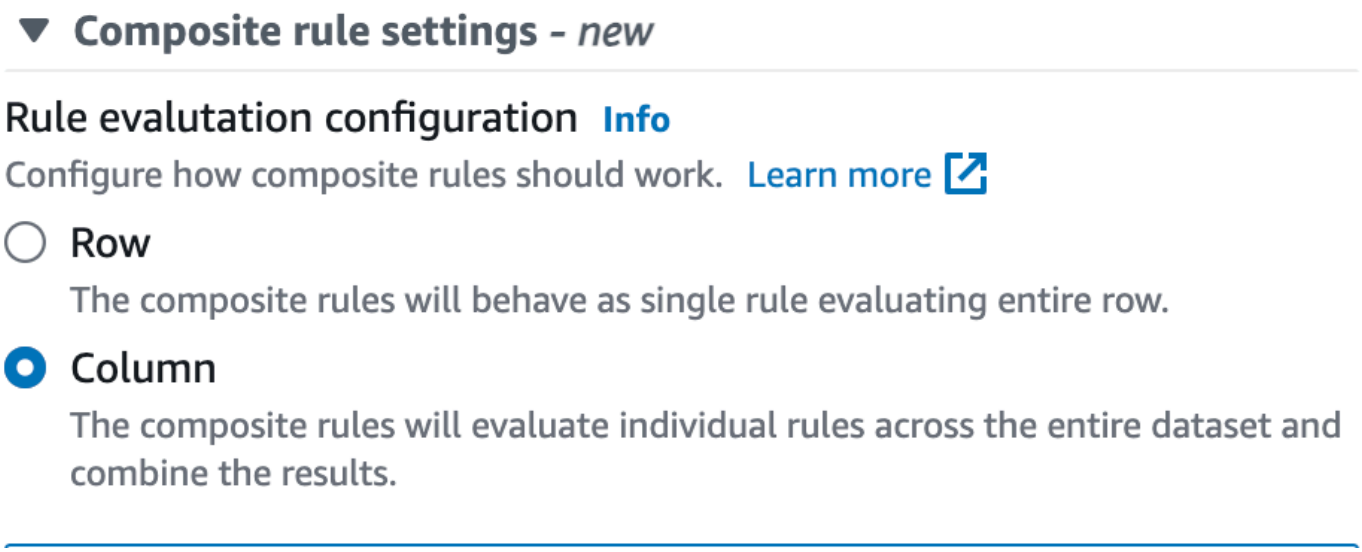
Nell'esempio precedente, AWS Glue Data Quality valuta innanzitutto `(ColumnValues "myCol1" > 1)`, che darà esito negativo. Quindi valuterà `(ColumnValues "myCol2" > 2)`, che darà anch'esso esito negativo. La combinazione di entrambi i risultati verrà contrassegnata come FAILED.

Tuttavia, se si preferisce un comportamento simile a SQL, in cui è necessario valutare l'intera riga, è necessario impostare esplicitamente il parametro `ruleEvaluation.scope` come mostrato in `additionalOptions` nel frammento di codice riportato di seguito.

```
object GlueApp {
  val datasource = glueContext.getCatalogSource(
    database="<db>",
    tableName="<table>",
    transformationContext="datasource"
  ).getDynamicFrame()
```

```
  val ruleset = """
    Rules = [
        (ColumnValues "age" >= 26) OR (ColumnLength "name" >= 4)
    ]
  """

  val dq_results = EvaluateDataQuality.processRows(
    frame=datasource,
    ruleset=ruleset,
    additionalOptions=JsonOptions("""
        {
          "compositeRuleEvaluation.method":"ROW"
        }
      """
    )
  )
}
```

In AWS Glue Data Catalog, è possibile configurare facilmente questa opzione nell'interfaccia utente come illustrato di seguito.




Una volta impostate, le regole composite si comporteranno come una singola regola che valuta l'intera riga. Nell'esempio seguente viene descritto questo comportamento.

```
# Row Level outcome

+------+------+-------------------------------------------------------------
+--------------------------+
|myCol1|myCol2|DataQualityRulesPass                                          |
DataQualityEvaluationResult|
+------+------+-------------------------------------------------------------
+--------------------------+
|2     |1     |[(ColumnValues "myCol1" > 1) OR (ColumnValues "myCol2" > 2)]|Passed
                |
|0     |3     |[(ColumnValues "myCol1" > 1) OR (ColumnValues "myCol2" > 2)]|Passed
                |
+------+------+-------------------------------------------------------------
+--------------------------+
```

Alcune regole non possono essere supportate in questa funzionalità perché il loro risultato complessivo dipende da soglie o coefficienti. Le soglie e i coefficienti sono elencati di seguito.

Regole che dipendono da coefficienti:

- Completezza
- DatasetMatch
- ReferentialIntegrity
- Univocità

Regole che dipendono da soglie:

Quando le seguenti regole includono una soglia, non sono supportate. Tuttavia, le regole che non prevedono `with threshold` continuano a essere supportate.

- ColumnDataType
- ColumnValues
- CustomSQL

## Espressioni

Se un tipo di regola non produce una risposta booleana, è necessario fornire un'espressione come parametro per creare una risposta booleana. Ad esempio, la regola seguente controlla la media di tutti i valori di una colonna rispetto a un'espressione per restituire un risultato vero o falso.

```
Mean "colA" between 80 and 100
```

Alcuni tipi di regole, ad esempio `IsUnique` e `IsComplete`, restituiscono già una risposta booleana.

Nella tabella seguente sono riportate le espressioni che è possibile utilizzare nelle regole DQDL.

Espressioni DQDL supportate

| Expression | Description | Esempio |
|---|---|---|
| =*x* | Risolve `true` se la risposta del tipo di regola è uguale a. *x* | `Completeness "colA" = "1.0", ColumnValues "colA" = "2022-06-30"` |
| !=*x* | x Risolve a true se la risposta del tipo di regola non è uguale a. *x* | `ColumnValues "colA" != "a", ColumnValues "colA" != "2022-06-30"` |
| > *x* | Risolve `true` se la risposta del tipo di regola è maggiore di. *x* | `ColumnValues "colA" > 10` |
| < *x* | Risolve `true` se la risposta del tipo di regola è inferiore a. *x* | `ColumnValues "colA" < 1000, ColumnValues "colA" < "2022-06-30"` |
| >= *x* | Risolve `true` se la risposta del tipo di regola è maggiore o uguale a. *x* | `ColumnValues "colA" >= 10` |

| Expression | Description | Esempio |
| --- | --- | --- |
| <= *x* | Risolve `true` se la risposta del tipo di regola è minore o uguale a. *x* | `ColumnValues "colA" <= 1000` |
| tra e *x* *y* | Si risolve in `true` se la risposta del tipo di regola rientra in un intervallo specificato (esclusivo). Utilizza questo tipo di espressione solo per i tipi numerici e data. | `Mean "colA" between 8 and 100, ColumnValues "colA" between "2022-05-31" and "2022-06-30"` |
| non tra *x* e *y* | Si risolve in true se il tipo di risposta del tipo di regola non rientra in un intervall o specificato (inclusivo). È necessario utilizzare questo tipo di espressione solo per i tipi numerici e data. | `ColumnValues "colA" not between "2022-05-31" and "2022-06-30"` |
| in [*a, b, c, ...*] | Si risolve in `true` se la risposta del tipo di regola è nel set specificato. | `ColumnValues "colA" in [ 1, 2, 3 ], ColumnValues "colA" in [ "a", "b", "c" ]` |
| non in [*a, b, c, ...*] | Si risolve in `true` se la risposta del tipo di regola non è nel set specificato. | `ColumnValues "colA" not in [ 1, 2, 3 ], ColumnValues "colA" not in [ "a", "b", "c" ]` |
| fiammiferi */ab+c/i* | Si risolve in `true` se la risposta del tipo di regola corrisponde a un'espressione regolare. | `ColumnValues "colA" matches "[a-zA-Z]*"` |

| Expression | Description | Esempio |
| --- | --- | --- |
| non corrisponde `/ab+c/i` | Si risolve in `true` se la risposta del tipo di regola non corrisponde a un'espressione regolare. | `ColumnValues "colA" not matches "[a-zA-Z]*"` |
| `now()` | Funziona solo con il tipo di regola `ColumnValues` per creare un'espressione di data. | `ColumnValues "load_date" > (now() - 3 days)` |
| matches/in […]/not matches/notin [...] `with threshold` | Specifica la percentuale di valori che corrispondono alle condizioni della regola. Funziona solo con i tipi di regola `ColumnValues` , `ColumnDataType` e `CustomSQL` . | `ColumnValues "colA" in ["A", "B"] with threshold > 0.8, ColumnValues "colA" matches "[a-zA-Z]*" with threshold between 0.2 and 0.9 ColumnDataType "colA" = "Timestamp" with threshold > 0.9` |

Parole chiave per NULL, EMPTY e WHITESPACES_ONLY

Se vuoi verificare se una colonna di stringhe contiene un valore null, vuoto o una stringa con solo spazi bianchi, puoi utilizzare le seguenti parole chiave:

- NULL / null: questa parola chiave restituisce il valore true per un valore `null` in una colonna di stringhe.

  `ColumnValues "colA" != NULL with threshold > 0.5` restituisce il valore true se oltre il 50% dei dati non contiene valori null.

  `(ColumnValues "colA" = NULL) or (ColumnLength "colA" > 5)` restituisce il valore true per tutte le righe che hanno un valore null o una lunghezza >5. Nota che ciò richiederà l'uso dell'opzione «compositeRuleEvaluation.method» = «ROW».

- EMPTY / empty: questa parola chiave restituisce il valore true per un valore di stringa vuoto ("") in una colonna di stringhe. Alcuni formati di dati trasformano i valori null di una colonna di stringhe in stringhe vuote. Questa parola chiave consente di filtrare le stringhe vuote presenti nei dati.

  `(ColumnValues "colA" = EMPTY) or (ColumnValues "colA" in ["a", "b"])` restituisce il valore true se una riga è vuota, "a" o "b". Nota che ciò richiede l'uso dell'opzione «compositeRuleEvaluation.method» = «ROW».
- WHITESPACES_ONLY / whitespaces_only: questa parola chiave restituisce il valore true per una stringa contenente solo spazi bianchi (" ") in una colonna di stringhe.

  `ColumnValues "colA" not in ["a", "b", WHITESPACES_ONLY]` restituisce il valore true se una riga non è né "a" né "b" né solo spazi bianchi.

  Regole supportate:

  - ColumnValues

Per un'espressione numerica o basata sulla data, se si desidera verificare se una colonna contiene un valore null, è possibile utilizzare le seguenti parole chiave.

- NULL / null: questa parola chiave restituisce il valore true per un valore null in una colonna di stringhe.

  `ColumnValues "colA" in [NULL, "2023-01-01"]` restituisce il valore true se una delle date nella colonna è `2023-01-01` o null.

  `(ColumnValues "colA" = NULL) or (ColumnValues "colA" between 1 and 9)` restituisce il valore true per tutte le righe che hanno un valore null o valori compresi tra 1 e 9. Nota che ciò richiederà l'uso dell'opzione «compositeRuleEvaluation.method» = «ROW».

  Regole supportate:

  - ColumnValues

Filtraggio con la clausola Where

> **Note**
>
> Where Clause è supportato solo in AWS Glue 4.0.

È possibile filtrare i dati durante la creazione delle regole. È utile quando si vuole applicare regole condizionali.

```
<DQDL Rule> where "<valid SparkSQL where clause> "
```

Il filtro deve essere specificato con la parola chiave `where` seguita da un'istruzione SparkSQL valida racchiusa tra virgolette (`""`).

Se si desidera aggiungere la clausola where a una regola con una soglia, la clausola where deve essere specificata prima della condizione di soglia.

```
<DQDL Rule> where "valid SparkSQL statement>" with threshold <threshold condition>
```

Con questa sintassi è possibile scrivere regole come quelle riportate di seguito.

```
Completeness "colA" > 0.5 where "colB = 10"
ColumnValues "colB" in ["A", "B"] where "colC is not null" with threshold > 0.9
ColumnLength "colC" > 10 where "colD != Concat(colE, colF)"
```

Valuteremo se l'istruzione SparkSQL fornita è valida. Se non è valida, la valutazione della regola avrà esito negativo e verrà generato un `IllegalArgumentException` con il seguente formato:

```
Rule <DQDL Rule> where "<invalid SparkSQL>" has provided an invalid where clause :
<SparkSQL Error>
```

Comportamento della clausola Where quando è attivata l'identificazione dei record di errore a livello di riga

Con AWS Glue Data Quality, puoi identificare record specifici che hanno avuto esito negativo. Quando si applica una clausola where alle regole che supportano risultati a livello di riga, le righe filtrate dalla clausola where vengono contrassegnate come `Passed`.

Se si preferisce contrassegnare separatamente le righe filtrate come `SKIPPED`, è possibile impostare le seguenti `additionalOptions` per il processo ETL.

```
object GlueApp {
  val datasource = glueContext.getCatalogSource(
    database="<db>",
    tableName="<table>",
```

```
    transformationContext="datasource"
  ).getDynamicFrame()

  val ruleset = """
    Rules = [
       IsComplete "att2" where "att1 = 'a'"
    ]
  """

  val dq_results = EvaluateDataQuality.processRows(
    frame=datasource,
    ruleset=ruleset,
    additionalOptions=JsonOptions("""
        {
          "rowLevelConfiguration.filteredRowLabel":"SKIPPED"
        }
      """
    )
  )
}
```

A titolo esemplificativo, fare riferimento alla regola e al dataframe riportati di seguito:

```
IsComplete att2 where "att1 = 'a'"
```

| id | att1 | att2 | Risultati a livello di riga (impostazione predefinita) | Risultati a livello di riga (opzione ignorata) | Commenti |
|---|---|---|---|---|---|
| 1 | a | f | PASSED | PASSED | |
| 2 | b | d | PASSED | SKIPPED | La riga viene esclusa in quanto `att1` non è `"a"` |
| 3 | a | null | NON RIUSCITO | NON RIUSCITO | |

| id | att1 | att2 | Risultati a livello di riga (impostazione predefinita) | Risultati a livello di riga (opzione ignorata) | Commenti |
|---|---|---|---|---|---|
| 4 | a | f | PASSED | PASSED | |
| 5 | b | null | PASSED | SKIPPED | La riga viene esclusa in quanto `att1` non è `"a"` |
| 6 | a | f | PASSED | PASSED | |

## Costanti

In DQDL, è possibile definire valori costanti e farvi riferimento in tutto lo script. Questo aiuta a prevenire problemi relativi ai limiti di dimensione delle query, ad esempio quando si lavora con istruzioni SQL di grandi dimensioni che potrebbero superare i limiti consentiti. Assegnando questi valori a Constants, è possibile semplificare il DQDL ed evitare di raggiungere tali limiti.

L'esempio seguente mostra come definire e utilizzare una costante:

```
mySql = "select count(*) from primary"

Rules = [
    CustomSql $mySql between 0 and 100
]
```

In questo esempio, la query SQL viene assegnata alla costante`mySql`, a cui viene quindi fatto riferimento nella regola utilizzando il $ prefisso.

## Etichette

Le etichette offrono un modo efficace per organizzare e analizzare i risultati di qualità dei dati. È possibile interrogare i risultati in base a etichette specifiche per identificare le regole non corrette all'interno di determinate categorie, contare i risultati delle regole per team o dominio e creare report mirati per le diverse parti interessate.

Ad esempio, puoi applicare tutte le regole relative al team finanziario con un'etichetta `"team=finance"` e generare un report personalizzato per mostrare le metriche di qualità specifiche del team finanziario. È possibile etichettare le regole ad alta priorità con cui dare priorità alle azioni `"criticality=high"` correttive. Le etichette possono essere create come parte del DQDL. È possibile interrogare le etichette come parte dei risultati delle regole, dei risultati a livello di riga e delle risposte delle API, semplificando l'integrazione con i flussi di lavoro di monitoraggio e reporting esistenti.

> **Note**
>
> Le etichette sono disponibili solo in AWS Glue ETL e non sono disponibili in Data Quality basato su AWS Glue Data Catalog.

### Sintassi per le etichette DQDL

DQDL supporta etichette predefinite e specifiche per regole. Le etichette predefinite sono definite a livello di set di regole e si applicano automaticamente a tutte le regole all'interno di quel set di regole. Le singole regole possono anche avere le proprie etichette e, poiché le etichette sono implementate come coppie chiave-valore, le etichette specifiche delle regole possono sovrascrivere le etichette predefinite quando si utilizza la stessa chiave.

L'esempio seguente mostra come utilizzare etichette predefinite e specifiche per regole:

```
DefaultLabels=["frequency"="monthly"]

Rules = [
    // Auto includes the default label ["frequency"="monthly"]
    ColumnValues "col" > 21,
    // Add ["foo"="bar"] to default label. Labels for this rule would be
 ["frequency"="monthly", "foo"="bar"]
    RowCount > 0 with threshold > 0.8 labels=["foo"="bar"],
    // Override default label. Labels for this rule would be ["frequency"="daily",
 "foo"="bar"]
    ColumnValues "colA" in ["A", "B"]  with threshold > 0.8 labels=["foo"="bar",
 "frequency"="daily"]
    // Labels must be applied to the entire composite rule (parentheses required)
    (isComplete "col" AND RowCount > 0) labels=["foo"="bar]
]
```

L'esempio seguente mostra una sintassi non valida con etichette e regole composite:

```
        (isComplete "colA") AND (RowCount > 0) labels=["foo"="bar"]
        (isComplete "colA" labels=["foo"="bar"]) AND (RowCount > 0)
        isComplete "col" AND RowCount > 0 labels=["foo"="bar]
```

Vincoli relativi alle etichette

Le etichette presentano i seguenti vincoli:

- Un massimo di 10 etichette per regola DQDL.
- Le etichette sono specificate come un elenco di coppie chiave-valore.
- La chiave dell'etichetta e il valore dell'etichetta fanno distinzione tra maiuscole e minuscole.
- La lunghezza massima della chiave dell'etichetta è di 128 caratteri. La chiave dell'etichetta non deve essere vuota o nulla.
- La lunghezza massima del valore dell'etichetta è 256 caratteri. Il valore dell'etichetta può essere vuoto o nullo.

Recupero delle etichette DQDL

È possibile recuperare le etichette DQDL dai risultati delle regole, dai risultati a livello di riga e dalle risposte delle API.

Risultati delle regole

Le etichette DQDL sono sempre visibili nei risultati delle regole. Non è necessaria alcuna configurazione aggiuntiva per abilitarle.

Risultati a livello di riga

Le etichette DQDL sono disabilitate per impostazione predefinita nei risultati a livello di riga, ma possono essere abilitate utilizzando in. `AdditionalOptions` `EvaluateDataQuality`

L'esempio seguente mostra come abilitare le etichette nei risultati a livello di riga:

```
val evaluateResult = EvaluateDataQuality.processRows(
    frame=AmazonS3_node1754591511068,
    ruleset=example_ruleset,
    publishingOptions=JsonOptions("""{
        "dataQualityEvaluationContext": "evaluateResult",
```

```
        "enableDataQualityCloudWatchMetrics": "true",
        "enableDataQualityResultsPublishing": "true"
    }"""),
    additionalOptions=JsonOptions("""{
        "performanceTuning.caching":"CACHE_NOTHING",
        "observations.scope":"ALL",
        "rowLevelConfiguration.ruleWithLabels":"ENABLED"
    }""")
)
```

Se abilitato, il dataframe dei risultati a livello di riga include le etichette per ogni regola nelle colonne, e. `DataQualityRulesPass` `DataQualityRulesFail` `DataQualityRulesSkip`

Risposta API

Le etichette DQDL sono sempre visibili nelle risposte API in un nuovo campo `Labels` dell'`RuleResults`oggetto.

L'esempio seguente mostra le etichette in una risposta API:

```
{
    "ResultId": "dqresult-example",
    "ProfileId": "dqprofile-example",
    "Score": 0.6666666666666666,
    "RulesetName": "EvaluateDataQuality_node1754591514205",
    "EvaluationContext": "EvaluateDataQuality_node1754591514205",
    "StartedOn": "2025-08-22T19:36:10.448000+00:00",
    "CompletedOn": "2025-08-22T19:36:16.368000+00:00",
    "JobName": "anniezc-test-labels",
    "JobRunId": "jr_068f6d7a45074d9105d14e4dee09db12c3b95664b45f6ee44fa29ed7e5619ba8",
    "RuleResults": [
        {
            "Name": "Rule_0",
            "Description": "IsComplete colA",
            "EvaluationMessage": "Input data does not include column colA!",
            "Result": "FAIL",
            "EvaluatedMetrics": {},
            "EvaluatedRule": "IsComplete colA",
            "Labels": {
                "frequency": "monthly"
            }
        },
        {
```

```
            "Name": "Rule_1",
            "Description": "Rule 1 with threshold > 0.8",
            "Result": "PASS",
            "EvaluatedMetrics": {},
            "EvaluatedRule": "Rule 1 with threshold > 0.8",
            "Labels": {
                "frequency": "monthly",
                "foo": "bar"
            }
        },
        {
            "Name": "Rule_3",
            "Description": "Rule 2 with threshold > 0.8",
            "Result": "PASS",
            "EvaluatedMetrics": {},
            "EvaluatedRule": "Rule 2 with threshold > 0.8",
            "Labels": {
                "frequency": "daily",
                "foo": "bar"
            }
        }
    ]
}
```

## Regole dinamiche

> **Note**
>
> Le regole dinamiche sono supportate solo in AWS Glue ETL e non sono supportate in AWS Glue Data Catalog.

Ora puoi creare regole dinamiche per confrontare le metriche correnti prodotte dalle tue regole con i relativi valori storici. Questi confronti storici sono abilitati utilizzando l'operatore `last()` nelle espressioni. Ad esempio, la regola `RowCount > last()` avrà esito positivo se il numero di righe nell'esecuzione corrente è maggiore del conteggio precedente più recente delle righe per lo stesso set di dati. `last()` utilizza un argomento facoltativo relativo ai numeri naturali che descrive il numero di metriche precedenti da prendere in considerazione; `last(k)` dove `k >= 1` farà riferimento alle ultime k metriche.

- Se non sono disponibili punti dati, `last(k)` restituirà il valore predefinito 0,0.

- Se sono disponibili meno di k metriche, `last(k)` restituirà tutte quelle precedenti.

Utilizza `last(k)` per formare espressioni valide, dove `k > 1` richiede una funzione di aggregazione per ridurre più risultati storici a un unico numero. Ad esempio, `RowCount > avg(last(5))` controllerà se il conteggio delle righe del set di dati corrente è strettamente maggiore della media dei conteggi delle ultime cinque righe per lo stesso set di dati. `RowCount > last(5)` produrrà un errore perché il conteggio delle righe del set di dati corrente non può essere confrontato in modo significativo con un elenco.

Funzioni di aggregazione supportate:

- `avg`
- `median`
- `max`
- `min`
- `sum`
- `std` (deviazione standard)
- `abs` (valore assoluto)
- `index(last(k), i)` consentirà di selezionare il `i`° valore più recente tra gli ultimi k. `i` è indicizzato a zero, quindi `index(last(3), 0)` restituirà il punto dati più recente e `index(last(3), 3)` genererà un errore poiché ci sono solo tre punti dati, mentre noi cerchiamo di indicizzare il 4° punto dati più recente.

Espressioni di esempio

ColumnCorrelation

- `ColumnCorrelation "colA" "colB" < avg(last(10))`

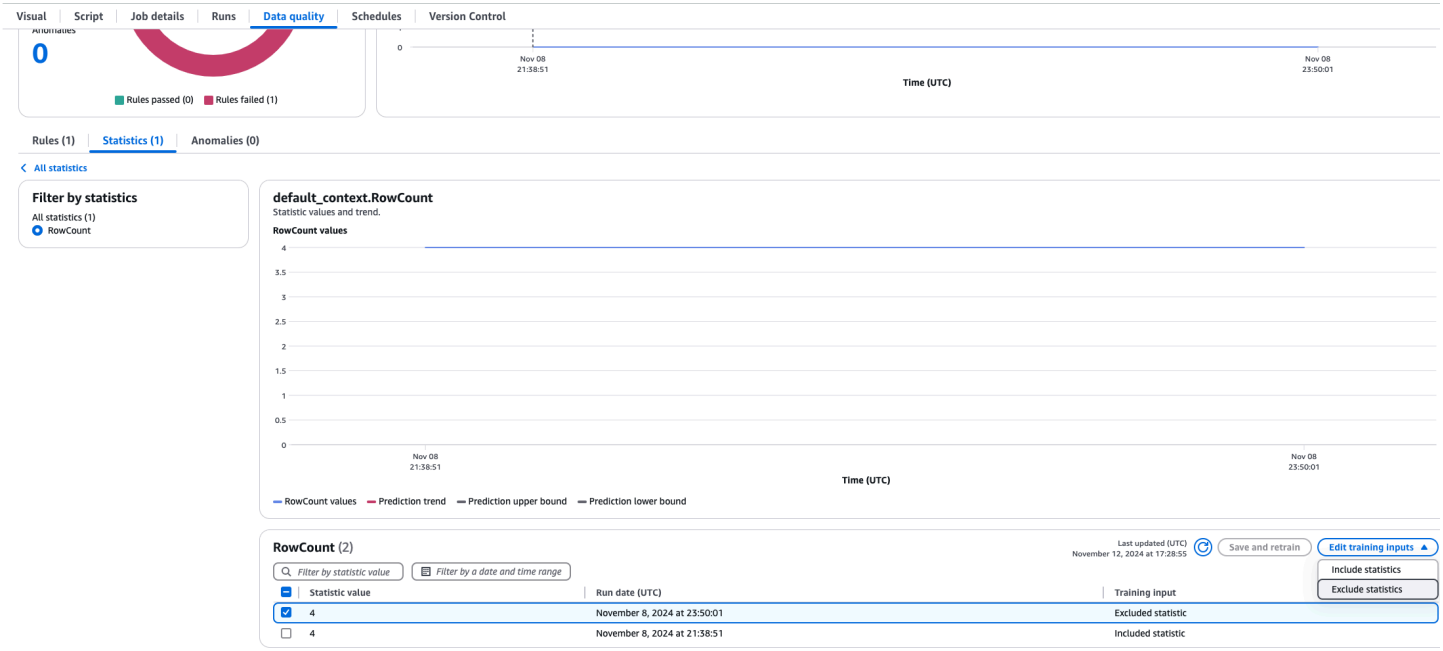
DistinctValuesCount

- `DistinctValuesCount "colA" between min(last(10))-1 and max(last(10))+1`

La maggior parte dei tipi di regole con condizioni o soglie numeriche supporta regole dinamiche; consulta la tabella fornita, Analizzatori e regole, per determinare se le regole dinamiche sono supportate per il tuo tipo di regola.

### Escludere le statistiche dalle regole dinamiche

A volte è necessario escludere le statistiche sui dati dai calcoli delle regole dinamiche. Supponiamo che tu abbia caricato dei dati storici e non desideri che ciò influisca sulle tue medie. Per fare ciò, apri il lavoro in AWS Glue ETL e scegli la scheda Data Quality, quindi scegli Statistiche e seleziona le statistiche che desideri escludere. Potrai visualizzare un grafico delle tendenze insieme a una tabella delle statistiche. Scegli i valori che desideri escludere e seleziona Escludi statistiche. Ora le statistiche escluse non saranno incluse nei calcoli delle regole dinamiche.




## Analizzatori

> **Note**
>
> Gli analizzatori non sono supportati in AWS Glue Data Catalog.

Le regole DQDL utilizzano funzioni chiamate analizzatori per raccogliere informazioni sui dati. Queste informazioni vengono utilizzate dall'espressione booleana di una regola per determinare se quest'ultima deve avere esito positivo o negativo. Ad esempio, la RowCount regola `RowCount > 5` utilizzerà un analizzatore del conteggio delle righe per scoprire il numero di righe nel set di dati e confronterà tale conteggio con l'espressione `> 5` per verificare se nel set di dati corrente esistono più di cinque righe.

A volte, invece di creare regole, consigliamo di creare analizzatori e fare in modo che generino statistiche da utilizzare per rilevare anomalie. In questi casi, puoi creare analizzatori. Gli analizzatori differiscono dalle regole nei modi indicati di seguito.

| Caratteristica | Analizzatori | Regole |
|---|---|---|
| Parte del set di regole | Sì | Sì |
| Genera statistiche | Sì | Sì |
| Genera osservazioni | Sì | Sì |
| Può valutare e verificare una condizione | No | Sì |
| È possibile configurare operazioni come l'interru zione dei processi in caso di errore o la prosecuzione di un processo di elaborazione | No | Sì |

Gli analizzatori possono esistere indipendentemente senza regole, quindi puoi configurarli in modo rapido e creare regole di qualità dei dati in modo progressivo.

Alcuni tipi di regole possono essere inseriti nel blocco `Analyzers` del set di regole per eseguire quelle richieste per gli analizzatori e raccogliere informazioni senza applicare controlli per alcuna condizione. Esistono analizzatori che non sono associati ad alcuna regola e che possono essere inseriti solo nel blocco `Analyzers`. La tabella seguente indica se ogni elemento è supportato come regola o come analizzatore autonomo, insieme a dettagli aggiuntivi per ogni tipo di regola.

Esempio di set di regole con Analyzer

Il seguente set di regole utilizza:

- una regola dinamica per verificare se un set di dati è in crescita rispetto alla media finale delle ultime tre esecuzioni del processo
- un analizzatore `DistinctValuesCount` per registrare il numero di valori distinti nella colonna del `Name` del set di dati

- un analizzatore `ColumnLength` per tracciare le dimensioni minime e massime del `Name` nel tempo

I risultati delle metriche dell'analizzatore per l'esecuzione del processo possono essere visualizzati nella scheda Qualità dei dati.

```
Rules = [
   RowCount > avg(last(3))
]
Analyzers = [
   DistinctValuesCount "Name",
   ColumnLength "Name"
]
```

AWS Glue Data Quality supporta i seguenti analizzatori.

| Nome dell'analizzatore | Funzionalità |
|---|---|
| `RowCount` | Calcola il conteggio delle righe per un set di dati |
| `Completeness` | Calcola la percentuale di completezza di una colonna |
| `Uniqueness` | Calcola la percentuale di unicità di una colonna |
| `Mean` | Calcola la media di una colonna numerica |
| `Sum` | Calcola la somma di una colonna numerica |
| `StandardDeviation` | Calcola la deviazione standard di una colonna numerica |
| `Entropy` | Calcola l'entropia di una colonna numerica |
| `DistinctValuesCount` | Calcola il numero di valori distinti in una colonna |
| `UniqueValueRatio` | Calcola il rapporto di valori univoci in una colonna |

| Nome dell'analizzatore | Funzionalità |
| --- | --- |
| `ColumnCount` | Calcola il numero di colonne in un set di dati |
| `ColumnLength` | Calcola la lunghezza di una colonna |
| `ColumnValues` | Calcola il valore minimo e il valore massimo per le colonne numeriche. Calcola il valore minimo ColumnLength e massimo ColumnLength per le colonne non numeriche |
| `ColumnCorrelation` | Calcola le correlazioni tra determinate colonne |
| `CustomSql` | Calcola le statistiche restituite da CustomSQL |
| `AllStatistics` | Calcola le seguenti statistiche:<br>• RowCount, ColumnCount<br>• Tutte le colonne: completezza, unicità<br>• Numeriche: minimo, massimo, entropia, media, sviluppo standard, somma<br>• Stringa:, MinLength MaxLength |

### Commenti

È possibile usare il carattere '#' per aggiungere un commento al documento DQDL. Tutto ciò che segue il carattere '#' e fino alla fine della riga viene ignorato da DQDL.

```
Rules = [
    # More items should generally mean a higher price, so correlation should be
 positive
    ColumnCorrelation "price" "num_items" > 0
]
```

## Documentazione di riferimento del tipo di regola DQDL

Questa sezione fornisce un riferimento per ogni tipo di regola supportato da AWS Glue Data Quality.

> **Note**
>
> - Attualmente DQDL non supporta dati di colonna annidati o di tipo elenco.
> - I valori tra parentesi nella tabella seguente verranno sostituiti con le informazioni fornite negli argomenti delle regole.
> - Le regole richiedono in genere un argomento aggiuntivo per l'espressione.

| RuleType | Description | Argumenti (Argomenti) | Metriche riportate | Supporta come regola? | Supporta come analizzatore? | Restituisce risultati a livello di riga? | Supporto per regole dinamiche? | Genera osservazioni | Supporta la sintassi della clausola Where? |
|---|---|---|---|---|---|---|---|---|---|
| AggregateMatch | Verifica se due set di dati corrispondono confrontando parametri di riepilogo come l'importo totale delle vendite. È utile agli istituti | Una o più aggregazioni | Quando i nomi della prima e della seconda colonna di aggregazione corrispondono: `Column.[Column].aggregateMatch` | Sì | No | No | No | No | No |

| RuleTyp | Descripti on | Argumer (Argome i) | Metriche riportate | Supporta o come regola? | Supporta o come analizzat ore? | Restituis ce risultati a livello di riga? | Supporto per regole dinamich ? | Genera osservaz oni | Supporta la sintassi della clausola Where? |
|---|---|---|---|---|---|---|---|---|---|
| | finanziar i per confronta re se tutti i dati vengono importati dai sistemi di origine. | | Quando i nomi della prima e della seconda colonna di aggrega one non corrispor dono: `Column [C olumn1, lumn2]. gregate tch` | | | | | | |

| RuleTyp | Descripti on | Argumer (Argome i) | Metriche riportate | Supporta o come regola? | Supporta o come analizzat ore? | Restituis ce risultati a livello di riga? | Supporto per regole dinamich ? | Genera osservaz oni | Supporta la sintassi della clausola Where? |
|---|---|---|---|---|---|---|---|---|---|
| AllStatis tics | Analizzat ore autonom per raccoglie re più metriche per la colonna fornita in un set di dati. | Un nome di colonna singolo | Per le colonne di tutti i tipi:<br>`Dataset .RowCou`<br>`Column. [C olumn]. mpleter s`<br>`Column. [C olumn]. iquenes`<br>Metriche aggiuntiv e per le colonne con valori stringa: | No | Sì | No | No | No | No |

| RuleTyp | Descripti on | Argumer (Argome i) | Metriche riportate | Supporta o come regola? | Supporta o come analizzat ore? | Restituis ce risultati a livello di riga? | Supporto per regole dinamich ? | Genera osservaz oni | Supporta la sintassi della clausola Where? |
|---|---|---|---|---|---|---|---|---|---|
| | | | `ColumnL gth metrics`<br><br>Metriche aggiuntiv e per le colonne con valori numerici:<br><br>`Column\ ues metrics` | | | | | | |
| ColumnC relation | Verifica il grado di correlazi one tra due colonne. | Esattame te due nomi di colonne | `Multicc mn. [Colum n1,Colu 2].Colu Correla on` | Sì | Sì | No | Sì | No | Sì |

| RuleTyp | Descripti on | Argumer (Argome i) | Metriche riportate | Supporta o come regola? | Supporta o come analizzat ore? | Restituis ce risultati a livello di riga? | Supporto per regole dinamich ? | Genera osservaz oni | Supporta la sintassi della clausola Where? |
|---|---|---|---|---|---|---|---|---|---|
| ColumnC nt | Verifica se delle colonne vengono eliminate . | Nessuno | `Dataset .Columr unt` | Sì | Sì | No | Sì | Sì | No |
| ColumnD aType | Verifica se una colonna è conforme a un tipo di dati. | Esattame te un nome di colonna | `Column. [C olumn]. lumnDat ype.Cor iance` | Sì | No | No | Sì, nell'espr essione di soglia a livello di riga | No | Sì |

| RuleType | Description | Argumenti (Argomenti) | Metriche riportate | Supporta o come regola? | Supporta o come analizzatore? | Restituisce risultati a livello di riga? | Supporto per regole dinamiche? | Genera osservazioni | Supporta la sintassi della clausola Where? |
|---|---|---|---|---|---|---|---|---|---|
| ColumnExists | Verifica se esistono colonne in un set di dati. Ciò consente ai clienti di creare piattaforme di dati self-service per garantire la disponibilità di determinate colonne. | Esattamente un nome di colonna | N/D | Sì | No | No | No | No | No |

| RuleTyp | Descripti on | Argumer (Argome i) | Metriche riportate | Supporta o come regola? | Supporta o come analizzat ore? | Restituis ce risultati a livello di riga? | Supporto per regole dinamich ? | Genera osservaz oni | Supporta la sintassi della clausola Where? |
|---|---|---|---|---|---|---|---|---|---|
| ColumnL gth | Verifica se la lunghezz dei dati è coerente | Esattame te un nome di colonna | `Column. [C olumn]. ximumLe th`<br><br>`Column. [C olumn]. nimumLe th`<br><br>Metrica aggiuntiv a quando viene fornita la soglia a livello di riga:<br><br>`Column. [C olumn]. lumnVal` | Sì | Sì | Sì, quando viene fornita la soglia a livello di riga | No | Sì. Genera solo osservaz oni analizzar do la lunghezz minima e quella massima | Sì |

| RuleTyp | Descripti on | Argumer (Argome i) | Metriche riportate | Supporta o come regola? | Supporta o come analizzat ore? | Restituis ce risultati a livello di riga? | Supportc per regole dinamich ? | Genera osservaz oni | Supporta la sintassi della clausola Where? |
|---|---|---|---|---|---|---|---|---|---|
| | | | `s.Compl nce` | | | | | | |
| ColumnN esMatchl ttern | Verifica se i nomi delle colonne corrispor dono ai modelli definiti. È utile ai team di governar e per far rispettar e la coerenza dei nomi delle colonne. | Un'espre sione regolare per i nomi delle colonne | `Dataset .Columr mesPatt nMatchF io` | Sì | No | No | No | No | No |

| RuleTyp | Descripti on | Argumer (Argome i) | Metriche riportate | Supporta o come regola? | Supporta o come analizzat ore? | Restituis ce risultati a livello di riga? | Supporto per regole dinamich ? | Genera osservaz oni | Supporta la sintassi della clausola Where? |
|---|---|---|---|---|---|---|---|---|---|
| ColumnV ues | Verifica se i dati sono coerenti per valori definiti. Questa regola supporta le espressi ni regolari. | Esattame te un nome di colonna | `Column. [C olumn]. ximum`<br><br>`Column. [C olumn]. nimum`<br><br>Metrica aggiuntiv a quando viene fornita la soglia a livello di riga:<br><br>`Column. [C olumn]. lumnVa s.Comp nce` | Sì | Sì | Sì, quando viene fornita la soglia a livello di riga | No | Sì. Genera solo osservaz oni analizzar do i valori minimi e quelli massimi | Sì |

| RuleTyp | Descripti on | Argumer (Argome i) | Metriche riportate | Supporta o come regola? | Supporta o come analizzat ore? | Restituis ce risultati a livello di riga? | Supporto per regole dinamich ? | Genera osservaz oni | Supporta la sintassi della clausola Where? |
|---|---|---|---|---|---|---|---|---|---|
| Complet za | Verifica la presenza di dati vuoti o NULLs mancant | Esattame te un nome di colonna | `Column [C olumn] mpleter s` | Sì | Sì | Sì | Sì | Sì | Sì |
| CustomS | I clienti possono implemer are quasi tutti i tipi di controlli di qualità dei dati in SQL. | Un'istruz ione SQL<br><br>(Facoltat ivo) Una soglia a livello di riga | `Dataset .Custor L`<br><br>Metrica aggiuntiv a quando viene fornita la soglia a livello di riga:<br><br>`Dataset .Custor L.Comp] nce` | Sì | No | Sì, quando viene fornita la soglia a livello di riga | Sì | No | No |

| RuleTyp | Descripti on | Argumer (Argome i) | Metriche riportate | Supporta o come regola? | Supporta o come analizzat ore? | Restituis ce risultati a livello di riga? | Supporto per regole dinamich ? | Genera osservaz oni | Supporta la sintassi della clausola Where? |
|---|---|---|---|---|---|---|---|---|---|
| DataFres ness | Verifica se i dati sono aggiorna i. | Esattame te un nome di colonna | `Column [C olumn]. taFresh ss.Comp ance` | Sì | No | Sì | No | No | Sì |
| DatasetM tch | Confront due set di dati e identific a se sono sincroniz zati. | Nome di un set di dati di riferimen to<br><br>Una mappatu delle colonne<br><br>(Facoltat ivo) Colonne da controlla re per cercare corrispor denze | `Dataset [ Referer Dataset ias].Da setMatc` | Sì | No | Sì | Sì | No | No |

| RuleTyp | Descripti on | Argumer (Argome i) | Metriche riportate | Supporta o come regola? | Supporta o come analizzat ore? | Restituis ce risultati a livello di riga? | Supporto per regole dinamich ? | Genera osservaz oni | Supporta la sintassi della clausola Where? |
|---|---|---|---|---|---|---|---|---|---|
| DistinctV aluesCou t | Verifica la presenza di valori duplicati . | Esattame te un nome di colonna | `Column [C olumn]. stinctV uesCour` | Sì | Sì | Sì | Sì | Sì | Sì |
| DetectAr malies | Verifica la presenza di anomalie nelle metriche riportate di un altro tipo di regola. | Un tipo di regola | Metriche riportate dall'argo mento del tipo di regola | Sì | No | No | No | No | No |
| Entropia | Verifica l'entropi a dei dati. | Esattame te un nome di colonna | `Column [C olumn]. tropy` | Sì | Sì | No | Sì | No | Sì |

| RuleTyp | Descripti on | Argumer (Argome i) | Metriche riportate | Supporta o come regola? | Supporta o come analizzat ore? | Restituis ce risultati a livello di riga? | Supporto per regole dinamich ? | Genera osservaz oni | Supporta la sintassi della clausola Where? |
|---|---|---|---|---|---|---|---|---|---|
| IsComple e | Verifica se il 100% dei dati è completo | Esattame te un nome di colonna | `Column. [C olumn]. mpleter s` | Sì | No | Sì | No | No | Sì |
| IsPrimary Key | Verifica se una colonna è una chiave primaria (non NULL e univoca). | Esattame te un nome di colonna | Per colonna singola: `Column. [C olumn]. iquenes` Per più colonne: `Multico mn. [Comma Delimit Columns Uniquer s` | Sì | No | Sì | No | No | Sì |

| RuleTyp | Descripti on | Argumer (Argome i) | Metriche riportate | Supporta o come regola? | Supporta o come analizzat ore? | Restituis ce risultati a livello di riga? | Supporto per regole dinamich ? | Genera osservaz oni | Supporta la sintassi della clausola Where? |
|---|---|---|---|---|---|---|---|---|---|
| IsUnique | Verifica se il 100% dei dati è univoco. | Esattame te un nome di colonna | `Column [C olumn] iquenes` | Sì | No | Sì | No | No | Sì |
| Media | Verifica se la media corrispor de alla soglia impostat . | Esattame te un nome di colonna | `Column [C olumn] an` | Sì | Sì | Sì | Sì | No | Sì |

| RuleTyp | Descripti on | Argumer (Argome i) | Metriche riportate | Supporta o come regola? | Supporta o come analizzat ore? | Restituis ce risultati a livello di riga? | Supporto per regole dinamich ? | Genera osservaz oni | Supporta la sintassi della clausola Where? |
|---|---|---|---|---|---|---|---|---|---|
| Referent alIntegri ty | Verifica se due set di dati hanno un'integr ità referenzi ale. | Uno o più nomi di colonne dal set di dati<br><br>Uno o più nomi di colonne dal set di dati di riferimen to | `Column. [R eferenc ataset/ as].Re ential egrity` | Sì | No | Sì | Sì | No | No |
| RowCou | Verifica se il conteggi dei record corrispor de a una soglia. | Nessuno | `Dataset .RowCou` | Sì | Sì | No | Sì | Sì | Sì |

| RuleTyp | Descripti on | Argumer (Argome i) | Metriche riportate | Supporta o come regola? | Supporta o come analizzat ore? | Restituis ce risultati a livello di riga? | Supporto per regole dinamich ? | Genera osservaz oni | Supporta la sintassi della clausola Where? |
|---|---|---|---|---|---|---|---|---|---|
| RowCou atch | Verifica se il conteggi dei record tra due set di dati corrispor de. | Alias del set di dati di riferimen to | `Dataset [ Referer Dataset ias].Ro ountMat` | Sì | No | No | Sì | No | No |
| Standarc eviation | Verifica se la deviazior e standard corrispor de alla soglia. | Esattame te un nome di colonna | `Column. [C olumn]. andardD iation` | Sì | Sì | Sì | Sì | No | Sì |

| RuleType | Description | Argumenti (Argomenti) | Metriche riportate | Supporta come regola? | Supporta come analizzatore? | Restituisce risultati a livello di riga? | Supporto per regole dinamiche? | Genera osservazioni | Supporta la sintassi della clausola Where? |
|---|---|---|---|---|---|---|---|---|---|
| SchemaMatch | Verifica se il numero di record tra due set di dati corrisponde. | Alias del set di dati di riferimento | `Dataset[Referer Dataset ias].SchemaMatch` | Sì | No | No | Sì | No | No |
| Somma | Verifica se la somma corrisponde a una soglia impostata. | Esattamente un nome di colonna | `Column.[Column].Sum` | Sì | Sì | No | Sì | No | Sì |
| Univocità | Verifica se l'unicità del set di dati corrisponde alla soglia. | Esattamente un nome di colonna | `Column.[Column].Uniqueness` | Sì | Sì | Sì | Sì | No | Sì |

| RuleType | Description | Argumenti (Argomenti) | Metriche riportate | Supporta come regola? | Supporta come analizzatore? | Restituisce risultati a livello di riga? | Supporto per regole dinamiche? | Genera osservazioni | Supporta la sintassi della clausola Where? |
|---|---|---|---|---|---|---|---|---|---|
| UniqueValueRatio | Verifica se la porzione di valore univoco corrisponde alla soglia. | Esattamente un nome di colonna | `Column.[Column].UniqueValueRatio` | Sì | Sì | Sì | Sì | No | Sì |
| FileFreshness | Controlla se i file in Amazon S3 sono aggiornati. | Percorso del file o della cartella e soglia. | `Dataset.FileFreshness.Compliance`<br>`Dataset.FileCount` | Sì | No | No | No | No | No |

| RuleType | Description | Argumenti (Argomenti) | Metriche riportate | Supporta o come regola? | Supporta o come analizzatore? | Restituisce risultati a livello di riga? | Supporto per regole dinamiche? | Genera osservazioni | Supporta la sintassi della clausola Where? |
|---|---|---|---|---|---|---|---|---|---|
| FileMatch | Controlla se il contenuto del file corrisponde a un checksum o a un altro file. Questa regola usa i checksum per verificare se due file sono uguali. | Percorso del file o della cartella di origine e percorso del file o della cartella di destinazione. | Non viene generata alcuna statistica. | Sì | No | No | No | No | No |

| RuleTyp | Descripti on | Argumer (Argome i) | Metriche riportate | Supporta o come regola? | Supporta o come analizzat ore? | Restituis ce risultati a livello di riga? | Supporto per regole dinamich ? | Genera osservaz oni | Supporta la sintassi della clausola Where? |
|---|---|---|---|---|---|---|---|---|---|
| FileSize | Controlla se la dimensio e di un file corrispor de a una condizior e specifica ta. | Percorso del file o della cartella e soglia. | `Dataset .FileSi .Compli ce`<br>`Dataset .FileCo t`<br>`Dataset .Maximu ileSize`<br>`Dataset .Minimu ileSize` | Sì | No | No | No | No | No |
| FileUniqu eness | Controlla se i file sono unici utilizzan do i checksur | Percorso del file o della cartella e soglia. | `Dataset .FileUn uenessR io`<br>`Dataset .FileCo t` | Sì | No | No | No | No | No |

Argomenti

- AggregateMatch
- ColumnCorrelation

- ColumnCount
- ColumnDataType
- ColumnExists
- ColumnLength
- ColumnNamesMatchPattern
- ColumnValues
- Completezza
- CustomSQL
- DataFreshness
- DatasetMatch
- DistinctValuesCount
- Entropia
- IsComplete
- IsPrimaryKey
- IsUnique
- Media
- ReferentialIntegrity
- RowCount
- RowCountMatch
- StandardDeviation
- Somma
- SchemaMatch
- Univocità
- UniqueValueRatio
- DetectAnomalies
- FileFreshness
- FileMatch
- FileUniqueness

- FileSize

## AggregateMatch

Verifica il rapporto di aggregazioni a due colonne rispetto a una determinata espressione. Questo tipo di regola funziona su più set di dati. Le aggregazioni a due colonne vengono valutate e viene prodotto un rapporto dividendo il risultato dell'aggregazione della prima colonna per il risultato dell'aggregazione della seconda colonna. Il rapporto viene confrontato con l'espressione fornita per produrre una risposta booleana.

Sintassi

Aggregazione di colonne

```
AggregateMatch <AGG_OPERATION> (<OPTIONAL_REFERENCE_ALIAS>.<COL_NAME>)
```

- AGG_OPERATION: l'operazione da utilizzare per l'aggregazione. Attualmente, sono supportati `sum` e `avg`.

  Tipi di colonna supportati: Byte, Decimal, Double, Float, Integer, Long, Short
- OPTIONAL_REFERENCE_ALIAS: è necessario fornire questo parametro se la colonna proviene da un set di dati di riferimento e non dal set di dati primario. <database_name>Se utilizzi questa regola nel AWS Glue Data Catalog, il tuo alias di riferimento deve seguire il formato ". <table_name>. <column_name>

  Tipi di colonna supportati: Byte, Decimal, Double, Float, Integer, Long, Short
- COL_NAME: il nome della colonna da aggregare.

  Tipi di colonna supportati: Byte, Decimal, Double, Float, Integer, Long, Short

Esempio: media

```
"avg(rating)"
```

Esempio: somma

```
"sum(amount)"
```

Esempio: media della colonna nel set di dati di riferimento

```
"avg(reference.rating)"
```

Regola

```
AggregateMatch <AGG_EXP_1> <AGG_EXP_2> <EXPRESSION>
```

- AGG_EXP_1: l'aggregazione della prima colonna.

  Tipi di colonna supportati: Byte, Decimal, Double, Float, Integer, Long, Short

  Tipi di colonna supportati: Byte, Decimal, Double, Float, Integer, Long, Short
- AGG_EXP_2: l'aggregazione della seconda colonna.

  Tipi di colonna supportati: Byte, Decimal, Double, Float, Integer, Long, Short

  Tipi di colonna supportati: Byte, Decimal, Double, Float, Integer, Long, Short
- EXPRESSION: un'espressione da eseguire sulla risposta del tipo di regola per produrre un valore booleano. Per ulteriori informazioni, consulta Espressioni.

Esempio: aggrega corrispondenza utilizzando la somma

La seguente regola di esempio verifica se la somma dei valori nella colonna `amount` è esattamente uguale alla somma dei valori nella colonna `total_amount`.

```
AggregateMatch "sum(amount)" "sum(total_amount)" = 1.0
```

Esempio: aggrega corrispondenza utilizzando la media

La seguente regola di esempio verifica se la media dei valori nella colonna `ratings` è pari almeno al 90% della media dei valori nella colonna `ratings` del set di dati `reference`. Il set di dati di riferimento viene fornito come origine dati aggiuntiva nell'esperienza ETL o Catalogo dati.

In AWS Glue ETL, puoi usare:

```
AggregateMatch "avg(ratings)" "avg(reference.ratings)" >= 0.9
```

Nel AWS Glue Data Catalog, puoi usare:

```
AggregateMatch "avg(ratings)" "avg(database_name.tablename.ratings)" >= 0.9
```

Comportamento nullo

La regola `AggregateMatch` ignorerà le righe con valori NULL nel calcolo dei metodi di aggregazione (somma/media). Esempio:

```
+---+-----------+
|id |units      |
+---+-----------+
|100|0          |
|101|null       |
|102|20         |
|103|null       |
|104|40         |
+---+-----------+
```

La media della colonna `units` sarà (0 + 20 + 40) / 3 = 20. Le righe 101 e 103 non vengono considerate in questo calcolo.

## ColumnCorrelation

Verifica la correlazione tra due colonne rispetto a una determinata espressione. AWS Glue Data Quality utilizza il coefficiente di correlazione di Pearson per misurare la correlazione lineare tra due colonne. Il risultato è un numero compreso tra -1 e 1 che misura la forza e la direzione della relazione.

Sintassi

```
ColumnCorrelation <COL_1_NAME> <COL_2_NAME> <EXPRESSION>
```

- COL_1_NAME: il nome della prima colonna in base alla quale si desidera valutare la regola di qualità dei dati.

  Tipi di colonna supportati: Byte, Decimal, Double, Float, Integer, Long, Short
- COL_2_NAME: il nome della seconda colonna in base alla quale si desidera valutare la regola di qualità dei dati.

  Tipi di colonna supportati: Byte, Decimal, Double, Float, Integer, Long, Short

- EXPRESSION: un'espressione da eseguire sulla risposta del tipo di regola per produrre un valore booleano. Per ulteriori informazioni, consulta Espressioni.

Esempio: correlazione tra colonne

La seguente regola di esempio verifica se il coefficiente di correlazione tra le colonne `height` e `weight` ha una forte correlazione positiva (un valore del coefficiente maggiore di 0,8).

```
ColumnCorrelation "height" "weight" > 0.8
```

```
ColumnCorrelation "weightinkgs" "Salary" > 0.8 where "weightinkgs > 40"
```

Regole dinamiche di esempio

- `ColumnCorrelation "colA" "colB" between min(last(10)) and max(last(10))`
- `ColumnCorrelation "colA" "colB" < avg(last(5)) + std(last(5))`

Comportamento nullo

La regola `ColumnCorrelation` ignorerà le righe con valori `NULL` nel calcolo della correlazione. Esempio:

```
+---+-----------+
|id |units      |
+---+-----------+
|100|0          |
|101|null       |
|102|20         |
|103|null       |
|104|40         |
+---+-----------+
```

Le righe 101 e 103 verranno ignorate e la `ColumnCorrelation` sarà 1,0.

## ColumnCount

Verifica il conteggio delle righe del set di dati primario rispetto a una determinata espressione. Nell'espressione, è possibile specificare il numero di colonne o un intervallo di colonne utilizzando operatori come > e <.

Sintassi

```
ColumnCount <EXPRESSION>
```

- EXPRESSION: un'espressione da eseguire sulla risposta del tipo di regola per produrre un valore booleano. Per ulteriori informazioni, consulta Espressioni.

Esempio: controllo numerico del conteggio delle colonne

La seguente regola di esempio controlla se il conteggio delle colonne rientra in un determinato intervallo.

```
ColumnCount between 10 and 20
```

Regole dinamiche di esempio

- `ColumnCount >= avg(last(10))`
- `ColumnCount between min(last(10))-1 and max(last(10))+1`

## ColumnDataType

Controlla se i valori in una determinata colonna possono essere trasmessi in Apache Spark nel tipo fornito. Accetta un'espressione `with threshold` per verificare la presenza di un sottoinsieme di valori nella colonna.

Sintassi

```
ColumnDataType <COL_NAME> = <EXPECTED_TYPE>
```

- COL_NAME: il nome della colonna in base alla quale si desidera valutare la regola di qualità dei dati.

  Tipi di colonne supportati: tipo stringa

  Tipi di colonna supportati: Byte, Decimal, Double, Float, Integer, Long, Short
- EXPECTED_TYPE: il tipo di valori previsto nella colonna.

  Valori supportati: Boolean, Date, Timestamp, Integer, Double, Float, Long

Tipi di colonna supportati: Byte, Decimal, Double, Float, Integer, Long, Short

- EXPRESSION: un'espressione facoltativa per specificare la percentuale di valori che devono essere del tipo previsto.

  Tipi di colonna supportati: Byte, Decimal, Double, Float, Integer, Long, Short

Esempio: il tipo di dato nella colonna rappresentato come stringa è intero

La seguente regola di esempio verifica se i valori nella colonna specificata, che è di tipo stringa, possono essere convertiti in numeri interi.

```
ColumnDataType "colA" = "INTEGER"
```

Esempio: verifica di un sottoinsieme dei valori delle colonne, di tipo intero ma rappresentati come stringhe

La seguente regola di esempio verifica se più del 90% dei valori nella colonna data, che è di tipo stringa, possono essere convertiti in numeri interi.

```
ColumnDataType "colA" = "INTEGER" with threshold > 0.9
```

## ColumnExists

Verifica se esiste una colonna.

Sintassi

```
ColumnExists <COL_NAME>
```

- COL_NAME: il nome della colonna in base alla quale si desidera valutare la regola di qualità dei dati.

  Tipi di colonna supportati: qualsiasi tipo di colonna

Esempio: la colonna esiste

La seguente regola di esempio verifica se la colonna denominata `Middle_Name` esiste.

```
ColumnExists "Middle_Name"
```

## ColumnLength

Verifica se la lunghezza di ogni riga in una colonna è conforme a una determinata espressione.

### Sintassi

```
ColumnLength <COL_NAME><EXPRESSION>
```

- COL_NAME: il nome della colonna in base alla quale si desidera valutare la regola di qualità dei dati.

  Tipi di colonne supportati: String
- EXPRESSION: un'espressione da eseguire sulla risposta del tipo di regola per produrre un valore booleano. Per ulteriori informazioni, consulta Espressioni.

### Esempio: lunghezza della riga della colonna

La seguente regola di esempio verifica se il valore in ogni riga della colonna denominata `Postal_Code` è lungo 5 caratteri.

```
ColumnLength "Postal_Code" = 5
ColumnLength "weightinkgs" = 2 where "weightinkgs > 10"
```

### Comportamento nullo

La regola `ColumnLength` considera i valori `NULL` come stringhe di lunghezza pari a 0. Per una riga `NULL`:

```
ColumnLength "Postal_Code" > 4 # this will fail
```

```
ColumnLength "Postal_Code" < 6 # this will succeed
```

Il seguente esempio di regola composta fornisce un modo per rifiutare esplicitamente i valori `NULL`:

```
(ColumnLength "Postal_Code" > 4) AND (ColumnValues "Postal_Code" != NULL)
```

## ColumnNamesMatchPattern

Verifica se i nomi di tutte le colonne del set di dati primario corrispondono all'espressione regolare specificata.

Sintassi

```
ColumnNamesMatchPattern <PATTERN>
```

- PATTERN: il modello in base al quale si desidera valutare la regola di qualità dei dati.

  Tipi di colonna supportati: Byte, Decimal, Double, Float, Integer, Long, Short

Esempio: i nomi delle colonne corrispondono al modello

La seguente regola di esempio verifica se tutte le colonne iniziano con il prefisso "aws_"

```
ColumnNamesMatchPattern "aws_.*"
ColumnNamesMatchPattern "aws_.*" where "weightinkgs > 10"
```

## ColumnValues

Esegue un'espressione rispetto ai valori di una colonna.

Sintassi

```
ColumnValues <COL_NAME> <EXPRESSION>
```

- COL_NAME: il nome della colonna in base alla quale si desidera valutare la regola di qualità dei dati.

  Tipi di colonna supportati: qualsiasi tipo di colonna
- EXPRESSION: un'espressione da eseguire sulla risposta del tipo di regola per produrre un valore booleano. Per ulteriori informazioni, consulta Espressioni.

Esempio: valori consentiti

La seguente regola di esempio verifica se ogni valore nella colonna specificata si trova in un set di valori consentiti (compresi valori null, vuoti e stringhe contenenti solo spazi bianchi).

```
ColumnValues "Country" in [ "US", "CA", "UK", NULL, EMPTY, WHITESPACES_ONLY ]
ColumnValues "gender" in ["F", "M"] where "weightinkgs < 10"
```

Esempio: espressione regolare

La seguente regola di esempio verifica i valori di una colonna rispetto a un'espressione regolare.

```
ColumnValues "First_Name" matches "[a-zA-Z]*"
```

Esempio: valori data

La seguente regola di esempio verifica i valori di una colonna data rispetto a un'espressione data.

```
ColumnValues "Load_Date" > (now() - 3 days)
```

Esempio: valori numerici

La seguente regola di esempio verifica se i valori delle colonne corrispondono a un determinato vincolo numerico.

```
ColumnValues "Customer_ID" between 1 and 2000
```

Comportamento nullo

Per tutte le regole `ColumnValues` (diverse da `!=` e`NOT IN`), le righe `NULL` non soddisfano la regola. Se la regola ha esito negativo a causa di un valore null, il motivo dell'errore è il seguente:

```
Value: NULL does not meet the constraint requirement!
```

Il seguente esempio di regola composta fornisce un modo per accettare esplicitamente i valori `NULL`:

```
(ColumnValues "Age" > 21) OR (ColumnValues "Age" = NULL)
```

ColumnValues Le regole negate che utilizzano la `not in` sintassi `!=` and verranno passate per le `NULL` righe. Esempio:

```
ColumnValues "Age" != 21
```

```
ColumnValues "Age" not in [21, 22, 23]
```

Gli esempi seguenti forniscono un modo per rifiutare esplicitamente i valori NULL

```
(ColumnValues "Age" != 21) AND (ColumnValues "Age" != NULL)
```

```
ColumnValues "Age" not in [21, 22, 23, NULL]
```

## Completezza

Verifica la percentuale di valori completi (non nulli) in una colonna rispetto a una determinata espressione.

Sintassi

```
Completeness <COL_NAME> <EXPRESSION>
```

- COL_NAME: il nome della colonna in base alla quale si desidera valutare la regola di qualità dei dati.

  Tipi di colonna supportati: qualsiasi tipo di colonna
- EXPRESSION: un'espressione da eseguire sulla risposta del tipo di regola per produrre un valore booleano. Per ulteriori informazioni, consulta Espressioni.

Esempio: percentuale di valore nullo

Le seguenti regole di esempio controllano se più del 95% dei valori in una colonna sono completi.

```
Completeness "First_Name" > 0.95
Completeness "First_Name" > 0.95 where "weightinkgs > 10"
```

Regole dinamiche di esempio

- `Completeness "colA" between min(last(5)) - 1 and max(last(5)) + 1`
- `Completeness "colA" <= avg(last(10))`

Comportamento nullo

Nota sui formati di dati CSV: le righe vuote nelle colonne CSV possono presentare diversi comportamenti.

- Se una colonna è di tipo `String`, la riga vuota verrà riconosciuta come una stringa vuota e non violerà la regola `Completeness`.
- Se una colonna è di un altro tipo di dati come `Int`, la riga vuota verrà riconosciuta come `NULL` e non rispetterà la regola `Completeness`.

## CustomSQL

Questo tipo di regola è stato esteso per supportare due casi d'uso:

- Esegui un'istruzione SQL personalizzata su un set di dati e controlla il valore restituito rispetto a una determinata espressione.
- Esegui un'istruzione SQL personalizzata specificando un nome di colonna nell'istruzione SELECT in base alla quale eseguire un confronto con alcune condizioni per ottenere risultati a livello di riga.

Sintassi

```
CustomSql <SQL_STATEMENT> <EXPRESSION>
```

- SQL_STATEMENT: un'istruzione SQL che restituisce un singolo valore numerico, racchiuso tra virgolette doppie.
- EXPRESSION: un'espressione da eseguire sulla risposta del tipo di regola per produrre un valore booleano. Per ulteriori informazioni, consulta Espressioni.

Esempio: SQL personalizzato per recuperare il risultato di una regola generale

Questa regola di esempio utilizza un'istruzione SQL per recuperare il numero di record per un set di dati. La regola verifica quindi che il conteggio dei record sia compreso tra 10 e 20.

```
CustomSql "select count(*) from primary" between 10 and 20
```

Esempio: SQL personalizzato per recuperare i risultati a livello di riga

Questa regola di esempio utilizza un'istruzione SQL personalizzata specificando un nome di colonna nell'istruzione SELECT in base alla quale eseguire un confronto con alcune condizioni per ottenere

risultati a livello di riga. Un'espressione di condizione di soglia definisce una soglia di quanti record devono avere esito negativo perché l'intera regola abbia esito negativo. Tieni presente che una regola non può contenere contemporaneamente una condizione e una parola chiave.

```
CustomSql "select Name from primary where Age  > 18"
```

or

```
CustomSql "select Name from primary where Age > 18" with threshold  > 3
```

> **Important**
>
> L'alias `primary` sostituisce il nome del set di dati che si desidera valutare. Quando lavori con job ETL visivi sulla console, `primary` rappresenta sempre il `DynamicFrame` passato alla trasformazione `EvaluateDataQuality.apply()`. Quando si utilizza il AWS Glue Data Catalog per eseguire attività di qualità dei dati su una tabella, `primary` rappresenta la tabella.

Se utilizzi Catalogo dati AWS Glue, puoi anche utilizzare i nomi effettivi delle tabelle:

```
CustomSql "select count(*) from database.table" between 10 and 20
```

È inoltre possibile effettuare il join di più tabelle per confrontare diversi elementi di dati:

```
CustomSql "select count(*) from database.table inner join database.table2 on id1 = id2"
 between 10 and 20
```

In ETL di AWS Glue, CustomSQL è in grado di identificare i record che non hanno superato i controlli di qualità dei dati. Affinché ciò funzioni, è necessario restituire i record che fanno parte della tabella primaria per la quale si sta valutando la qualità dei dati. I record restituiti come parte della query sono considerati riusciti, mentre i record che non vengono restituiti sono considerati non riusciti. Ciò funziona unendo il risultato della query CustomSQL con il set di dati originale. In base alla complessità della query SQL, potrebbero esserci implicazioni sulle prestazioni.

Per farlo:

- Bisogna selezionare almeno 1 colonna dalla tabella principale.

  - `select count(*) from primary` è una query valida per la regola OVERALL CustomSQL DQ, ma non per Row Level Custom SQL.
  - Questa regola genererà un errore durante la valutazione: `The output from CustomSQL must contain at least one column that matches the input dataset for AWS Glue Data Quality to provide row level results. The SQL query is a valid query but the columns from the SQL result are not present in the Input Dataset. Ensure that matching columns are returned from the SQL.`
- Nella query SQL, seleziona una "chiave primaria" dalla tabella oppure seleziona un set di colonne che formano una chiave composita. Altrimenti, potrebbero verificarsi risultati incoerenti, a causa della corrispondenza di righe duplicate, e un calo delle prestazioni.
- Seleziona le chiavi SOLO dalla tabella principale e non dalle tabelle di riferimento.

La regola seguente garantirà che i record con età < 100 vengano identificati come riusciti e i record al di sopra vengano contrassegnati come non riusciti.

```
CustomSql "select id from primary where age < 100"
```

Questa regola CustomSQL sarà soddisfatta quando il 50% dei record hanno un'età > 10 e identificherà anche i record che non soddisfano la regola. I record restituiti da questa regola CustomSQL verranno considerati superati, mentre quelli non restituiti verranno considerati non superati.

```
CustomSQL "select ID, CustomerID from primary where age > 10" with threshold > 0.5
```

Nota: la regola CustomSQL avrà esito negativo se si restituiscono record che non sono disponibili nel set di dati.

## DataFreshness

Verifica l'aggiornamento dei dati in una colonna valutando la differenza tra l'ora corrente e i valori di una colonna di date. È possibile specificare un'espressione basata sul tempo per questo tipo di regola per assicurarsi che i valori delle colonne siano aggiornati.

Sintassi

```
DataFreshness <COL_NAME> <EXPRESSION>
```

- COL_NAME: il nome della colonna in base alla quale si desidera valutare la regola di qualità dei dati.

  Tipi di colonne supportati: Data
- EXPRESSION: un'espressione numerica in ore o giorni. È necessario specificare l'unità di tempo nell'espressione.

Esempio: freschezza dei dati

Le seguenti regole di esempio controllano la freschezza dei dati, ovvero quanto sono aggiornati.

```
DataFreshness "Order_Date" <= 24 hours
DataFreshness "Order_Date" between 2 days and 5 days
```

Comportamento nullo

Le regole `DataFreshness` avranno esito negativo per le righe con valori `NULL`. Se la regola ha esito negativo a causa di un valore null, il motivo dell'errore è il seguente:

```
80.00 % of rows passed the threshold
```

dove il 20% delle righe che hanno dato esito negativo include le righe con valori `NULL`.

Il seguente esempio di regola composta fornisce un modo per accettare esplicitamente i valori `NULL`:

```
(DataFreshness "Order_Date" <= 24 hours) OR (ColumnValues "Order_Date" = NULL)
```

Aggiornamento dei dati per oggetti Amazon S3

A volte è necessario verificare l'aggiornamento dei dati in base alla data e all'ora di creazione del file Amazon S3. A tale scopo, puoi usare il seguente codice per ottenere il timestamp, aggiungerlo al dataframe e poi applicare i controlli relativi all'aggiornamento dei dati.

```
df = glueContext.create_data_frame.from_catalog(database = "default", table_name =
 "mytable")
df = df.withColumn("file_ts", df["_metadata.file_modification_time"])

Rules = [
 DataFreshness "file_ts" < 24 hours
```

```
]
```

## DatasetMatch

Verifica se i dati nel set di dati primario corrispondono ai dati in un set di dati di riferimento. Il join dei due set di dati viene effettuato utilizzando le mappature delle colonne chiave fornite. È possibile fornire mappature di colonne aggiuntive se si desidera verificare l'uguaglianza dei dati solo in quelle colonne. Nota che, DataSetMatchper funzionare, le tue chiavi di join devono essere uniche e non devono essere NULL (deve essere una chiave primaria). Se non soddisfi queste condizioni, riceverai il messaggio di errore "Provided key map not suitable for given data frames". Nei casi in cui non è possibile disporre di chiavi di unione univoche, si consiglia di utilizzare altri tipi di regole, ad esempio la corrispondenza nei dati AggregateMatchdi riepilogo.

Sintassi

```
DatasetMatch <REFERENCE_DATASET_ALIAS> <JOIN CONDITION WITH
 MAPPING> <OPTIONAL_MATCH_COLUMN_MAPPINGS> <EXPRESSION>
```

- REFERENCE_DATASET_ALIAS: l'alias del set di dati di riferimento con cui confronti i dati del set di dati primario.
- KEY_COLUMN_MAPPINGS: un elenco separato da virgole di nomi di colonne che formano una chiave nei set di dati. Se i nomi delle colonne non sono uguali in entrambi i set di dati, è necessario separarli con un `->`
- OPTIONAL_MATCH_COLUMN_MAPPINGS: puoi fornire questo parametro se desideri verificare la corrispondenza dei dati solo in determinate colonne. Utilizza la stessa sintassi delle mappature delle colonne chiave. Se questo parametro non viene fornito, abbineremo i dati in tutte le colonne rimanenti. Le colonne rimanenti (non chiave) devono avere gli stessi nomi in entrambi i set di dati.
- EXPRESSION: un'espressione da eseguire sulla risposta del tipo di regola per produrre un valore booleano. Per ulteriori informazioni, consulta Espressioni.

Esempio: abbina i set di dati definiti utilizzando la colonna ID

La seguente regola di esempio verifica che più del 90% del set di dati primario corrisponda al set di dati di riferimento, utilizzando la colonna "ID" per il join dei due set di dati. In questo caso confronta tutte le colonne.

```
DatasetMatch "reference" "ID" >= 0.9
```

Esempio: abbina i set di dati del set utilizzando più colonne chiave

Nell'esempio seguente, il set di dati primario e il set di dati di riferimento hanno nomi diversi per le colonne chiave. `ID_1` e `ID_2` insieme formano una chiave composita nel set di dati primario. `ID_ref1` e `ID_ref2` insieme formano una chiave composita nel set di dati di riferimento. In questo scenario, è possibile utilizzare la sintassi speciale per fornire i nomi delle colonne.

```
DatasetMatch "reference" "ID_1->ID_ref1,ID_2->ID_ref2" >= 0.9
```

Esempio: abbina i set di dati del set utilizzando più colonne chiave e verifica che le colonne specifiche corrispondano

Questo esempio si basa sull'esempio precedente. Vogliamo verificare che solo la colonna contenente gli importi corrisponda. Questa colonna è denominata `Amount1` nel set di dati primario e `Amount2` nel set di dati di riferimento. Vuoi una corrispondenza esatta.

```
DatasetMatch "reference" "ID_1->ID_ref1,ID_2->ID_ref2" "Amount1->Amount2" >= 0.9
```

## DistinctValuesCount

Seleziona il numero di valori distinti in una colonna rispetto a una determinata espressione.

Sintassi

```
DistinctValuesCount <COL_NAME> <EXPRESSION>
```

- COL_NAME: il nome della colonna in base alla quale si desidera valutare la regola di qualità dei dati.

  Tipi di colonna supportati: qualsiasi tipo di colonna
- EXPRESSION: un'espressione da eseguire sulla risposta del tipo di regola per produrre un valore booleano. Per ulteriori informazioni, consulta Espressioni.

Esempio: conteggio dei valori distinti delle colonne

La seguente regola di esempio verifica che la colonna denominata `State` contenga più di 3 valori distinti.

```
DistinctValuesCount "State" > 3
DistinctValuesCount "Customer_ID" < 6  where "Customer_ID < 10"
```

Regole dinamiche di esempio

- `DistinctValuesCount "colA" between avg(last(10))-1 and avg(last(10))+1`
- `DistinctValuesCount "colA" <= index(last(10),2) + std(last(5))`

## Entropia

Verifica se il valore di entropy di una colonna corrisponde a una determinata espressione. L'entropia misura il livello di informazioni contenute in un messaggio. Data la distribuzione della probabilità sui valori in una colonna, l'entropia descrive quanti bit sono necessari per identificare un valore.

Sintassi

```
Entropy <COL_NAME> <EXPRESSION>
```

- COL_NAME: il nome della colonna in base alla quale si desidera valutare la regola di qualità dei dati.

  Tipi di colonna supportati: qualsiasi tipo di colonna
- EXPRESSION: un'espressione da eseguire sulla risposta del tipo di regola per produrre un valore booleano. Per ulteriori informazioni, consulta Espressioni.

Esempio: entropia delle colonne

La seguente regola di esempio verifica che la colonna denominata `Feedback` abbia un valore di entropia maggiore di uno.

```
Entropy "Star_Rating" > 1
Entropy "First_Name" > 1  where "Customer_ID < 10"
```

Regole dinamiche di esempio

- `Entropy "colA" < max(last(10))`
- `Entropy "colA" between min(last(10)) and max(last(10))`

## IsComplete

Verifica se tutti i valori in una colonna sono completi (non nulli).

Sintassi

```
IsComplete <COL_NAME>
```

- COL_NAME: il nome della colonna in base alla quale si desidera valutare la regola di qualità dei dati.

  Tipi di colonna supportati: qualsiasi tipo di colonna

Esempio: valori nulli

L'esempio seguente verifica se tutti i valori in una colonna denominata `email` non sono nulli.

```
IsComplete "email"
IsComplete "Email" where "Customer_ID between 1 and 50"
IsComplete "Customer_ID"  where "Customer_ID < 16 and Customer_ID != 12"
IsComplete "passenger_count" where "payment_type<>0"
```

Comportamento nullo

Nota sui formati di dati CSV: le righe vuote nelle colonne CSV possono presentare diversi comportamenti.

- Se una colonna è di tipo `String`, la riga vuota verrà riconosciuta come una stringa vuota e non violerà la regola `Completeness`.
- Se una colonna è di un altro tipo di dati come `Int`, la riga vuota verrà riconosciuta come `NULL` e non rispetterà la regola `Completeness`.

## IsPrimaryKey

Verifica se una colonna contiene una chiave primaria. Una colonna contiene una chiave primaria se tutti i valori nella colonna sono univoci e completi (non nulli). Puoi anche verificare la presenza di chiavi primarie con più colonne.

Sintassi

```
IsPrimaryKey <COL_NAME>
```

- COL_NAME: il nome della colonna in base alla quale si desidera valutare la regola di qualità dei dati.

  Tipi di colonna supportati: qualsiasi tipo di colonna

Esempio: chiave primaria

La seguente regola di esempio verifica se la colonna denominata `Customer_ID` contiene una chiave primaria.

```
IsPrimaryKey "Customer_ID"
IsPrimaryKey "Customer_ID" where "Customer_ID < 10"
```

Esempio: chiave primaria con più colonne. Ognuno degli esempi seguenti è valido.

```
IsPrimaryKey "colA" "colB"
IsPrimaryKey "colA" "colB" "colC"
IsPrimaryKey colA "colB" "colC"
```

## IsUnique

Verifica se tutti i valori in una colonna sono univoci e restituisce un valore booleano.

Sintassi

```
IsUnique <COL_NAME>
```

- COL_NAME: il nome della colonna in base alla quale si desidera valutare la regola di qualità dei dati.

  Tipi di colonna supportati: qualsiasi tipo di colonna

Esempi

La seguente regola di esempio verifica se tutti i valori in una colonna denominata `email` sono univoci.

```
IsUnique "email"
IsUnique "Customer_ID" where "Customer_ID < 10"]
```

La seguente regola di esempio controlla più colonne.

```
IsUnique "vendorid" "tpep_pickup_datetime"
```

## Media

Verifica se la media di tutti i valori in una colonna corrisponde a una determinata espressione.

Sintassi

```
Mean <COL_NAME> <EXPRESSION>
```

- COL_NAME: il nome della colonna in base alla quale si desidera valutare la regola di qualità dei dati.

  Tipi di colonna supportati: Byte, Decimal, Double, Float, Integer, Long, Short
- EXPRESSION: un'espressione da eseguire sulla risposta del tipo di regola per produrre un valore booleano. Per ulteriori informazioni, consulta Espressioni.

Esempio: valore medio

La seguente regola di esempio verifica se la media di tutti i valori di una colonna supera una soglia.

```
Mean "Star_Rating" > 3
Mean "Salary" < 6200 where "Customer_ID < 10"
```

Regole dinamiche di esempio

- `Mean "colA" > avg(last(10)) + std(last(2))`
- `Mean "colA" between min(last(5)) - 1 and max(last(5)) + 1`

Comportamento nullo

La regola `Mean` ignorerà le righe con valori `NULL` nel calcolo della media. Esempio:

```
+---+-----------+
|id |units      |
+---+-----------+
|100|0          |
|101|null       |
|102|20         |
|103|null       |
|104|40         |
+---+-----------+
```

La media della colonna `units` sarà (0 + 20 + 40) / 3 = 20. Le righe 101 e 103 non vengono considerate in questo calcolo.

## ReferentialIntegrity

Verifica in che misura i valori di un set di colonne nel set di dati primario siano un sottoinsieme dei valori di un set di colonne in un set di dati di riferimento.

Sintassi

```
ReferentialIntegrity <PRIMARY_COLS> <REFERENCE_DATASET_COLS> <EXPRESSION>
```

- PRIMARY_COLS: un elenco separato da virgole dei nomi delle colonne nel set di dati primario.

  Tipi di colonna supportati: Byte, Decimal, Double, Float, Integer, Long, Short
- REFERENCE_DATASET_COLS: questo parametro contiene due parti separate da un punto. La prima parte è l'alias del set di dati di riferimento. La seconda parte è l'elenco separato da virgole dei nomi delle colonne nel set di dati di riferimento racchiuso tra parentesi.

  Tipi di colonna supportati: Byte, Decimal, Double, Float, Integer, Long, Short
- EXPRESSION: un'espressione da eseguire sulla risposta del tipo di regola per produrre un valore booleano. Per ulteriori informazioni, consulta Espressioni.

Esempio: verifica l'integrità referenziale di una colonna di codice postale

La seguente regola di esempio verifica che più del 90% dei valori nella colonna `zipcode` del set di dati primario siano presenti nella colonna `zipcode` del set di dati `reference`.

```
ReferentialIntegrity "zipcode" "reference.zipcode" >= 0.9
```

Esempio: verifica l'integrità referenziale delle colonne relative alla città e allo stato

Nell'esempio seguente, nel set di dati primario e nel set di dati di riferimento sono presenti colonne contenenti informazioni sulla città e sullo stato. I nomi delle colonne sono diversi in entrambi i set di dati. La regola verifica se il set di valori delle colonne nel set di dati primario è esattamente uguale al set di valori delle colonne nel set di dati di riferimento.

```
ReferentialIntegrity "city,state" "reference.{ref_city,ref_state}" = 1.0
```

Regole dinamiche di esempio

- `ReferentialIntegrity "city,state" "reference.{ref_city,ref_state}" > avg(last(10))`
- `ReferentialIntegrity "city,state" "reference.{ref_city,ref_state}" between min(last(10)) - 1 and max(last(10)) + 1`

## RowCount

Verifica il numero di righe di un set di dati rispetto a una determinata espressione. Nell'espressione, è possibile specificare il numero di righe o un intervallo di righe utilizzando operatori come > e <.

Sintassi

```
RowCount <EXPRESSION>
```

- EXPRESSION: un'espressione da eseguire sulla risposta del tipo di regola per produrre un valore booleano. Per ulteriori informazioni, consulta Espressioni.

Esempio: controllo numerico del conteggio delle righe

La seguente regola di esempio controlla se il conteggio delle righe rientra in un determinato intervallo.

```
RowCount between 10 and 100
RowCount between 1 and 50 where "Customer_ID < 10"
```

Regole dinamiche di esempio

```
RowCount > avg(last(10)) *0.8
```

## RowCountMatch

Verifica il rapporto tra il conteggio delle righe del set di dati primario e il conteggio delle righe di un set di dati di riferimento rispetto all'espressione data.

Sintassi

```
RowCountMatch <REFERENCE_DATASET_ALIAS> <EXPRESSION>
```

- REFERENCE_DATASET_ALIAS: l'alias del set di dati di riferimento con cui confrontare il conteggio delle righe.

  Tipi di colonna supportati: Byte, Decimal, Double, Float, Integer, Long, Short
- EXPRESSION: un'espressione da eseguire sulla risposta del tipo di regola per produrre un valore booleano. Per ulteriori informazioni, consulta Espressioni.

Esempio: controllo del conteggio delle righe rispetto a un set di dati di riferimento

La seguente regola di esempio verifica se il conteggio delle righe del set di dati primario è almeno il 90% del conteggio delle righe del set di dati di riferimento.

```
RowCountMatch "reference" >= 0.9
```

## StandardDeviation

Verifica la deviazione standard di tutti i valori di una colonna rispetto a una determinata espressione.

Sintassi

```
StandardDeviation <COL_NAME> <EXPRESSION>
```

- COL_NAME: il nome della colonna in base alla quale si desidera valutare la regola di qualità dei dati.

  Tipi di colonna supportati: Byte, Decimal, Double, Float, Integer, Long, Short

- EXPRESSION: un'espressione da eseguire sulla risposta del tipo di regola per produrre un valore booleano. Per ulteriori informazioni, consulta Espressioni.

Esempio: deviazione standard

La seguente regola di esempio verifica se la deviazione standard dei valori in una colonna denominata `colA` è inferiore a un valore specificato.

```
StandardDeviation "Star_Rating" < 1.5
StandardDeviation "Salary" < 3500 where "Customer_ID < 10"
```

Regole dinamiche di esempio

- `StandardDeviation "colA" > avg(last(10) + 0.1`
- `StandardDeviation "colA" between min(last(10)) - 1 and max(last(10)) + 1`

Comportamento nullo

La regola `StandardDeviation` ignorerà le righe con valori `NULL` nel calcolo della deviazione standard. Esempio:

```
+---+-----------+-----------+
|id |units1     |units2     |
+---+-----------+-----------+
|100|0          |0          |
|101|null       |0          |
|102|20         |20         |
|103|null       |0          |
|104|40         |40         |
+---+-----------+-----------+
```

La deviazione standard della colonna `units1` non terrà conto delle righe 101 e 103 e risulterà pari a 16,33. La deviazione standard per la colonna `units2` darà come risultato 16.

## Somma

Verifica la somma di tutti i valori in una colonna rispetto a una determinata espressione.

Sintassi

```
Sum <COL_NAME> <EXPRESSION>
```

- COL_NAME: il nome della colonna in base alla quale si desidera valutare la regola di qualità dei dati.

  Tipi di colonna supportati: Byte, Decimal, Double, Float, Integer, Long, Short
- EXPRESSION: un'espressione da eseguire sulla risposta del tipo di regola per produrre un valore booleano. Per ulteriori informazioni, consulta Espressioni.

Esempio: somma

La seguente regola di esempio verifica se la somma di tutti i valori in una colonna supera una determinata soglia.

```
Sum "transaction_total" > 500000
Sum "Salary" < 55600 where "Customer_ID < 10"
```

Regole dinamiche di esempio

- `Sum "ColA" > avg(last(10))`
- `Sum "colA" between min(last(10)) - 1 and max(last(10)) + 1`

Comportamento nullo

La regola `Sum` ignorerà le righe con valori `NULL` nel calcolo della somma. Esempio:

```
+---+-----------+
|id |units      |
+---+-----------+
|100|0          |
|101|null       |
|102|20         |
|103|null       |
|104|40         |
+---+-----------+
```

La somma della colonna `units` non terrà conto delle righe 101 e 103 e darà come risultato (0 + 20 + 40) = 60.

## SchemaMatch

Verifica se lo schema del set di dati primario corrisponde allo schema del set di dati di riferimento. Il controllo dello schema viene eseguito colonna per colonna. Lo schema di due colonne corrisponde se i nomi sono identici e i tipi sono identici. L'ordine delle colonne non è rilevante.

Sintassi

```
SchemaMatch <REFERENCE_DATASET_ALIAS> <EXPRESSION>
```

- REFERENCE_DATASET_ALIAS: l'alias del set di dati di riferimento con cui confrontare gli schemi.

  Tipi di colonna supportati: Byte, Decimal, Double, Float, Integer, Long, Short
- EXPRESSION: un'espressione da eseguire sulla risposta del tipo di regola per produrre un valore booleano. Per ulteriori informazioni, consulta Espressioni.

Esempio SchemaMatch:

La seguente regola di esempio verifica se lo schema del set di dati primario corrisponde esattamente allo schema di un set di dati di riferimento.

```
SchemaMatch "reference" = 1.0
```

## Univocità

Verifica la percentuale di valori univoci in una colonna rispetto a una determinata espressione. I valori univoci si verificano esattamente una volta.

Sintassi

```
Uniqueness <COL_NAME> <EXPRESSION>
```

- COL_NAME: il nome della colonna in base alla quale si desidera valutare la regola di qualità dei dati.

  Tipi di colonna supportati: qualsiasi tipo di colonna
- EXPRESSION: un'espressione da eseguire sulla risposta del tipo di regola per produrre un valore booleano. Per ulteriori informazioni, consulta Espressioni.

Esempio

La seguente regola di esempio verifica se la percentuale di valori univoci in una colonna corrisponde a determinati criteri numerici.

```
Uniqueness "email" = 1.0
Uniqueness "Customer_ID" != 1.0 where "Customer_ID < 10"
```

La seguente regola di esempio controlla più colonne.

```
Uniqueness "vendorid" "tpep_pickup_datetime" = 1
```

Regole dinamiche di esempio

- `Uniqueness "colA" between min(last(10)) and max(last(10))`
- `Uniqueness "colA" >= avg(last(10))`

## UniqueValueRatio

Verifica il rapporto di valori univoci in una colonna rispetto a una determinata espressione. Un rapporto di valori univoci è la frazione di valori univoci divisa per il numero di tutti i valori distinti in una colonna. I valori univoci si verificano esattamente una volta, mentre i valori distinti si verificano almeno una volta.

Ad esempio, il set `[a, a, b]` contiene un valore univoco (b) e due valori distinti (a e b). Quindi il rapporto di valori univoci del set è ½ = 0,5.

Sintassi

```
UniqueValueRatio <COL_NAME> <EXPRESSION>
```

- COL_NAME: il nome della colonna in base alla quale si desidera valutare la regola di qualità dei dati.

  Tipi di colonna supportati: qualsiasi tipo di colonna
- EXPRESSION: un'espressione da eseguire sulla risposta del tipo di regola per produrre un valore booleano. Per ulteriori informazioni, consulta Espressioni.

Esempio: rapporto di valori univoci

Questo esempio controlla il rapporto tra i valori univoci di una colonna rispetto a un intervallo di valori.

```
UniqueValueRatio "test_score" between 0 and 0.5
UniqueValueRatio "Customer_ID" between 0 and 0.9 where "Customer_ID < 10"
```

Regole dinamiche di esempio

- `UniqueValueRatio "colA" > avg(last(10))`
- `UniqueValueRatio "colA" <= index(last(10),2) + std(last(5))`

## DetectAnomalies

Rileva le anomalie per una determinata regola relativa alla qualità dei dati. Ogni esecuzione della DetectAnomalies regola comporta il salvataggio del valore valutato per la regola data. Quando vengono raccolti dati sufficienti, l'algoritmo di rilevamento delle anomalie prende tutti i dati storici relativi a quella determinata regola ed esegue il rilevamento delle anomalie. DetectAnomalies la regola fallisce quando viene rilevata un'anomalia. È possibile ottenere ulteriori informazioni sull'anomalia rilevata tramite le osservazioni.

Sintassi

```
      DetectAnomalies <RULE_NAME> <RULE_PARAMETERS>
```

`RULE_NAME`: il nome della regola che desideri valutare e per la quale desideri rilevare le anomalie. Regole supportate:

- "RowCount"
- "Completezza"
- "Univocità"
- "Media"
- "Somma"
- "StandardDeviation"
- "Entropia"

- "DistinctValuesCount"
- "UniqueValueRatio"
- "ColumnLength"
- "ColumnValues"
- "ColumnCorrelation"
- "CustomSQL"
- "ColumnCount"

`RULE_PARAMETERS`: alcune regole richiedono parametri aggiuntivi per l'esecuzione. Fai riferimento alla documentazione fornita sulle regole per visualizzare i parametri richiesti.

Esempio: anomalie per RowCount

Ad esempio, se vogliamo rilevare RowCount anomalie, forniamo RowCount come regola il nome.

```
DetectAnomalies "RowCount"
```

Esempio: anomalie per ColumnLength

Ad esempio, se vogliamo rilevare ColumnLength anomalie, forniamo ColumnLength come regola il nome e il nome della colonna.

```
DetectAnomalies "ColumnLength" "id"
```

## FileFreshness

FileFreshness assicura che i file di dati siano aggiornati in base alle condizioni fornite. Utilizza l'ora dell'ultima modifica dei file per garantire che i file di dati o l'intera cartella lo siano up-to-date.

Questa regola comprende due metriche:

- FileFreshness conformità basata sulla regola impostata
- Numero di file sottoposti a scansione dalla regola

```
Dataset.*.FileFreshness.Compliance: 1, Dataset.*.FileCount: 1
```

Il rilevamento delle anomalie non tiene in considerazione questi parametri.

Controllo dell'aggiornamento dei file

La seguente regola garantisce che tickets.parquet sia stato creato nelle ultime 24 ore.

```
FileFreshness "s3://amzn-s3-demo-bucket/artifacts/file/tickets/tickets.parquet" >
 (now() - 24 hours)
```

Controllo dell'aggiornamento delle cartelle

La seguente regola viene applicata se tutti i file nella cartella sono stati creati o modificati nelle ultime 24 ore.

```
FileFreshness "s3://bucket/" >= (now() - 1 days)
FileFreshness "s3://amzn-s3-demo-bucket/artifacts/file/tickets/" >= (now() - 24 hours)
```

Controllo dell'aggiornamento di cartelle o file con soglia

La seguente regola viene applicata se il 10% dei file nella cartella "tickets" è stato creato o modificato negli ultimi 10 giorni.

```
FileFreshness "s3://amzn-s3-demo-bucket/artifacts/file/tickets/" > (now() - 10 days)
 with threshold > 0.1
```

Controllo di file o cartelle con date specifiche

Puoi controllare l'aggiornamento dei file per giorni specifici.

```
FileFreshness "s3://amzn-s3-demo-bucket/artifacts/file/tickets/" > "2020-01-01"
FileFreshness "s3://amzn-s3-demo-bucket/artifacts/file/tickets/" between "2023-01-01"
 and "2024-01-01"
```

Controllo di file o cartelle con l'orario

Puoi utilizzarlo FileFreshness per assicurarti che i file arrivino in determinati orari.

```
FileFreshness "s3://amzn-s3-demo-bucket/artifacts/file/tickets/" between (now() - 45
 minutes) and now()
FileFreshness "s3://amzn-s3-demo-bucket/artifacts/file/tickets/" between "9:30 AM" and
 "9:30 PM"
```

```
FileFreshness "s3://amzn-s3-demo-bucket/artifacts/file/tickets/" > (now() - 10 minutes)
FileFreshness "s3://amzn-s3-demo-bucket/artifacts/file/tickets/" > now()
FileFreshness "s3://amzn-s3-demo-bucket/artifacts/file/tickets/" between (now() - 2
 hours) and (now() + 15 minutes)
FileFreshness "s3://amzn-s3-demo-bucket/artifacts/file/tickets/" between (now() - 3
 days) and (now() + 15 minutes)
FileFreshness "s3://amzn-s3-demo-bucket/artifacts/file/tickets/" between "2001-02-07"
 and (now() + 15 minutes)
FileFreshness "s3://amzn-s3-demo-bucket/artifacts/file/tickets/" > "21:45"
FileFreshness "s3://amzn-s3-demo-bucket/artifacts/file/tickets/" > "2024-01-01"
FileFreshness "s3://amzn-s3-demo-bucket/artifacts/file/tickets/" between "02:30" and
 "04:30"
FileFreshness "s3://amzn-s3-demo-bucket/artifacts/file/tickets/" between "9:30 AM" and
 "22:15"
```

Considerazioni chiave:

- FileFreshness può valutare i file utilizzando unità di giorni, ore e minuti
- Per l'ora, supporta il formato AM/PM e il formato a 24 ore
- Gli orari sono calcolati in UTC, salvo diversamente specificato
- Le date sono calcolate in UTC alle ore 00:00

FileFreshness che sono lavori basati sul tempo come segue:

```
FileFreshness "s3://amzn-s3-demo-bucket/artifacts/file/tickets/" > "21:45"
```

- Per prima cosa, l'ora "21:45" viene combinata con la data odierna in formato UTC per creare un campo data-ora
- Successivamente, se viene specificata un'alternativa al fuso orario, la data-ora viene convertita in quel fuso orario (altrimenti viene utilizzato UTC)
- Infine, la regola viene valutata

Tag delle regole opzionali basati su file:

I tag permettono di controllare il comportamento delle regole.

recentFiles

Questo tag limita il numero di file elaborati mantenendo il file più recente per primo.

```
FileFreshness "s3://amzn-s3-demo-bucket/" between (now() - 100 minutes) and (now() + 10
 minutes) with recentFiles = 1
```

uriRegex

> **Note**
>
> Il `uriRegex` tag è disponibile in AWS Glue 5.0 e versioni successive.

Questo tag filtra i file applicando un pattern regex al percorso del file. Vengono elaborati solo i file i cui percorsi corrispondono al modello. Puoi anche usare un lookahead negativo per escludere i file che corrispondono a uno schema.

```
# Match only files with a .csv extension
FileFreshness "s3://amzn-s3-demo-bucket/" > (now() - 24 hours) with uriRegex = "\.csv$"
# Match Parquet files that contain "orders_" in the path
FileFreshness "s3://amzn-s3-demo-bucket/" > (now() - 24 hours) with uriRegex =
 ".*orders_.*\.parquet"
# Exclude files ending in .tmp using a negative lookahead
FileFreshness "s3://amzn-s3-demo-bucket/" > (now() - 24 hours) with uriRegex = "(?!.*
\.tmp$).*"
```

Ordinamento del filtro

> **Note**
>
> Il `filterOrder` tag è disponibile in AWS Glue 5.0 e versioni successive.

Quando si utilizzano più tag di filtro `uriRegex` contemporaneamente, il `filterOrder` tag controlla l'ordine in cui vengono applicati. `recentFiles` L'ordine predefinito è quindi il `recentFiles` primo`uriRegex`.

```
FileFreshness "s3://amzn-s3-demo-bucket/" > (now() - 24 hours) with recentFiles = 1
 with uriRegex = "inventory_" with filterOrder = ["uriRegex","recentFiles"]
```

Nell'esempio precedente, il `uriRegex` filtro viene applicato prima per selezionare solo i file che corrispondono a «inventory_», quindi `recentFiles = 1` prende il file più recente da quel set filtrato.

In caso contrario`filterOrder`, il comportamento predefinito prenderebbe prima il singolo file più recente e poi applicherebbe l'espressione regolare, il che potrebbe non avere alcuna corrispondenza tra i file se il file più recente non corrisponde al modello.

> **Note**
>
> Tutti i valori dell'`filterOrder`elenco devono fare riferimento ad altri tag di filtro (`recentFilesouriRegex`) presenti anche nella stessa regola. Tag non filtranti come `timeZone` o non `failFast` sono validi in`filterOrder`.

FailFast

Se impostata su`"true"`, la regola restituisce immediatamente l'errore sul primo file che non soddisfa la condizione di aggiornamento, anziché valutare tutti i file e calcolare un rapporto di conformità.

```
FileFreshness "s3://amzn-s3-demo-bucket/" > (now() - 24 hours) with failFast = "true"
```

timezone

Sostituzioni di fuso orario accettate. Consulta Fusi orari consentiti per i fusi orari supportati.

```
FileFreshness "s3://path/" > "21:45" with timeZone = "America/New_York"
```

```
FileFreshness "s3://path/" > "21:45" with timeZone = "America/Chicago"
```

```
FileFreshness "s3://path/" > "21:45" with timeZone = "Europe/Paris"
```

```
FileFreshness "s3://path/" > "21:45" with timeZone = "Asia/Shanghai"
```

```
FileFreshness "s3://path/" > "21:45" with timeZone = "Australia/Darwin"
```

Dedurre i nomi dei file direttamente dai data frame

Non è sempre necessario fornire un percorso del file. Ad esempio, quando si crea la regola nel AWS Glue Data Catalog, potrebbe essere difficile trovare le cartelle utilizzate dalle tabelle del catalogo.

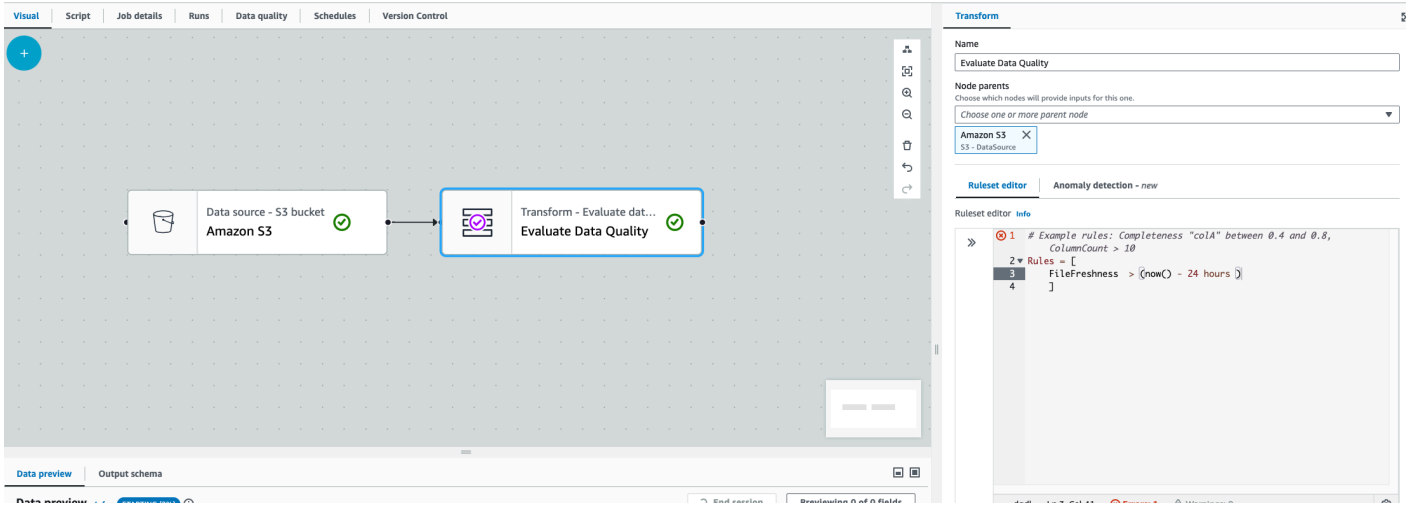
AWS Glue Data Quality è in grado di trovare le cartelle o i file specifici utilizzati per popolare il tuo dataframe e può rilevare se sono nuovi.

> **Note**
>
> Questa funzione funziona solo quando i file vengono letti correttamente in sala operatoria. DynamicFrame DataFrame

```
FileFreshness > (now() - 24 hours)
```

Questa regola individuerà il percorso della cartella o i file usati per popolare il frame dinamico o il frame di dati. Funziona con i percorsi Amazon S3 o le tabelle AWS Glue Data Catalog basate su Amazon S3. Occorre tenere presente alcuni aspetti:

1. In AWS Glue ETL, è necessario disporre di EvaluateDataQualityTransform subito dopo una trasformazione di Amazon S3 o AWS Glue Data Catalog.




2. Questa regola non funzionerà nelle sessioni interattive di AWS Glue.

Se provi a farlo in entrambi i casi o quando AWS Glue non riesce a trovare i file, AWS Glue genererà il seguente errore: "`Unable to parse file path from DataFrame`"

## FileMatch

La FileMatch regola consente di confrontare i file con altri file o checksum. Questo può essere utile in alcuni scenari:

1. Convalida dei file ricevuti da fonti esterne: è possibile eseguire il confronto con i checksum FileMatch per assicurarsi di aver ricevuto i file corretti da fonti esterne. Ciò contribuisce a verificare l'integrità dei dati che stai acquisendo.
2. Confronto dei dati in due cartelle diverse: FileMatch può essere utilizzato per confrontare i file tra due cartelle.

Questa regola include una metrica: il numero di file che sono stati scansionati dalla regola.

```
Dataset.*.FileCount: 1
```

Convalidare i file con un checksum:

FileMatch accetta un file e un set di checksum per garantire che almeno un checksum corrisponda al file.

```
FileMatch "s3://amzn-s3-demo-bucket/file.json" in ["3ee0d8617ac041793154713e5ef8f319"]
 with hashAlgorithm = "MD5"
FileMatch "s3://amzn-s3-demo-bucket/file.json" in ["3ee0d8617ac041793154713e5ef8f319"]
 with hashAlgorithm = "SHA-1"
FileMatch "s3://amzn-s3-demo-bucket/file.json" in ["3ee0d8617ac041793154713e5ef8f319"]
 with hashAlgorithm = "SHA-256"
FileMatch "s3://amzn-s3-demo-bucket/file.json" in ["3ee0d8617ac041793154713e5ef8f319"]
```

Sono supportati i seguenti algoritmi standard:

- MD5
- SHA-1
- SHA-256

Se non fornisci un algoritmo, quello predefinito è SHA-256.

Convalidare tutti i file in una cartella con un set di checksum:

```
FileMatch "s3://amzn-s3-demo-bucket /" in ["3ee0d8617ac041793154713e5ef8f319",
 "7e8617ac041793154713e5ef8f319"] with hashAlgorithm = "MD5"
FileMatch "s3://amzn-s3-demo-bucket /internal-folder/" in
 ["3ee0d8617ac041793154713e5ef8f319", "7e8617ac041793154713e5ef8f319"]
```

Confrontare i file in diverse cartelle

```
# Compare all files across two buckets
FileMatch "s3://original_bucket/" "s3://archive_bucket/"
# Compare files within specific subfolders
FileMatch "s3://original_bucket/internal-folder/" "s3://original_bucket/other-folder/"
# Compare only .json files across two folders
FileMatch "s3://original_bucket/" "s3://archive_bucket/" with uriRegex = "\.json$"
# Compare only the 5 most recent .csv files
FileMatch "s3://original_bucket/" "s3://archive_bucket/" with recentFiles = 5 with
 uriRegex = "\.csv$" with filterOrder = ["uriRegex","recentFiles"]
```

FileMatch controllerà il contenuto dei file `original_bucket` e si assicurerà che corrispondano a ciò che c'è dentro. `archive_bucket` Se non corrispondono esattamente, la regola non verrà applicata. Può anche verificare il contenuto delle cartelle interne o dei singoli file.

FileMatch possono anche confrontare i singoli file l'uno con l'altro.

```
FileMatch "s3://amzn-s3-demo-bucket /file_old.json" "s3://amzn-s3-demo-bucket /
file_new.json"
```

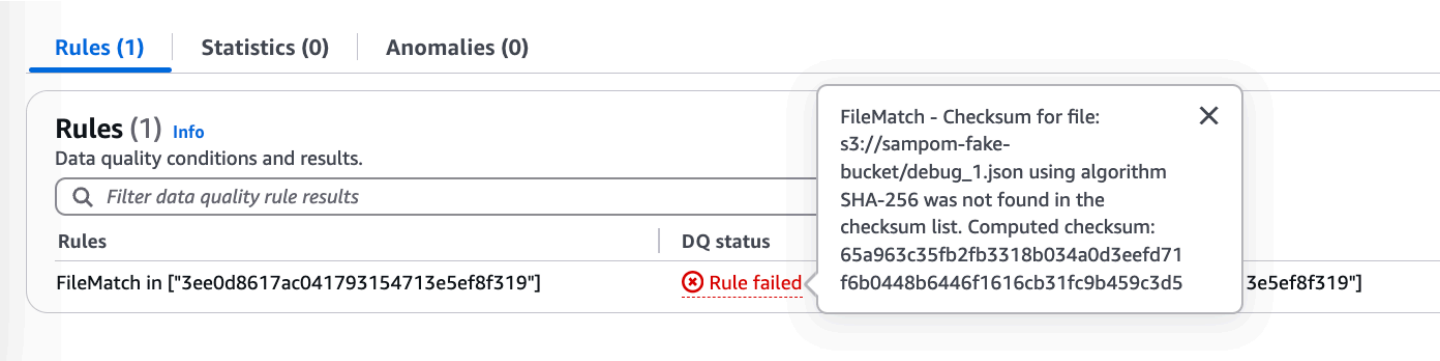
Dedurre i nomi dei file direttamente dai data frame

Non è sempre necessario fornire un percorso del file. Ad esempio, quando crei la regola nel AWS Glue Data Catalog (supportato da Amazon S3), potrebbe essere difficile trovare le cartelle utilizzate dalle tabelle del catalogo. AWS Glue Data Quality può trovare le cartelle o i file specifici utilizzati per popolare il tuo frame di dati.

> **Note**
>
> Questa funzione funziona solo quando i file vengono letti correttamente in sala operatoria DynamicFrame . DataFrame

```
FileMatch in ["3ee0d8617ac041793154713e5ef8f319"] with hashAlgorithm = "MD5"
FileMatch in ["3ee0d8617ac041793154713e5ef8f319"] with hashAlgorithm = "SHA-1"
FileMatch in ["3ee0d8617ac041793154713e5ef8f319"] with hashAlgorithm = "SHA-256"
FileMatch in ["3ee0d8617ac041793154713e5ef8f319"]
```

Se il checksum fornito è diverso da quello calcolato, ti FileMatch avviserà della differenza.




Tag delle regole opzionali basati su file:

I tag permettono di controllare il comportamento delle regole.

recentFiles

Questo tag limita il numero di file elaborati mantenendo il file più recente per primo.

```
FileMatch "s3://bucket/" in ["3ee0d8617ac04179sam4713e5ef8f319"] with recentFiles = 1
```

uriRegex

> **Note**
>
> Il `uriRegex` tag è disponibile in AWS Glue 5.0 e versioni successive.

Questo tag filtra i file applicando un pattern regex al percorso del file. Vengono elaborati solo i file i cui percorsi corrispondono al modello. Puoi anche usare un lookahead negativo per escludere i file che corrispondono a uno schema.

```
# Match only files with a .json extension
FileMatch "s3://bucket/" in ["3ee0d8617ac04179sam4713e5ef8f319"] with uriRegex =
 "\.json$"
# Exclude files ending in .tmp using a negative lookahead
FileMatch "s3://bucket/" in ["3ee0d8617ac04179sam4713e5ef8f319"] with uriRegex = "(?!.*
\.tmp$).*"
```

Ordinamento del filtro

> **Note**
>
> Il `filterOrder` tag è disponibile in AWS Glue 5.0 e versioni successive.

Quando si utilizzano più tag di filtro, `recentFiles` `uriRegex` singolarmente e contemporaneamente, il `filterOrder` tag controlla l'ordine in cui vengono applicati. L'ordine predefinito è quindi il `recentFiles` primo`uriRegex`.

```
FileMatch "s3://bucket/" in ["3ee0d8617ac04179sam4713e5ef8f319"] with recentFiles = 1
 with uriRegex = "\.json$" with filterOrder = ["uriRegex","recentFiles"]
```

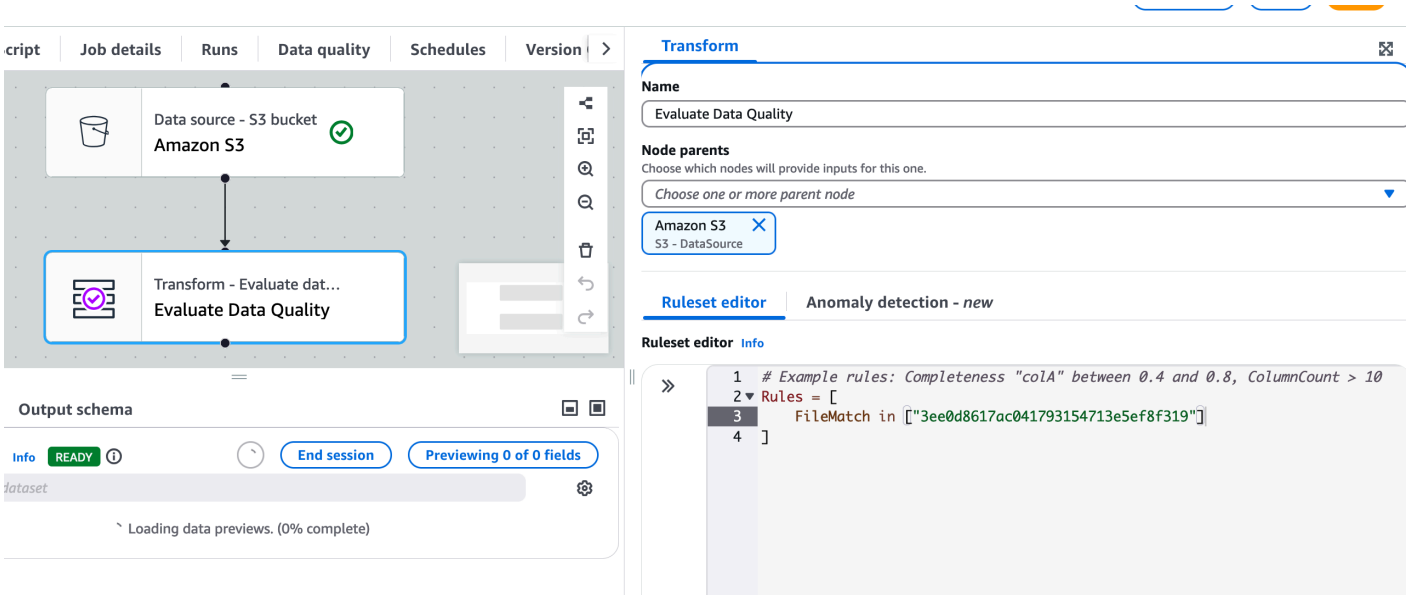
matchFileName

Questo tag garantisce che i file non abbiano nomi duplicati. Il comportamento predefinito è false.

```
FileMatch "s3://amzn-s3-demo-bucket/file.json" in ["3ee0d8617ac04179sam4713e5ef8f319"]
 with matchFileName = "true"
```

Occorre tenere presente alcuni aspetti:

1. In AWS Glue ETL, è necessario disporre di EvaluateDataQualityTransform subito dopo una trasformazione di Amazon S3 o AWS Glue Data Catalog.




2. Questa regola non funzionerà nelle sessioni interattive di AWS Glue.

## FileUniqueness

FileUniqueness consente di garantire che non vi siano file duplicati nei dati ricevuti dai produttori di dati.

Raccoglie le seguenti statistiche sui dati:

1. Numero di file sottoposti a scansione dalla regola
2. Il livello di unicità dei file

```
Dataset.*.FileUniquenessRatio: 1.00, Dataset.*.FileCount: 8
```

Trovare file duplicati in una cartella:

```
FileUniqueness "s3://bucket/" > 0.5
FileUniqueness "s3://bucket/folder/" = 1
```

Inferire i nomi delle cartelle direttamente dai data frame per rilevare i duplicati:

Non è sempre necessario fornire un percorso del file. Ad esempio, quando si crea la regola nel AWS Glue Data Catalog, potrebbe essere difficile trovare le cartelle utilizzate dalle tabelle del catalogo. AWS Glue Data Quality può trovare le cartelle o i file specifici utilizzati per popolare il tuo frame di dati.

> **Note**
>
> Quando si utilizza l'inferenza, le regole basate su file possono rilevare solo i file letti correttamente in sala operatoria. DynamicFrame DataFrame

```
FileUniqueness > 0.5
```

Tag delle regole opzionali basati su file:

I tag permettono di controllare il comportamento delle regole.

recentFiles

Questo tag limita il numero di file elaborati mantenendo il file più recente per primo.

```
FileUniqueness "s3://amzn-s3-demo-bucket/" > 0.5 with recentFiles = 1
```

uriRegex

> **Note**
>
> Il `uriRegex` tag è disponibile in AWS Glue 5.0 e versioni successive.

Questo tag filtra i file applicando un pattern regex al percorso del file. Vengono elaborati solo i file i cui percorsi corrispondono al modello. Puoi anche usare un lookahead negativo per escludere i file che corrispondono a uno schema.

```
# Match only files with a .csv extension
FileUniqueness "s3://bucket/" > 0.5 with uriRegex = "\.csv$"
# Exclude files ending in .tmp using a negative lookahead
FileUniqueness "s3://bucket/" > 0.5 with uriRegex = "(?!.*\.tmp$).*"
```

Ordinamento del filtro

> **Note**
>
> Il `filterOrder` tag è disponibile in AWS Glue 5.0 e versioni successive.

Quando si utilizzano più tag di filtro, `recentFiles` `uriRegex` singolarmente e contemporaneamente, il `filterOrder` tag controlla l'ordine in cui vengono applicati. L'ordine predefinito è quindi il `recentFiles` primo`uriRegex`.

```
FileUniqueness "s3://bucket/" > 0.5 with recentFiles = 5 with uriRegex = "\.csv$" with
 filterOrder = ["uriRegex","recentFiles"]
```

matchFileName

Questo tag garantisce che i file non abbiano nomi duplicati. Il comportamento predefinito è false.

```
FileUniqueness "s3://amzn-s3-demo-bucket/" > 0.5 with matchFileName = "true"
```

Occorre tenere presente alcuni aspetti:

1. In AWS Glue ETL, è necessario disporre di EvaluateDataQualityTransform subito dopo una trasformazione di Amazon S3 o AWS Glue Data Catalog.
2. Questa regola non funzionerà nelle sessioni interattive di AWS Glue.

## FileSize

Il FileSize tipo di regola consente di garantire che i file soddisfino determinati criteri di dimensione dei file. Ciò è utile nei seguenti casi d'uso:

1. Assicurarsi che i produttori non inviino file vuoti o di dimensioni notevolmente inferiori per l'elaborazione.
2. Assicurarsi che i bucket di destinazione non contengano file di dimensioni ridotte che potrebbero causare problemi di prestazioni.

FileSize raccoglie le seguenti metriche:

1. Conformità: restituisce la percentuale di file che soddisfano la soglia della regola stabilita
2. Numero di file: numero dei file sottoposti a scansione dalla regola
3. Dimensione minima dei file in byte
4. Dimensione massima dei file in byte

```
Dataset.*.FileSize.Compliance: 1, Dataset.*.FileCount: 8, Dataset.*.MaximumFileSize:
 327413121, Dataset.*.MinimumFileSize: 204558920
```

Il rilevamento delle anomalie non è supportato per queste metriche.

Convalidare la dimensione dei file

Questa regola viene applicata quando file.dat è superiore a 2 MB.

```
FileSize "s3://amzn-s3-demo-bucket/file.dat" > 2 MB
```

Le unità supportate includono B (byte), KB (kilobyte), MB (megabyte), GB (gigabyte) e TB (terabyte).

Convalidare la dimensione dei file nelle cartelle

```
FileSize "s3://bucket/" > 5 B
FileSize "s3://bucket/" < 2 GB
```

Questa regola viene applicata se il 70% dei file in s3://amzn-s3-demo-bucket ha dimensioni comprese tra 2 GB e 1 TB.

```
FileSize "s3://amzn-s3-demo-bucket/" between 2 GB and 1 TB with threshold > 0.7
```

Dedurre i nomi dei file direttamente dai data frame

Non è sempre necessario fornire un percorso del file. Ad esempio, quando si crea la regola nel Data Catalog, può essere difficile individuare le cartelle utilizzate dalle tabelle del catalogo. AWS Glue Data Quality può trovare le cartelle o i file specifici utilizzati per popolare il tuo frame di dati.

> **Note**
>
> Questa funzione funziona solo quando i file vengono letti correttamente in sala operatoria DynamicFrame . DataFrame

```
FileSize < 10 MB with threshold > 0.7
```

Tag delle regole opzionali basati su file:

I tag permettono di controllare il comportamento delle regole.

recentFiles

Questo tag limita il numero di file elaborati mantenendo il file più recente per primo.

```
FileSize "s3://amzn-s3-demo-bucket/" > 5 B with recentFiles = 1
```

uriRegex

> **Note**
>
> Il `uriRegex` tag è disponibile in AWS Glue 5.0 e versioni successive.

Questo tag filtra i file applicando un pattern regex al percorso del file. Vengono elaborati solo i file i cui percorsi corrispondono al modello. Puoi anche usare un lookahead negativo per escludere i file che corrispondono a uno schema.

```
# Match only files with a .dat extension
FileSize "s3://bucket/" > 5 B with uriRegex = "\.dat$"
# Exclude files ending in .tmp using a negative lookahead
FileSize "s3://bucket/" > 5 B with uriRegex = "(?!.*\.tmp$).*"
```

Ordinamento del filtro

> **Note**
>
> Il `filterOrder` tag è disponibile in AWS Glue 5.0 e versioni successive.

Quando si utilizzano più tag di filtro `uriRegex` contemporaneamente, il `filterOrder` tag controlla l'ordine in cui vengono applicati. `recentFiles` L'ordine predefinito è quindi il `recentFiles` primo`uriRegex`.

```
FileSize "s3://bucket/" > 5 B with recentFiles = 5 with uriRegex = "\.dat$" with
 filterOrder = ["uriRegex","recentFiles"]
```

FailFast

Se impostata su`"true"`, la regola restituisce immediatamente l'errore sul primo file che non soddisfa la condizione di dimensione, anziché valutare tutti i file e calcolare un rapporto di conformità.

```
FileSize "s3://bucket/" > 2 MB with failFast = "true"
```

Occorre tenere presente alcuni aspetti:

1. In AWS Glue ETL, devi avere Evaluate DataQuality Transform subito dopo la trasformazione di Amazon S3 o Data Catalog.
2. Questa regola non funzionerà nelle sessioni interattive di AWS Glue.

# Utilizzo APIs per misurare e gestire la qualità dei dati

Questo argomento descrive come utilizzare per APIs misurare e gestire la qualità dei dati.

Indice


- Prerequisiti
- Utilizzo dei consigli di AWS Glue Data Quality
- Utilizzo dei set di regole AWS Glue Data Quality
- L'utilizzo di AWS Glue Data Quality funziona
- Utilizzo dei risultati di AWS Glue Data Quality


## Prerequisiti

- Assicurati che la tua versione di boto3 sia aggiornata in modo che includa l'ultima API AWS Glue Data Quality.
- Assicurati che la tua versione AWS CLI sia aggiornata, in modo da includere la CLI più recente.

Se stai usando un lavoro AWS Glue per eseguirli APIs, puoi utilizzare la seguente opzione per aggiornare la libreria boto3 alla versione più recente:

```
–additional-python-modules boto3==<version>
```

## Utilizzo dei consigli di AWS Glue Data Quality

Per iniziare una raccomandazione di AWS Glue Data Quality, esegui:

```
class GlueWrapper:
    """Encapsulates AWS Glue actions."""
    def __init__(self, glue_client):
        """
        :param glue_client: A Boto3 AWS Glue client.
        """
        self.glue_client = glue_client

    def start_data_quality_rule_recommendation_run(self, database_name, table_name,
 role_arn):
        """
        Starts a recommendation run that is used to generate rules when you don't know
 what rules to write. AWS Glue Data Quality
        analyzes the data and comes up with recommendations for a potential ruleset.
 You can then triage the ruleset
        and modify the generated ruleset to your liking.
```

```
        :param database_name: The name of the AWS Glue database which contains the
dataset.
        :param table_name: The name of the AWS Glue table against which we want a
recommendation
        :param role_arn: The Amazon Resource Name (ARN) of an AWS Identity and Access
Management (IAM) role that grants permission to let AWS Glue access the resources it
needs.

        """
        try:
            response = self.client.start_data_quality_rule_recommendation_run(
                DataSource={
                    'GlueTable': {
                        'DatabaseName': database_name,
                        'TableName': table_name
                    }
                },
                Role=role_arn
            )
        except ClientError as err:
            logger.error(
                "Couldn't start data quality recommendation run %s. Here's why: %s:
%s", name,
                err.response['Error']['Code'], err.response['Error']['Message'])
            raise
        else:
            return response['RunId']
```

Per l'esecuzione di un suggerimento, puoi utilizzare `pushDownPredicates` o per `catalogPartitionPredicates` migliorare le prestazioni ed eseguire i suggerimenti solo su partizioni specifiche delle origini del catalogo.

```
client.start_data_quality_rule_recommendation_run(
            DataSource={
                'GlueTable': {
                    'DatabaseName': database_name,
                    'TableName': table_name,
                    'AdditionalOptions': {
                        'pushDownPredicate': "year=2022"
                    }
                }
            },
```

```
            Role=role_arn,
            NumberOfWorkers=2,
            CreatedRulesetName='<rule_set_name>'
  )
```

Per ottenere i risultati di una raccomandazione di AWS Glue Data Quality, esegui:

```
class GlueWrapper:
    """Encapsulates AWS Glue actions."""
    def __init__(self, glue_client):
        """
        :param glue_client: A Boto3 AWS Glue client.
        """
        self.glue_client = glue_client

    def get_data_quality_rule_recommendation_run(self, run_id):
        """
        Gets the specified recommendation run that was used to generate rules.

        :param run_id: The id of the data quality recommendation run

        """
        try:
            response =
 self.client.get_data_quality_rule_recommendation_run(RunId=run_id)
        except ClientError as err:
            logger.error(
                "Couldn't get data quality recommendation run %. Here's why: %s: %s",
 run_id,
                err.response['Error']['Code'], err.response['Error']['Message'])
            raise
        else:
            return response
```

Dall'oggetto di risposta precedente, è possibile estrarre RuleSet quanto consigliato dall'esecuzione, da utilizzare nei passaggi successivi:

```
print(response['RecommendedRuleset'])

Rules = [
    RowCount between 2000 and 8000,
    IsComplete "col1",
    IsComplete "col2",
```

```
    StandardDeviation "col3" between 58138330.8 and 64258155.09,
    ColumnValues "col4" between 1000042965 and 1214474826,
    IsComplete "col5"
]
```

Per ottenere un elenco di tutte le esecuzioni suggerite che è possibile filtrare ed elencare:

```
response = client.list_data_quality_rule_recommendation_runs(
    Filter={
        'DataSource': {
            'GlueTable': {
                'DatabaseName': '<database_name>',
                'TableName': '<table_name>'
            }
        }
)
```

Per annullare le attività di raccomandazione esistenti di AWS Glue Data Quality:

```
response = client.cancel_data_quality_rule_recommendation_run(
    RunId='dqrun-d4b6b01957fdd79e59866365bf9cb0e40fxxxxxxx'
)
```

## Utilizzo dei set di regole AWS Glue Data Quality

Per creare un set di regole AWS Glue Data Quality:

```
response = client.create_data_quality_ruleset(
    Name='<ruleset_name>',
    Ruleset='Rules = [IsComplete "col1", IsPrimaryKey "col2", RowCount between 2000 and
 8000]',
    TargetTable={
        'TableName': '<table_name>',
        'DatabaseName': '<database_name>'
    }
)
```

Per ottenere un set di regole di qualità dei dati:

```
response = client.get_data_quality_ruleset(
    Name='<ruleset_name>'
```

```
)
print(response)
```

Successivamente, puoi utilizzare quest'API per estrarre il set di regole:

```
print(response['Ruleset'])
```

Per elencare tutti i set di regole sulla qualità dei dati per una tabella:

```
response = client.list_data_quality_rulesets()
```

È possibile utilizzare la condizione di filtro all'interno dell'API per filtrare tutti i set di regole collegati a un database o una tabella specifici:

```
response = client.list_data_quality_rulesets(
    Filter={
        'TargetTable': {
            'TableName': '<table_name>',
            'DatabaseName': '<database_name>'
        }
    },
)
```

Per aggiornare un set di regole di qualità dei dati:

```
class GlueWrapper:
    """Encapsulates AWS Glue actions."""
    def __init__(self, glue_client):
        """
        :param glue_client: A Boto3 AWS Glue client.
        """
        self.glue_client = glue_client

    def update_data_quality_ruleset(self, ruleset_name, ruleset_string):
        """
        Update an AWS Glue Data Quality Ruleset

        :param ruleset_name: The name of the AWS Glue Data Quality ruleset to update
        :param ruleset_string: The DQDL ruleset string to update the ruleset with

        """
```

```
        try:
            response = self.client.update_data_quality_ruleset(
                Name=ruleset_name,
                Ruleset=ruleset_string
            )
        except ClientError as err:
            logger.error(
                "Couldn't update the AWS Glue Data Quality ruleset. Here's why: %s:
 %s",
                err.response['Error']['Code'], err.response['Error']['Message'])
            raise
        else:
            return response
```

Per eliminare un set di regole di qualità dei dati:

```
class GlueWrapper:
    """Encapsulates AWS Glue actions."""
    def __init__(self, glue_client):
        """
        :param glue_client: A Boto3 AWS Glue client.
        """
        self.glue_client = glue_client

    def delete_data_quality_ruleset(self, ruleset_name):
        """
        Delete a AWS Glue Data Quality Ruleset

        :param ruleset_name: The name of the AWS Glue Data Quality ruleset to delete

        """
        try:
            response = self.client.delete_data_quality_ruleset(
                Name=ruleset_name
            )
        except ClientError as err:
            logger.error(
                "Couldn't delete the AWS Glue Data Quality ruleset. Here's why: %s:
 %s",
                err.response['Error']['Code'], err.response['Error']['Message'])
            raise
        else:
            return response
```

# L'utilizzo di AWS Glue Data Quality funziona

Per avviare un'esecuzione di AWS Glue Data Quality:

```
class GlueWrapper:
    """Encapsulates AWS Glue actions."""
    def __init__(self, glue_client):
        """
        :param glue_client: A Boto3 AWS Glue client.
        """
        self.glue_client = glue_client

    def start_data_quality_ruleset_evaluation_run(self, database_name, table_name,
 role_name, ruleset_list):
        """
        Start an AWS Glue Data Quality evaluation run

        :param database_name: The name of the AWS Glue database which contains the
 dataset.
        :param table_name: The name of the AWS Glue table against which we want to
 evaluate.
        :param role_arn: The Amazon Resource Name (ARN) of an AWS Identity and Access
 Management (IAM) role that grants permission to let AWS Glue access the resources it
 needs.
        :param ruleset_list: The list of AWS Glue Data Quality ruleset names to
 evaluate.

        """
        try:
            response = client.start_data_quality_ruleset_evaluation_run(
                DataSource={
                    'GlueTable': {
                        'DatabaseName': database_name,
                        'TableName': table_name
                    }
                },
                Role=role_name,
                RulesetNames=ruleset_list
            )
        except ClientError as err:
            logger.error(
                "Couldn't start the AWS Glue Data Quality Run. Here's why: %s: %s",
                err.response['Error']['Code'], err.response['Error']['Message'])
```

```
            raise
        else:
            return response['RunId']
```

Ricorda che puoi trasmettere un parametro `pushDownPredicate` o `catalogPartitionPredicate` per garantire che l'esecuzione della qualità dei dati riguardi solo un set specifico di partizioni all'interno della tabella del catalogo. Esempio:

```
response = client.start_data_quality_ruleset_evaluation_run(
    DataSource={
        'GlueTable': {
            'DatabaseName': '<database_name>',
            'TableName': '<table_name>',
            'AdditionalOptions': {
                'pushDownPredicate': 'year=2023'
            }
        }
    },
    Role='<role_name>',
    NumberOfWorkers=5,
    Timeout=123,
    AdditionalRunOptions={
        'CloudWatchMetricsEnabled': False
    },
    RulesetNames=[
        '<ruleset_name>',
    ]
)
```

Puoi anche configurare la modalità di valutazione delle regole composite nel set di regole, a livello di RIGA o COLONNA. Per maggiori informazioni sul funzionamento delle regole composite, consulta la sezione Come funzionano le regole composite nella documentazione.

Esempio su come impostare il metodo di valutazione delle regole composite nella richiesta:

```
response = client.start_data_quality_ruleset_evaluation_run(
    DataSource={
        'GlueTable': {
            'DatabaseName': '<database_name>',
            'TableName': '<table_name>',
            'AdditionalOptions': {
                'pushDownPredicate': 'year=2023'
```

```
            }
        }
    },
    Role='<role_name>',
    NumberOfWorkers=5,
    Timeout=123,
    AdditionalRunOptions={
        'CompositeRuleEvaluationMethod':ROW
    },
    RulesetNames=[
        '<ruleset_name>',
    ]
)
```

Per ottenere informazioni su un programma AWS Glue Data Quality esegui:

```
class GlueWrapper:
    """Encapsulates AWS Glue actions."""
    def __init__(self, glue_client):
        """
        :param glue_client: A Boto3 AWS Glue client.
        """
        self.glue_client = glue_client

    def get_data_quality_ruleset_evaluation_run(self, run_id):
        """
        Get details about an AWS Glue Data Quality Run

        :param run_id: The AWS Glue Data Quality run ID to look up

        """
        try:
            response = self.client.get_data_quality_ruleset_evaluation_run(
                RunId=run_id
            )
        except ClientError as err:
            logger.error(
                "Couldn't look up the AWS Glue Data Quality run ID. Here's why: %s:
 %s",
                err.response['Error']['Code'], err.response['Error']['Message'])
            raise
        else:
            return response
```

Per ottenere i risultati di un AWS Glue Data Quality esegui:

Per una determinata esecuzione di AWS Glue Data Quality, puoi estrarre i risultati della valutazione dell'esecuzione utilizzando il seguente metodo:

```
response = client.get_data_quality_ruleset_evaluation_run(
    RunId='d4b6b01957fdd79e59866365bf9cb0e40fxxxxxxx'
)

resultID = response['ResultIds'][0]

response = client.get_data_quality_result(
    ResultId=resultID
)

print(response['RuleResults'])
```

Per elencare tutte le tue esecuzioni di AWS Glue Data Quality:

```
class GlueWrapper:
    """Encapsulates AWS Glue actions."""
    def __init__(self, glue_client):
        """
        :param glue_client: A Boto3 AWS Glue client.
        """
        self.glue_client = glue_client

    def list_data_quality_ruleset_evaluation_runs(self, database_name, table_name):
        """
        Lists all the AWS Glue Data Quality runs against a given table

        :param database_name: The name of the database where the data quality runs
        :param table_name: The name of the table against which the data quality runs
 were created

        """
        try:
            response = self.client.list_data_quality_ruleset_evaluation_runs(
                Filter={
                    'DataSource': {
                        'GlueTable': {
                            'DatabaseName': database_name,
                            'TableName': table_name
```

```
                        }
                    }
                }
            )
        except ClientError as err:
            logger.error(
                "Couldn't list the AWS Glue Quality runs. Here's why: %s: %s",
                err.response['Error']['Code'], err.response['Error']['Message'])
            raise
        else:
            return response
```

È possibile modificare la clausola di filtro in modo che mostri solo i risultati compresi entro orari specifici o in base a tabelle specifiche.

Per interrompere una corsa in corso di AWS Glue Data Quality:

```
class GlueWrapper:
    """Encapsulates AWS Glue actions."""
    def __init__(self, glue_client):
        """
        :param glue_client: A Boto3 AWS Glue client.
        """
        self.glue_client = glue_client

    def cancel_data_quality_ruleset_evaluation_run(self, result_id):
        """
        Cancels a given AWS Glue Data Quality run

        :param result_id: The result id of a AWS Glue Data Quality run to cancel

        """
        try:
            response = self.client.cancel_data_quality_ruleset_evaluation_run(
                ResultId=result_id
            )
        except ClientError as err:
            logger.error(
                "Couldn't cancel the AWS Glue Data Quality run. Here's why: %s: %s",
                err.response['Error']['Code'], err.response['Error']['Message'])
            raise
        else:
            return response
```

# Utilizzo dei risultati di AWS Glue Data Quality

Per ottenere i risultati della tua corsa AWS Glue Data Quality:

```
class GlueWrapper:
    """Encapsulates AWS Glue actions."""
    def __init__(self, glue_client):
        """
        :param glue_client: A Boto3 AWS Glue client.
        """
        self.glue_client = glue_client

    def get_data_quality_result(self, result_id):
        """
        Outputs the result of an AWS Glue Data Quality Result

        :param result_id: The result id of an AWS Glue Data Quality run

        """
        try:
            response = self.client.get_data_quality_result(
                ResultId=result_id
            )
        except ClientError as err:
            logger.error(
                "Couldn't get the AWS Glue Data Quality result. Here's why: %s: %s",
                err.response['Error']['Code'], err.response['Error']['Message'])
            raise
        else:
            return response
```

Per visualizzare le statistiche raccolte per un determinato risultato relativo a Qualità dei dati:

```
import boto3
from botocore.exceptions import ClientError
import logging

logger = logging.getLogger(__name__)
class GlueWrapper:
    """Encapsulates AWS Glue actions."""
    def __init__(self, glue_client):
        """
        :param glue_client: A Boto3 AWS Glue client.
```

```
        """
        self.glue_client = glue_client

    def get_profile_for_data_quality_result(self, result_id):
        """
        Outputs the statistic profile for a AWS Glue Data Quality Result

        :param result_id: The result id of a AWS Glue Data Quality run

        """
        try:
            response = self.glue_client.get_data_quality_result(
                ResultId=result_id
            )

            # the profile contains all statistics gathered for the result
            profile_id = response['ProfileId']
            profile = self.glue_client.list_data_quality_statistics(
                ProfileId = profile_id
            )
            return profile
        except ClientError as err:
            logger.error(
                "Couldn't retrieve Data Quality profile. Here's why: %s: %s",
                err.response['Error']['Code'], err.response['Error']['Message'])
            raise
```

Per visualizzare la serie temporale di una statistica raccolta durante più cicli di Qualità dei dati:

```
class GlueWrapper:
    """Encapsulates AWS Glue actions."""
    def __init__(self, glue_client):
        """
        :param glue_client: A Boto3 AWS Glue client.
        """
        self.glue_client = glue_client

    def get_statistics_for_data_quality_result(self, profile_id):
        """
        Outputs an array of datapoints for each statistic in the input result.

        :param result_id: The profile id of a AWS Glue Data Quality run
```

```
        """
        try:
            profile = self.glue_client.list_data_quality_statistics(
                ProfileId = profile_id
            )
            statistics = [self.glue_client.list_data_quality_statistics(
                StatisticId = s['StatisticId']
            ) for s in profile['Statistics']]
            return statistics
        except ClientError as err:
            logger.error(
                "Couldn't retrieve Data Quality statistics. Here's why: %s: %s",
                err.response['Error']['Code'], err.response['Error']['Message'])
            raise
```

Per visualizzare il modello di rilevamento delle anomalie per una specifica statistica:

```
class GlueWrapper:
    """Encapsulates AWS Glue actions."""
    def __init__(self, glue_client):
        """
        :param glue_client: A Boto3 AWS Glue client.
        """
        self.glue_client = glue_client

    def get_model_training_result_for_statistic(self, statistic_id, profile_id):
        """
        Outputs the details (bounds) of anomaly detection training for the given
 statistic at the given profile.

        :param statistic_id the model's statistic (the timeseries it is tracking)
        :param profile_id the profile associated with the model (a point in the
 timeseries)

        """
        try:
            model = self.glue_client.get_data_quality_model_result(
                ProfileId = profile_id, StatisticId = statistic_id
            )
            return model
        except ClientError as err:
            logger.error(
                "Couldn't retrieve Data Quality model results. Here's why: %s: %s",
```

```
                err.response['Error']['Code'], err.response['Error']['Message'])
            raise
```

Per escludere un punto dati dalla linea di base di rilevamento delle anomalie del relativo modello statistico:

```
class GlueWrapper:
    """Encapsulates AWS Glue actions."""
    def __init__(self, glue_client):
        """
        :param glue_client: A Boto3 AWS Glue client.
        """
        self.glue_client = glue_client

    def apply_exclusions_to_statistic(self, statistic_id, profile_ids):
        """
        Annotate some points along a given statistic timeseries.

        This example excludes the provided values; INCLUDE can also be used to undo
 this action.

        :param statistic_id the statistic timeseries to annotate
        :param profile_id the profiles we want to exclude (points in the timeseries)

        """

        try:
            response = self.glue_client.batch_put_data_quality_statistic_annotation(
                    InclusionAnnotations = [
                        {'ProfileId': prof_id,
                        'StatisticId': statistic_id,
                        'InclusionAnnotation': 'EXCLUDE'} for prof_id in profile_ids
                    ]
            )
            return response['FailedInclusionAnnotations']
        except ClientError as err:
            logger.error(
                "Couldn't store Data Quality annotations. Here's why: %s: %s",
                err.response['Error']['Code'], err.response['Error']['Message'])
            raise
```

Per visualizzare lo stato dell'addestramento dei modelli di rilevamento delle anomalie per una specifica statistica:

```
class GlueWrapper:
    """Encapsulates AWS Glue actions."""
    def __init__(self, glue_client):
        """
        :param glue_client: A Boto3 AWS Glue client.
        """
        self.glue_client = glue_client

    def get_model_training_status_for_statistic(self, statistic_id, profile_id):
        """
        Outputs the status of anomaly detection training for the given statistic at the
 given profile.

        :param statistic_id the model's statistic (the timeseries it is tracking)
        :param profile_id the profile associated with the model (a point in the
 timeseries)

        """
        try:
            model = self.glue_client.get_data_quality_model(
                ProfileId = profile_id, StatisticId = statistic_id
            )
            return model
        except ClientError as err:
            logger.error(
                "Couldn't retrieve Data Quality statistics. Here's why: %s: %s",
                err.response['Error']['Code'], err.response['Error']['Message'])
            raise
```

Per escludere tutti i risultati di una specifica esecuzione di Qualità dei dati dalle linee di base per il rilevamento delle anomalie:

```
class GlueWrapper:
    """Encapsulates AWS Glue actions."""
    def __init__(self, glue_client):
        """
        :param glue_client: A Boto3 AWS Glue client.
        """
        self.glue_client = glue_client

    def apply_exclusions_to_profile(self, profile_id):
        """
        Exclude datapoints produced by a run across statistic timeseries.
```

```
        This example excludes the provided values; INCLUDE can also be used to undo
 this action.

        :param profile_id the profiles we want to exclude (points in the timeseries)

        """
        try:
            response = self.glue_client.put_data_quality_profile_annotation(
                    ProfileId = profile_id,
                    InclusionAnnotation = "EXCLUDE"
            )
            return response
        except ClientError as err:
            logger.error(
                "Couldn't store Data Quality annotations. Here's why: %s: %s",
                err.response['Error']['Code'], err.response['Error']['Message'])
            raise
```

Per ottenere i risultati di un determinato ciclo di Qualità dei dati e visualizzare i risultati:

Con un AWS Glue Data Quality`runID`, puoi estrarre i dati `resultID` per poi ottenere i risultati effettivi, come mostrato di seguito:

```
response = client.get_data_quality_ruleset_evaluation_run(
    RunId='dqrun-abca77ee126abe1378c1da1ae0750d7dxxxx'
)

resultID = response['ResultIds'][0]

response = client.get_data_quality_result(
    ResultId=resultID
)

print(resp['RuleResults'])
```

# Configurazione di avvisi, implementazioni e pianificazioni

Questo argomento descrive come configurare avvisi, distribuzioni e pianificazione per AWS Glue Data Quality.

Indice

- Configurazione di avvisi e notifiche nell'integrazione con Amazon EventBridge
  - Opzioni di configurazione aggiuntive per il modello di evento
  - Formattazione delle notifiche in formato e-mail
- Imposta avvisi e notifiche in integrazione CloudWatch
- Esecuzione di query di risultati sulla qualità dei dati per creare pannelli di controllo
- Implementazione delle regole di qualità dei dati utilizzando AWS CloudFormation
- Pianificazione delle regole di qualità dei dati


## Configurazione di avvisi e notifiche nell'integrazione con Amazon EventBridge

AWS Glue Data Quality supporta la pubblicazione di EventBridge eventi, che vengono emessi al termine di un ciclo di valutazione del set di regole di Data Quality. In questo modo, è possibile impostare facilmente avvisi quando le regole di qualità dei dati non vengono soddisfatte.

Ecco un esempio di evento relativo alla valutazione dei set di regole sulla qualità dei dati in Catalogo dati. Con queste informazioni, puoi esaminare i dati resi disponibili con Amazon EventBridge. È possibile effettuare chiamate API aggiuntive per ottenere maggiori dettagli. Ad esempio, chiama l'API `get_data_quality_result` con l'ID del risultato per ottenere i dettagli di una particolare esecuzione.

```
{
    "version":"0",
    "id":"abcdef00-1234-5678-9abc-def012345678",
    "detail-type":"Data Quality Evaluation Results Available",
    "source":"aws.glue-dataquality",
    "account":"123456789012",
    "time":"2017-09-07T18:57:21Z",
    "region":"us-west-2",
    "resources":[],
    "detail":{
        "context": {
                    "contextType": "GLUE_DATA_CATALOG",
                    "runId":"dqrun-12334567890",
                    "databaseName": "db-123",
                    "tableName": "table-123",
                    "catalogId": "123456789012"
                    },
```

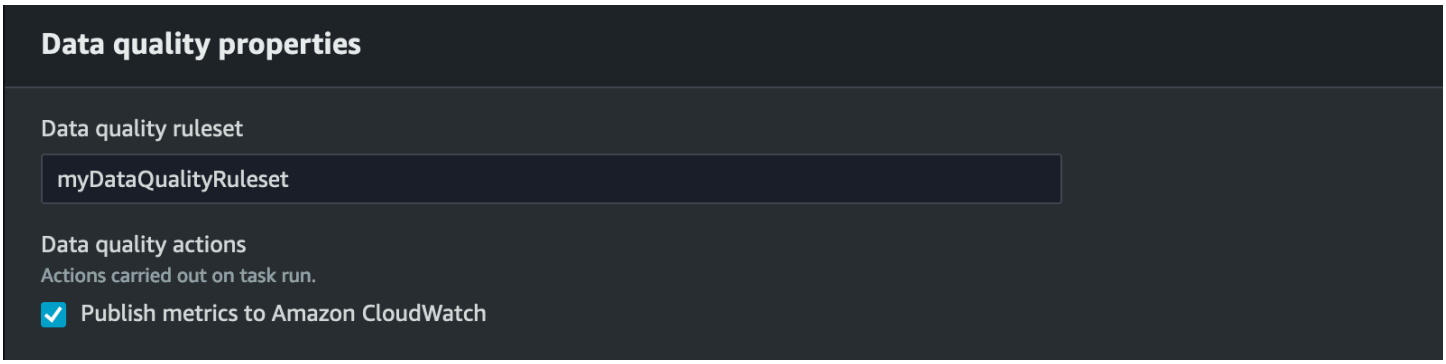
```
        "resultID": "dqresult-12334567890",
        "rulesetNames": ["rulset1"],
        "state":"SUCCEEDED",
        "score": 1.00,
        "rulesSucceeded": 100,
        "rulesFailed": 0,
        "rulesSkipped": 0
    }
}
```

Ecco un esempio di evento che viene pubblicato quando si valutano i set di regole di qualità dei dati nei notebook Glue AWS ETL o AWS Glue Studio.

```
{
    "version":"0",
    "id":"abcdef00-1234-5678-9abc-def012345678",
    "detail-type":"Data Quality Evaluation Results Available",
    "source":"aws.glue-dataquality",
    "account":"123456789012",
    "time":"2017-09-07T18:57:21Z",
    "region":"us-west-2",
    "resources":[],
    "detail":{
        "context": {
                    "contextType": "GLUE_JOB",
                    "jobId": "jr-12334567890",
                    "jobName": "dq-eval-job-1234",
                    "evaluationContext": "",
                    }
        "resultID": "dqresult-12334567890",
        "rulesetNames": ["rulset1"],
        "state":"SUCCEEDED",
        "score": 1.00
        "rulesSucceeded": 100,
        "rulesFailed": 0,
        "rulesSkipped": 0
    }
}
```

Affinché la valutazione della qualità dei dati venga eseguita sia nel Data Catalog che nei job ETL, l' Amazon CloudWatch opzione Pubblica metriche su, selezionata per impostazione predefinita, deve rimanere selezionata affinché la pubblicazione funzioni. EventBridge

Configurazione delle notifiche EventBridge




Per ricevere gli eventi emessi e definire gli obiettivi, devi configurare EventBridge le regole di Amazon. Per creare regole:

1. Apri la EventBridge console Amazon.
2. Scegli Regole nella sezione Router della barra di navigazione.
3. Selezionare Create Rule (Crea regola).
4. In Definisci i dettagli della regola:
   a. In Nome, inserisci `myDQRule`.
   b. Inserisci una descrizione (facoltativa).
   c. Per Router di eventi, seleziona il tuo router. Se non ne hai uno, lascia quello predefinito.
   d. Per Tipo di regola, scegli Regola con un modello di eventi, quindi scegli Successivo.
5. In Crea un modello di eventi:
   a. Per l'origine dell'evento, seleziona AWS eventi o eventi dei EventBridge partner.
   b. Puoi saltare la sezione Evento di esempio.
   c. Per il metodo di creazione, seleziona Utilizza modulo del modello.
   d. Per il modello dell'evento:
      i. Seleziona Servizi AWS come Origine dell'evento.
      ii. Seleziona Glue Data Quality per l' AWS assistenza.
      iii. Seleziona Risultati di valutazione della qualità dei dati disponibili per Tipo di evento.
      iv. Seleziona NON RIUSCITO per Stati specifici. Viene quindi visualizzato un modello di eventi simile al seguente:

```
{
  "source": ["aws.glue-dataquality"],
  "detail-type": ["Data Quality Evaluation Results Available"],
```

```
  "detail": {
    "state": ["FAILED"]
  }
}
```

v. Per ulteriori opzioni di connessione, consulta la sezione Opzioni di configurazione aggiuntive per il modello di evento.

6. Su Seleziona destinazioni:
   a. Per Tipi di destinazione, seleziona Servizio AWS .
   b. Utilizza il menu a discesa Seleziona una destinazione per scegliere il AWS servizio desiderato a cui connetterti (SNS, Lambda, SQS, ecc.), quindi scegli Avanti.
7. In Configura tag, fai clic su Aggiungi nuovi tag per aggiungere tag facoltativi, quindi scegli Successivo.
8. Viene visualizzata una pagina di riepilogo di tutte le selezioni. Scegli Crea regola in basso.

## Opzioni di configurazione aggiuntive per il modello di evento

Oltre a filtrare l'evento in base alla riuscita o al fallimento, potresti voler filtrare ulteriormente gli eventi in base a parametri diversi.

Per fare ciò, vai alla sezione Modello di evento e seleziona Modifica modello per specificare parametri aggiuntivi. Nota che i campi del modello di evento fanno distinzione tra maiuscole e minuscole. Di seguito sono riportati alcuni esempi di configurazione del modello di evento.

Per acquisire eventi da una particolare tabella valutando set di regole specifici, utilizza questo tipo di modello:

```
{
  "source": ["aws.glue-dataquality"],
  "detail-type": ["Data Quality Evaluation Results Available"],
  "detail": {
    "context": {
      "contextType": ["GLUE_DATA_CATALOG"],
      "databaseName": "db-123",
       "tableName": "table-123",
    },
    "rulesetNames": ["ruleset1", "ruleset2"]
    "state": ["FAILED"]
  }
```

```
}
```

Per acquisire eventi da processi specifici nell'esperienza ETL, utilizza questo tipo di modello:

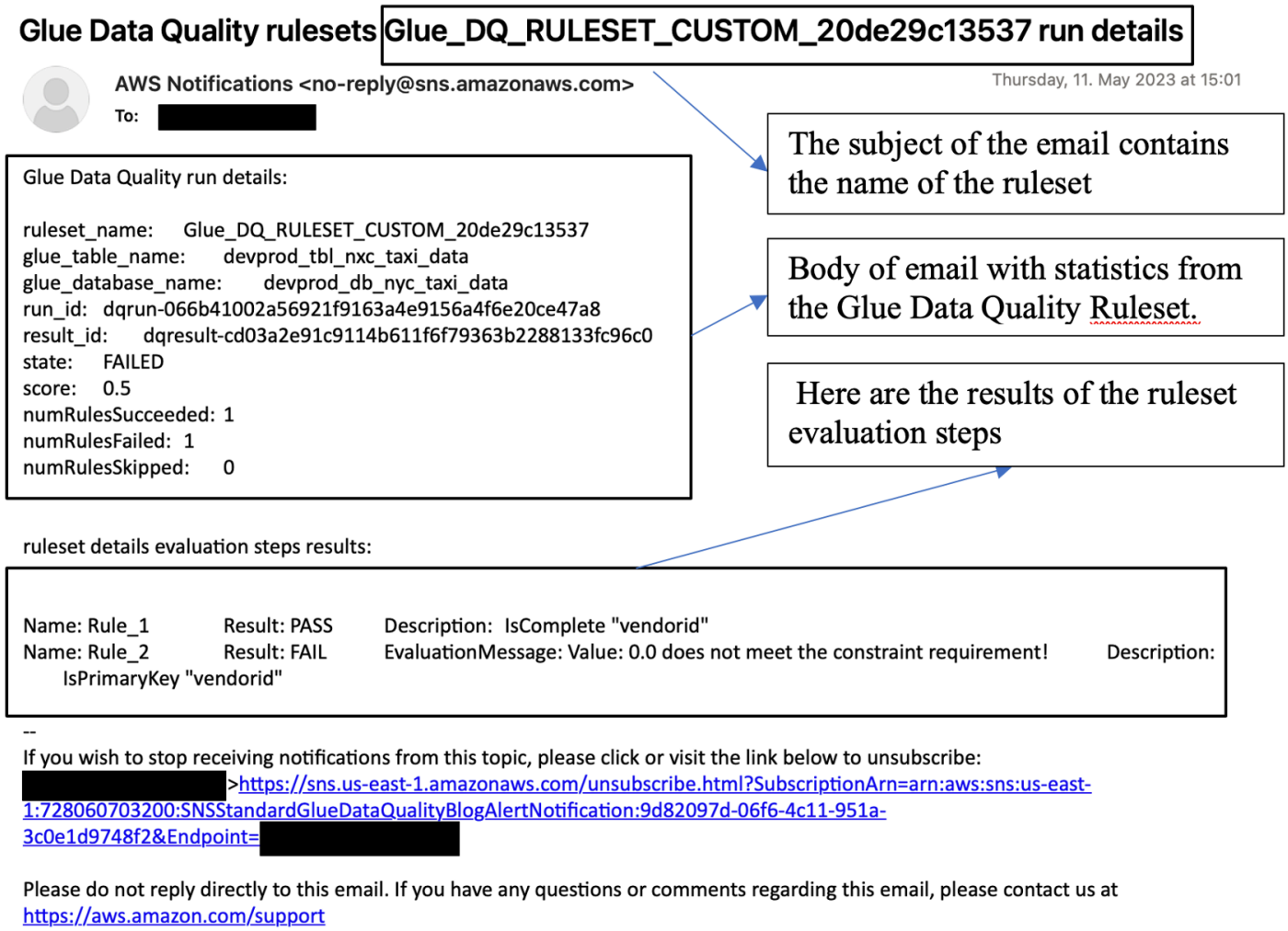
```
{
  "source": ["aws.glue-dataquality"],
  "detail-type": ["Data Quality Evaluation Results Available"],
  "detail": {
    "context": {
      "contextType": ["GLUE_JOB"],
      "jobName": ["dq_evaluation_job1", "dq_evaluation_job2"]
    },
    "state": ["FAILED"]
  }
}
```

Per registrare eventi con un punteggio inferiore a una soglia specifica (ad esempio 70%):

```
{
  "source": ["aws.glue-dataquality"],
  "detail-type": ["Data Quality Evaluation Results Available"],
  "detail": {
    "score": [{
      "numeric": ["<=", 0.7]
    }]
  }
}
```

## Formattazione delle notifiche in formato e-mail

A volte è necessario inviare una notifica e-mail ben formattata ai team aziendali. Puoi usare Amazon EventBridge e AWS Lambda per raggiungere questo obiettivo.




Il seguente codice di esempio può essere utilizzato per formattare le notifiche sulla qualità dei dati per generare e-mail.

```
import boto3
import json
from datetime import datetime

sns_client = boto3.client('sns')
glue_client = boto3.client('glue')

sns_topic_arn = 'arn:aws:sns:<region-code>:<account-id>:<sns-topic-name>'



def lambda_handler(event, context):
    log_metadata = {}
```

```
    message_text = ""
    subject_text = ""

    if event['detail']['context']['contextType'] == 'GLUE_DATA_CATALOG':
        log_metadata['ruleset_name'] = str(event['detail']['rulesetNames'][0])
        log_metadata['tableName'] = str(event['detail']['context']['tableName'])
        log_metadata['databaseName'] = str(event['detail']['context']['databaseName'])
        log_metadata['runId'] = str(event['detail']['context']['runId'])
        log_metadata['resultId'] = str(event['detail']['resultId'])
        log_metadata['state'] = str(event['detail']['state'])
        log_metadata['score'] = str(event['detail']['score'])
        log_metadata['numRulesSucceeded'] = str(event['detail']['numRulesSucceeded'])
        log_metadata['numRulesFailed'] = str(event['detail']['numRulesFailed'])
        log_metadata['numRulesSkipped'] = str(event['detail']['numRulesSkipped'])

        message_text += "Glue Data Quality run details:\n"
        message_text += "ruleset_name: {}\n".format(log_metadata['ruleset_name'])
        message_text += "glue_table_name: {}\n".format(log_metadata['tableName'])
        message_text += "glue_database_name: {}\n".format(log_metadata['databaseName'])
        message_text += "run_id: {}\n".format(log_metadata['runId'])
        message_text += "result_id: {}\n".format(log_metadata['resultId'])
        message_text += "state: {}\n".format(log_metadata['state'])
        message_text += "score: {}\n".format(log_metadata['score'])
        message_text += "numRulesSucceeded:
{}\n".format(log_metadata['numRulesSucceeded'])
        message_text += "numRulesFailed: {}\n".format(log_metadata['numRulesFailed'])
        message_text += "numRulesSkipped: {}\n".format(log_metadata['numRulesSkipped'])

        subject_text = "Glue Data Quality ruleset {} run
details".format(log_metadata['ruleset_name'])

    else:
        log_metadata['ruleset_name'] = str(event['detail']['rulesetNames'][0])
        log_metadata['jobName'] = str(event['detail']['context']['jobName'])
        log_metadata['jobId'] = str(event['detail']['context']['jobId'])
        log_metadata['resultId'] = str(event['detail']['resultId'])
        log_metadata['state'] = str(event['detail']['state'])
        log_metadata['score'] = str(event['detail']['score'])

        log_metadata['numRulesSucceeded'] = str(event['detail']['numRulesSucceeded'])
        log_metadata['numRulesFailed'] = str(event['detail']['numRulesFailed'])
        log_metadata['numRulesSkipped'] = str(event['detail']['numRulesSkipped'])

        message_text += "Glue Data Quality run details:\n"
```

```
        message_text += "ruleset_name: {}\n".format(log_metadata['ruleset_name'])
        message_text += "glue_job_name: {}\n".format(log_metadata['jobName'])
        message_text += "job_id: {}\n".format(log_metadata['jobId'])
        message_text += "result_id: {}\n".format(log_metadata['resultId'])
        message_text += "state: {}\n".format(log_metadata['state'])
        message_text += "score: {}\n".format(log_metadata['score'])
        message_text += "numRulesSucceeded:
{}\n".format(log_metadata['numRulesSucceeded'])
        message_text += "numRulesFailed: {}\n".format(log_metadata['numRulesFailed'])
        message_text += "numRulesSkipped: {}\n".format(log_metadata['numRulesSkipped'])

        subject_text = "Glue Data Quality ruleset {} run
details".format(log_metadata['ruleset_name'])

    resultID = str(event['detail']['resultId'])
    response = glue_client.get_data_quality_result(ResultId=resultID)
    RuleResults = response['RuleResults']
    message_text += "\n\nruleset details evaluation steps results:\n\n"
    subresult_info = []

    for dic in RuleResults:
        subresult = "Name: {}\t\tResult: {}\t\tDescription: \t{}".format(dic['Name'],
dic['Result'], dic['Description'])
        if 'EvaluationMessage' in dic:
            subresult += "\t\tEvaluationMessage: {}".format(dic['EvaluationMessage'])
        subresult_info.append({
            'Name': dic['Name'],
            'Result': dic['Result'],
            'Description': dic['Description'],
            'EvaluationMessage': dic.get('EvaluationMessage', '')
        })
        message_text += "\n" + subresult

    log_metadata['resultrun'] = subresult_info



    sns_client.publish(
        TopicArn=sns_topic_arn,
        Message=message_text,
        Subject=subject_text
    )

    return {
```

```
        'statusCode': 200,
        'body': json.dumps('Message published to SNS topic')
    }
```

## Imposta avvisi e notifiche in integrazione CloudWatch

Il nostro approccio consigliato consiste nell'impostare avvisi sulla qualità dei dati utilizzando Amazon EventBridge, poiché Amazon EventBridge richiede una configurazione unica per avvisare i clienti. Tuttavia, alcuni clienti preferiscono Amazon CloudWatch per familiarità. Per tali clienti, offriamo l'integrazione con Amazon CloudWatch.

Ogni valutazione di AWS Glue Data Quality emette un paio di metriche denominate `glue.data.quality.rules.passed` (che indicano un numero di regole passate) e `glue.data.quality.rules.failed` (che indica il numero di regole non riuscite) per ogni esecuzione sulla qualità dei dati. È possibile utilizzare questo parametro emesso per creare allarmi per avvisare gli utenti se un determinato ciclo di qualità dei dati scende al di sotto di una soglia. Per iniziare a configurare un allarme che invii un'e-mail tramite una notifica Amazon SNS, procedi nel modo seguente:

Per iniziare a configurare un allarme che invii un'e-mail tramite una notifica Amazon SNS, procedi nel modo seguente:

1. Apri la CloudWatch console Amazon.
2. Scegli Tutti i parametri in Parametri. Vedrai uno spazio dei nomi aggiuntivo in Spazi dei nomi personalizzati intitolato Qualità dei dati di Glue.

   > **Note**
   >
   > Quando avvii un'esecuzione di AWS Glue Data Quality, assicurati che la CloudWatch casella di controllo Publish metrics to Amazon sia abilitata. Altrimenti, le metriche per quella particolare corsa non verranno pubblicate su Amazon CloudWatch.

   Nello spazio del nome `Glue Data Quality`, puoi visualizzare i parametri emessi per tabella e per set di regole. Ai fini di questo argomento, useremo la regola `glue.data.quality.rules.failed` e attiveremo l'allarme se questo valore supera 1 (indicando che, se riscontriamo un numero di valutazioni delle regole non riuscite superiore a 1, vogliamo ricevere una notifica).

3. Per creare l'allarme, scegli Tutti gli allarmi in Allarmi.
4. Scegli Crea allarme.
5. Scegli Seleziona metrica.
6. Seleziona il parametro `glue.data.quality.rules.failed` corrispondente alla tabella che hai creato, quindi scegli Seleziona parametro.
7. Nella scheda Specifica parametri e condizioni, nella sezione Parametri:
   a. Per Statistic (Statistica), scegliere Sum (Somma).
   b. Per Periodo, scegli 1 minuto.
8. Nella sezione Condizioni:
   a. For Threshold type (Tipo di soglia), scegli Static (Statica).
   b. Per Quando glue.data.quality.rules.failed è..., seleziona Maggiore di/uguale a.
   c. Per Oltre..., inserisci 1 come valore di soglia.

   Queste selezioni implicano che se il parametro `glue.data.quality.rules.failed` emette un valore maggiore o uguale a 1, attiveremo un allarme. Tuttavia, se non ci sono dati, lo considereremo accettabile.
9. Scegli Next (Successivo).
10. In Configura operazioni:
    a. Per Attivazione dello stato di allarme, scegli In allarme.
    b. Nella sezione Invia una notifica al seguente argomento SNS, scegli Crea un nuovo argomento per inviare una notifica tramite un nuovo argomento SNS.
    c. In Endpoint e-mail che riceveranno la notifica, inserisci il tuo indirizzo e-mail. Quindi, fai clic su Crea argomento.
    d. Scegli Next (Successivo).
11. In Nome dell'allarme, inserisci `myFirstDQAlarm`, quindi seleziona Successivo.
12. Viene visualizzata una pagina di riepilogo di tutte le selezioni. Scegli Crea allarme in basso.

Ora puoi vedere l'allarme creato dalla dashboard degli CloudWatch allarmi di Amazon.

# Esecuzione di query di risultati sulla qualità dei dati per creare pannelli di controllo

Potresti voler creare un pannello di controllo per visualizzare i risultati di qualità dei dati. Ci sono due modi per effettuare questa operazione:

Configura Amazon EventBridge con il seguente codice per scrivere i dati su Amazon S3:

```
import boto3
import json
from datetime import datetime


s3_client = boto3.client('s3')
glue_client = boto3.client('glue')


s3_bucket = 's3-bucket-name'

def write_logs(log_metadata):
    try:
        filename = datetime.now().strftime("%m%d%Y%H%M%S") + ".json"
        key_opts = {
            'year': datetime.now().year,
            'month': "{:02d}".format(datetime.now().month),
            'day': "{:02d}".format(datetime.now().day),
            'filename': filename
        }
        s3key = "gluedataqualitylogs/year={year}/month={month}/day={day}/
{filename}".format(**key_opts)
        s3_client.put_object(Bucket=s3_bucket, Key=s3key,
 Body=json.dumps(log_metadata))
    except Exception as e:
        print(f'Error writing logs to S3: {e}')


def lambda_handler(event, context):
    log_metadata = {}
    message_text = ""
    subject_text = ""

    if event['detail']['context']['contextType'] == 'GLUE_DATA_CATALOG':
        log_metadata['ruleset_name'] = str(event['detail']['rulesetNames'][0])
```

```
        log_metadata['tableName'] = str(event['detail']['context']['tableName'])
        log_metadata['databaseName'] = str(event['detail']['context']['databaseName'])
        log_metadata['runId'] = str(event['detail']['context']['runId'])
        log_metadata['resultId'] = str(event['detail']['resultId'])
        log_metadata['state'] = str(event['detail']['state'])
        log_metadata['score'] = str(event['detail']['score'])
        log_metadata['numRulesSucceeded'] = str(event['detail']['numRulesSucceeded'])
        log_metadata['numRulesFailed'] = str(event['detail']['numRulesFailed'])
        log_metadata['numRulesSkipped'] = str(event['detail']['numRulesSkipped'])

        message_text += "Glue Data Quality run details:\n"
        message_text += "ruleset_name: {}\n".format(log_metadata['ruleset_name'])
        message_text += "glue_table_name: {}\n".format(log_metadata['tableName'])
        message_text += "glue_database_name: {}\n".format(log_metadata['databaseName'])
        message_text += "run_id: {}\n".format(log_metadata['runId'])
        message_text += "result_id: {}\n".format(log_metadata['resultId'])
        message_text += "state: {}\n".format(log_metadata['state'])
        message_text += "score: {}\n".format(log_metadata['score'])
        message_text += "numRulesSucceeded:
{}\n".format(log_metadata['numRulesSucceeded'])
        message_text += "numRulesFailed: {}\n".format(log_metadata['numRulesFailed'])
        message_text += "numRulesSkipped: {}\n".format(log_metadata['numRulesSkipped'])

        subject_text = "Glue Data Quality ruleset {} run
details".format(log_metadata['ruleset_name'])

    else:
        log_metadata['ruleset_name'] = str(event['detail']['rulesetNames'][0])
        log_metadata['jobName'] = str(event['detail']['context']['jobName'])
        log_metadata['jobId'] = str(event['detail']['context']['jobId'])
        log_metadata['resultId'] = str(event['detail']['resultId'])
        log_metadata['state'] = str(event['detail']['state'])
        log_metadata['score'] = str(event['detail']['score'])

        log_metadata['numRulesSucceeded'] = str(event['detail']['numRulesSucceeded'])
        log_metadata['numRulesFailed'] = str(event['detail']['numRulesFailed'])
        log_metadata['numRulesSkipped'] = str(event['detail']['numRulesSkipped'])

        message_text += "Glue Data Quality run details:\n"
        message_text += "ruleset_name: {}\n".format(log_metadata['ruleset_name'])
        message_text += "glue_job_name: {}\n".format(log_metadata['jobName'])
        message_text += "job_id: {}\n".format(log_metadata['jobId'])
        message_text += "result_id: {}\n".format(log_metadata['resultId'])
        message_text += "state: {}\n".format(log_metadata['state'])
```

```
        message_text += "score: {}\n".format(log_metadata['score'])
        message_text += "numRulesSucceeded:
{}\n".format(log_metadata['numRulesSucceeded'])
        message_text += "numRulesFailed: {}\n".format(log_metadata['numRulesFailed'])
        message_text += "numRulesSkipped: {}\n".format(log_metadata['numRulesSkipped'])

        subject_text = "Glue Data Quality ruleset {} run
details".format(log_metadata['ruleset_name'])

    resultID = str(event['detail']['resultId'])
    response = glue_client.get_data_quality_result(ResultId=resultID)
    RuleResults = response['RuleResults']
    message_text += "\n\nruleset details evaluation steps results:\n\n"
    subresult_info = []

    for dic in RuleResults:
        subresult = "Name: {}\t\tResult: {}\t\tDescription: \t{}".format(dic['Name'],
dic['Result'], dic['Description'])
        if 'EvaluationMessage' in dic:
            subresult += "\t\tEvaluationMessage: {}".format(dic['EvaluationMessage'])
        subresult_info.append({
            'Name': dic['Name'],
            'Result': dic['Result'],
            'Description': dic['Description'],
            'EvaluationMessage': dic.get('EvaluationMessage', '')
        })
        message_text += "\n" + subresult

    log_metadata['resultrun'] = subresult_info

    write_logs(log_metadata)


    return {
        'statusCode': 200,
        'body': json.dumps('Message published to SNS topic')
    }
```

Dopo aver scritto su Amazon S3, puoi usare i crawler AWS Glue per registrarti su Athena e interrogare le tabelle.

Configurazione di una posizione Amazon S3 durante una valutazione della qualità dei dati:

Quando esegui attività di qualità dei dati in AWS Glue Data Catalog o AWS Glue ETL, puoi fornire una posizione Amazon S3 per scrivere i risultati sulla qualità dei dati su Amazon S3. Per creare una tabella facendo riferimento alla destinazione per leggere i risultati di qualità dei dati, puoi utilizzare la sintassi seguente.

Tieni presente che è necessario eseguire le query `CREATE EXTERNAL TABLE` e `MSCK REPAIR TABLE` separatamente.

```
CREATE EXTERNAL TABLE <my_table_name>(
    catalogid string,
    databasename string,
    tablename string,
    dqrunid string,
    evaluationstartedon timestamp,
    evaluationcompletedon timestamp,
    rule string,
    outcome string,
    failurereason string,
    evaluatedmetrics string)
PARTITIONED BY (
    `year` string,
    `month` string,
    `day` string)
ROW FORMAT SERDE 'org.openx.data.jsonserde.JsonSerDe'
WITH SERDEPROPERTIES (

 'paths'='catalogId,databaseName,dqRunId,evaluatedMetrics,evaluationCompletedOn,evaluationStart
STORED AS INPUTFORMAT 'org.apache.hadoop.mapred.TextInputFormat'
OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
LOCATION 's3://glue-s3-dq-bucket-us-east-2-results/'
TBLPROPERTIES (
    'classification'='json',
    'compressionType'='none',
    'typeOfData'='file');
```

```
MSCK REPAIR TABLE <my_table_name>;
```

Dopo aver creato la tabella precedente, puoi eseguire query analitiche utilizzando Amazon Athena.

## Implementazione delle regole di qualità dei dati utilizzando AWS CloudFormation

È possibile utilizzare AWS CloudFormation per creare regole di qualità dei dati. Per ulteriori informazioni, vedere AWS CloudFormation AWS Glue.

## Pianificazione delle regole di qualità dei dati

È possibile pianificare le regole di qualità dei dati utilizzando i seguenti metodi:

- Pianifica le regole di qualità dei dati dal Data Catalog: gli utenti senza codice possono utilizzare questa opzione per pianificare facilmente le scansioni della qualità dei dati. AWS Glue Data Quality creerà la pianificazione in Amazon EventBridge. Per pianificare le regole di qualità dei dati:
  - Vai al set di regole e fai clic su Esegui.
  - In Frequenza di esecuzione, seleziona la pianificazione desiderata e fornisci un Nome dell'attività. Questo nome dell'attività è il nome della tua pianificazione in EventBridge.
- Usa Amazon EventBridge e AWS Step Functions per orchestrare valutazioni e raccomandazioni per le regole di qualità dei dati.

# Crittografia dei dati a riposo per AWS Glue Data Quality

AWS Glue Data Qualityfornisce la crittografia di default per proteggere i dati sensibili dei clienti archiviati utilizzando chiavi AWS di crittografia proprietarie.

## AWS chiavi di proprietà

AWS Glue Data Quality utilizza queste chiavi per crittografare automaticamente le risorse Data Quality dei clienti. Non è possibile visualizzare, gestire o utilizzare chiavi AWS di proprietà o controllarne l'utilizzo. Tuttavia, non è necessario effettuare alcuna operazione o modificare programmi per proteggere le chiavi che eseguono la crittografia dei dati. Per ulteriori informazioni, consulta le chiavi AWS possedute nella Guida per AWS KMS gli sviluppatori.

La crittografia predefinita dei dati a riposo aiuta a ridurre il sovraccarico operativo e la complessità associati alla protezione dei dati sensibili. Allo stesso tempo, consente di creare applicazioni sicure che soddisfano i rigorosi requisiti normativi e di conformità alla crittografia.

Sebbene non sia possibile disabilitare questo livello di crittografia o selezionare un tipo di crittografia alternativo, è possibile aggiungere un secondo livello di crittografia rispetto alle chiavi di crittografia

di AWS proprietà esistenti scegliendo una chiave gestita dal cliente quando si creano le risorse Data Quality.

## Chiavi gestite dal cliente

Chiavi gestite dal cliente: AWS Glue Data Quality supporta l'uso di una chiave simmetrica gestita dal cliente che puoi creare, possedere e gestire. Ciò aggiunge un secondo livello di crittografia rispetto alla crittografia di proprietà esistente. AWS Avendo il pieno controllo di questo livello di crittografia, è possibile eseguire operazioni quali:

- Stabilire e mantenere le policy delle chiavi
- Stabilire e mantenere le policy IAM
- Abilitare e disabilitare le policy delle chiavi
- Ruotare i materiali crittografici delle chiavi
- Aggiungere tag
- Creare alias delle chiavi
- Pianificare l'eliminazione delle chiavi

Per ulteriori informazioni, consulta Customer managed keys nella AWS KMS Developer Guide.

La tabella seguente riassume il modo in cui AWS Glue Data Quality crittografa diverse risorse di Data Quality.

| Tipo di dati | AWS crittografia a chiave proprietaria | Crittografia con chiavi gestite dal cliente |
|---|---|---|
| Set di regole su Data Quality | Abilitato | Abilitato |
| Stringa del set di regole DQDL a cui fa riferimento il set di regole DQ persistente. Per il momento, questi set di regole persistenti vengono usati solo nell'esperienza Catalogo dati AWS Glue. | | |

| Tipo di dati | AWS crittografia a chiave proprietaria | Crittografia con chiavi gestite dal cliente |
| --- | --- | --- |
| Rule/Analyzer Risultati sulla qualità dei dati<br><br>Elementi di risultato che contengono lo pass/fail stato di ogni regola in un set di regole e le metriche raccolte sia dalle regole che dagli analizzatori. | Abilitato | Abilitato |
| Osservazioni<br><br>Le osservazioni vengono generate quando si rileva un'anomalia nei dati. Contengono informazioni sui limiti superiore e inferiore previsti e una regola suggerita basata su tali limiti. Se generati, vengono visualizzati con i risultati di Data Quality. | Abilitato | Abilitato |
| Statistiche<br><br>Contiene informazioni sulle metriche raccolte dopo la valutazione dei dati forniti da un set di regole, come il valore della metrica (ad esempio, Completezza), i nomi delle colonne e altri metadati. RowCount | Abilitato | Abilitato |

| Tipo di dati | AWS crittografia a chiave proprietaria | Crittografia con chiavi gestite dal cliente |
| --- | --- | --- |
| Modelli statistici di rilevamento delle anomalie<br><br>I modelli statistici contengon o le serie temporali dei limiti superiore e inferiore per una determinata metrica generata sulla base delle precedenti valutazioni dei dati dei clienti. | Abilitato | Abilitato |

> **Note**
>
> AWS Data Quality abilita automaticamente la crittografia dei dati inattivi utilizzando chiavi AWS proprietarie per proteggere gratuitamente i dati di identificazione personale. Tuttavia, l'utilizzo di una chiave gestita dal cliente comporta dei AWS KMS costi. Per ulteriori informazioni sui prezzi, consulta Prezzi di AWS KMS.
> Per ulteriori informazioni su AWS KMS, vedere AWS KMS.

## Creazione di una chiave gestita dal cliente

È possibile creare una chiave simmetrica gestita dal cliente utilizzando Console di gestione AWS, o il. AWS KMS APIs

Per creare una chiave simmetrica gestita dal cliente:

- Segui i passaggi per la creazione di AWS KMS chiavi di crittografia simmetriche nella Guida per gli sviluppatori. AWS Key Management Service

### Policy della chiave

Le policy della chiave controllano l'accesso alla chiave gestita dal cliente. Ogni chiave gestita dal cliente deve avere esattamente una policy della chiave, che contiene istruzioni che determinano chi può usare la chiave e come la possono usare. Quando crei la chiave gestita dal cliente, è possibile

specificare una policy della chiave. Per ulteriori informazioni, consulta Key Policy in AWS KMS keys nella AWS Key Management Service Developer Guide.

Per utilizzare la chiave gestita dal cliente con le risorse di Data Quality, le seguenti operazioni API devono essere permesse nella policy della chiave:

- `kms:Decrypt`— Decripta il testo cifrato che è stato crittografato da una chiave utilizzando AWS KMS `GenerateDataKeyWithoutPlaintext`
- `kms:DescribeKey`: fornisce i dettagli della chiave gestiti dal cliente per consentire ad Amazon Location di convalidare la chiave.
- `kms:GenerateDataKeyWithoutPlaintext`— Restituisce una chiave dati simmetrica unica da utilizzare all'esterno di. AWS KMS Questa operazione restituisce una chiave dati crittografata con una chiave KMS di crittografia simmetrica specificata. I byte nella chiave sono casuali e non sono correlati al chiamante o alla chiave KMS. Usata per ridurre le chiamate KMS che il cliente deve effettuare.
- `kms:ReEncrypt*`: deecrittografa il testo criptato e poi lo ricrittografa interamente all'interno di AWS KMS. È possibile sfruttare questa operazione per modificare la chiave KMS con cui sono crittografati i dati, ad esempio quando si ruota manualmente una chiave KMS o si modifica la chiave KMS che protegge un testo criptato. È inoltre possibile utilizzarla per crittografare nuovamente il testo criptato con la stessa chiave KMS, ad esempio per modificare il contesto di crittografia di un testo criptato.

Di seguito sono riportati alcuni esempi di istruzioni di policy che puoi aggiungere per Amazon Location:

```
"Statement" : [
    {
        "Sid" : "Allow access to principals authorized to use AWS Glue Data Quality",
        "Effect" : "Allow",
        "Principal" : {
            "AWS": "arn:aws:iam::<account_id>:role/ExampleRole"
        },
        "Action" : [
            "kms:Decrypt",
            "kms:DescribeKey",
            "kms:GenerateDataKeyWithoutPlaintext",
            "kms:ReEncrypt*"
        ],
        "Resource" : "*",
```

```
        "Condition" : {
            "StringEquals" : {
                "kms:ViaService" : "glue.amazonaws.com",
                "kms:CallerAccount" : "111122223333"
            }
        },
    {
        "Sid": "Allow access for key administrators",
        "Effect": "Allow",
        "Principal": {
            "AWS": "arn:aws:iam::111122223333:root"
          },
        "Action" : [
            "kms:*"
         ],
        "Resource": "arn:aws:kms:region:111122223333:key/key_ID"
    },
    {
        "Sid" : "Allow read-only access to key metadata to the account",
        "Effect" : "Allow",
        "Principal" : {
            "AWS" : "arn:aws:iam::111122223333:root"
        },
        "Action" : [
            "kms:Describe*",
            "kms:Get*",
            "kms:List*",
        ],
        "Resource" : "*"
    }
]
```

## Note sull'uso delle chiavi KMS in AWS Glue Data Quality

AWS Glue Data Quality non supporta le transizioni chiave. Ciò significa che se crittografi le tue risorse Data Quality con la chiave A e decidi di passare alla chiave B, non crittograferemo nuovamente i dati crittografati con la chiave A per usare la chiave B. Puoi comunque passare alla chiave B, ma dovrai mantenere l'accesso alla chiave A per accedere ai dati precedentemente crittografati con la chiave A.

Per ulteriori informazioni sulla specificazione delle autorizzazioni in una politica, consulta Autorizzazioni per AWS i servizi nelle politiche chiave nella Guida per gli sviluppatori. AWS Key Management Service

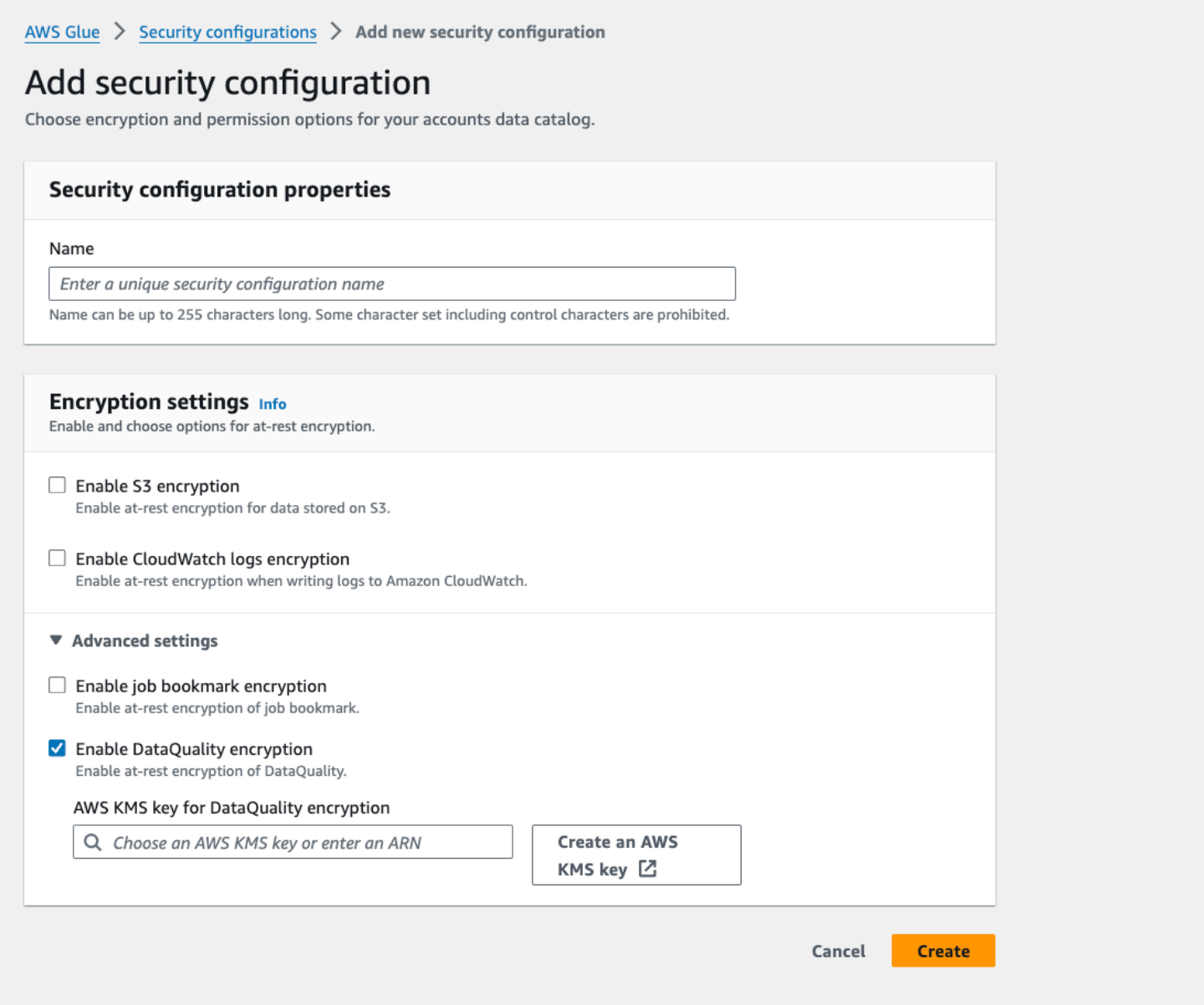
Per ulteriori informazioni sulla risoluzione dei problemi di accesso tramite chiave, consulta Risoluzione dei problemi di accesso tramite chiave nella Guida per gli AWS Key Management Service sviluppatori.

## Creazione di una configurazione di sicurezza

In AWS Glue la risorsa Security Configurations contiene le proprietà necessarie quando si scrivono dati crittografati.

Per crittografare le risorse relative alla qualità dei dati:

1. Nelle impostazioni di crittografia, in Impostazioni avanzate, seleziona Abilita crittografia della qualità dei dati
2. Seleziona la tua chiave KMS o scegli Crea una chiave AWS KMS




# AWS Contesto di crittografia Glue Data Quality

Un contesto di crittografia è un set facoltativo di coppie chiave-valore che contengono ulteriori informazioni contestuali sui dati.

AWS KMS utilizza il contesto di crittografia come dati autenticati aggiuntivi per supportare la crittografia autenticata. Quando includi un contesto di crittografia in una richiesta di crittografia dei dati, AWS KMS associa il contesto di crittografia ai dati crittografati. Per decrittografare i dati, nella richiesta deve essere incluso lo stesso contesto di crittografia.

### AWS Esempio di contesto di crittografia Glue Data Quality

```
"encryptionContext": {
    "kms-arn": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
    "branch-key-id": "111122223333+arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
    "hierarchy-version": "1",
    "aws-crypto-ec:aws:glue:securityConfiguration": "111122223333:customer-security-
configuration-name",
    "create-time": "2024-06-07T13:47:23:000861Z",
    "tablename": "AwsGlueMlEncryptionKeyStore",
    "type": "beacon:ACTIVE"
}
```

### Utilizzo del contesto di crittografia per il monitoraggio

Quando si utilizza una chiave simmetrica gestita dal cliente per crittografare il tracker o la collezione di recinzioni geografiche virtuali, è possibile utilizzare il contesto di crittografia anche nei record e nei log di controllo per identificare come viene utilizzata la chiave gestita dal cliente. Il contesto di crittografia appare anche nei log generati da AWS CloudTrail o Amazon CloudWatch Logs.

## Monitoraggio delle chiavi di crittografia per AWS Glue Data Quality

Quando utilizzi una chiave gestita AWS KMS dal cliente con le tue risorse AWS Glue Data Quality, puoi utilizzare AWS CloudTrail o tenere traccia delle richieste Amazon CloudWatch Logs a cui AWS Glue Data Quality invia AWS KMS.

Gli esempi seguenti sono AWS CloudTrail eventi per `GenerateDataKeyWithoutPlainText` e per `Decrypt` monitorare le operazioni KMS chiamate da AWS Glue Data Quality per accedere ai dati crittografati dalla chiave gestita dal cliente.

Decrypt

```
{
    "eventVersion": "1.09",
    "userIdentity": {
        "type": "AssumedRole",
        "arn": "arn:aws:sts::111122223333:role/CustomerRole",
        "accountId": "111122223333",
        "invokedBy": "glue.amazonaws.com"
    },
```

```
    "eventTime": "2024-07-02T20:03:10Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "glue.amazonaws.com",
    "userAgent": "glue.amazonaws.com",
    "requestParameters": {
        "keyId": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
        "encryptionContext": {
            "kms-arn": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
            "branch-key-id": "111122223333+arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
            "hierarchy-version": "1",
            "aws-crypto-ec:aws:glue:securityConfiguration": "111122223333:customer-
security-configuration-name",
            "create-time": "2024-06-07T13:47:23:000861Z",
            "tablename": "AwsGlueMlEncryptionKeyStore",
            "type": "branch:ACTIVE",
            "version": "branch:version:ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE"
        }
    },
    "responseElements": null,
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

### GenerateDataKeyWithoutPlaintext

```
{
    "eventVersion": "1.09",
    "userIdentity": {
        "type": "AssumedRole",
        "arn": "arn:aws:sts::111122223333:role/CustomerRole",
        "accountId": "111122223333",
        "invokedBy": "glue.amazonaws.com"
    },
    "eventTime": "2024-07-02T20:03:10Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKeyWithoutPlaintext",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "glue.amazonaws.com",
    "userAgent": "glue.amazonaws.com",
    "requestParameters": {
        "keyId": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
        "encryptionContext": {
            "kms-arn": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
            "branch-key-id": "111122223333+arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
            "hierarchy-version": "1",
            "aws-crypto-ec:aws:glue:securityConfiguration": "111122223333:customer-
security-configuration-name",
            "create-time": "2024-06-07T13:47:23:000861Z",
            "tablename": "AwsGlueMlEncryptionKeyStore",
            "type": "branch:version:ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE"
        }
    },
    "responseElements": null,
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
```

```
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

## ReEncyrpt

```
{
    "eventVersion": "1.09",
    "userIdentity": {
        "type": "AssumedRole",
        "arn": "arn:aws:sts::111122223333:role/CustomerRole",
        "accountId": "111122223333",
        "invokedBy": "glue.amazonaws.com"
    },
    "eventTime": "2024-07-17T21:34:41Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "ReEncrypt",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "glue.amazonaws.com",
    "userAgent": "glue.amazonaws.com",
    "requestParameters": {
        "destinationEncryptionContext": {
             "kms-arn": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
            "branch-key-id": "111122223333+arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
            "hierarchy-version": "1",
            "aws-crypto-ec:aws:glue:securityConfiguration": "111122223333:customer-
security-configuration-name",
            "create-time": "2024-06-07T13:47:23:000861Z",
            "tablename": "AwsGlueMlEncryptionKeyStore",
            "type": "branch:ACTIVE"
            "version": "branch:version:12345678-SAMPLE"
        },
        "destinationKeyId": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
        "sourceAAD": "1234567890-SAMPLE+Z+lqoYOHj7VtWxJLrvh+biUFbliYDAQkobM=",
        "sourceKeyId": "arn:aws:kms:ap-southeast-2:585824196334:key/17ca05ca-a8c1-40d7-
b7fd-30abb569a53a",
        "destinationEncryptionAlgorithm": "SYMMETRIC_DEFAULT",
        "sourceEncryptionContext": {
```

```
            "kms-arn": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
            "branch-key-id": "111122223333+arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
            "hierarchy-version": "1",
            "aws-crypto-ec:aws:glue:securityConfiguration": "111122223333:customer-
security-configuration-name",
            "create-time": "2024-06-07T13:47:23:000861Z",
            "tablename": "AwsGlueMlEncryptionKeyStore",
            "type": "branch:version:ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE"
        },
        "destinationAAD": "1234567890-SAMPLE",
        "sourceEncryptionAlgorithm": "SYMMETRIC_DEFAULT"
    },
    "responseElements": null,
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        },
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```


## Ulteriori informazioni

Le seguenti risorse forniscono ulteriori informazioni sulla crittografia dei dati a riposo.

- Per ulteriori informazioni sui concetti di AWS Key Management Service base, consulta la Guida per gli AWS Key Management Service sviluppatori.
- Per ulteriori informazioni sulle migliori pratiche di sicurezza, AWS Key Management Service consulta la Guida per gli AWS Key Management Service sviluppatori.

# Risoluzione degli errori di AWS Glue Data Quality

Se riscontri errori in AWS Glue Data Quality, utilizza le seguenti soluzioni per aiutarti a trovare l'origine dei problemi e risolverli.

Indice


- Errore: modulo AWS Glue Data Quality mancante
- Errore: permessi di AWS Lake Formation insufficienti
- Errore: i set di regole non hanno un nome univoco
- Errore: tabelle con caratteri speciali
- Errore: errore di overflow con un set di regole di grandi dimensioni
- Errore: lo stato generale della regola è non riuscito
- AnalysisException: impossibile verificare l'esistenza del database predefinito
- Messaggio di errore: Provided key map not suitable for given data frames
- Eccezione nella classe utente: java.lang. RuntimeException : Impossibile recuperare i dati. Controlla i log in CloudWatch per avere maggiori dettagli
- ERRORE DI AVVIO: errore durante il download da S3 per il bucket
- InvalidInputException (status: 400): DataQuality le regole non possono essere analizzate
- Errore: EventBridge non attiva i processi Qualità dei dati di Glue in base alla pianificazione che ho impostato
- Errori CustomSQL
- Regole dinamiche
- Eccezione nella classe utente: org.apache.spark.sql. AnalysisException: org.apache.hadoop.hive.ql.metadata. HiveException
- UNCLASSIFIED_ERROR; IllegalArgumentException: Errore di analisi: non sono state fornite regole o analizzatori., nessuna valida alternativa in fase di input

## Errore: modulo AWS Glue Data Quality mancante

Messaggio di errore: No module named 'awsgluedq'.

Risoluzione: questo errore si verifica quando si esegue AWS Glue Data Quality in una versione non supportata. AWS Glue Data Quality è supportato solo nella versione 3.0 e successive di Glue.

## Errore: permessi di AWS Lake Formation insufficienti

Messaggio di errore: Eccezione nella classe utente:`com.amazonaws.services.glue.model.AccessDeniedException`: Autorizzazioni Lake Formation insufficienti su impact_sdg_engagement (Service:; Codice di stato: 400; Codice di errore:AWS Glue; ID richiesta: AccessDeniedException 465ae693-b7ba-4df0-a4e4-6b17xxxxxxxx; Proxy: null).

Risoluzione: è necessario fornire autorizzazioni sufficienti in AWS Lake Formation.

## Errore: i set di regole non hanno un nome univoco

Messaggio di errore: eccezione nella classe utente:... services.glue.model. AlreadyExistsException: Esiste già un altro set di regole con lo stesso nome.

Risoluzione: i set di regole sono globali e devono essere univoci.

## Errore: tabelle con caratteri speciali

Messaggio di errore: eccezione nella classe utente: org.apache.spark.sql. AnalysisException: impossibile risolvere le colonne di input «C» fornite: [primary.data_end_time, primary.data_start_time, primary.end_time, primary.last_updated, primary.message, primary.process_date, primary.rowhash, primary.run_by, primary.run_id, primary.start_time, primary.status]; riga 1 pos 44;.

Risoluzione: attualmente esiste una limitazione per cui AWS Glue Data Quality non può essere eseguito su tabelle che contengono caratteri speciali come «.».

## Errore: errore di overflow con un set di regole di grandi dimensioni

Messaggio di errore: Eccezione nella classe utente: java.lang. StackOverflowError.

Risoluzione: se disponi di un set di regole di grandi dimensioni con più di 2.000 regole, potresti riscontrare questo problema. Suddividi le tue regole in più set di regole.

## Errore: lo stato generale della regola è non riuscito

Condizione di errore: il mio set di regole ha esito positivo, ma lo stato generale delle regole non è riuscito.

Risoluzione: questo errore si è probabilmente verificato perché hai scelto l'opzione per pubblicare le metriche su Amazon CloudWatch durante la pubblicazione. Se il tuo set di dati è in un VPC, il tuo VPC potrebbe non consentire a AWS Glue di pubblicare metriche su Amazon. CloudWatch In questo caso, devi >configurare un endpoint per consentire al tuo VPC di accedere ad Amazon. CloudWatch

## AnalysisException: impossibile verificare l'esistenza del database predefinito

Condizione di errore AnalysisException: impossibile verificare l'esistenza del database predefinito: com.amazonaws.services.glue.model. AccessDeniedException: Autorizzazioni Lake Formation insufficienti per impostazione predefinita (Service:AWS Glue; Codice di stato: 400; Codice di errore:; ID richiesta: XXXXXXXX-XXXX-XXXX-XXXX -XXXXXXXXXXXX AccessDeniedException; Proxy: null)

Risoluzione: nell'integrazione del catalogo del processo AWS Glue, AWS Glue cerca sempre di verificare se il database predefinito esiste o meno utilizzando l'`GetDatabase API` AWS Glue. Quando l'autorizzazione `DESCRIBE` Lake Formation non viene concessa o viene concessa l'autorizzazione `GetDatabase IAM`, il processo ha esito negativo durante la verifica dell'esistenza del database predefinito.

Per risolvere:

1. Aggiungi l'autorizzazione `DESCRIBE` in Lake Formation per il database predefinito.
2. Configura il ruolo IAM associato al processo AWS Glue come Creatore di database in Lake Formation. Questo creerà automaticamente un database predefinito e concederà le autorizzazioni Lake Formation richieste per il ruolo.
3. Disabilita l'opzione `--enable-data-catalog`. Viene visualizzato come Utilizza Data Catalog come metastore Hive in AWS Glue Studio.

   Se l'integrazione Data Catalog Spark SQL non è necessaria nel processo, è possibile disabilitarla.

## Messaggio di errore: Provided key map not suitable for given data frames

Condizione di errore: la mappa delle chiavi fornita non è adatta a determinati frame di dati.

Risoluzione: stai usando il tipo di DataSetMatchregola e le chiavi di join hanno dei duplicati. Le tue chiavi di join devono essere univoche e non possono essere NULL. Nei casi in cui non puoi avere chiavi di join univoche, prendi in considerazione l'utilizzo di altri tipi di regole, AggregateMatchad esempio la corrispondenza nei dati di riepilogo.

## Eccezione nella classe utente: java.lang. RuntimeException : Impossibile recuperare i dati. Controlla i log in CloudWatch per avere maggiori dettagli

Condizione di errore: eccezione nella classe utente: java.lang. RuntimeException : Impossibile recuperare i dati. Controlla i log in CloudWatch per avere maggiori dettagli.

Risoluzione: questo accade quando crei regole DQ su una tabella basata su Amazon S3 confrontabile con Amazon RDS o. Amazon Redshift In questi casi, AWS Glue non è in grado di caricare la connessione. Prova invece a configurare la regola DQ sul set di dati Amazon Redshift o Amazon RDS. Si tratta di un bug noto.

## ERRORE DI AVVIO: errore durante il download da S3 per il bucket

Condizione di errore: ERRORE DI AVVIO: Errore durante il download da S3 per il bucket: `aws-glue-ml-data-quality-assets-us-east-1, key: jars/aws-glue-ml-data-quality-etl.jar.Access Denied (Service: Amazon S3; Status Code: 403; Please refer logs for details)` .

Risoluzione: le autorizzazioni relative al ruolo passate a AWS Glue Data Quality devono consentire la lettura dalla precedente posizione Amazon S3. Al ruolo deve essere collegata questa policy IAM:

```
{
  "Sid": "allowS3",
  "Effect": "Allow",
  "Action": "s3:GetObject",
  "Resource": "arn:aws:s3:::aws-glue-ml-data-quality-assets-<region>/*"
}
```

Fai riferimento all'Autorizzazione di Qualità dei dati per le autorizzazioni dettagliate. Queste librerie sono necessarie per valutare la qualità dei dati per i tuoi set di dati.

## InvalidInputException (status: 400): DataQuality le regole non possono essere analizzate

Condizione di errore: InvalidInputException (status: 400): DataQuality le regole non possono essere analizzate.

Risoluzione: sono molte le possibili cause di questo errore. Una possibilità è che le regole siano racchiuse tra virgolette singole. Verifica che siano racchiuse tra virgolette doppie. Esempio:

```
Rules = [
ColumnValues "tipo_vinculo" in ["CODO", "DOCO", "COCO", "DODO"] AND "categoria" = 'ES'
    AND "cod_bandera" = 'CEP'
```

Modificalo con:

```
Rules = [
(ColumnValues "tipovinculo" in [ "CODO", "DOCO", "COCO", "DODO"]) AND (ColumnValues
 "categoria" = "ES")
    AND (ColumnValues "codbandera" = "CEP")
]
```

## Errore: EventBridge non attiva i processi Qualità dei dati di Glue in base alla pianificazione che ho impostato

Condizione di errore: EventBridge non attiva i processi AWS Glue Data Quality in base alla pianificazione che ho impostato.

Risoluzione: il ruolo che attiva il processo potrebbe non avere le autorizzazioni corrette. Assicurati che il ruolo che stai utilizzando per avviare i processi disponga delle autorizzazioni menzionate nella sezione Configurazione IAM richiesta per la pianificazione delle esecuzioni di valutazione.

## Errori CustomSQL

Condizione di errore: `The output from CustomSQL must contain at least one column that matches the input dataset for AWS Glue Data Quality to provide row level results. The SQL query is a valid query but no columns from the SQL`

`result are present in the Input Dataset. Ensure that matching columns are returned from the SQL`.

Risoluzione: la query SQL è valida, ma assicurati di selezionare solo le colonne della tabella primaria. La selezione di funzioni aggregate come somma o conteggio delle colonne della tabella primaria può causare questo errore.

Condizione di errore: `There was a problem when executing your SQL statement: cannot resolve "Col"`.

Risoluzione: questa colonna non è presente nella tabella primaria.

Condizione di errore: `The columns that are returned from the SQL statement should only belong to the primary table. "In this case, some columns ( Col ) belong to reference table"`.

Risoluzione: nelle query SQL, quando esegui il join della tabella primaria con altre tabelle di riferimento, assicurati che l'istruzione select contenga solo i nomi di colonna della tabella primaria per generare risultati a livello di riga per tale tabella.

## Regole dinamiche

Condizione di errore: `Dynamic rules require job context, and cannot be evaluated in interactive session or data preview.`.

Causa: questo messaggio di errore potrebbe apparire nei risultati dell'anteprima dei dati, o in altre sessioni interattive, quando nel set di regole sono presenti regole di qualità dei dati dinamiche. Le regole dinamiche fanno riferimento alle metriche storiche associate a un particolare nome di processo e contesto di valutazione, quindi non possono essere valutate nelle sessioni interattive.

Risoluzione: l'esecuzione del processo AWS Glue produrrà metriche storiche, a cui è possibile fare riferimento nelle esecuzioni successive per lo stesso processo.

Condizione di errore:

- `[RuleType] rule only supports simple atomic operands in thresholds.`.
- `Function last not yet implemented for [RuleType] rule.`

Risoluzione: le regole dinamiche sono generalmente supportate per tutti i tipi di regole DQDL nelle espressioni numeriche (consulta il Riferimento a Data Quality Definition Language (DQDL)). Tuttavia, alcune regole che producono più metriche non sono ancora supportate. ColumnValues ColumnLength

Condizione di errore: `Binary expression operands must resolve to a single number.`.

Causa: le regole dinamiche supportano le espressioni binarie, come `RowCount > avg(last(5)) * 0.9`. In questo caso, l'espressione binaria è `avg(last(5)) * 0.9`. Questa regola è valida perché entrambi gli operandi `avg(last(5))` e `0.9` si risolvono in un unico numero. Un esempio errato è `RowCount > last(5) * 0.9`, perché `last(5)` produrrà un elenco che non può essere confrontato in modo significativo con il conteggio delle righe corrente.

Risoluzione: utilizza le funzioni di aggregazione per ridurre un operando con valori di elenco a un unico numero.

Condizione di errore:

- `Rule threshold results in list, and a single value is expected. Use aggregation functions to produce a single value. Valid example: sum(last(10)), avg(last(10)).`
- `Rule threshold results in empty list, and a single value is expected.`

Causa: è possibile utilizzare regole dinamiche per confrontare alcune funzionalità del set di dati con i valori storici corrispondenti. L'ultima funzione consente il recupero di più valori storici, se viene fornito un argomento intero positivo. Ad esempio, `last(5)` recupererà i cinque valori più recenti osservati nelle esecuzioni dei processi per la regola.

Risoluzione: è necessario utilizzare una funzione di aggregazione per ridurre questi valori a un unico numero per effettuare un confronto significativo con il valore osservato nell'esecuzione del processo corrente.

Esempi validi:

- `RowCount >= avg(last(5))`
- `RowCount > last(1)`
- `RowCount < last()`

Esempio non valido: `RowCount > last(5)`.

Condizione di errore:

- `Function index used in threshold requires positive integer argument.`
- `Index argument must be an integer. Valid syntax example: RowCount > index(last(10, 2)), which means RowCount must be greater than third most recent execution from last 10 job runs.`

Risoluzione: durante la creazione di regole dinamiche, è possibile utilizzare la funzione di aggregazione `index` per selezionare un valore storico da un elenco. Ad esempio, `RowCount > index(last(5)`, 1) controllerà se il conteggio delle righe osservato nel processo corrente è strettamente maggiore del secondo conteggio di righe più recente osservato per il processo. `index` è indicizzato a zero.

Condizione di errore: `IllegalArgumentException: Parsing Error: Rule Type: DetectAnomalies is not valid`.

Risoluzione: il rilevamento delle anomalie è disponibile solo in AWS Glue 4.0.

Condizione di errore: `IllegalArgumentException: Parsing Error: Unexpected condition for rule of type ... no viable alternative at input ...`.

Nota: `...` è dinamico. Esempio: `IllegalArgumentException: Parsing Error: Unexpected condition for rule of type RowCount with number return type, line 4:19 no viable alternative at input '>last'`.

Risoluzione: il rilevamento delle anomalie è disponibile solo in AWS Glue 4.0.

## Eccezione nella classe utente: org.apache.spark.sql. AnalysisException: org.apache.hadoop.hive.ql.metadata. HiveException

Condizione di errore `: Exception in User Class: org.apache.spark.sql.AnalysisException: org.apache.hadoop.hive.ql.metadata.HiveException: Unable to fetch table mailpiece_submitted. StorageDescriptor#InputFormat cannot be null for table: mailpiece_submitted (Service: null; Status Code: 0; Error Code: null; Request ID: null; Proxy: null)`

Causa: stai usando Apache Iceberg in AWS Glue Data Catalog e l'attributo Input Format in AWS Glue Data Catalog è vuoto.

Soluzione: questo problema si verifica quando usi il tipo di regola CustomSQL nella tua regola DQ. Un modo per risolvere questo problema è utilizzare "primario" o aggiungere il nome del catalogo `glue_catalog.` a `<database>.<table> in Custom ruletype` nel tipo di regola personalizzata.

## UNCLASSIFIED_ERROR; IllegalArgumentException: Errore di analisi: non sono state fornite regole o analizzatori., nessuna valida alternativa in fase di input

Condizione di errore : `UNCLASSIFIED_ERROR; IllegalArgumentException: Parsing Error: No rules or analyzers provided., no viable alternative at input`

Risoluzione: DQDL non è analizzabile. Ci sono alcuni casi in cui ciò può verificarsi. Se usi regole composite, assicurati che le parentesi giuste siano inserite correttamente.

```
(RowCount >= avg(last(10)) * 0.6) and (RowCount <= avg(last(10)) * 1.4) instead of
      RowCount >= avg(last(10)) * 0.6 and RowCount <= avg(last(10)) * 1.4
```

# Integrazione dei dati di Amazon Q in AWS Glue

L'integrazione dei dati di Amazon Q AWS Glue è una nuova funzionalità di intelligenza artificiale generativa AWS Glue che consente ai data engineer e agli sviluppatori ETL di creare lavori di integrazione dei dati utilizzando il linguaggio naturale. Ingegneri e sviluppatori possono chiedere ad Amazon Q di creare lavori, risolvere problemi e rispondere a domande sull' AWS Glue integrazione dei dati.

## Che cos'è Amazon Q?

> **Note**
>
> Realizzato da Amazon Bedrock: AWS implementa il rilevamento automatico degli abusi. Poiché l'integrazione dei dati di Amazon Q è basata su Amazon Bedrock, gli utenti possono sfruttare appieno i controlli implementati in Amazon Bedrock per garantire la sicurezza, la protezione e l'uso responsabile dell'intelligenza artificiale (IA).

Amazon Q è un assistente conversazionale basato sull'intelligenza artificiale generativa (AI) che può aiutarti a comprendere, creare, estendere e utilizzare le applicazioni. AWS Il modello alla base di Amazon Q è stato arricchito con AWS contenuti di alta qualità per fornirti risposte più complete, attuabili e referenziate per accelerare la tua crescita. AWS Per ulteriori informazioni, consultare Che cos'è Amazon Q?

## Che cos'è l'integrazione dei dati di Amazon Q in AWS Glue?

L'integrazione dei dati di Amazon Q in AWS Glue include le seguenti funzionalità:

- Chat: Amazon Q Data Integration in AWS Glue può rispondere a domande in linguaggio naturale in inglese su AWS Glue domini di integrazione dei dati come connettori di AWS Glue origine e destinazione, job AWS Glue ETL, Data Catalog, crawler e AWS Lake Formation altra documentazione sulle funzionalità e best practice. L'integrazione dei dati di Amazon Q AWS Glue risponde con step-by-step istruzioni e include riferimenti alle sue fonti di informazioni.
- Generazione di codice di integrazione dei dati: Amazon Q Data Integration in AWS Glue può rispondere a domande sugli script AWS Glue ETL e generare nuovo codice in base a una domanda in linguaggio naturale in inglese.

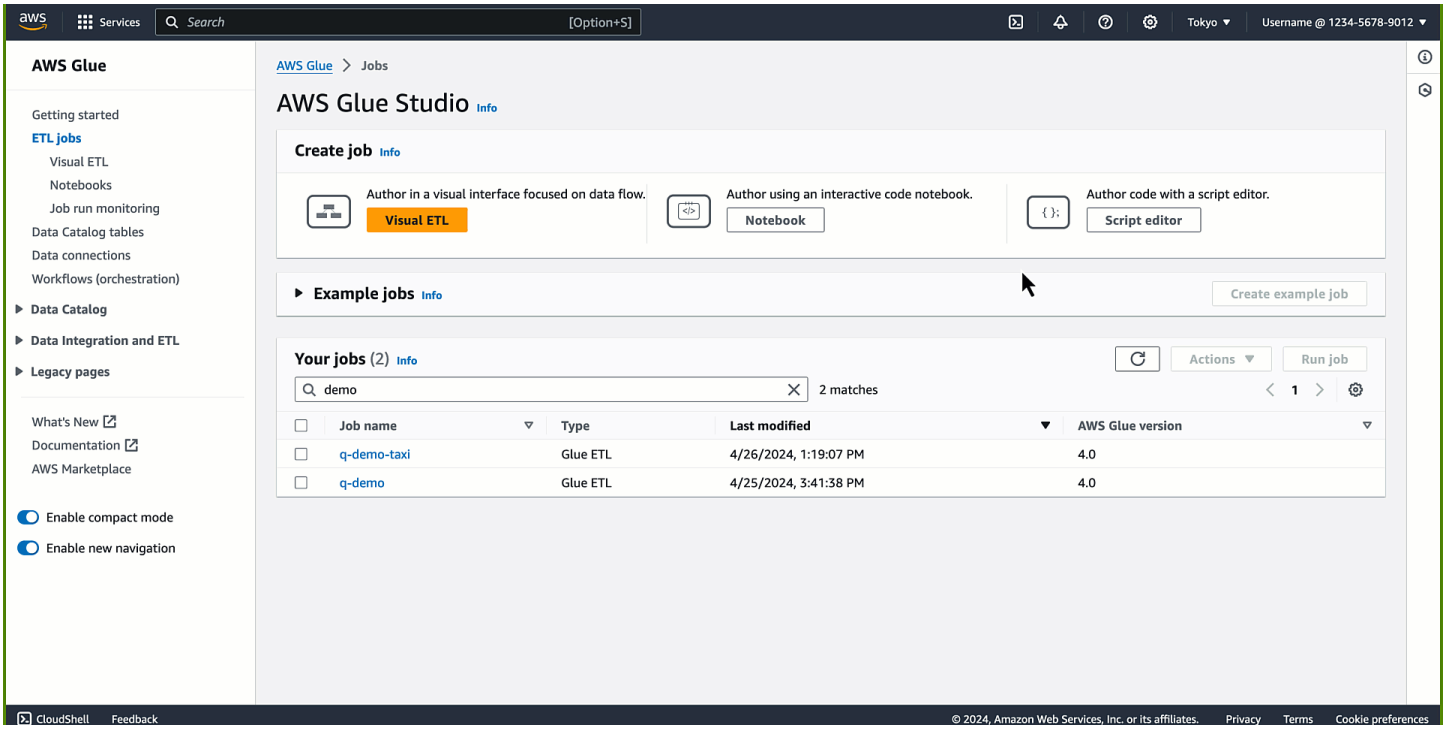
- Risoluzione dei problemi: l'integrazione dei dati di Amazon Q AWS Glue è stata creata appositamente per aiutarti a comprendere gli errori nei AWS Glue lavori e fornisce step-by-step istruzioni per la causa principale e la risoluzione dei problemi.

> **Note**
>
> L'integrazione dei dati di Amazon Q in AWS Glue non utilizza il contesto della conversazione per fornire informazioni sulle risposte future per tutta la durata della conversazione. Ogni conversazione con Amazon Q Data Integration in AWS Glue è indipendente dalle conversazioni precedenti o future.

## Utilizzi l'integrazione dei dati di Amazon Q in AWS Glue?

Nel pannello Amazon Q puoi richiedere ad Amazon Q di generare codice per uno script AWS Glue ETL o rispondere a una domanda sulle AWS Glue funzionalità o risolvere un errore. La risposta è uno script ETL PySpark con step-by-step istruzioni per personalizzare lo script, esaminarlo ed eseguirlo. Per le domande, la risposta viene generata sulla base della knowledge base sull'integrazione dei dati con un riepilogo e un URL di origine per i riferimenti.

Ad esempio, puoi chiedere ad Amazon Q di "Fornisci uno script Glue che legga da Snowflake, rinomini i campi e scriva su Redshift" e, in risposta, l'integrazione dei dati di Amazon Q AWS Glue restituirà uno script di AWS Glue lavoro in grado di eseguire l'azione richiesta. Puoi esaminare il codice generato per assicurarti che soddisfi l'intento richiesto. Se sei soddisfatto, puoi implementarlo come processo di produzione. AWS Glue È possibile risolvere i problemi relativi ai processi chiedendo all'integrazione di spiegare gli errori e di proporre soluzioni. Amazon Q può rispondere a domande sulle nostre AWS Glue best practice di integrazione dei dati.




Di seguito sono riportati alcuni esempi di domande che dimostrano come l'integrazione dei dati di Amazon Q in AWS Glue può aiutarti a sviluppare AWS Glue:

AWS Glue Generazione di codice ETL:

- Scrivi uno AWS Glue script che legga JSON da S3, trasformi i campi utilizzando Apply Mapping e scriva su Amazon Redshift
- Come posso scrivere uno AWS Glue script per leggere da DynamoDB, applicare DropNullFields la trasformazione e scrivere su S3 come Parquet?
- Dammi uno AWS Glue script che legga da MySQL, rilasci alcuni campi in base alla mia logica aziendale e scriva su Snowflake
- Scrivi un AWS Glue lavoro da leggere da DynamoDB e scrivi su S3 come JSON
- Aiutami a sviluppare uno AWS Glue script per AWS Glue Data Catalog to S3
- Scrivi un AWS Glue lavoro per leggere JSON da S3, elimina i valori nulli e scrivi su Redshift

AWS Glue spiegazioni delle funzionalità:

- Come posso usare AWS Glue Data Quality?
- Come usare i segnalibri di AWS Glue lavoro?
- Come posso abilitare la scalabilità AWS Glue automatica?

- Qual è la differenza tra frame AWS Glue dinamici e frame di dati Spark?
- Quali sono i diversi tipi di connessioni supportati AWS Glue?

AWS Glue risoluzione dei problemi:

- Come risolvere gli errori di memoria esaurita (OOM) nei AWS Glue job?
- Quali sono alcuni messaggi di errore che potresti visualizzare durante la configurazione della qualità AWS Glue dei dati e come puoi risolverli?
- Come posso correggere un AWS Glue lavoro con l'errore Accesso negato ad Amazon S3?
- Come posso risolvere i problemi relativi allo shuffle dei dati nei lavori? AWS Glue

# Best practice per interagire con l'integrazione dei dati di Amazon Q

Di seguito sono riportate le best practice per interagire con l'integrazione dei dati di Amazon Q:

- Quando si interagisce con l'integrazione dei dati di Amazon Q, occorre porre domande specifiche, applicare l'iterazione in caso di richieste complesse e verificare l'accuratezza delle risposte.
- Quando si forniscono prompt di integrazione dei dati in linguaggio naturale, bisogna essere il più specifico possibile per aiutare l'assistente a capire esattamente di cosa si ha bisogno. Invece di chiedere «estrai dati da S3", fornisci maggiori dettagli come «scrivi AWS Glue uno script che estragga i file JSON da S3".
- Esaminare lo script generato prima di eseguirlo per verificarne l'accuratezza. Se lo script generato contiene errori o non corrisponde al tuo intento, fornire all'assistente le istruzioni per correggerlo.
- La tecnologia di IA generativa è nuova e nelle risposte possono esserci errori, a volte chiamati allucinazioni. Testa e rivedi tutto il codice per individuare errori e vulnerabilità prima di utilizzarlo nell'ambiente o nel carico di lavoro.

# L'integrazione dei dati di Amazon Q nel miglioramento dei AWS Glue servizi

Per aiutare l'integrazione dei dati di Amazon Q a AWS Glue fornire le informazioni più pertinenti sui AWS servizi, possiamo utilizzare determinati contenuti di Amazon Q, come le domande che poni ad Amazon Q e le relative risposte, per migliorare il servizio.

Per informazioni sui contenuti che utilizziamo e su come disattivarli, consultare Miglioramento del servizio di Amazon Q Developer nella Guida per l'utente di Amazon Q Developer.

# Considerazioni

Considera i seguenti elementi prima di utilizzare l'integrazione dei dati di Amazon Q in AWS Glue:

- Attualmente, la generazione di codice funziona solo con il PySpark kernel. Il codice generato è per i AWS Glue lavori basati su Python Spark.
- Per informazioni sulle combinazioni supportate di capacità di generazione di codice dell'integrazione dei dati di Amazon Q in AWS Glue, consultaCapacità di generazione di codice supportate.

# Configurazione dell'integrazione dei dati di Amazon Q in AWS Glue

Nelle sezioni seguenti vengono fornite informazioni sulla configurazione dell'integrazione dei dati di Amazon Q in AWS Glue.

Argomenti

- Configurazione delle autorizzazioni IAM

## Configurazione delle autorizzazioni IAM

Questo argomento descrive le autorizzazioni IAM che configuri per l'esperienza di chat di Amazon Q e l'esperienza con i notebook AWS Glue Studio.

Argomenti

- Configurazione delle autorizzazioni IAM per la chat di Amazon Q
- Configurazione delle autorizzazioni IAM per i notebook Studio AWS Glue

### Configurazione delle autorizzazioni IAM per la chat di Amazon Q

La concessione delle autorizzazioni per l'integrazione dei dati APIs utilizzati da Amazon Q AWS Glue richiede le autorizzazioni IAM ( AWS Identity and Access Management) appropriate. Puoi ottenere le autorizzazioni allegando la seguente AWS politica personalizzata alla tua identità IAM (ad esempio un utente, un ruolo o un gruppo):

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:StartCompletion",
        "glue:GetCompletion"
      ],
      "Resource": [
        "arn:aws:glue:*:*:completion/*"
      ]
    }
  ]
}
```

## Configurazione delle autorizzazioni IAM per i notebook Studio AWS Glue

Per abilitare l'integrazione dei dati di Amazon Q nei notebook AWS Glue Studio, assicurati che al ruolo IAM del notebook sia associata la seguente autorizzazione:

> **Note**
>
> Il prefisso `codewhisperer` è un nome legacy di un servizio che si è unito ad Amazon Q Developer. Per ulteriori informazioni, consultare Ridenonimazione di Amazon Q Developer – Riepilogo delle modifiche.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:StartCompletion",
```

```
        "glue:GetCompletion"
      ],
      "Resource": [
        "arn:aws:glue:*:*:completion/*"
      ]
    },
    {
      "Sid": "AmazonQDeveloperPermissions",
      "Effect": "Allow",
      "Action": [
        "codewhisperer:GenerateRecommendations"
      ],
      "Resource": "*"
    }
  ]
}
```

> **Note**
>
> L'integrazione dei dati in Amazon Q AWS Glue non è APIs disponibile tramite l' AWS SDK utilizzabile a livello di codice. I due seguenti APIs vengono utilizzati nella policy IAM per abilitare questa esperienza tramite il pannello di chat di Amazon Q o i notebook AWS Glue Studio: e. `StartCompletion` `GetCompletion`

Assegnare le autorizzazioni

Per fornire l'accesso, aggiungi autorizzazioni agli utenti, gruppi o ruoli:

- Utenti e gruppi in AWS IAM Identity Center: crea un set di autorizzazioni. Seguire le istruzioni riportate nella pagina Creazione di un set di autorizzazioni nella Guida per l'utente di Centro identitàAWS IAM.
- Utenti gestiti in IAM tramite un provider di identità: creare un ruolo per la federazione delle identità. Segui le istruzioni riportate nella pagina Creazione di un ruolo per un provider di identità di terze parti (federazione) nella Guida per l'utente di IAM.
- Utenti IAM:
  - Crea un ruolo che l'utente possa assumere. Per istruzioni, consulta la pagina Creating a role for an IAM user (Creazione di un ruolo per un utente IAM) nella Guida per l'utente di IAM.

- (Non consigliato) Collega una policy direttamente a un utente o aggiungi un utente a un gruppo di utenti. Segui le istruzioni riportate nella pagina Aggiunta di autorizzazioni a un utente (console) nella Guida per l'utente IAM.

## Capacità di generazione di codice supportate

Di seguito sono elencate le combinazioni delle capacità di generazione di codice dell'integrazione dei dati di Amazon Q.

| Origini e destinazioni | Trasformazione |
|---|---|
| S3 con i seguenti tipi di formato: json, csv, parquet, hudi, delta | Drop (E-mail eliminata) |
| AWS Glue Data Catalog | Aggregazione |
| Redlake | DropDuplicates |
| Amazon DynamoDB | Join |
| MySQL | Filtro |
| Oracle | RenameColumns |
| PostgresSQL | FillNull |
| Microsoft SQL Server | DropNull |
| Amazon DocumentDB/MongoDB | WithColumns |
| Snowflake | Query SQL |
| Google BigQuery | Union |
| Teradata | Select |
| OpenSearch Servizio Amazon | |
| Vertica | |

| Origini e destinazioni | Trasformazione |
| --- | --- |
| SAP HANA | |
| Amazon Redshift | |

# Interazioni di esempio

L'integrazione dei dati di Amazon Q ti AWS Glue consente di inserire la tua domanda nel pannello Amazon Q. È possibile inserire una domanda relativa alla funzionalità di integrazione dei dati fornita da AWS Glue. Verrà restituita una risposta dettagliata, insieme ai documenti di riferimento.

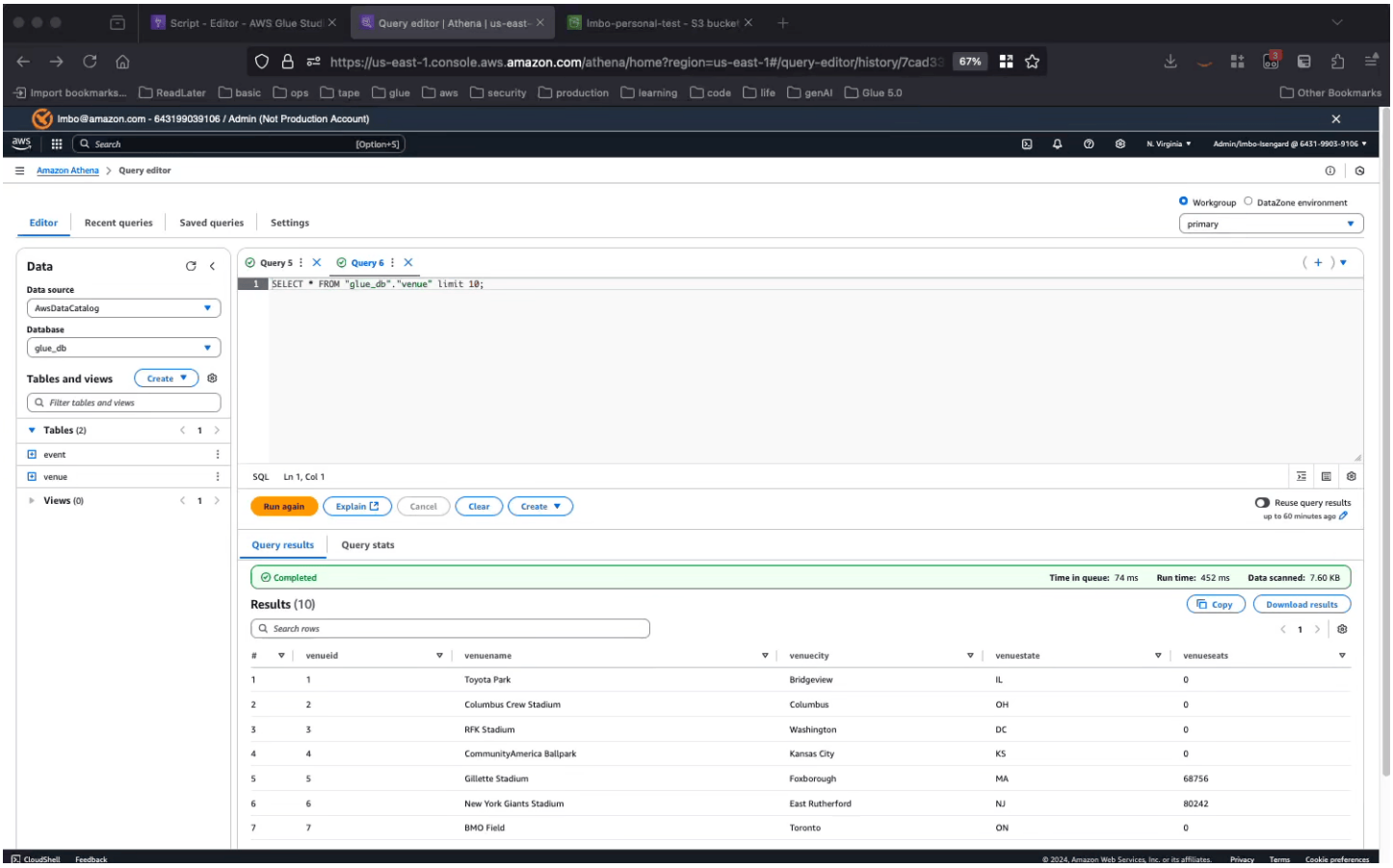
Un altro caso d'uso è la generazione di script di lavoro AWS Glue ETL. È possibile porre una domanda su come eseguire un processo di estrazione, trasformazione e caricamento dei dati. Verrà restituito PySpark uno script generato.

Argomenti

- Interazioni con la chat di Amazon Q
- AWS Glue Interazioni con i notebook da Studio

## Interazioni con la chat di Amazon Q

Sulla AWS Glue console, inizia a creare un nuovo lavoro e chiedi ad Amazon Q: «Crea un flusso ETL Glue connettiti a due tabelle del catalogo Glue, sede ed evento nel mio database glue_db, unisci i risultati sul venueid della sede e e_venueid dell'evento, quindi filtra in base allo stato della sede con condizione come venueState=='dc' e scrivi a s3://amzn-s3-demo -/in formato CSV."» bucket/codegen/BDB-9999/output



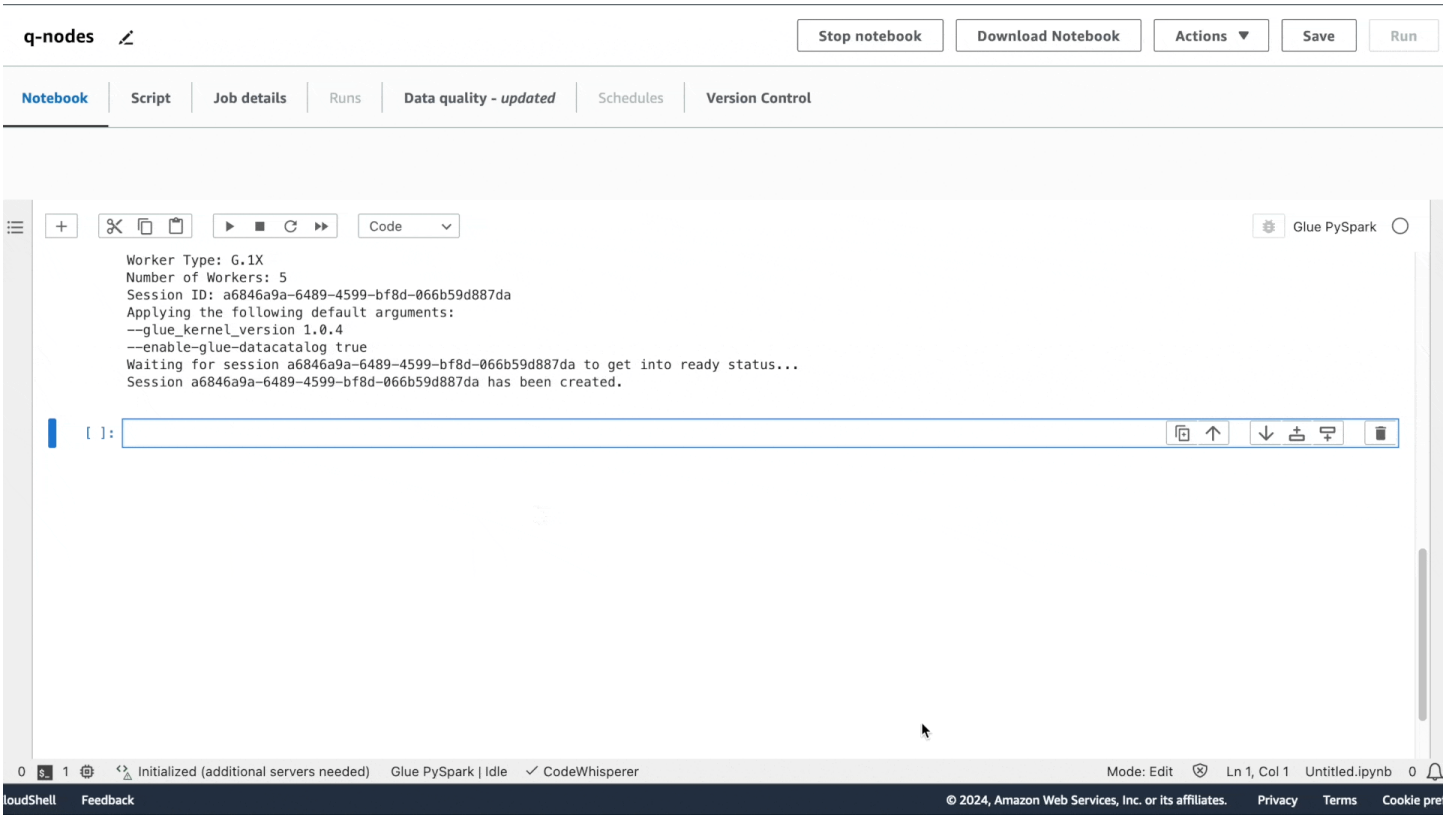

Viene generato un codice. Con questa risposta, puoi imparare e capire come creare AWS Glue codice per i tuoi scopi. Puoi inserire copy/paste il codice generato nell'editor di script e configurare i segnaposto. Dopo aver configurato un ruolo IAM e AWS Glue le connessioni sul job, salva ed esegui il job. Una volta completato il processo, si può verificare che i dati di riepilogo vengano mantenuti in Amazon S3 come previsto e possano essere utilizzati dai carichi di lavoro a valle.

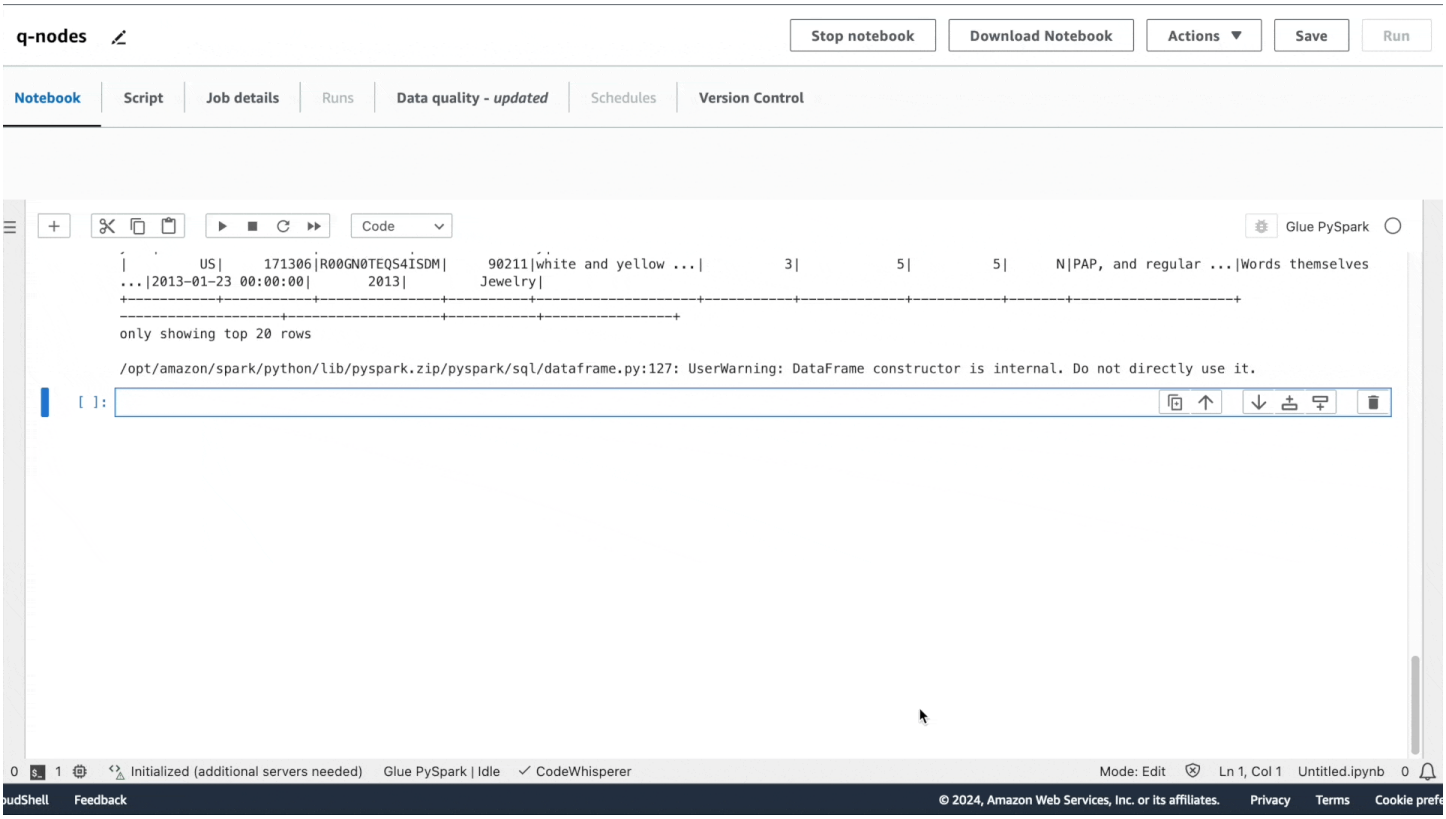
## AWS Glue Interazioni con i notebook da Studio

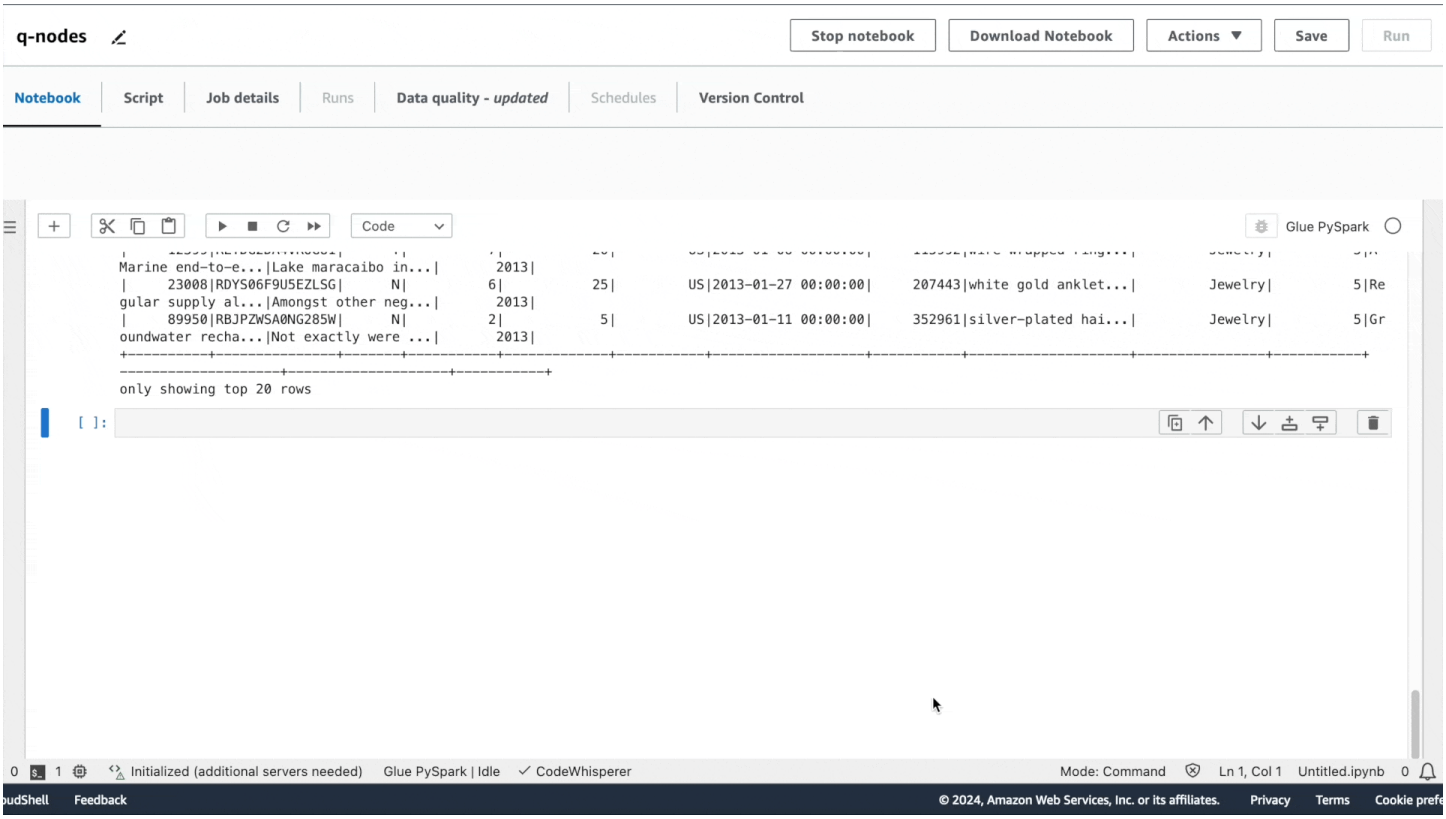
> **Note**
>
> L'esperienza di integrazione di Amazon Q Data nei AWS Glue Studio notebook si concentra ancora sul flusso di integrazione dei dati DynamicFrame basato.

Aggiungere una nuova cella e inserire il commento per descrivere ciò che si desidera ottenere. Dopo aver premuto Tab e Invio, viene visualizzato il codice consigliato.

Il primo tentativo è estrarre i dati: "Dammi un codice che legga una tabella del catalogo dati di Glue", seguito da "Dammi un codice per applicare una trasformazione del filtro con star_rating>3" e "Dammi un codice che scriva il frame su S3 in formato Parquet".




q-nodes
Stop notebook
Download Notebook
Actions
Save
Run
Notebook
Script
Job details
Runs
Data quality - updated
Schedules
Version Control
Code
Glue PySpark
only showing top 20 rows
/opt/amazon/spark/python/lib/pyspark.zip/pyspark/sql/dataframe.py:127: UserWarning: DataFrame constructor is internal. Do not directly use it.
[ ]:
Initialized (additional servers needed)
Glue PySpark | Idle
CodeWhisperer
Mode: Edit
Ln 1, Col 1
Untitled.ipynb
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms


q-nodes
Stop notebook
Download Notebook
Actions
Save
Run
Notebook
Script
Job details
Runs
Data quality - updated
Schedules
Version Control
Code
Glue PySpark
only showing top 20 rows
[ ]:
Initialized (additional servers needed)
Glue PySpark | Idle
CodeWhisperer
Mode: Command
Ln 1, Col 1
Untitled.ipynb
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms

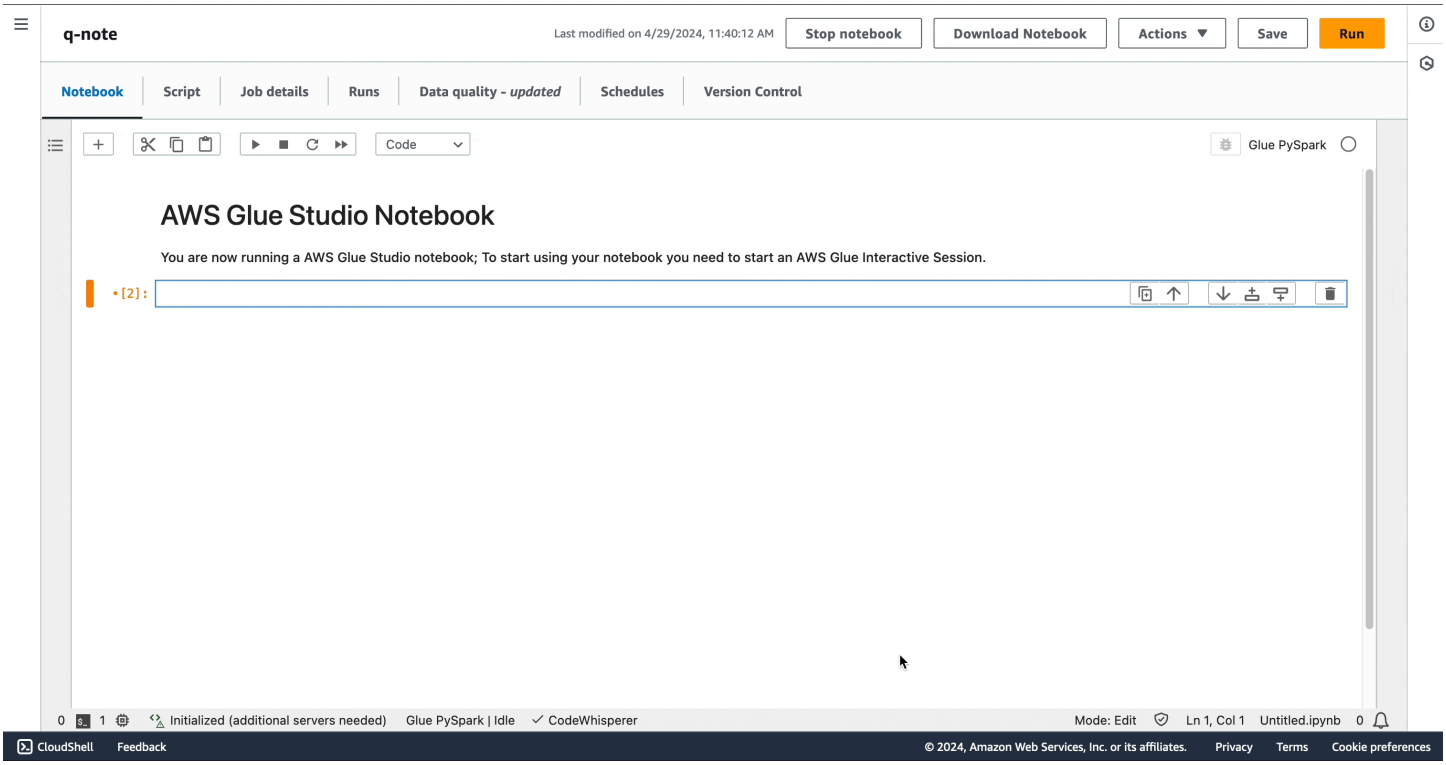
Analogamente all'esperienza di chat di Amazon Q, il codice è consigliato. Se si preme Tab, viene scelto il codice consigliato.

È possibile eseguire ogni cella compilando le opzioni appropriate per le origini nel codice generato. In qualsiasi momento nelle esecuzioni, si può anche visualizzare in anteprima un campione del set di dati utilizzando il metodo `show()`.

È possibile eseguire il notebook come processo, a livello di programmazione o scegliendo Esegui.

## Prompt complessi

È possibile generare uno script completo con un solo prompt complesso. “Dispongo di dati JSON in S3 e dati in Oracle che devono essere combinati. Fornisci uno script Glue che legga da entrambe le origini, esegua un'unione e poi scriva i risultati su Redshift.”




Potresti notare che, sul notebook, l'integrazione dei dati di Amazon Q AWS Glue ha generato lo stesso frammento di codice generato nella chat di Amazon Q.

È possibile eseguire il notebook come processo, o scegliendo Esegui o a livello di programmazione.

# Utilizzo della consapevolezza del contesto con Amazon Q Data Integration in AWS Glue

Ora puoi creare processi di elaborazione dati in modo più efficiente con la generazione di PySpark DataFrame codice e con riconoscimento del contesto basati su query in Amazon Q Data Integration. Ad esempio, puoi utilizzare questo prompt per generare PySpark codice: «crea un lavoro per caricare i dati di vendita dalla tabella Redshift 'analytics.salesorder' utilizzando la connessione 'erp_conn', filtra order_amount al di sotto di 50 dollari e salva su Amazon S3 in formato parquet».

Amazon Q genererà lo script in base alla configurazione del flusso di lavoro di integrazione dei dati di prompt e setup con i dettagli forniti dalla domanda, come configurazioni di connessione, dettagli dello schema, database/table nomi e specifiche delle colonne per le trasformazioni. Le informazioni sensibili, come le password delle opzioni di connessione, continuano a essere oscurate.

Se le informazioni richieste non vengono fornite nel prompt, Amazon Q inserirà dei segnaposto, di cui dovrà essere aggiornato il codice generato con i valori appropriati prima di eseguire il codice.

Di seguito sono riportati alcuni esempi su come utilizzare la consapevolezza del contesto.

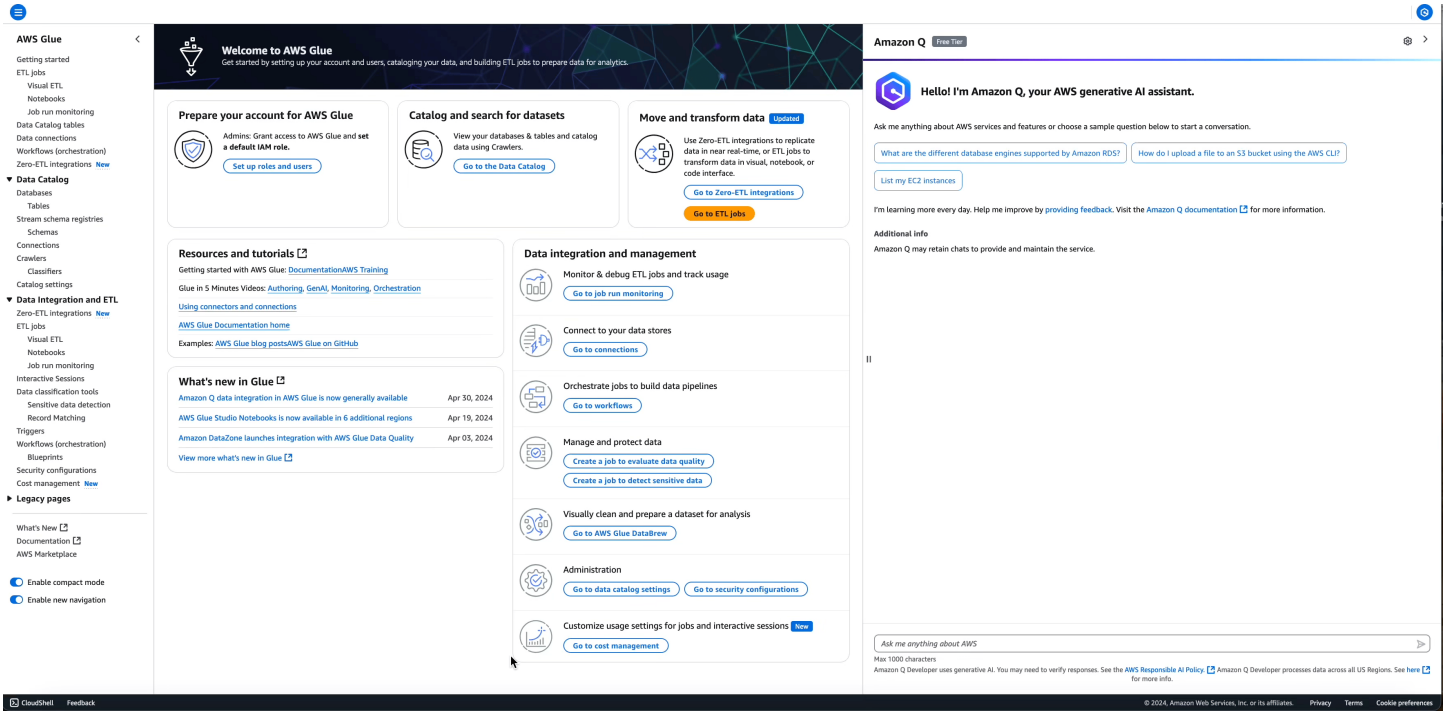
## Interazioni di esempio

Leggere un file Amazon S3 e scrivere in un bucket Amazon S3

Prompt: `Create an AWS Glue spark job that reads a file s3://amzn-s3-demo-bucket-input-bucket-name/input-filename.csv and saves it into the parquet file under directory s3://amzn-s3-demo-bucket-output-bucket-name/output-s3-prefix/`

```
import sys
from awsglue.transforms import *
from pyspark.context import SparkContext
from pyspark.sql import SparkSession

sc = SparkContext.getOrCreate()
spark = SparkSession.builder.getOrCreate()

# Script generated for node S3DataSource
S3DataSource_dsource1 = spark.read.format("csv").load(
    "s3://amzn-s3-demo-bucket-input-bucket-name/input-filename.csv"
)
```

```
# Script generated for node S3DataSink
S3DataSource_dsource1.write.format("parquet").mode("append").save(
    "s3://amzn-s3-demo-bucket-output-bucket-name/output-s3-prefix/"
)
```




Ottenere dati da Lakehouse e scriverli nel database

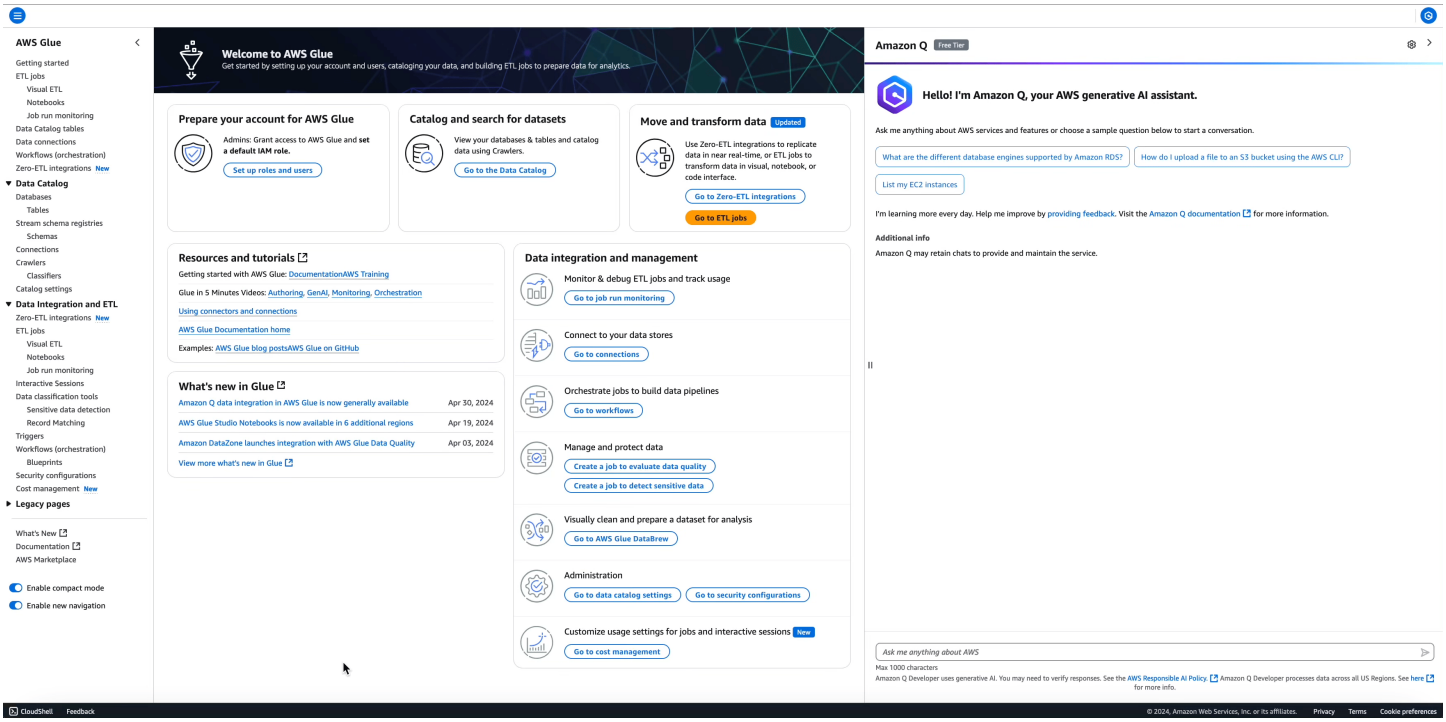
Prompt: `write an ETL script to read from a Lakehouse table my-table in database my-database and write it to a RDS MySQL table my-target-table`

Per i campi in cui non sono state fornite informazioni (ad esempio, il connectionName obbligatorio è per il data sink MySQL e l'impostazione predefinita con un segnaposto <connection-name> nel codice generato), viene mantenuto un segnaposto per inserire le informazioni richieste prima di eseguire lo script.

Script generato:

```
import sys
from awsglue.transforms import *
from pyspark.context import SparkContext
from pyspark.sql import SparkSession
from connectivity.adapter import CatalogConnectionHelper
```

```
sc = SparkContext.getOrCreate()
spark = SparkSession.builder.getOrCreate()

# Script generated for node S3DataSource
S3DataSource_dsource1 = spark.read.format("parquet").load(
    "s3://amzn-lakehouse-demo-bucket/my-database/my-table"
)

# Script generated for node ConnectionV2DataSink
ConnectionV2DataSink_dsink1_additional_options = {"dbtable": "my-target-table"}
CatalogConnectionHelper(spark).write(
    S3DataSource_dsource1,
    "mysql",
    "<connection-name>",
    ConnectionV2DataSink_dsink1_additional_options,
)
```




## Esempio: flusso di lavoro ETL completo

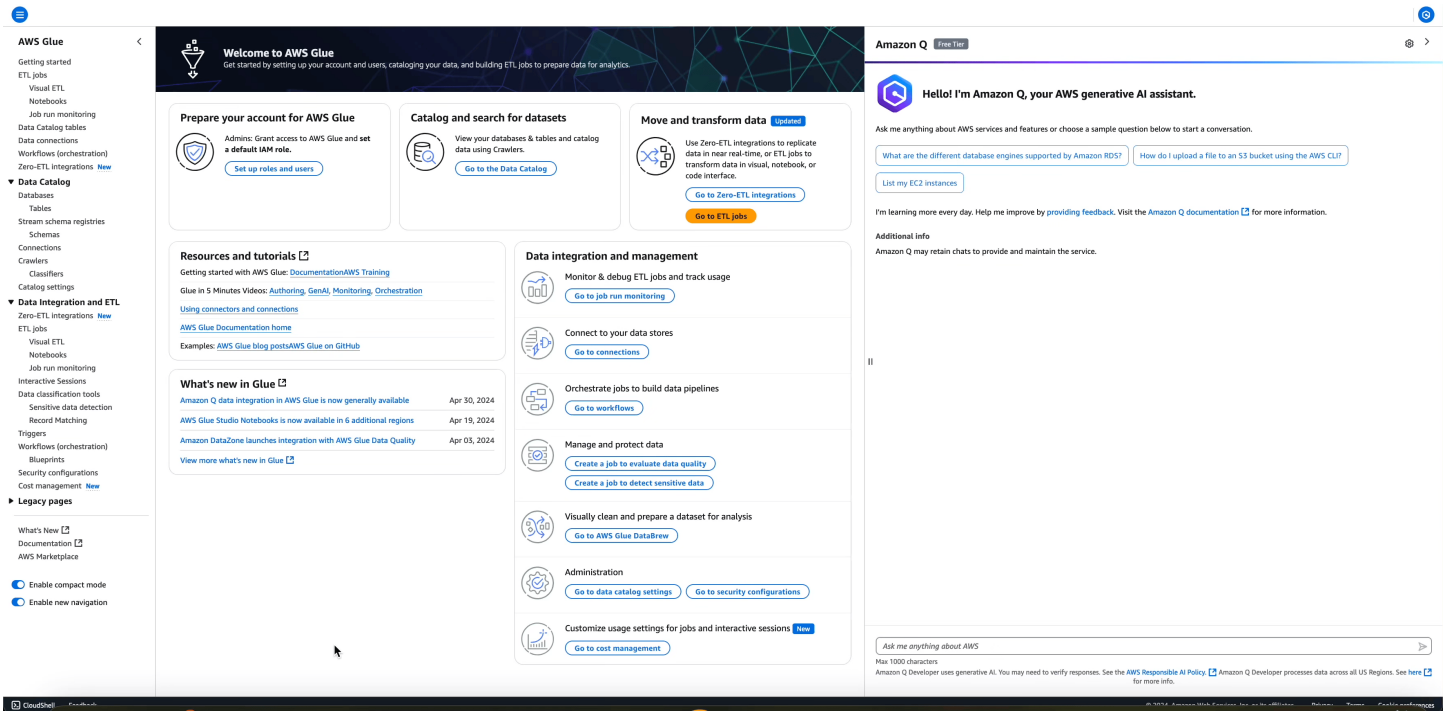
L'esempio seguente dimostra come è possibile chiedere a AWS Glue di creare uno script AWS Glue per completare un flusso di lavoro ETL completo con il seguente prompt:. `Create a AWS Glue ETL Script read from two AWS Glue Data Catalog tables venue and event in my database glue_db_4fthqih3vvk1if, join the results on the field venueid, filter on venue state with condition as venuestate=='DC' after joining`

`the results and write output to an Amazon S3 S3 location s3://amz-s3-demo-bucket/output/ in CSV format`

Il flusso di lavoro contiene la lettura da diverse fonti di dati (due tabelle AWS Glue Data Catalog) e un paio di trasformazioni dopo la lettura unendo il risultato di due letture, filtrando in base a determinate condizioni e scrivendo l'output trasformato in una destinazione Amazon S3 in formato CSV.

Il processo generato compilerà le informazioni dettagliate nell'origine dati, nella trasformazione e nell'operazione di sink con le informazioni corrispondenti estratte dalla domanda dell'utente, come di seguito.

```
import sys
from awsglue.transforms import *
from pyspark.context import SparkContext
from pyspark.sql import SparkSession

sc = SparkContext.getOrCreate()
spark = SparkSession.builder.getOrCreate()

# Script generated for node CatalogDataSource
CatalogDataSource_dsource1 = spark.sql("select * from
 `glue_db_4fthqih3vvk1if`.`venue`")

# Script generated for node CatalogDataSource
CatalogDataSource_dsource2 = spark.sql("select * from
 `glue_db_4fthqih3vvk1if`.`event`")

# Script generated for node JoinTransform
JoinTransform_transform1 = CatalogDataSource_dsource1.join(
    CatalogDataSource_dsource2,
    (CatalogDataSource_dsource1["venueid"] == CatalogDataSource_dsource2["venueid"]),
    "inner",
)

# Script generated for node FilterTransform
FilterTransform_transform2 = JoinTransform_transform1.filter("venuestate=='DC'")

# Script generated for node S3DataSink
FilterTransform_transform2.write.format("csv").mode("append").save(
    "s3://amz-s3-demo-bucket/output//output/"
)
```




# Limitazioni

- Restituzione del contesto:
  - La funzionalità di consapevolezza del contesto restituisce solo il contesto della precedente query dell'utente all'interno della stessa conversazione. Non conserva il contesto oltre la query immediatamente precedente.
- Supporto per le configurazioni dei nodi:
  - Attualmente, la consapevolezza del contesto supporta solo un sottoinsieme di configurazioni richieste per vari nodi.
  - Il supporto per i campi opzionali è previsto nelle prossime versioni.
- Disponibilità:
  - La consapevolezza del contesto e DataFrame il supporto sono disponibili nei notebook Q Chat e SageMaker Unified Studio. Tuttavia, queste funzionalità non sono ancora disponibili nei notebook AWS Glue Studio.

# Orchestrazione in AWS Glue

Nelle sezioni seguenti vengono fornite informazioni sull'orchestrazione dei processi in AWS Glue.

Argomenti


- Avvio di lavori e crawler utilizzando i trigger
- Esecuzione di attività ETL complesse utilizzando gli schemi e i flussi di lavoro in AWS Glue
- Sviluppo di schemi in AWS Glue


## Avvio di lavori e crawler utilizzando i trigger

In AWS Glue, è possibile creare oggetti del catalogo dati chiamati trigger, che è possibile utilizzare per avviare manualmente o automaticamente uno o più crawler o estrarre, trasformare e caricare lavori (ETL). Utilizzando i trigger, è possibile progettare una catena di crawler e lavori dipendenti.

> **Note**
>
> È possibile eseguire la stessa procedura definendo i flussi di lavoro. I flussi di lavoro sono i preferiti nella creazione di complesse operazioni ETL multi-processo. Per ulteriori informazioni, consulta the section called "Esecuzione di attività ETL complesse utilizzando gli schemi e i flussi di lavoro".

Argomenti


- Trigger di AWS Glue
- Aggiunta di trigger
- Attivazione e disattivazione dei trigger


### Trigger di AWS Glue

Quando viene attivato, un trigger può avviare processi e crawler specificati. Un trigger viene attivato on demand, in base a una pianificazione o in base a una combinazione di eventi.

> **Note**
>
> Solo due crawler possono essere attivati da un singolo trigger. Se vuoi eseguire il crawling di più datastore, utilizza più fonti per ogni crawler anziché eseguire più crawler contemporaneamente.

Esistono diversi stati di trigger. Un trigger è `CREATED`, `ACTIVATED` o `DEACTIVATED`. Esistono anche stati transitori, come ad esempio `ACTIVATING`. Per interrompere temporaneamente l'attivazione di un trigger, è possibile disattivarlo. È quindi possibile riattivarlo in un secondo momento.

Esistono tre tipi di trigger:

Pianificato

Un trigger basato sul tempo in `cron`.

È possibile creare un trigger per un set di lavori o crawler in base a una pianificazione. È possibile specificare i vincoli, ad esempio la frequenza in cui vengono eseguiti i lavori o i crawler, i giorni della settimana in cui vengono eseguiti e a che ora. Questi vincoli si basano sul comando `cron`. Quando si configura una pianificazione per un trigger, tenere conto delle caratteristiche e delle limitazioni del cron. Ad esempio, se vuoi eseguire il crawler il giorno 31 di ogni mese, devi ricordare che alcuni mesi non sono di 31 giorni. Per ulteriori informazioni sul cron, consulta Pianificazioni basate sul tempo per processi e crawler.

Condizionale

Un trigger che viene attivato quando un lavoro o crawler o più lavori o crawler precedenti soddisfano un elenco di condizioni.

Quando si crea un trigger condizionale, si specificano un elenco di lavori e un elenco di crawler da controllare. Per ogni lavoro o crawler controllato, è necessario specificare uno stato da controllare, ad esempio riuscito, non riuscito, timeout e così via. Il trigger viene attivato se i lavori o i crawler controllati terminano con gli stati specificati. È possibile configurare il trigger per l'attivazione quando si verificano uno o tutti gli eventi osservati.

Ad esempio, è possibile configurare un trigger T1 per avviare il lavoro J3 quando entrambi i lavori J1 e J2 vengono completati correttamente e un altro trigger T2 per avviare il lavoro J4 se il lavoro J1 o J2 non riesce.

Nella tabella seguente sono elencati gli stati di completamento del lavoro e del crawler (eventi) controllati dai trigger.

| Stati di completamento del lavoro | Stati di completamento del crawler |
| --- | --- |
| • `SUCCEEDED`<br>• `STOPPED`<br>• `FAILED`<br>• `TIMEOUT` | • `SUCCEEDED`<br>• `FAILED`<br>• `CANCELLED` |

On demand

Un trigger che si attiva quando viene acceso. I trigger su richiesta non entrano mai nello stato `ACTIVATED` o `DEACTIVATED`. Rimangono sempre nello stato `CREATED`.

Affinché si attivino non appena creati, è possibile impostare un flag per attivare i trigger pianificati e condizionali al momento della creazione.

> **Important**
>
> I lavori o i crawler eseguiti dopo il completamento di altri processi o crawler vengono definiti dipendenti. I lavori o i crawler dipendenti vengono avviati solo se il lavoro o il crawler completato è stato avviato da un trigger. Tutti i lavori o i crawler in una catena di dipendenze devono discendere da una singola pianificazione o da un singolo trigger on-demand.

Passare i parametri del lavoro con i trigger

Un trigger può passare parametri ai lavori che avvia. I parametri includono argomenti del lavoro, valore di timeout, configurazione di sicurezza e altro ancora. Se il trigger avvia più lavori, i parametri vengono passati a ciascun lavoro.

Di seguito sono riportate le regole per argomenti di lavoro passati da un trigger:

- Se la chiave nella coppia chiave-valore corrisponde a un argomento di lavoro predefinito, l'argomento passato sostituisce l'argomento predefinito. Se la chiave non corrisponde a un argomento predefinito, l'argomento viene passato come argomento aggiuntivo al lavoro.

- Se la chiave nella coppia chiave-valore corrisponde a un argomento non sovrascrivibile, l'argomento passato viene ignorato.

Per ulteriori informazioni, consulta the section called "Triggers" nell'API AWS Glue.

## Aggiunta di trigger

Puoi aggiungere un trigger utilizzando la AWS Glue console, il AWS Command Line Interface (AWS CLI) o l'AWS GlueAPI.

Per aggiungere un trigger (console)

1. Accedi a Console di gestione AWS e apri la AWS Glue console all'indirizzo https://console.aws.amazon.com/glue/.
2. Nel riquadro di navigazione, in ETL, scegliere Triggers (Trigger). Selezionare Add trigger (Aggiungi trigger).
3. Specificare le proprietà seguenti:

   Name

   Assegna al trigger un nome univoco.

   Tipo di trigger

   Specifica una delle seguenti proprietà:

   - Schedule (Pianifica): il trigger si attiva a una frequenza e un tempo specifici.
   - Job events (Eventi di lavoro): un trigger condizionale. Il trigger viene attivato quando uno o tutti i lavori nell'elenco corrispondono agli stati designati. Per consentire l'attivazione del trigger, il lavoro in questione deve essere stato avviato da un trigger. Per qualsiasi lavoro selezionato, è possibile osservare un solo evento (stato di completamento).
   - On-demand (On demand): il trigger funziona se attivato.
4. Completare la procedura guidata del trigger. Nella pagina Review (Revisione) è possibile attivare immediatamente i trigger Schedule (Pianifica) e Job events (Eventi di lavoro) (condizionali), selezionando Enable trigger on creation (Attiva trigger alla creazione).

Per aggiungere un trigger (AWS CLI)

- Utilizzare un comando simile al seguente:

```
aws glue create-trigger --name MyTrigger --type SCHEDULED --schedule  "cron(0 12 *
 * ? *)" --actions CrawlerName=MyCrawler --start-on-creation
```

Questo comando crea un trigger di pianificazione denominato `MyTrigger`, che viene eseguito ogni giorno alle 12:00 UTC e avvia un crawler denominato `MyCrawler`. Il trigger viene creato nello stato attivato.

Per ulteriori informazioni, consulta the section called "Trigger di AWS Glue".

## Pianificazioni basate sul tempo per processi e crawler

Puoi definire una pianificazione basata sul tempo per i crawler e i processi in AWS Glue. La definizione di queste pianificazioni usa la sintassi cron di tipo Unix. Specifichi il tempo in Coordinated Universal Time (UTC) e la precisione minima per una pianificazione è 5 minuti.

Per ulteriori informazioni sulla configurazione di processi e crawler da eseguire utilizzando una pianificazione, consulta Avvio di lavori e crawler utilizzando i trigger.

Espressioni Cron

Le espressioni Cron hanno sei campi obbligatori separati da uno spazio vuoto.

Sintassi

```
cron(Minutes Hours Day-of-month Month Day-of-week Year)
```

| Campi | Valori | Caratteri jolly |
|---|---|---|
| Minuti | 0-59 | , - * / |
| Ore | 0-23 | , - * / |
| D ay-of-month | 1-31 | , - * ? / L W |
| Mese | 1-12 o JAN-DEC | , - * / |
| D ay-of-week | 1-7 o SUN-SAT | , - * ? / L |
| Anno | 1970–2199 | , - * / |

### Caratteri jolly

- Il carattere jolly , (virgola) include valori aggiuntivi. Nel campo `Month`, `JAN,FEB,MAR` includono gennaio, febbraio e marzo.
- Il carattere jolly - (trattino) specifica gli intervalli. Nel campo `Day`, 1-15 include i giorni dall'1 al 15 del mese specificato.
- Il carattere jolly * (asterisco) include tutti i valori nel campo. Nel campo `Hours`, * include ogni ora.
- Il carattere jolly / (barra) specifica gli incrementi. Nel campo `Minutes` puoi immettere **`1/10`** per specificare ogni decimo minuto, a partire dal primo minuto dell'ora (ad esempio, l'11°, il 21° e il 31° minuto).
- Il carattere jolly ? (punto interrogativo) specifica un valore. Nel **`Day-of-month`** campo puoi inserire 7, e se non ti interessa in che giorno della settimana è il settimo, puoi inserire? sul Day-of-week campo.
- Il carattere jolly L nel campo `Day-of-month` o `Day-of-week` specifica l'ultimo giorno del mese o della settimana.
- Il carattere jolly W nel campo `Day-of-month` specifica un giorno feriale. Nel campo `Day-of-month`, `3W` specifica il giorno più vicino al terzo giorno feriale del mese.

### Limits

- Non puoi specificare i campi `Day-of-month` e `Day-of-week` nella stessa espressione cron. Se specifichi un valore in uno dei campi, devi usare un carattere ? nell'altro campo.
- Le espressioni cron che indicano frequenze più rapide di 5 minuti non sono supportate.

### Esempi

Quando crei una pianificazione puoi utilizzare le seguenti stringhe cron di esempio.

| Minuti | Ore | Giorno del mese | Mese | Giorno della settimana | Anno | Significato |
|---|---|---|---|---|---|---|
| 0 | 10 | * | * | ? | * | Esegui ogni giorno |

| Minuti | Ore | Giorno del mese | Mese | Giorno della settimana | Anno | Significato |
|---|---|---|---|---|---|---|
| | | | | | | alle 10:00 (UTC) |
| 15 | 12 | * | * | ? | * | Esegui ogni giorno alle 12:15 (UTC) |
| 0 | 18 | ? | * | LUN-VEN | * | Esegui dal lunedì al venerdì alle 18:00 (UTC) |
| 0 | 8 | 1 | * | ? | * | Esegui ogni primo giorno del mese alle 8.00 (UTC) |
| 0/15 | * | * | * | ? | * | Esegui ogni 15 minuti |
| 0/10 | * | ? | * | LUN-VEN | * | Esegui dal lunedì al venerdì ogni 10 minuti |

| Minuti | Ore | Giorno del mese | Mese | Giorno della settimana | Anno | Significato |
|---|---|---|---|---|---|---|
| 0/5 | 8-17 | ? | * | LUN-VEN | * | Esegui dal lunedì al venerdì dalle 8:00 alle 17:55 (UTC) ogni 5 minuti |

Ad esempio, per eseguire una pianificazione ogni giorno alle 12:15 UTC, specifica:

```
cron(15 12 * * ? *)
```

## Attivazione e disattivazione dei trigger

Puoi attivare o disattivare un trigger utilizzando la AWS Glue console, il AWS Command Line Interface (AWS CLI) o l'AWS GlueAPI.

Per attivare o disattivare un trigger (console)

1. Accedi Console di gestione AWS e apri la AWS Glue console all'indirizzo https://console.aws.amazon.com/glue/.
2. Nel riquadro di navigazione, in ETL, scegliere Triggers (Trigger).
3. Selezionare la casella di controllo accanto al trigger desiderato e nel menu Action (Operazione) scegliere Enable trigger (Abilita trigger) per attivare il trigger o Disable trigger (Disattiva trigger) per disattivare il trigger.

Per attivare o disattivare un trigger (AWS CLI)

- Inserisci uno dei comandi seguenti.

```
aws glue start-trigger --name MyTrigger
```

```
aws glue stop-trigger --name MyTrigger
```

Un trigger viene attivato con l'avvio e viene disattivato con l'arresto. Quando un trigger si attiva on demand, il funzionamento è immediato.

Per ulteriori informazioni, consulta the section called "Trigger di AWS Glue".

# Esecuzione di attività ETL complesse utilizzando gli schemi e i flussi di lavoro in AWS Glue

Alcuni dei processi di estrazione, trasformazione e caricamento (ETL) della tua organizzazione potrebbero essere implementati utilizzando più processi e crawler dipendenti AWS Glue. Con i flussi di lavoro di AWS Glue, puoi progettare una complessa operazione ETL multi-processo e multi-crawler che AWS Glue può eseguire e tracciare come singola entità. Dopo aver creato un flusso di lavoro e averne specificato i processi, i crawler e i trigger, puoi eseguire il flusso di lavoro on demand o in base a una pianificazione.

Argomenti

- Panoramica di flussi di lavoro in AWS Glue
- Creazione e costruzione manuale di un flusso di lavoro in AWS Glue
- Avvio di un AWS Glue flusso di lavoro con un EventBridge evento Amazon
- Visualizzazione degli EventBridge eventi che hanno avviato un flusso di lavoro
- Esecuzione e monitoraggio di un flusso di lavoro in AWS Glue
- Arresto dell'esecuzione di un flusso di lavoro
- Ripresa e ripristino dell'esecuzione di un flusso di lavoro
- Recupero e impostazione delle proprietà di esecuzione del flusso di lavoro in AWS Glue
- Interrogazione dei flussi di lavoro utilizzando AWS Glue API
- Restrizioni dei flussi di lavoro e degli schemi in AWS Glue
- Risoluzione degli errori relativi agli schemi in AWS Glue
- Autorizzazioni per utenti e ruoli per gli schemi AWS Glue

# Panoramica di flussi di lavoro in AWS Glue

In AWS Glue, è possibile utilizzare i flussi di lavoro per creare e visualizzare complesse attività di estrazione, trasformazione e caricamento (ETL) che coinvolgono più crawler, processi e trigger. Ogni flusso di lavoro gestisce l'esecuzione e il monitoraggio di tutti i suoi processi e crawler. Poiché un flusso di lavoro esegue ogni componente, registra l'avanzamento e lo stato di esecuzione. In questo modo viene fornita una panoramica dell'attività complessiva e i dettagli di ciascuna fase. La console di AWS Glue offre una rappresentazione visiva di un flusso di lavoro sotto forma di diagramma.

È possibile creare un flusso di lavoro da un AWS Glue blueprint oppure è possibile creare manualmente un flusso di lavoro un componente alla volta utilizzando Console di gestione AWS o il AWS Glue API. Per ulteriori informazioni sui piani, consulta the section called "Panoramica degli schemi".

I trigger all'interno dei flussi di lavoro possono attivare sia processi che crawler e possono attivarsi quando i processi o i crawler vengono completati. Utilizzando i trigger è possibile creare grandi catene di processi e crawler interdipendenti. Oltre ai trigger all'interno di un flusso di lavoro che definiscono le dipendenze dei processi e dei crawler, ogni flusso di lavoro dispone di un trigger di avvio. Esistono tre tipi di trigger di avvio:

- Pianificazione: il flusso di lavoro viene avviato secondo una pianificazione definita. La pianificazione può essere giornaliera, settimanale, mensile e così via oppure può essere una personalizzata in base a un'espressione `cron`.
- Su richiesta: il flusso di lavoro viene avviato manualmente dalla AWS Glue console, dall'API o AWS CLI.
- EventBridge evento: il flusso di lavoro viene avviato al verificarsi di un singolo EventBridge evento Amazon o di un batch di EventBridge eventi Amazon. Con questo tipo di trigger, AWS Glue può essere un consumer di eventi in un'architettura basata su eventi. Qualsiasi tipo di EventBridge evento può avviare un flusso di lavoro. Un caso d'uso ìcomune è l'arrivo di un nuovo oggetto in un bucket Amazon S3 (l'operazione `PutObject` di S3).

  Avviare un flusso di lavoro con un batch di eventi significa attendere fino a quando non è stato ricevuto un numero specificato di eventi o fino a quando non è trascorso un determinato periodo di tempo. Quando crei il trigger EventBridge dell'evento, puoi facoltativamente specificare le condizioni del batch. Se si specificano le condizioni del batch, è necessario specificare la dimensione del batch (numero di eventi) e, facoltativamente, è possibile specificare una finestra batch (numero di secondi). La dimensione massima di default della finestra è di 900 secondi (15 minuti). La condizione batch che viene soddisfatta per prima avvia il flusso di lavoro. La finestra

batch si avvia all'arrivo del primo evento. Se durante la creazione di un trigger non si specificano le condizioni di batch, la dimensione del batch viene impostata automaticamente su 1.

All'avvio del flusso di lavoro, le condizioni di batch vengono reimpostate e il trigger di evento inizia a monitorare la condizione di batch successiva da soddisfare per avviare nuovamente il flusso di lavoro.

Nella tabella seguente viene illustrato il modo in cui le dimensioni batch e la finestra batch operano insieme per attivare un flusso di lavoro.

| Dimensione batch | Finestra batch | Condizione di attivazione risultante |
|---|---|---|
| 10 | | Il flusso di lavoro viene attivato all'arrivo di 10 EventBridge eventi o 15 minuti dopo l'arrivo del primo evento, a seconda di quale evento si verifica per primo. (Se la dimensione della finestra non viene specificata, il valore predefinito è 15 minuti.) |
| 10 | 2 minuti | Il flusso di lavoro viene attivato all'arrivo di 10 EventBridge eventi o 2 minuti dopo l'arrivo del primo evento, a seconda di quale evento si verifica per primo. |
| 1 | | Il flusso di lavoro viene attivato all'arrivo del primo evento. La dimensione della finestra è irrilevante. La dimensione predefinita del batch è 1 se non si specifica no le condizioni del batch quando si crea il trigger dell'evento. EventBridge |

L'operazione API `GetWorkflowRun` restituisce la condizione batch che ha attivato il flusso di lavoro.

Indipendentemente dalla modalità di avvio di un flusso di lavoro, è possibile specificare il numero massimo di esecuzioni simultanee durante la creazione del flusso di lavoro.

Se un evento o un batch di eventi avvia un'esecuzione del flusso di lavoro che alla fine ha esito negativo, tale evento o batch di eventi non viene più considerato per l'avvio di un'esecuzione del

flusso di lavoro. Un nuovo flusso di lavoro viene avviato solo quando arriva l'evento o il batch di eventi successivo.

> **Important**
>
> Limita il numero totale di processi, crawler e attivazioni all'interno di un flusso di lavoro a 100 o meno. Se includi più di 100, potresti riscontrare errori durante il tentativo di riprendere o interrompere l'esecuzione del flusso di lavoro.

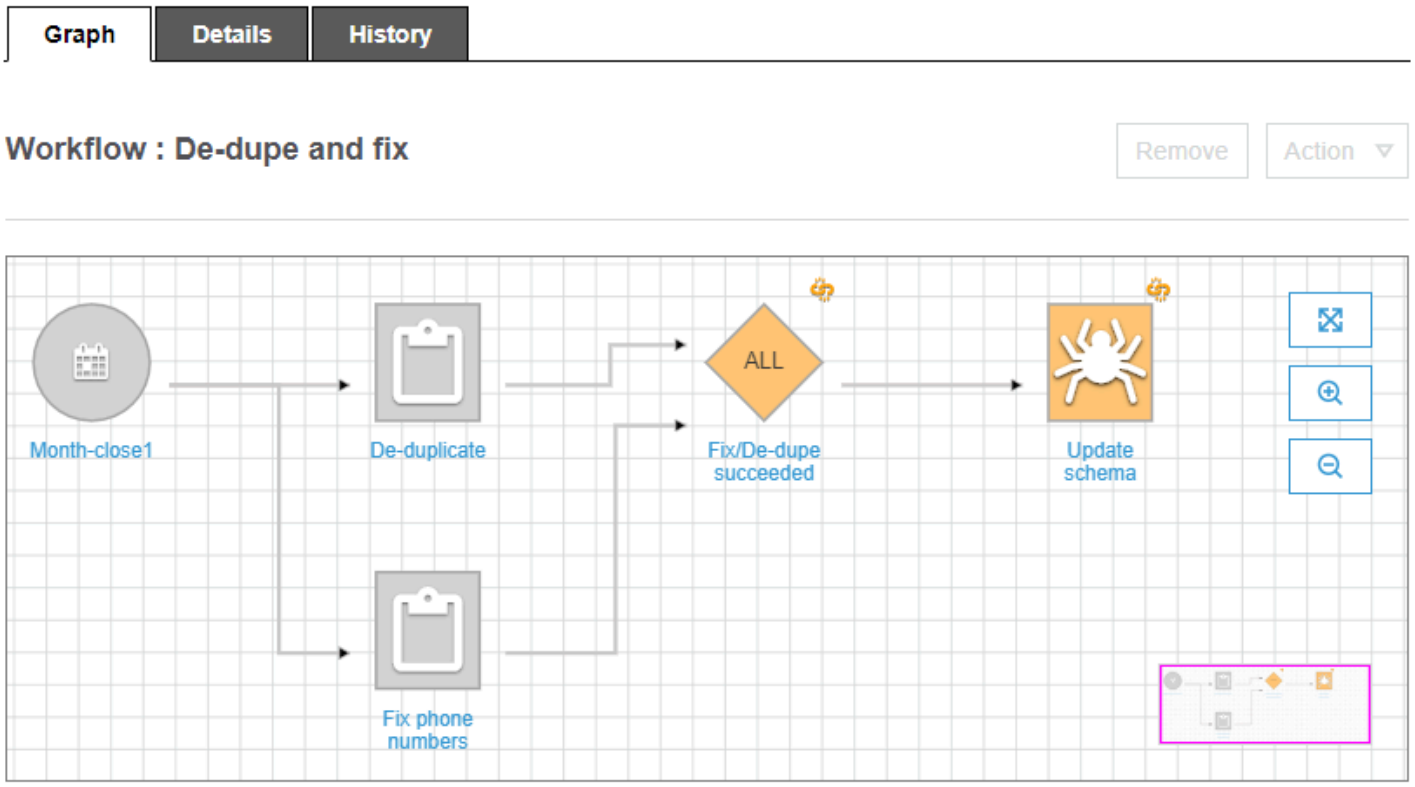
Un'esecuzione del flusso di lavoro non verrà avviata se supererà il limite di concorrenza impostato per il flusso di lavoro, anche se la condizione dell'evento è soddisfatta. È consigliabile modificare i limiti di simultaneità del flusso di lavoro in base al volume degli eventi previsti. AWS Glue non ritenta le esecuzioni del flusso di lavoro che non riescono a causa di limiti di simultaneità superati. Allo stesso modo, è consigliabile modificare i limiti di simultaneità per i processi e i crawler all'interno dei flussi di lavoro in base al volume degli eventi previsto.

Proprietà esecuzione flusso di lavoro

Per condividere e gestire lo stato di un flusso di lavoro in esecuzione, è possibile definire le proprietà dell'esecuzione di flussi di lavoro di default. Queste proprietà, che sono name/value coppie, sono disponibili per tutti i lavori del flusso di lavoro. Utilizzando AWS Glue API, i job possono recuperare le proprietà di esecuzione del flusso di lavoro e modificarle per i lavori successivi nel flusso di lavoro.

Grafico del flusso di lavoro

L'immagine seguente mostra il grafico di un flusso di lavoro basato sulla console di AWS Glue. Un flusso di lavoro potrebbe essere composto da dozzine di componenti.




Questo flusso di lavoro viene avviato da un trigger di pianificazione, `Month-close1`, che avvia due processi, `De-duplicate` e `Fix phone numbers`. Al corretto completamento di entrambi i processi, un trigger di evento, `Fix/De-dupe succeeded`, avvia un crawler, `Update schema`.

Visualizzazioni del flusso di lavoro statica e dinamica

Per ogni flusso di lavoro, esiste il concetto di visualizzazione statica e visualizzazione dinamica. La visualizzazione statica descrive la struttura del flusso di lavoro. La visualizzazione dinamica è una visualizzazione in fase di runtime che include le informazioni sull'ultima esecuzione di ognuno dei processi e dei crawler. Le informazioni sull'esecuzione includono l'esito finale e i dettagli degli errori.

Quando un flusso di lavoro è in esecuzione, la console mostra la visualizzazione dinamica, che indica graficamente i processi che si sono conclusi e quelli che devono ancora essere eseguiti. È anche possibile recuperare una visualizzazione dinamica di un flusso di lavoro in esecuzione utilizzando la AWS Glue API. Per ulteriori informazioni, consulta Interrogazione dei flussi di lavoro utilizzando AWS Glue API.

Consulta anche

- the section called "Creazione di un flusso di lavoro da uno schema"

- the section called "Creazione e costruzione manuale di un flusso di lavoro"
- Flussi di lavoro (per l'API dei flussi di lavoro)

## Creazione e costruzione manuale di un flusso di lavoro in AWS Glue

Puoi utilizzare la console AWS Glue per creare e costruire manualmente un flusso di lavoro, un nodo alla volta.

Un flusso di lavoro contiene processi, crawler e trigger. Prima di creare un flusso di lavoro manualmente, è necessario creare i processi e i crawler che devono essere inclusi nel flusso di lavoro. È preferibile specificare i run-on-demand crawler per i flussi di lavoro. È possibile creare nuovi trigger durante la creazione del flusso di lavoro oppure è possibile clonare i trigger già esistenti nel flusso di lavoro. Quando si clona un trigger, tutti gli oggetti catalogo associati al trigger, ovvero i processi o i crawler che lo attivano e i processi o crawler che avvia, vengono aggiunti al flusso di lavoro.

> **Important**
>
> Limita il numero totale di processi, crawler e attivazioni all'interno di un flusso di lavoro a 100 o meno. Se includi più di 100, potresti riscontrare errori durante il tentativo di riprendere o interrompere l'esecuzione del flusso di lavoro.

È possibile creare il proprio flusso di lavoro aggiungendo trigger al diagramma del flusso di lavoro e definendo gli eventi osservati e le operazioni di ogni trigger. Si inizia con un trigger di attivazione, che può essere un trigger on demand o pianificato, e si completa il diagramma aggiungendo trigger basati su evento (condizionali).

### Fase 1: creazione del flusso di lavoro

1. Accedi a Console di gestione AWS e apri la console all' AWS Glue indirizzo. https://console.aws.amazon.com/glue/
2. Nel pannello di navigazione, in ETL, scegliere Workflows (Flussi di lavoro).
3. Scegliere Add workflow (Aggiungi flusso di lavoro) e completare il modulo Add a new ETL workflow (Aggiungi un nuovo flusso di lavoro ETL).

Qualsiasi proprietà opzionale di default per l'esecuzione aggiuntiva viene resa disponibile come argomento a tutti i processi del flusso di lavoro. Per ulteriori informazioni, consulta Recupero e impostazione delle proprietà di esecuzione del flusso di lavoro in AWS Glue.

4. Scegliere Add workflow (Aggiungi flusso di lavoro).

   Il nuovo flusso di lavoro verrà visualizzato nell'elenco sulla pagina Workflows (Flussi di lavoro).

## Fase 2: aggiunta di un trigger di attivazione

1. Nella pagina Workflows (Flussi di lavoro), selezionare il nuovo flusso di lavoro. Quindi, nella parte inferiore della pagina, assicurarsi che grafico  sia selezionato.
2. Scegliere Add trigger (Aggiungi trigger) e nella finestra di dialogo Add trigger (Aggiungi trigger), procedere in uno dei seguenti modi:

   - Scegliere Clone existing (Clona esistente) e scegliere un trigger da clonare. Quindi scegliere Add (Aggiungi).

     Il trigger viene visualizzato sul diagramma, insieme ai processi e ai crawler che lo attivano e i processi o i crawler che lancia.

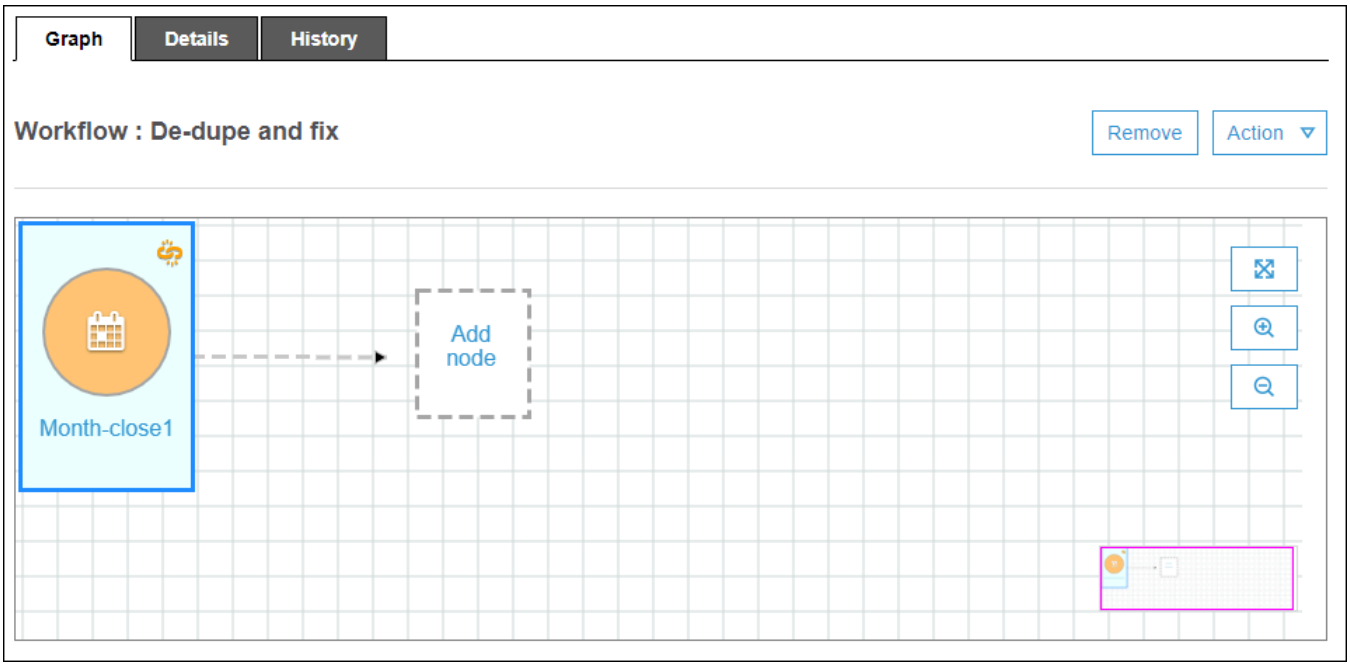
     Se è stato selezionato inavvertitamente il trigger sbagliato, selezionare il trigger nel diagramma e quindi scegliere Remove (Rimuovi).
   - Scegliere Add new (Aggiungi nuovo) e completare il modulo Add trigger (Aggiungi trigger).
     1. Per il tipo di trigger, seleziona Pianificazione, Su richiesta o EventBridgeEvento.

        Per il tipo di trigger Schedule (Pianificazione), scegliere una delle opzioni di Frequency (Frequenza). Scegliere Custom (Personalizza) per immettere un'espressione `cron`.

        Per l'EventBridge evento di tipo trigger, inserisci Numero di eventi (dimensione del batch) e, facoltativamente, inserisci Ritardo temporale (finestra del batch). Omettendo Time delay (Ritardo), la finestra batch viene impostata per impostazione predefinita su 15 minuti. Per ulteriori informazioni, consulta Panoramica di flussi di lavoro in AWS Glue.
     2. Scegliere Aggiungi.

     Il trigger viene visualizzato sul diagramma, insieme a un nodo segnaposto nodo (etichettato Add node (Aggiungi nodo)). Nell'esempio seguente, il trigger di avvio è un trigger di pianificazione denominato `Month-close1`.

A questo punto, il trigger non è ancora salvato.




3. Se è stato aggiunto un nuovo trigger, completare i seguenti passaggi:

   a. Esegui una delle seguenti operazioni:

      - Scegliere il nodo segnaposto (Add node (Aggiungi nodo)).
      - Assicurati che il trigger di avvio sia selezionato e nel menu Azione sopra il grafico, scegli Aggiungi jobs/crawlers al trigger.

   b. Nella finestra di dialogo Add jobs(s) and crawler(s) to trigger (Aggiungi processo/i e crawler al trigger) selezionare uno o più processi o crawler e quindi scegliere Add (Aggiungi).

      Il trigger viene salvato e i processi o crawler selezionati vengono visualizzati sul diagramma con connettori che hanno origine dal trigger.

      Se sono stati aggiunti inavvertitamente processi o crawler sbagliati, è possibile selezionare il trigger o un connettore e scegliere Remove (Rimuovi).

## Fase 3: aggiunta di più trigger

Continuare a creare il flusso di lavoro aggiungendo ulteriori trigger di tipo Event (Evento). Per incrementare o ridurre il livello di dettaglio o per ingrandire l'area di disegno del diagramma, utilizzare le icone a destra. Per ogni trigger da aggiungere, completare i seguenti passaggi:

> **Note**
>
> Non viene eseguita alcuna azione per salvare il flusso di lavoro. Dopo aver aggiunto l'ultimo trigger e assegnato azioni al trigger, il flusso di lavoro è completato e salvato. È sempre possibile tornare successivamente e aggiungere altri nodi.

1. Esegui una delle seguenti operazioni:

   - Per clonare un trigger esistente, accertarsi che non sia selezionato nessun nodo del diagramma e, nel menu Action (Operazione), scegliere Add trigger (Aggiungi trigger).
   - Per aggiungere un nuovo trigger che monitori un determinato processo o crawler del diagramma, selezionare il nodo del processo o del crawler e quindi scegliere il nodo segnaposto Add trigger (Aggiungi trigger).

     È possibile aggiungere ulteriori processi o crawler da far monitorare a questo trigger in un secondo momento.

2. Nella finestra di dialogo Add Folder (Aggiungi cartella), effettuare una delle operazioni indicate di seguito:

   - Scegliere Add new (Aggiungi nuovo) e completare il modulo Add trigger (Aggiungi trigger). Quindi scegliere Add (Aggiungi).

     Il trigger viene visualizzato sul diagramma. È possibile completare il trigger in un secondo momento.
   - Scegliere Clone existing (Clona esistente) e scegliere un trigger da clonare. Quindi scegliere Add (Aggiungi).

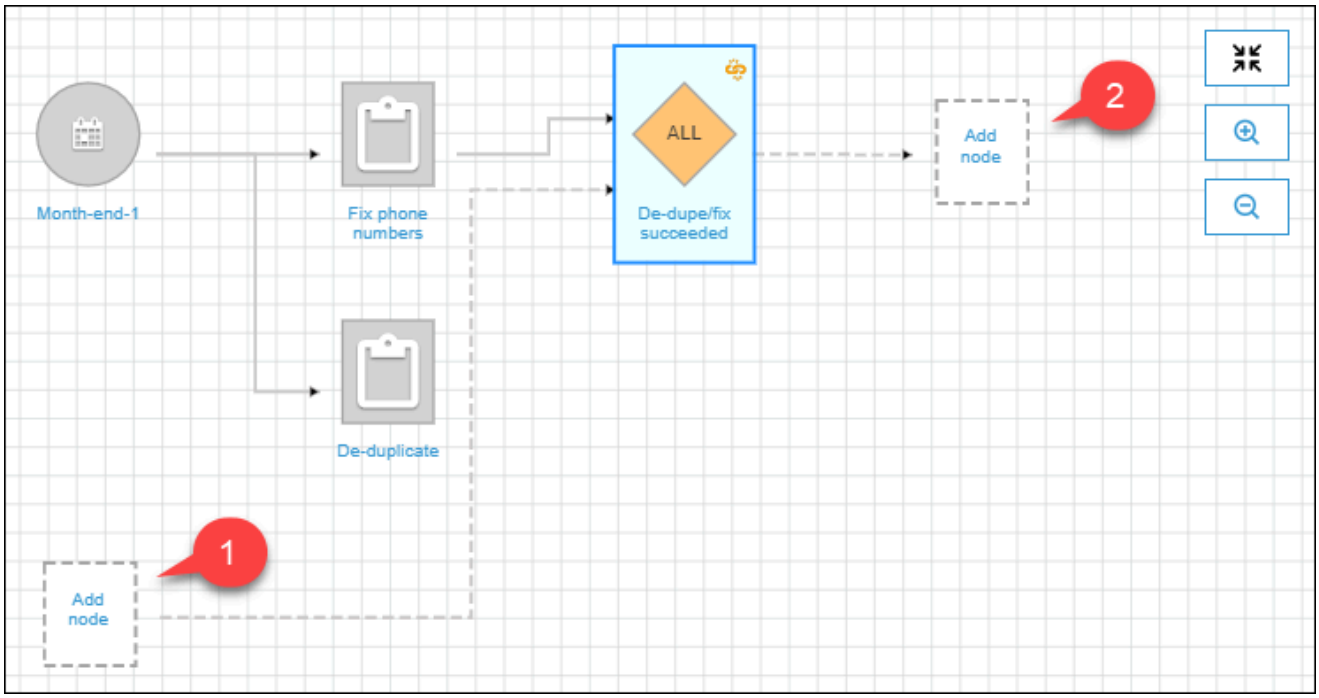
     Il trigger viene visualizzato sul diagramma, insieme ai processi e ai crawler che lo attivano e i processi o i crawler che lancia.

     Se è stato selezionato inavvertitamente il trigger sbagliato, selezionare il trigger nel diagramma e quindi scegliere Remove (Rimuovi).

3. Se è stato aggiunto un nuovo trigger, completare i seguenti passaggi:

   a. Selezionare il nuovo trigger.

Come mostra il seguente diagramma, il trigger `De-dupe/fix succeeded` è selezionato e i nodi segnaposto appaiono per (1) eventi da monitorare e (2) operazioni.




b. (Facoltativo se il trigger monitora già un evento e si desidera aggiungere ulteriori processi o crawler da monitorare) Scegliete il nodo events-to-watch segnaposto e nella finestra di dialogo Aggiungi lavori e crawler da visualizzare, selezionate uno o più lavori o crawler. Scegli un evento da monitorare (SUCCEEDED, FAILED, ecc.) e scegliere Add (Aggiungi).

c. Verificare che il trigger sia selezionato e scegliere il nodo segnaposto delle operazioni.

d. Nella finestra di dialogo Add job(s) and crawler(s) to watch (Aggiungi processo/i e crawler da monitorare), selezionare uno o più processi o crawler e scegliere Add (Aggiungi).

   I processi e crawler selezionati vengono visualizzati sul diagramma con connettori che hanno origine dal trigger.

Per ulteriori informazioni sui flussi di lavoro e sui progetti, consulta i seguenti argomenti.

- Panoramica di flussi di lavoro in AWS Glue
- Esecuzione e monitoraggio di un flusso di lavoro in AWS Glue
- Creazione di un flusso di lavoro da uno schema in AWS Glue

# Avvio di un AWS Glue flusso di lavoro con un EventBridge evento Amazon

Amazon EventBridge, noto anche come CloudWatch Events, ti consente di automatizzare AWS i tuoi servizi e rispondere automaticamente a eventi di sistema come problemi di disponibilità delle applicazioni o modifiche delle risorse. Gli eventi dei AWS servizi vengono forniti quasi EventBridge in tempo reale. Puoi compilare regole semplici che indichino quali eventi sono considerati di interesse per te e quali azioni automatizzate intraprendere quando un evento corrisponde a una regola.

Con EventBridge il supporto, AWS Glue può fungere da produttore e consumatore di eventi in un'architettura basata sugli eventi. Per quanto riguarda i flussi di lavoro, AWS Glue supporta qualsiasi tipo di EventBridge evento come consumatore. Il caso d'uso più comune è l'arrivo di un nuovo oggetto in un bucket Amazon S3. Se si dispone di dati che arrivano a intervalli irregolari o non definiti, è possibile elaborare questi dati il più vicino possibile al loro arrivo.

> **Note**
>
> AWS Gluenon fornisce la consegna garantita dei EventBridge messaggi. AWS Gluenon esegue alcuna deduplicazione se EventBridge recapita messaggi duplicati. È necessario gestire l'idempotenza in base al proprio caso d'uso.
> Assicurati di configurare correttamente EventBridge le regole per evitare l'invio di eventi indesiderati.

Prima di iniziare

Se desideri avviare un flusso di lavoro con gli eventi relativi ai dati di Amazon S3, devi assicurarti che gli eventi per il bucket S3 di interesse siano registrati su e. AWS CloudTrail EventBridge A tale scopo, è necessario creare un percorso. CloudTrail Per ulteriori informazioni, vedi Creazione di un percorso per il tuo AWS account.

Per avviare un flusso di lavoro con un EventBridge evento

> **Note**
>
> Nei comandi seguenti, sostituisci:
>
> - *<workflow-name>*con il nome da assegnare al flusso di lavoro.
> - *<trigger-name>*con il nome da assegnare al trigger.
> - *<bucket-name>*con il nome del bucket Amazon S3.

- *<account-id>*con un ID AWS account valido.
- *<region>*con il nome della regione (ad esempio,`us-east-1`).
- *<rule-name>*con il nome da assegnare alla EventBridge regola.

1. Assicurati di disporre delle autorizzazioni AWS Identity and Access Management (IAM) per creare e visualizzare EventBridge regole e obiettivi. Di seguito è riportato una policy di esempio che è possibile allegare. Potresti ridurre il suo ambito per applicare restrizioni alle operazioni e alle risorse.

   JSON

   ```
   {
     "Version":"2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Action": [
           "events:PutRule",
           "events:DisableRule",
           "events:DeleteRule",
           "events:PutTargets",
           "events:RemoveTargets",
           "events:EnableRule",
           "events:List*",
           "events:Describe*"
         ],
         "Resource": "*"
       }
     ]
   }
   ```

2. Crea un ruolo IAM che il EventBridge servizio possa assumere quando trasmette un evento. AWS Glue

   a. Nella pagina Create role (Crea ruolo) della console IAM, seleziona Service AWS (Servizio AWS ). Quindi scegli il servizio CloudWatch Eventi.

b. Completa la procedura guidata Create role (Crea ruolo). La procedura guidata allega automaticamente il `CloudWatchEventsBuiltInTargetExecutionAccess` e le policy `CloudWatchEventsInvocationAccess`.

c. Allega la seguente policy inline al ruolo. Questa politica consente al EventBridge servizio di indirizzare gli eventi versoAWS Glue.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:notifyEvent"
      ],
      "Resource": [
        "arn:aws:glue:us-east-1:111122223333:workflow/workflow-name"
      ]
    }
  ]
}
```

3. Inserisci il seguente comando per creare il flusso di lavoro.

Per ulteriori informazioni sui parametri della riga di comando aggiuntivi, consulta create-workflow nel Riferimento ai comandi AWS CLI .

```
aws glue create-workflow --name <workflow-name>
```

4. Immettere il comando seguente per creare un trigger di EventBridge evento per il flusso di lavoro. Questo sarà il trigger di avvio per il flusso di lavoro. Sostituiscilo *<actions>* con le azioni da eseguire (i job e i crawler da avviare).

Consulta create-trigger nel Riferimento ai comandi AWS CLI per informazioni su come codificare l'argomento `actions`.

```
aws glue create-trigger --workflow-name <workflow-name> --type EVENT --
name <trigger-name> --actions <actions>
```

Se desideri che il flusso di lavoro venga attivato da un batch di eventi anziché da un singolo EventBridge evento, inserisci invece il comando seguente.

```
aws glue create-trigger --workflow-name <workflow-name> --type EVENT
 --name <trigger-name> --event-batching-condition BatchSize=<number-of-
events>,BatchWindow=<seconds> --actions <actions>
```

Per l'argomento `event-batching-condition`, `BatchSize` è obbligatorio e `BatchWindow` è facoltativo. Se `BatchWindow` viene omesso, il valore predefinito della finestra è 900 secondi, ovvero la dimensione massima della finestra.

Example

L'esempio seguente crea un trigger che avvia il `eventtest` flusso di lavoro dopo l'arrivo di tre EventBridge eventi o cinque minuti dopo l'arrivo del primo evento, a seconda dell'evento che si verifica per primo.

```
aws glue create-trigger --workflow-name eventtest --type EVENT --name objectArrival
 --event-batching-condition BatchSize=3,BatchWindow=300 --actions JobName=test1
```

5. Crea una regola in Amazon EventBridge.

   a. Crea l'oggetto JSON per i dettagli della regola nell'editor di testo che preferisci.

      Nell'esempio seguente viene specificato Amazon S3 come origine evento, `PutObject`come nome dell'evento e il nome del bucket come parametro di richiesta. Questa regola avvia un flusso di lavoro quando un nuovo oggetto arriva nel bucket.

```
{
  "source": [
    "aws.s3"
  ],
  "detail-type": [
    "AWS API Call via CloudTrail"
  ],
  "detail": {
    "eventSource": [
      "s3.amazonaws.com"
    ],
    "eventName": [
```

```
      "PutObject"
    ],
    "requestParameters": {
      "bucketName": [
        "<bucket-name>"
      ]
    }
  }
}
```

Per avviare il flusso di lavoro quando un nuovo oggetto arriva in una cartella all'interno del bucket, è possibile sostituire il codice seguente con `requestParameters`.

```
    "requestParameters": {
      "bucketName": [
        "<bucket-name>"
      ]
      "key" : [{ "prefix" : "<folder1>/<folder2>/*"}}]
  }
```

b. Utilizza lo strumento che preferisci per convertire l'oggetto JSON regola in una stringa di escape.

```
{\n  \"source\": [\n    \"aws.s3\"\n  ],\n  \"detail-type\": [\n    \"AWS API
 Call via CloudTrail\"\n  ],\n  \"detail\": {\n    \"eventSource\": [\n
 \"s3.amazonaws.com\"\n    ],\n    \"eventName\": [\n      \"PutObject\"\n
  ],\n    \"requestParameters\": {\n      \"bucketName\": [\n        \"<bucket-
name>\"\n      ]\n    }\n  }\n}
```

c. Esegui il comando seguente per creare un modello di parametro JSON che puoi modificare per specificare i parametri di input in un comando `put-rule`successivo. Salva l'output in un file. Per questo esempio, il file è denominato `ruleCommand`.

```
aws events put-rule --name <rule-name> --generate-cli-skeleton >ruleCommand
```

Per ulteriori informazioni sul parametro `--generate-cli-skeleton`, consulta Generazione di parametri di input e skeleton AWS CLI da un file di input JSON o YAML nella Guida per l'utente di AWS .

Il file di output deve essere simile al seguente.

```
{
    "Name": "",
    "ScheduleExpression": "",
    "EventPattern": "",
    "State": "ENABLED",
    "Description": "",
    "RoleArn": "",
    "Tags": [
        {
            "Key": "",
            "Value": ""
        }
    ],
    "EventBusName": ""
}
```

d. Modifica il file per rimuovere facoltativamente i parametri e specificare come minimo i parametri `Name`, `EventPattern`, e`State`. Per il parametro `EventPattern`, specifica la stringa di escape per i dettagli della regola creati in un passaggio precedente.

```
{
    "Name": "<rule-name>",
    "EventPattern": "{\n  \"source\": [\n    \"aws.s3\"\n  ],\n  \"detail-
type\": [\n    \"AWS API Call via CloudTrail\"\n  ],\n  \"detail\": {\n
 \"eventSource\": [\n      \"s3.amazonaws.com\"\n    ],\n    \"eventName\": [\n
      \"PutObject\"\n    ],\n    \"requestParameters\": {\n      \"bucketName
\": [\n        \"<bucket-name>\"\n      ]\n    }\n  }\n}",
    "State": "DISABLED",
    "Description": "Start an AWS Glue workflow upon new file arrival in an
 Amazon S3 bucket"
}
```

> **Note**
>
> È consigliabile lasciare disabilitata la regola fino a quando non si completa la creazione del flusso di lavoro.

e. Immetti la seguente comando `put-rule`, che legge i parametri di input dal file `ruleCommand`.

```
aws events put-rule --name <rule-name> --cli-input-json file://ruleCommand
```

Il seguente output indica il successo.

```
{
    "RuleArn": "<rule-arn>"
}
```

6. Immetti il seguente comando per collegare la regola alla destinazione. La destinazione è il flusso di lavoro in AWS Glue. Sostituiscila *<role-name>* con il ruolo che hai creato all'inizio di questa procedura.

```
aws events put-targets --rule <rule-name> --targets
 "Id"="1","Arn"="arn:aws:glue:<region>:<account-id>:workflow/<workflow-
name>","RoleArn"="arn:aws:iam::<account-id>:role/<role-name>" --region <region>
```

Il seguente output indica il successo.

```
{
    "FailedEntryCount": 0,
    "FailedEntries": []
}
```

7. Conferma la corretta connessione della regola e della destinazione inserendo il comando seguente.

```
aws events list-rule-names-by-target --target-arn arn:aws:glue:<region>:<account-
id>:workflow/<workflow-name>
```

L'output seguente indica l'esito positivo, *<rule-name>* dov'è il nome della regola creata.

```
{
    "RuleNames": [
        "<rule-name>"
    ]
}
```

8. Accedi a Console di gestione AWS e apri la AWS Glue console all'indirizzo https://console.aws.amazon.com/glue/.

9. Seleziona il flusso di lavoro e verifica che il trigger di avvio e le relative azioni, ovvero i processi o i crawler che avvia, vengano visualizzati nel grafico del flusso di lavoro. Poi, continua con la procedura in Fase 3: aggiunta di più trigger. In alternativa, aggiungi altri componenti al flusso di lavoro utilizzando l'API AWS Glue o AWS Command Line Interface.
10. Quando il flusso di lavoro è specificato completamente, abilita la regola.

```
aws events enable-rule --name <rule-name>
```

Il flusso di lavoro è ora pronto per essere avviato da un EventBridge evento o da un batch di eventi.

Consulta anche

- Guida per EventBridge l'utente di Amazon
- Panoramica di flussi di lavoro in AWS Glue
- Creazione e costruzione manuale di un flusso di lavoro in AWS Glue

## Visualizzazione degli EventBridge eventi che hanno avviato un flusso di lavoro

Puoi visualizzare l'ID evento dell' EventBridge evento Amazon che ha avviato il tuo flusso di lavoro. Se il flusso di lavoro è stato avviato da un batch di eventi, puoi visualizzare l'evento IDs di tutti gli eventi del batch.

Per i flussi di lavoro con una dimensione batch maggiore di uno, è inoltre possibile verificare quale condizione batch ha avviato il flusso di lavoro: l'arrivo del numero di eventi nella dimensione batch o la scadenza del periodo batch.

Per visualizzare gli EventBridge eventi che hanno avviato un flusso di lavoro (console)

1. Accedi a Console di gestione AWS e apri la AWS Glue console all'indirizzo https://console.aws.amazon.com/glue/.
2. Nel pannello di navigazione, scegli Workflows (Flussi di lavoro).
3. Seleziona un flusso di lavoro. Quindi, nella parte inferiore, scegli la scheda History (Cronologia).

4. Seleziona un'esecuzione del flusso di lavoro, quindi scegli View run details (Visualizza i dettagli dell'esecuzione).
5. Nella pagina dei dettagli di esecuzione, individua il campo Run propertiese cerca il campo (Proprietà di esecuzione) e cerca la chiave aws:eventIds.

   Il valore di quella chiave è un elenco di EventBridge eventi IDs.

Per visualizzare gli EventBridge eventi che hanno avviato un flusso di lavoro (AWS API)

- Includi il seguente codice nello script Python.

```
workflow_params =
 glue_client.get_workflow_run_properties(Name=workflow_name,RunId=workflow_run_id)
batched_events = workflow_params['aws:eventIds']
```

  `batched_events` sarà un elenco di stringhe, in cui ogni stringa è un ID evento.

Consulta anche

- Guida per EventBridge l'utente di Amazon
- the section called “Panoramica di flussi di lavoro”

## Esecuzione e monitoraggio di un flusso di lavoro in AWS Glue

Se il trigger di attivazione di un flusso di lavoro è un trigger on-demand, è possibile avviare il flusso di lavoro dalla console di AWS Glue. Utilizza la procedura seguente per eseguire e monitorare un flusso di lavoro. Se il flusso di lavoro non riesce, puoi visualizzare il grafico di esecuzione per individuare il nodo non riuscito. Per facilitare la risoluzione dei problemi, se il flusso di lavoro è stato creato da un piano, puoi visualizzare l'esecuzione del piano per vedere i valori dei parametri del piano utilizzati per creare il flusso di lavoro. Per ulteriori informazioni, consulta the section called “Visualizzazione delle esecuzioni dello schema”.

Puoi eseguire e monitorare l'esecuzione di un flusso di lavoro utilizzando la console AWS Glue, l'API o AWS Command Line Interface (AWS CLI).

Per eseguire e monitorare un flusso di lavoro (console)

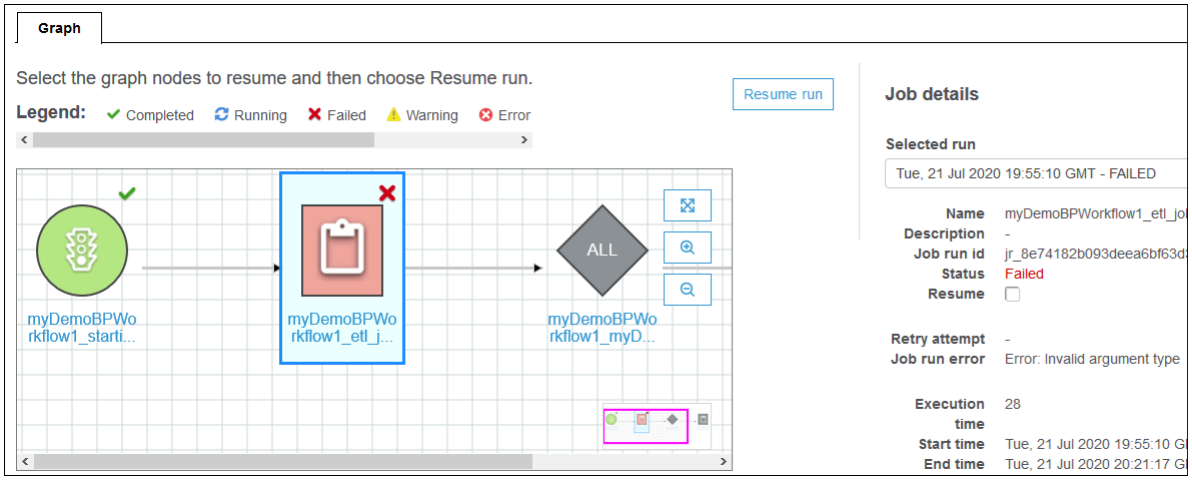
1. Apri la AWS Glue console all'indirizzo https://console.aws.amazon.com/glue/.
2. Nel pannello di navigazione, in ETL, scegliere Workflows (Flussi di lavoro).
3. Selezionare un flusso di lavoro. Nel menu Actions (Operazioni), scegliere Run (Esegui).
4. Controlla la colonna Last run status (Stato dell'ultima esecuzione) nell'elenco dei flussi di lavoro. Scegli il pulsante di aggiornamento per visualizzare lo stato del flusso di lavoro in corso.
5. Mentre il flusso di lavoro è in esecuzione o dopo che è stato completato (o non riuscito), visualizza i dettagli dell'esecuzione completando la procedura seguente.

   a. Verifica che il flusso di lavoro sia selezionato e scegli la scheda History (Cronologia).

   b. Seleziona l'esecuzione del flusso di lavoro corrente o più recente, quindi seleziona View run details (Visualizza i dettagli dell'esecuzione).

      Il grafico del tempo di esecuzione del flusso di lavoro mostra lo stato corrente dell'esecuzione.

   c. Scegli un nodo nel grafico per visualizzare relativi i dettagli e lo stato.




Per eseguire e monitorare un flusso di lavoro (AWS CLI)

1. Inserire il seguente comando. Sostituisci *<workflow-name>* con il flusso di lavoro da eseguire.

```
aws glue start-workflow-run --name <workflow-name>
```

   Se il flusso di lavoro viene avviato correttamente, il comando restituisce l'ID di esecuzione.

2. Visualizza lo stato di esecuzione del flusso di lavoro utilizzando il comando `get-workflow-run`. Specifica il nome del flusso di lavoro e l'ID di esecuzione.

```
aws glue get-workflow-run --name myWorkflow --run-id
 wr_d2af14217e8eae775ba7b1fc6fc7a42c795aed3cbcd8763f9415452e2dbc8705
```

Di seguito è riportato un output del comando di esempio.

```
{
    "Run": {
        "Name": "myWorkflow",
        "WorkflowRunId":
 "wr_d2af14217e8eae775ba7b1fc6fc7a42c795aed3cbcd8763f9415452e2dbc8705",
        "WorkflowRunProperties": {
            "run_state": "COMPLETED",
            "unique_id": "fee63f30-c512-4742-a9b1-7c8183bdaae2"
        },
        "StartedOn": 1578556843.049,
        "CompletedOn": 1578558649.928,
        "Status": "COMPLETED",
        "Statistics": {
            "TotalActions": 11,
            "TimeoutActions": 0,
            "FailedActions": 0,
            "StoppedActions": 0,
            "SucceededActions": 9,
            "RunningActions": 0,
            "ErroredActions": 0
        }
    }
}
```

Consulta anche:

- the section called “Panoramica di flussi di lavoro”
- the section called “Panoramica degli schemi”

# Arresto dell'esecuzione di un flusso di lavoro

Puoi utilizzare la AWS Glue console, AWS Command Line Interface (AWS CLI) o l'AWS GlueAPI per interrompere l'esecuzione di un flusso di lavoro. Quando si interrompe l'esecuzione di un flusso di lavoro, tutti i processi e i crawler in esecuzione vengono immediatamente terminati e i processi e i crawler non ancora avviati non vengono mai avviati. Potrebbe essere necessario fino a un minuto prima che tutti i processi in esecuzione e i crawler si fermino. Lo stato di esecuzione del flusso di lavoro passa da Running (In esecuzione) a Stopping (Arresto in corso), e quando l'esecuzione del flusso di lavoro è completamente interrotta, lo stato passa a Stopped (Arrestato).

Dopo l'interruzione dell'esecuzione del flusso di lavoro, è possibile visualizzare il grafico di esecuzione per verificare quali processi e crawler sono stati completati e quali non sono mai stati avviati. È quindi possibile determinare se è necessario eseguire qualsiasi procedura per garantire l'integrità dei dati. L'arresto di un'esecuzione del flusso di lavoro non comporta l'esecuzione di operazioni di rollback automatico.

Per interrompere l'esecuzione di un flusso di lavoro (console)

1. Apri la AWS Glue console all'indirizzo https://console.aws.amazon.com/glue/.
2. Nel pannello di navigazione, in ETL, scegliere Workflows (Flussi di lavoro).
3. Scegliere un flusso di lavoro in esecuzione, quindi scegliere la scheda History (Cronologia) .
4. Scegliere l'esecuzione del flusso di lavoro, quindi scegliere Stop run (Arresta esecuzione).

   Lo stato di esecuzione cambia in Stopping (Arresto in corso).
5. (Facoltativo) Scegliere l'esecuzione del flusso di lavoro, scegliere View run details (Visualizza dettagli esecuzione), ed esaminare il grafico di esecuzione.

Per interrompere l'esecuzione di un flusso di lavoro (AWS CLI)

- Inserire il seguente comando. Sostituisci *<workflow-name>* con il nome del flusso di lavoro e *<run-id>* con l'ID di esecuzione del flusso di lavoro esegui per interrompere.

  ```
  aws glue stop-workflow-run --name <workflow-name> --run-id <run-id>
  ```

  Di seguito è riportato un comando stop-workflow-run di esempio.

```
aws glue stop-workflow-run --name my-workflow --run-id
 wr_137b88917411d128081069901e4a80595d97f719282094b7f271d09576770354
```

## Ripresa e ripristino dell'esecuzione di un flusso di lavoro

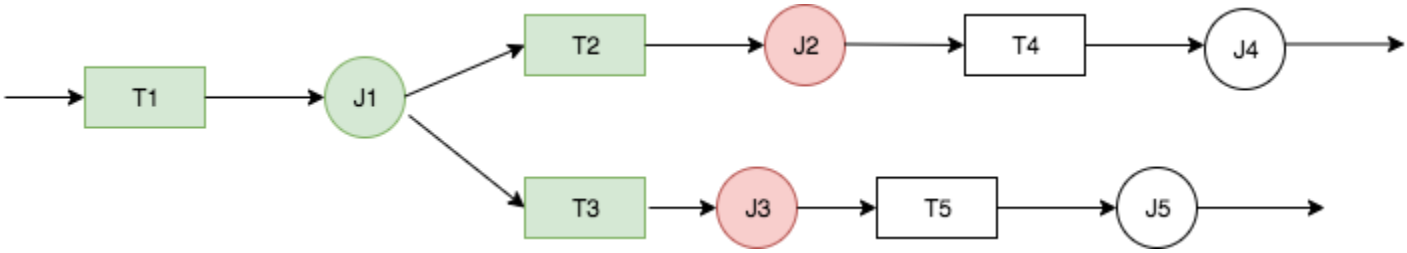
Se uno o più nodi (processi o crawler) in un flusso di lavoro non vengono completati correttamente, ciò significa che il flusso di lavoro è stato eseguito solo parzialmente. Dopo aver individuato le cause principali e apportato le correzioni, è possibile selezionare uno o più nodi da cui riprendere l'esecuzione del flusso di lavoro e quindi riprendere l'esecuzione del flusso di lavoro. Vengono eseguiti i nodi selezionati e tutti i nodi che sono a valle dei nodi selezionati.

Argomenti

- Riprendere un'esecuzione del flusso di lavoro: come funziona
- Riprendere un'esecuzione di flusso di lavoro
- Note e limitazioni per la ripresa delle esecuzioni del flusso di lavoro

### Riprendere un'esecuzione del flusso di lavoro: come funziona

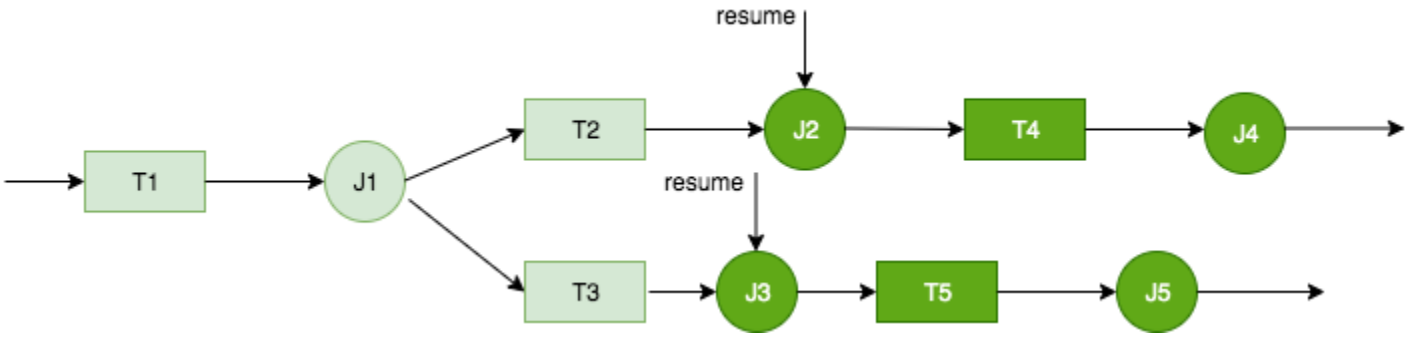
Considera il flusso di lavoro W1 nel diagramma seguente.




L'esecuzione del flusso di lavoro si svolge come segue:

1. Il trigger T1 avvia il processo J1.
2. Il completamento riuscito di J1 attiva i trigger T2 e T3, che eseguono i processi J2 e J3, rispettivamente.
3. I processi J2 e J3 non riescono.
4. I trigger T4 e T5 dipendono dal completamento riuscito di J2 e J3, quindi non si attivano e i processi J4 e J5 non vengono eseguiti. Il flusso di lavoro W1 viene eseguito solo parzialmente.

Ora supponiamo che i problemi che hanno causato la non riuscita di J2 e J3 vengano corretti. J2 e J3 sono selezionati come punti di partenza da cui riprendere l'esecuzione del flusso di lavoro.




L'esecuzione del flusso di lavoro riprende come segue:

1. I processi J2 e J3 vengono eseguiti correttamente.
2. I trigger T4 e T5 si attivano.
3. I processi J4 e J5 vengono eseguiti correttamente.

L'esecuzione del flusso di lavoro ripreso viene registrata come un'esecuzione separata del flusso di lavoro con un nuovo ID di esecuzione. Nella cronologia del flusso di lavoro, è possibile visualizzare l'ID dell' esecuzione precedente per qualsiasi esecuzione del flusso di lavoro. Nell'esempio nello screenshot seguente, un nodo dell'esecuzione del flusso di lavoro con ID di esecuzione `wr_c7a22...` (la seconda riga) non è stato completato. L'utente ha risolto il problema e ha ripreso l'esecuzione del flusso di lavoro, con ID di esecuzione `wr_a07e55...` (la prima riga).

> **Note**
>
> Per il resto di questo articolo, il termine "esecuzione del flusso di lavoro ripreso" si riferisce all'esecuzione del flusso di lavoro creata quando è stata ripresa l'esecuzione precedente del flusso di lavoro. L'"esecuzione del flusso di lavoro originale" indica l'esecuzione del flusso di lavoro che è stata eseguita solo parzialmente e che doveva essere ripresa.

Grafico dell'esecuzione del flusso di lavoro ripreso

In un'esecuzione del flusso di lavoro ripreso, nonostante venga eseguito solo un sottoinsieme di nodi, il grafico di esecuzione è un grafico completo. Cioè, i nodi che non sono stati eseguiti nel flusso di lavoro ripreso vengono copiati dal grafico dell'esecuzione del flusso di lavoro originale. I nodi del processo e del crawler copiati eseguiti nell'esecuzione del flusso di lavoro originale includono dettagli di esecuzione.

Considera nuovamente il flusso di lavoro W1 nel diagramma precedente. Quando l'esecuzione del flusso di lavoro viene ripresa a partire da J2 e J3, il grafico di esecuzione per l'esecuzione del flusso di lavoro ripreso mostra tutti i processi, da J1 a J5, e tutti i trigger, da T1 a T5. I dettagli del processo per J1 vengono copiati dall'esecuzione del flusso di lavoro originale.

Snapshot dell'esecuzione del flusso di lavoro

All'avvio di un'esecuzione del flusso di lavoro, AWS Glue acquisisce uno snapshot del grafico di progettazione del flusso di lavoro in quel momento. Lo snapshot viene utilizzato per la durata dell'esecuzione del flusso di lavoro. Se si apportano modifiche a qualsiasi trigger dopo l'avvio dell'esecuzione, tali modifiche non influiscono sull'esecuzione del flusso di lavoro corrente. Gli snapshot garantiscono che le esecuzioni del flusso di lavoro procedano in modo coerente.

Gli snapshot rendono immutabili solo i trigger. Le modifiche apportate ai processi a valle e ai crawler durante l'esecuzione del flusso di lavoro diventano effettive per l'esecuzione corrente.

## Riprendere un'esecuzione di flusso di lavoro

Segui questi passaggi per riprendere un'esecuzione di flusso di lavoro. Puoi riprendere l'esecuzione di un flusso di lavoro utilizzando la AWS Glue console, l'API o AWS Command Line Interface (AWS CLI).

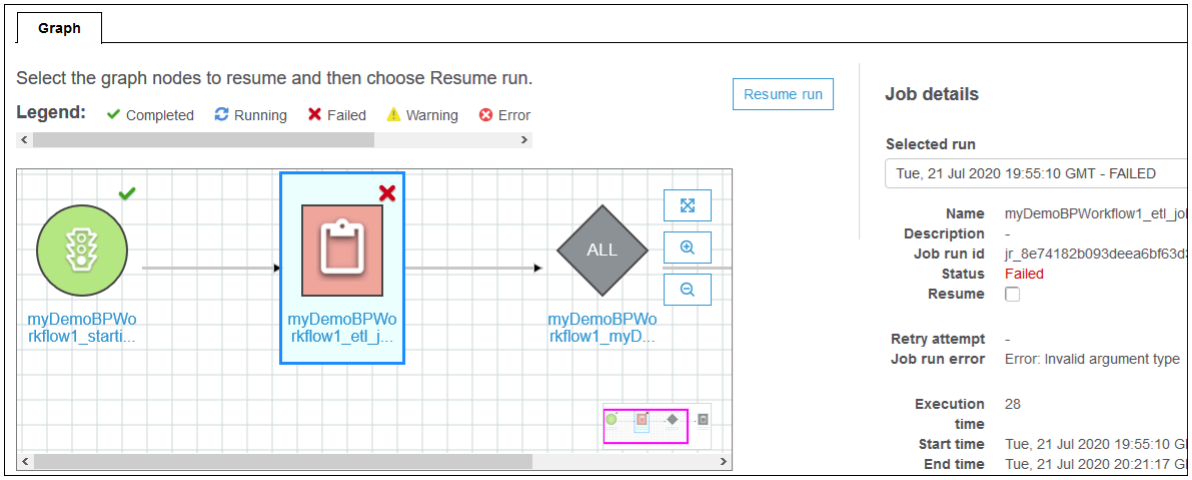
Per riprendere un flusso di lavoro (console)

1. Apri la AWS Glue console all'indirizzo https://console.aws.amazon.com/glue/.

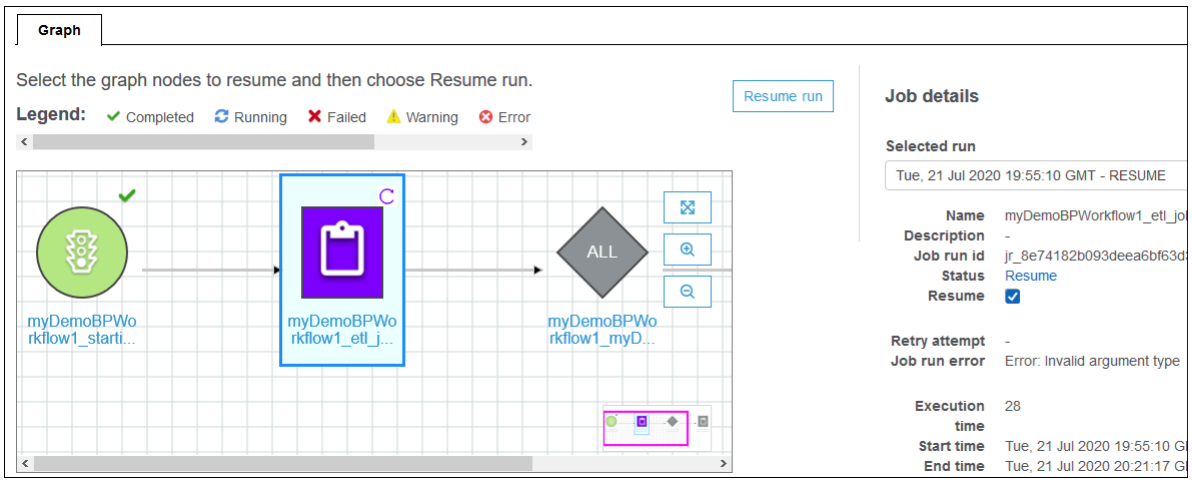
   Accedi come utente che dispone delle autorizzazioni per visualizzare i flussi di lavoro e riprendere le esecuzioni dei flussi di lavoro.

> **Note**
>
> Per riprendere l'esecuzione del flusso di lavoro, è necessaria l'autorizzazione `glue:ResumeWorkflowRun` AWS Identity and Access Management (IAM).

2. Nel pannello di navigazione, scegli Workflows (Flussi di lavoro).
3. Seleziona un flusso di lavoro, quindi la scheda History (Cronologia).
4. Seleziona un'esecuzione del flusso di lavoro eseguita solo parzialmente, quindi seleziona View run details (Visualizza i dettagli dell'esecuzione).
5. Nel grafico di esecuzione seleziona il primo (o solo) nodo da riavviare e da cui riprendere l'esecuzione del flusso di lavoro.
6. Nel riquadro dei dettagli a destra del grafico, seleziona la casella di controlloResume (Riprendi).




Il nodo cambia colore e mostra una piccola icona di ripresa in alto a destra.




7. Completa i due passaggi precedenti per riavviare eventuali nodi aggiuntivi.

8. Seleziona Resume (Riprendi).

Per riprendere un flusso di lavoro (AWS CLI)

1. Assicurati di disporre dell'autorizzazione `glue:ResumeWorkflowRun` IAM.
2. Recupera il nodo IDs per i nodi che desideri riavviare.

   a. Esegui il comando `get-workflow-run` per l'esecuzione del flusso di lavoro originale. Fornisci il nome del flusso di lavoro e l'ID di esecuzione e aggiungi l'opzione `--include-graph`, come mostrato nell'esempio seguente. Recupera l'ID di esecuzione dalla scheda History (Cronologia) sulla console o eseguendo il comando `get-workflow`.

   ```
   aws glue get-workflow-run --name cloudtrailtest1 --run-id
    wr_a07e55f2087afdd415a404403f644a4265278f68b13ba3da08c71924ebe3c3a8 --include-
   graph
   ```

   Il comando restituisce i nodi e gli estremi del grafico come un oggetto JSON di grandi dimensioni.

   b. Individua i nodi di interesse dalle proprietà `Type` e `Name` degli oggetti nodo.

   Il seguente è un esempio di oggetto nodo dell'output.

   ```
   {
       "Type": "JOB",
       "Name": "test1_post_failure_4592978",
       "UniqueId":
    "wnode_d1b2563c503078b153142ee76ce545fe5ceef66e053628a786ddd74a05da86fd",
       "JobDetails": {
           "JobRuns": [
               {
                   "Id":
    "jr_690b9f7fc5cb399204bc542c6c956f39934496a5d665a42de891e5b01f59e613",
                   "Attempt": 0,
                   "TriggerName": "test1_aggregate_failure_649b2432",
                   "JobName": "test1_post_failure_4592978",
                   "StartedOn": 1595358275.375,
                   "LastModifiedOn": 1595358298.785,
                   "CompletedOn": 1595358298.785,
                   "JobRunState": "FAILED",
                   "PredecessorRuns": [],
   ```

```
                "AllocatedCapacity": 0,
                "ExecutionTime": 16,
                "Timeout": 2880,
                "MaxCapacity": 0.0625,
                "LogGroupName": "/aws-glue/python-jobs"
            }
        ]
    }
}
```

c. Recupera l'ID nodo dalla proprietà `UniqueId` dell'oggetto nodo.

3. Esegui il comando `resume-workflow-run`. Fornite il nome del flusso di lavoro, l'ID di esecuzione e l'elenco dei nodi IDs separati da spazi, come illustrato nell'esempio seguente.

```
aws glue resume-workflow-run --name cloudtrailtest1 --run-id
 wr_a07e55f2087afdd415a404403f644a4265278f68b13ba3da08c71924ebe3c3a8 --node-
ids wnode_ca1f63e918fb855e063aed2f42ec5762ccf71b80082ae2eb5daeb8052442f2f3
 wnode_d1b2563c503078b153142ee76ce545fe5ceef66e053628a786ddd74a05da86fd
```

Il comando restituisce l'ID di esecuzione della (nuova) esecuzione del flusso di lavoro ripresa e un elenco di nodi che verranno avviati.

```
{
    "RunId": "wr_2ada0d3209a262fc1156e4291134b3bd643491bcfb0ceead30bd3e4efac24de9",
    "NodeIds": [
        "wnode_ca1f63e918fb855e063aed2f42ec5762ccf71b80082ae2eb5daeb8052442f2f3"
    ]
}
```

Nota che anche se l'esempio `resume-workflow-run` elencava due nodi da riavviare, l'output di esempio indicava che un solo nodo sarebbe stato riavviato. Questo perché un nodo era a valle dell'altro nodo e il nodo a valle viene comunque riavviato dal flusso normale del flusso di lavoro.

## Note e limitazioni per la ripresa delle esecuzioni del flusso di lavoro

Tieni presente le seguenti note e limitazioni per la ripresa delle esecuzioni del flusso di lavoro.

- È possibile riprendere l'esecuzione di un flusso di lavoro solo se si trova nello stato `COMPLETED`.

> **Note**
>
> Anche se uno o più nodi in un'esecuzione del flusso di lavoro non vengono completati, lo stato di esecuzione del flusso di lavoro viene visualizzato come `COMPLETED`. Accertati di controllare il grafico di esecuzione per individuare eventuali nodi che non sono stati completati correttamente.

- È possibile riprendere l'esecuzione di un flusso di lavoro da qualsiasi nodo del processo o del crawler che il flusso di lavoro originale ha tentato di eseguire. Non è possibile riprendere l'esecuzione di un flusso di lavoro da un nodo trigger.
- Il riavvio di un nodo non ne reimposta lo stato. Tutti i dati parzialmente elaborati non vengono ripristinati.
- È possibile riprendere un workflow non riuscito eseguito più volte. Tuttavia, un'esecuzione ripresa può essere ripresa solo un'altra volta. Per ulteriori tentativi, riprendi invece l'esecuzione originale non riuscita
- Se si selezionano due nodi da riavviare e dipendono l'uno dall'altro, il nodo a monte viene eseguito prima del nodo a valle. Di fatto, la selezione del nodo a valle è ridondante, perché viene eseguito in base al normale flusso di lavoro.

## Recupero e impostazione delle proprietà di esecuzione del flusso di lavoro in AWS Glue

Utilizzare le proprietà di esecuzione dei flussi di lavoro per condividere e gestire lo stato tra i processi del flusso di lavoro di AWS Glue. È possibile impostare proprietà di esecuzione di default al momento della creazione del flusso di lavoro. Quindi, nel momento in cui i processi sono eseguiti, è possibile recuperare i valori delle proprietà di esecuzione e, se necessario, modificarli come input per i processi successivi nel flusso di lavoro. Quando un processo modifica una proprietà di esecuzione, il nuovo valore esiste solo per il flusso di lavoro in esecuzione. Le proprietà di esecuzione predefinite non sono interessate.

Se il lavoro AWS Glue non fa parte di un workflow, queste proprietà non verranno impostate.

Il codice Python di esempio seguente relativo a un processo di estrazione, trasformazione e caricamento (ETL) mostra come recuperare le proprietà di esecuzione del flusso di lavoro.

```
import sys
```

```
import boto3
from awsglue.transforms import *
from awsglue.utils import getResolvedOptions
from awsglue.context import GlueContext
from pyspark.context import SparkContext

glue_client = boto3.client("glue")
args = getResolvedOptions(sys.argv, ['JOB_NAME','WORKFLOW_NAME', 'WORKFLOW_RUN_ID'])
workflow_name = args['WORKFLOW_NAME']
workflow_run_id = args['WORKFLOW_RUN_ID']
workflow_params = glue_client.get_workflow_run_properties(Name=workflow_name,
                                       RunId=workflow_run_id)["RunProperties"]

target_database = workflow_params['target_database']
target_s3_location = workflow_params['target_s3_location']
```

Il codice seguente prosegue impostando la proprietà di esecuzione `target_format` al valore `'csv'`.

```
workflow_params['target_format'] = 'csv'
glue_client.put_workflow_run_properties(Name=workflow_name, RunId=workflow_run_id,
 RunProperties=workflow_params)
```

Per ulteriori informazioni, consulta gli argomenti seguenti:

- GetWorkflowRunProperties azione (Python: get_workflow_run_properties)
- PutWorkflowRunProperties azione (Python: put_workflow_run_properties)

## Interrogazione dei flussi di lavoro utilizzando AWS Glue API

AWS Glue fornisce una ricca API per la gestione dei flussi di lavoro. Utilizzando la AWS Glue APIè possibile recuperare una visualizzazione statica o dinamica di un flusso di lavoro in esecuzione. Per ulteriori informazioni, consulta Flussi di lavoro.

Argomenti

- Eseguire query sulle visualizzazioni statiche
- Eseguire query sulle visualizzazioni dinamiche

## Eseguire query sulle visualizzazioni statiche

Per ottenere una visualizzazione statica che descrive la configurazione di un flusso di lavoro, utilizzare l'operazione API `GetWorkflow`. Questa operazione restituisce un diagramma orientato composto da nodi e archi, nel quale ogni nodo rappresenta un trigger, un processo o un crawler. Gli archi definiscono le relazioni tra i nodi. Sono rappresentati da connettori (frecce) all'interno del diagramma visualizzato nella console di AWS Glue.

È inoltre possibile utilizzare questa operazione con le librerie di elaborazione grafica più diffuse come NetworkX, igraph, JGraph T e Java Universal (JUNG) Framework. Network/Graph Poiché tutte queste librerie rappresentano i diagrammi in modo analogo, sono necessarie trasformazioni minime.

La visualizzazione statica restituita da questa API è la più up-to-date visualizzata secondo l'ultima definizione di trigger associati al flusso di lavoro.

Definizione del diagramma

Un diagramma di un flusso di lavoro G è una coppia ordinata (N, E), dove N è costituito da un insieme di nodi ed E è costituito da un insieme di archi. Un Nodo è un vertice del diagramma identificato da un numero univoco. Un nodo può essere di tipo trigger, processo o crawler.
Ad esempio: `{name:T1, type:Trigger, uniqueId:1}, {name:J1, type:Job, uniqueId:2}`.

Un Edge è una tupla nella forma (`src, dest`), dove `src` e `dest` sono nodi ed è presente un arco diretto che va da `src` a `dest`.

Esempio di esecuzione di query di una visualizzazione statica.

Considerare un trigger condizionale T, che attiva il processo J2 al completamento del processo J1.

```
J1 ---> T ---> J2
```

Nodi: J1, T, J2

Archi: (J1, T), (T, J2)

## Eseguire query sulle visualizzazioni dinamiche

Per ottenere una visualizzazione dinamica di un flusso di lavoro in esecuzione, utilizzare l'operazione API `GetWorkflowRun`. Questa operazione restituisce una visualizzazione identica a quella statica con i metadati correlati all'esecuzione del flusso di lavoro.

Per l'esecuzione, i nodi che rappresentano i processi nella chiamata `GetWorkflowRun` sono associati a un elenco di esecuzioni dei processi derivanti dall'ultima esecuzione del flusso di lavoro. È possibile usare questo elenco per visualizzare lo stato di esecuzione di ogni processo nel diagramma stesso. Per dipendenze a valle non ancora eseguite, questo campo è impostato su `null`. Le informazioni rappresentate sul diagramma permettono di conoscere lo stato attuale di qualsiasi flusso di lavoro in qualsiasi momento.

La visualizzazione dinamica restituita da questa API si basa sulla visualizzazione statica presente al momento dell'avvio dell'esecuzione del flusso di lavoro.

Esempio del tempo di esecuzione dei nodi: `{name:T1, type: Trigger, uniqueId:1}`, `{name:J1, type:Job, uniqueId:2, jobDetails:{jobRuns}}`, `{name:C1, type:Crawler, uniqueId:3, crawlerDetails:{crawls}}`

Esempio 1: visualizzazione dinamica

L'esempio seguente illustra un semplice flusso di lavoro costituito da due trigger.

- Nodi: t1, j1, t2, j2
- Archi: (t1, j1), (j1, t2), (t2, j2)

La risposta di `GetWorkflow` contiene quanto segue.

```
{
    Nodes : [
        {
            "type" : Trigger,
            "name" : "t1",
            "uniqueId" : 1
        },
        {
            "type" : Job,
            "name" : "j1",
            "uniqueId" : 2
        },
        {
            "type" : Trigger,
            "name" : "t2",
            "uniqueId" : 3
        },
        {
```

```
            "type" : Job,
            "name" : "j2",
            "uniqueId" : 4
        }
    ],
    Edges : [
        {
            "sourceId" : 1,
            "destinationId" : 2
        },
        {
            "sourceId" : 2,
            "destinationId" : 3
        },
        {
            "sourceId" : 3,
            "destinationId" : 4
        }
}
```

La risposta di `GetWorkflowRun` contiene quanto segue.

```
{
    Nodes : [
        {
            "type" : Trigger,
            "name" : "t1",
            "uniqueId" : 1,
            "jobDetails" : null,
            "crawlerDetails" : null
        },
        {
            "type" : Job,
            "name" : "j1",
            "uniqueId" : 2,
            "jobDetails" : [
                {
                    "id" : "jr_12334",
                    "jobRunState" : "SUCCEEDED",
                    "errorMessage" : "error string"
                }
            ],
            "crawlerDetails" : null
```

```
        },
        {
            "type" : Trigger,
            "name" : "t2",
            "uniqueId" : 3,
            "jobDetails" : null,
            "crawlerDetails" : null
        },
        {
            "type" : Job,
            "name" : "j2",
            "uniqueId" : 4,
            "jobDetails" : [
                {
                    "id" : "jr_1233sdf4",
                    "jobRunState" : "SUCCEEDED",
                    "errorMessage" : "error string"
                }
            ],
            "crawlerDetails" : null
        }
    ],
    Edges : [
        {
            "sourceId" : 1,
            "destinationId" : 2
        },
        {
            "sourceId" : 2,
            "destinationId" : 3
        },
        {
            "sourceId" : 3,
            "destinationId" : 4
        }
}
```

Esempio 2: processi multipli con un trigger condizionale.

L'esempio seguente mostra un flusso di lavoro con processi multipli e un trigger condizionale (t3).

```
Consider Flow:
T(t1) ---> J(j1) ---> T(t2) ---> J(j2)
             |                    |
```

```
             |                     |
             >+------> T(t3) <-----+
                        |
                        |
                      J(j3)

Graph generated:
Nodes: t1, t2, t3, j1, j2, j3
Edges: (t1, j1), (j1, t2), (t2, j2), (j1, t3), (j2, t3), (t3, j3)
```

# Restrizioni dei flussi di lavoro e degli schemi in AWS Glue

Di seguito sono riportate le restrizioni imposte sui piani e sui flussi di lavoro.

## Restrizioni degli schemi

Tieni a mente le seguenti restrizioni relative al piano:

- Il blueprint deve essere registrato nella stessa AWS regione in cui risiede il bucket Amazon S3.
- Per condividere i blueprint tra AWS account, devi concedere le autorizzazioni di lettura sull'archivio ZIP del blueprint in Amazon S3. I clienti che dispongono dell'autorizzazione di lettura su un archivio ZIP del blueprint possono registrare il blueprint nel proprio account e utilizzarlo. AWS
- L'insieme di parametri del piano viene memorizzato come un singolo oggetto JSON. La lunghezza massima di questo oggetto è 128 KB.
- La dimensione massima non compressa dell'archivio ZIP del piano è 5 MB. La dimensione massima compressa è 1 MB.
- Limita il numero totale di processi, crawler e attivazioni all'interno di un flusso di lavoro a 100 o meno. Se includi più di 100, potresti riscontrare errori durante il tentativo di riprendere o interrompere l'esecuzione del flusso di lavoro.

## Restrizioni dei flussi di lavoro

Tieni a mente le seguenti restrizioni relative ai flussi di lavoro. Alcuni di questi commenti sono principalmente indirizzati a utenti che creano flussi di lavoro manualmente.

- La dimensione massima del batch per un trigger di EventBridge eventi Amazon è 100. La dimensione massima della finestra è di 900 secondi (15 minuti).
- Un trigger può essere associato a un solo flusso di lavoro.

- È permessa la configurazione di un solo trigger di attivazione (on demand o pianificato).
- Se un processo o un crawler in un flusso di lavoro viene avviato da un trigger esterno al flusso di lavoro, eventuali trigger all'interno del flusso di lavoro che dipendono dal completamento del processo o del crawler (completato o meno) non vengono attivati.
- Analogamente, se un processo o crawler in un flusso di lavoro dispone di trigger che dipendono dal completamento del processo o del crawler (completato o meno) sia all'interno del flusso di lavoro che all'esterno del flusso di lavoro e se il processo o il crawler viene avviato da un flusso di lavoro, solo i trigger all'interno del flusso di lavoro si attivano al completamento del processo o del crawler.

## Risoluzione degli errori relativi agli schemi in AWS Glue

Se riscontri errori durante l'utilizzo dei piani AWS Glue, usa le seguenti soluzioni per trovare l'origine dei problemi e correggerli.

Argomenti


- Errore: PySpark modulo mancante
- Errore: file di configurazione del piano mancante
- Errore: file importato mancante
- Errore: non autorizzato a iamPassRole eseguire sulla risorsa
- Errore: pianificazione cron non valida
- Errore: esiste già un trigger con questo nome
- Errore: un flusso di lavoro con nome “foo” esiste già.
- Errore: modulo non trovato nel percorso layoutGenerator specificato
- Errore: errore di convalida nel campo Connections (Connessioni)


### Errore: PySpark modulo mancante

AWS Gluerestituisce l'errore «Errore sconosciuto nell'esecuzione della funzione del generatore di layout ModuleNotFoundError: nessun modulo chiamato 'pyspark'».

Quando si decomprime l'archivio del piano, potrebbe verificarsi uno dei seguenti casi:

```
$ unzip compaction.zip
Archive:  compaction.zip
   creating: compaction/
  inflating: compaction/blueprint.cfg
```

```
  inflating: compaction/layout.py
  inflating: compaction/README.md
  inflating: compaction/compaction.py

$ unzip compaction.zip
Archive:  compaction.zip
  inflating: blueprint.cfg
  inflating: compaction.py
  inflating: layout.py
  inflating: README.md
```

Nel primo caso, tutti i file relativi al piano sono stati collocati in una cartella denominata compattazione convertita poi in un file zip denominato compaction.zip.

Nel secondo caso, tutti i file necessari per il piano non sono stati inclusi in una cartella e sono stati aggiunti come file root sotto il file zip compaction.zip.

È consentita la creazione di un file in uno dei formati sopra indicati. Accertati che `blueprint.cfg` abbia il percorso corretto al nome della funzione nello script che genera il layout.

Esempi

Nel caso 1: `blueprint.cfg` deve avere `layoutGenerator` come segue:

```
layoutGenerator": "compaction.layout.generate_layout"
```

Nel caso 2: `blueprint.cfg` deve avere `layoutGenerator` come segue

```
layoutGenerator": "layout.generate_layout"
```

Se questo percorso non è incluso correttamente, è possibile che venga visualizzato l'errore indicato. Ad esempio, se disponi della struttura di cartelle come indicato nel caso 2 e `layoutGenerator` come indicato come nel caso 1, potresti visualizzare l'errore di cui sopra.

## Errore: file di configurazione del piano mancante

AWS Gluerestituisce l'errore «Errore sconosciuto nell'esecuzione della funzione del generatore di layout FileNotFoundError: [Errno 2] Nessun file o directory di questo tipo: tmp/compaction/blueprint '/ .cfg'».

blueprint.cfg deve essere posizionato al livello root dell'archivio ZIP o all'interno di una cartella che ha lo stesso nome dell'archivio ZIP.

Quando si estrae l'archivio ZIP del piano, blueprint.cfg deve essere trovato in uno dei seguenti percorsi. Se non viene trovato in uno dei seguenti percorsi, potresti visualizzare l'errore di cui sopra.

```
$ unzip compaction.zip
Archive:  compaction.zip
   creating: compaction/
  inflating: compaction/blueprint.cfg

$ unzip compaction.zip
Archive:  compaction.zip
  inflating: blueprint.cfg
```

## Errore: file importato mancante

AWS Gluerestituisce l'errore «Errore sconosciuto nell'esecuzione della funzione del generatore di layout FileNotFoundError: [Errno 2] Nessun file o directory di questo tipo: * *'demo-project/foo.py'».

Se lo script di generazione del layout dispone di funzionalità per leggere altri file, accertati di fornire un percorso completo per il file da importare. Ad esempio, Layout.py potrebbe fare riferimento allo script Conversion.py. Per ulteriori informazioni, consulta Progetto di schema di esempio.

## Errore: non autorizzato a iamPassRole eseguire sulla risorsa

AWS Gluerestituisce l'errore «User: arn:aws:sts: :123456789012:assumed-» role/ AWSGlueServiceRole/GlueSession is not authorized to perform: iam:PassRole on resource: arn:aws:iam::123456789012:role/AWSGlueServiceRole

Se i processi e i crawler nel flusso di lavoro assumono lo stesso ruolo del ruolo passato per creare il flusso di lavoro dal piano, il ruolo del piano deve includere l'autorizzazione `iam:PassRole` su se stesso.

Se i processi e i crawler nel flusso di lavoro assumono un ruolo diverso da quello passato per creare le entità del flusso di lavoro dal piano, il ruolo del piano deve includere l'autorizzazione `iam:PassRole` sull'altro ruolo invece che sul ruolo del piano.

Per ulteriori informazioni, consulta Autorizzazioni per i ruoli degli schemi.

## Errore: pianificazione cron non valida

AWS Glue restituisce l'errore "The schedule cron(0 0 * * * *) is invalid." (La pianificazione cron(0 0 * * * *) non è valida)

Fornisci un'espressione cron valida. Per ulteriori informazioni, consulta Pianificazioni basate sul tempo per processi e crawler.

## Errore: esiste già un trigger con questo nome

AWS Glue restituisce l'errore "Trigger with name 'foo_starting_trigger' already submitted with different configuration" (Trigger con nome foo_starting_trigger' già inviato con configurazione diversa).

Un piano non richiede la definizione dei trigger nello script di layout per la creazione del flusso di lavoro. La creazione dei trigger viene gestita dalla libreria del piano in base alle dipendenze definite tra due operazioni.

La denominazione per i trigger è la seguente:

- Per il trigger iniziale nel flusso di lavoro, la denominazione è <workflow_name>_starting_trigger.
- Per un nodo (processo/crawler) nel flusso di lavoro che dipende dal completamento di uno o più nodi a monte; AWS Glue definisce un trigger con il nome <workflow_name>_<node_name>_trigger

Questo errore indica che esiste già un trigger con lo stesso nome. È possibile eliminare il trigger esistente ed eseguire nuovamente la creazione del flusso di lavoro.

> **Note**
>
> L'eliminazione di un flusso di lavoro non comporta l'eliminazione dei nodi all'interno del flusso di lavoro. È possibile che i trigger vengano lasciati anche se il flusso di lavoro viene eliminato. Per questo motivo, nel caso in cui crei un flusso di lavoro, lo elimini e quindi provi a ricrearlo con lo stesso nome dallo stesso piano, potresti non ricevere un errore che indica che il flusso di lavoro esiste già, ma potresti ricevere un errore che indica che il trigger esiste già.

## Errore: un flusso di lavoro con nome “foo” esiste già.

Il nome del flusso di lavoro deve essere univoco. Prova con un nome diverso.

## Errore: modulo non trovato nel percorso layoutGenerator specificato

AWS Gluerestituisce l'errore «Errore sconosciuto nell'esecuzione della funzione del generatore di layout: nessun modulo denominato 'crawl_s3_locations'». ModuleNotFoundError

```
layoutGenerator": "crawl_s3_locations.layout.generate_layout"
```

Ad esempio, disponi del percorso LayoutGenerator di cui sopra, quando decomprimi l'archivio del piano, questo deve essere simile al seguente:

```
$ unzip crawl_s3_locations.zip
Archive:  crawl_s3_locations.zip
   creating: crawl_s3_locations/
  inflating: crawl_s3_locations/blueprint.cfg
  inflating: crawl_s3_locations/layout.py
  inflating: crawl_s3_locations/README.md
```

Quando decomprimi l'archivio, se l'archivio del piano è simile al seguente, potresti visualizzare l'errore precedente.

```
$ unzip crawl_s3_locations.zip
Archive:  crawl_s3_locations.zip
  inflating: blueprint.cfg
  inflating: layout.py
  inflating: README.md
```

Non esiste una cartella denominata `crawl_s3_locations` e quando il percorso `layoutGenerator` fa riferimento al file di layout tramite il modulo `crawl_s3_locations`, potresti visualizzare l'errore di cui sopra.

## Errore: errore di convalida nel campo Connections (Connessioni)

AWS Glue<class 'dict'>restituisce l'errore «Errore sconosciuto nell'esecuzione della funzione del generatore di layout TypeError: il valore ['foo'] per le connessioni chiave dovrebbe essere di tipo!».

Si tratta di un errore di convalida. Il campo `Connections` nella classe `Job` si aspetta un dizionario e invece viene fornito un elenco di valori che causa l'errore.

```
User input was list of values
Connections= ['string']

Should be a dict like the following
Connections*=*{'Connections': ['string']}
```

Per evitare questi errori di runtime durante la creazione di un flusso di lavoro da uno schema, puoi convalidare le definizioni del flusso di lavoro, del processo e del crawler come descritto in Test di uno schema.

Fai riferimento alla sintassi in Riferimento alle classi di schemaAWS Glue per definire processo, crawler e flusso di lavoro AWS Glue nello script di layout.

# Autorizzazioni per utenti e ruoli per gli schemi AWS Glue

Di seguito sono riportati i personaggi tipici e le politiche di autorizzazione suggerite AWS Identity and Access Management (IAM) per i personaggi e i ruoli per i blueprint. AWS Glue

Argomenti


- Utenti dello schema
- Autorizzazioni per gli utenti dei piani
- Autorizzazioni per i ruoli degli schemi


## Utenti dello schema

Di seguito sono riportati i utenti generalmente coinvolti nel ciclo di vita dei piani AWS Glue.

| Utente | Description |
|---|---|
| Sviluppatore AWS Glue | Sviluppa, verifica e pubblica progetti. |
| Amministratore di AWS Glue | Registra, mantiene e concede le autorizzazioni per i piani. |
| Analista dei dati | Esegue i piani per creare flussi di lavoro. |

Per ulteriori informazioni, consulta the section called "Panoramica degli schemi".

## Autorizzazioni per gli utenti dei piani

Di seguito sono riportate le autorizzazioni suggerite per ogni utente del piano.

### Autorizzazioni per i progetti per lo sviluppatore di AWS Glue per gli schemi

Lo sviluppatore di AWS Glue deve disporre delle autorizzazioni di scrittura sul bucket Amazon S3 utilizzato per pubblicare il progetto. Spesso, lo sviluppatore registra il piano dopo averlo caricato. In

tal caso, lo sviluppatore necessita delle autorizzazioni elencate in the section called "Autorizzazioni per i progetti per l'amministratore di AWS Glue per gli schemi". Inoltre, se lo sviluppatore desidera testare il piano dopo la registrazione, ha bisogno anche delle autorizzazioni elencate in the section called "Autorizzazioni per gli schemi per l'analista dati".

Autorizzazioni per i progetti per l'amministratore di AWS Glue per gli schemi

La policy seguente concede le autorizzazioni per la registrazione, la visualizzazione e la gestione dei progetti AWS Glue.

> **Important**
>
> Nella seguente politica, sostituisci *<s3-bucket-name>* e *<prefix>* con il percorso Amazon S3 per caricare gli archivi ZIP del blueprint da registrare.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:CreateBlueprint",
        "glue:UpdateBlueprint",
        "glue:DeleteBlueprint",
        "glue:GetBlueprint",
        "glue:ListBlueprints",
        "glue:BatchGetBlueprints"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/prefix/*"
    }
  ]
```

```
}
```

Autorizzazioni per gli schemi per l'analista dati

Il criterio seguente concede le autorizzazioni per eseguire i piani e per visualizzare il flusso di lavoro e i relativi i componenti risultanti. Concede inoltre `PassRole` al ruolo che AWS Glue assume per creare il flusso di lavoro e i relativi componenti.

La policy concede le autorizzazioni su qualsiasi risorsa. Se desideri configurare un accesso granulare ai singoli blueprint, utilizza il seguente formato per il blueprint: ARNs

```
arn:aws:glue:<region>:<account-id>:blueprint/<blueprint-name>
```

> **Important**
>
> Nella seguente politica, sostituiscilo *<account-id>* con un AWS account valido e sostituiscilo *<role-name>* con il nome del ruolo usato per eseguire un blueprint. Consulta the section called "Autorizzazioni per i ruoli degli schemi" per le autorizzazioni richieste da questo ruolo.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:ListBlueprints",
        "glue:GetBlueprint",
        "glue:StartBlueprintRun",
        "glue:GetBlueprintRun",
        "glue:GetBlueprintRuns",
        "glue:GetCrawler",
        "glue:ListTriggers",
        "glue:ListJobs",
        "glue:BatchGetCrawlers",
        "glue:GetTrigger",
```

```
        "glue:BatchGetWorkflows",
        "glue:BatchGetTriggers",
        "glue:BatchGetJobs",
        "glue:BatchGetBlueprints",
        "glue:GetWorkflowRun",
        "glue:GetWorkflowRuns",
        "glue:ListCrawlers",
        "glue:ListWorkflows",
        "glue:GetJob",
        "glue:GetWorkflow",
        "glue:StartWorkflowRun"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::111122223333:role/role-name"
    }
  ]
}
```

## Autorizzazioni per i ruoli degli schemi

Di seguito sono riportate le autorizzazioni suggerite per il ruolo IAM utilizzato per creare un flusso di lavoro da un piano. Il ruolo deve avere una relazione di trust con `glue.amazonaws.com`.

> **Important**
>
> Nella seguente politica, sostituiscilo *<account-id>* con un AWS account valido e *<role-name>* sostituiscilo con il nome del ruolo.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
```

```
      "Action": [
        "glue:CreateJob",
        "glue:GetCrawler",
        "glue:GetTrigger",
        "glue:DeleteCrawler",
        "glue:CreateTrigger",
        "glue:DeleteTrigger",
        "glue:DeleteJob",
        "glue:CreateWorkflow",
        "glue:DeleteWorkflow",
        "glue:GetJob",
        "glue:GetWorkflow",
        "glue:CreateCrawler"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::111122223333:role/role-name"
    }
  ]
}
```

> **Note**
>
> Se i processi e i crawler nel flusso di lavoro assumono un ruolo diverso da questo, questa policy deve includere l'autorizzazione `iam:PassRole` su quel ruolo invece che sul ruolo del piano.

# Sviluppo di schemi in AWS Glue

L'organizzazione potrebbe avere un set di casi di utilizzo ETL simili che potrebbero trarre vantaggio dalla possibilità di definire i parametri di un singolo flusso di lavoro in grado di gestirli tutti. A questo scopo, AWS Glue ti consente di definire dei progetti, che puoi utilizzare per generare i flussi di lavoro. Un progetto accetta i parametri, in modo che da un singolo progetto, un analista di dati possa creare diversi flussi di lavoro per gestire casi di utilizzo ETL simili. Una volta creato un progetto, puoi riutilizzarlo per reparti, team e progetti diversi.

Argomenti


- Panoramica degli schemi in AWS Glue
- Sviluppo di schemi in AWS Glue
- Registrazione di uno schema in AWS Glue
- Visualizzazione degli schemi in AWS Glue
- Aggiornamento di uno schema in AWS Glue
- Creazione di un flusso di lavoro da uno schema in AWS Glue
- Visualizzazione delle esecuzioni dello schema in AWS Glue


## Panoramica degli schemi in AWS Glue

> **Note**
>
> La funzionalità blueprints non è attualmente disponibile nelle seguenti regioni della console AWS Glue: Asia Pacifico (Giacarta) e Medio Oriente (Emirati Arabi Uniti).

I progetti AWS Glue offrono un modo per creare e condividere i flussi di lavoro AWS Glue. Quando esiste un processo ETL complesso che potrebbe essere utilizzato per casi d'uso simili, piuttosto che creare un flusso di lavoro AWS Glue per ogni caso d'uso, è possibile creare un singolo progetto.

Il piano specifica i processi e i crawler da includere in un flusso di lavoro e specifica i parametri che l'utente fornisce quando esegue il piano per creare un flusso di lavoro. L'uso di parametri consente a un singolo piano di generare flussi di lavoro per vari casi d'uso simili. Per ulteriori informazioni sui flussi di lavoro, consulta Panoramica di flussi di lavoro in AWS Glue.

Di seguito sono riportati esempi di casi d'uso per i piani:

- Vuoi partizionare un set di dati esistente. I parametri di input del piano sono i percorsi di origine e di destinazione Amazon Simple Storage Service (Amazon S3) e un elenco di colonne di partizione.
- Vuoi creare uno snapshot di una tabella Amazon DynamoDB in un archivio dati SQL come Amazon Redshift. I parametri di input per il progetto sono il nome della tabella DynamoDB e una connessione AWS Glue, che indica un cluster Amazon Redshift e un database di destinazione.
- Vuoi convertire i dati CSV in più percorsi Amazon S3 in Parquet. È consigliabile che il flusso di lavoro AWS Glue includa un crawler e un processo separati per ogni percorso. I parametri di input sono il database di destinazione in AWS Glue Data Catalog e un elenco di percorsi Amazon S3

delimitati da virgola. In questo caso, il numero di crawler e processi creati dal flusso di lavoro è variabile.

### Componenti dello schema

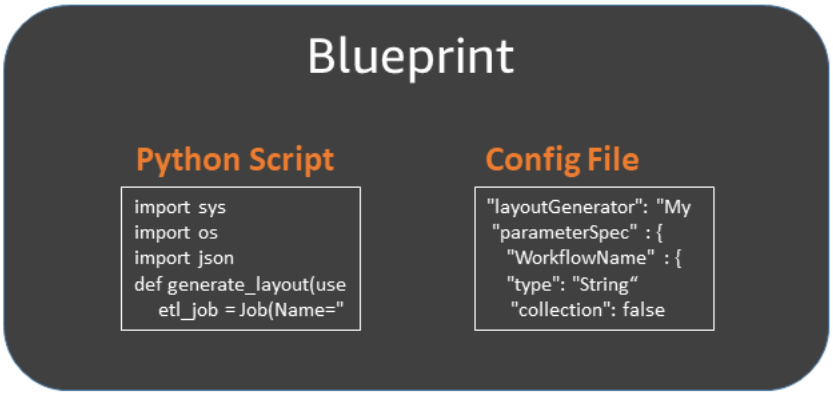
Un piano è un archivio ZIP contenente i seguenti componenti:

- Uno script generatore di layout Python

  Contiene una funzione che specifica il layout del flusso di lavoro: i crawler e i processi da creare per il flusso di lavoro, le proprietà del processo e del crawler e le dipendenze tra i processi e i crawler. La funzione accetta i parametri di progetto e restituisce una struttura del flusso di lavoro (oggetto JSON) che AWS Glue utilizza per generare il flusso di lavoro. Utilizzando uno script Python per generare il flusso di lavoro, puoi aggiungere la logica adatta ai tuoi casi d'uso.
- Un file di configurazione

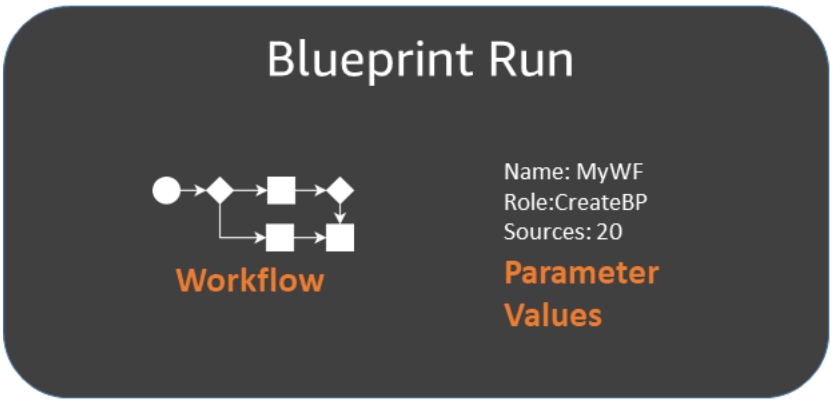
  Specifica il nome completo della funzione Python che genera il layout del flusso di lavoro. Specifica inoltre i nomi, i tipi di dati e le altre proprietà di tutti i parametri del piano utilizzati dallo script.
- (Facoltativo) Script ETL e file di supporto

  Come caso d'uso avanzato, è possibile definire i parametri della posizione degli script ETL utilizzati dai processi. Puoi includere i file di script di processo nell'archivio ZIP e specificare un parametro del piano per una posizione Amazon S3 in cui gli script devono essere copiati. Lo script generatore di layout può copiare gli script ETL nella posizione indicata e specificare tale posizione come proprietà della posizione dello script di processo. È inoltre possibile includere qualsiasi libreria o altri file di supporto, a condizione che lo script li gestisca.




### Esecuzioni del piano

Quando crei un flusso di lavoro da un progetto, AWS Glue lo esegue, il che avvia un processo asincrono per creare il flusso di lavoro e i processi, i crawler e i trigger incapsulati dal flusso di lavoro. AWS Glue usa l'esecuzione del progetto per orchestrare la creazione del flusso di lavoro e dei suoi componenti. Puoi vedere lo stato del processo di creazione attraverso lo stato di esecuzione del piano. L'esecuzione del piano memorizza anche i valori forniti per i parametri del piano.




È possibile visualizzare le esecuzioni dei blueprint utilizzando la AWS Glue console o (). AWS Command Line Interface AWS CLI Durante la visualizzazione o la risoluzione dei problemi di un flusso di lavoro, puoi sempre tornare all'esecuzione del piano per visualizzare i valori dei parametri del piano utilizzati per creare il flusso di lavoro.

Ciclo di vita di uno schema

Gli schemi sono sviluppati, testati, registrati con AWS Glue ed eseguiti per creare flussi di lavoro. In genere tre utenti sono coinvolti nel ciclo di vita del piano.

| Utente | Processi |
| --- | --- |
| Sviluppatore AWS Glue | • Scrive lo script del layout del flusso di lavoro e crea il file di configurazione.<br>• Testa il progetto in locale utilizzando le librerie fornite dal servizio AWS Glue.<br>• Crea un archivio ZIP dello script, del file di configura zione e dei file di supporto e pubblica l'archivio in una posizione in Amazon S3.<br>• Aggiunge una policy bucket al bucket Amazon S3 che concede le autorizzazioni di lettura sugli oggetti bucket all'account dell'amministratore. AWS Glue AWS |

| Utente | Processi |
| --- | --- |
| | • Concede le autorizzazioni di lettura IAM per l'archivio ZIP in Amazon S3 all'amministratore AWS Glue. |
| Amministratore di AWS Glue | • Registra il progetto con AWS Glue. AWS Glue crea una copia dell'archivio ZIP in una posizione Amazon S3 riservata.<br>• Concede le autorizzazioni IAM per il piano agli analisti dei dati. |
| Analista dei dati | • Esegue il piano per creare un flusso di lavoro e fornisce i valori dei parametri del piano. Controlla lo stato di esecuzione del piano per assicurarsi che il flusso di lavoro e i relativi componenti siano stati generati correttamente.<br>• Esegue e risolve i problemi relativi al flusso di lavoro. Prima di eseguire il flusso di lavoro, può verificarlo visualizzando il grafico di progettazione del flusso di lavoro nella console AWS Glue. |

Consulta anche

- Sviluppo di schemi in AWS Glue
- Creazione di un flusso di lavoro da uno schema in AWS Glue
- Autorizzazioni per utenti e ruoli per gli schemi AWS Glue

## Sviluppo di schemi in AWS Glue

In qualità di sviluppatore AWS Glue, puoi creare e pubblicare progetti che gli analisti di dati potranno utilizzare per generare flussi di lavoro.

Argomenti

- Panoramica sullo sviluppo di schemi
- Prerequisiti per lo sviluppo degli schemi

- Scrittura del codice dello schema
- Progetto di schema di esempio
- Test di uno schema
- Pubblicazione di uno schema
- Riferimento alle classi di schema AWS Glue
- Esempi di schema

Consulta anche

- Panoramica degli schemi in AWS Glue

## Panoramica sullo sviluppo di schemi

Il primo passo del processo di sviluppo consiste nell'identificare un caso d'uso comune che possa trarre vantaggio da un piano. Un tipico caso d'uso comporta un problema ETL ricorrente che ritieni debba essere risolto in modo generale. Quindi, progetta un piano che implementi il caso d'uso generalizzato e definisci i parametri di input del piano che insieme possono definire un caso d'uso specifico a partire dal caso d'uso generalizzato.

Un piano è costituito da un progetto che contiene un file di configurazione dei parametri del piano e uno script che definisce la proprietà di layout del flusso di lavoro da generare. Il layout definisce i processi e i crawler (o entità, nella terminologia dello script del piano) da creare.

Non è possibile specificare direttamente alcun trigger nello script di layout. È invece possibile scrivere codice per specificare le dipendenze tra i processi e i crawler creati dallo script. AWS Glue genera le attivazioni basate sulle specifiche delle dipendenze. L'output dello script di layout è un oggetto di flusso di lavoro che contiene le specifiche per tutte le entità del flusso di lavoro.

È possibile costruire l'oggetto flusso di lavoro utilizzando le seguenti librerie dei progetti AWS Glue:

- `awsglue.blueprint.base_resource`— Una libreria di risorse di base utilizzate dalle librerie.
- `awsglue.blueprint.workflow`— Una libreria per definire una classe di `Workflow`.
- `awsglue.blueprint.job`— Una libreria per definire una classe di `Job`.
- `awsglue.blueprint.crawler`— Una libreria per definire una classe di `Crawler`.

Le uniche altre librerie supportate per la generazione del layout sono quelle disponibili per la shell Python.

Prima di pubblicare il piano, è possibile utilizzare i metodi definiti nelle librerie dei piani per testarlo localmente.

Quando si è pronti a rendere il piano disponibile agli analisti dei dati, è possibile creare un pacchetto dello script, del file di configurazione dei parametri e di tutti i file di supporto, ad esempio script e librerie aggiuntivi, in un'unica risorsa distribuibile. Quindi si carica la risorsa su Amazon S3 e si chiede a un amministratore di registrarla su AWS Glue.

Per ulteriori informazioni su altri piani di esempio, consulta Progetto di schema di esempio e Esempi di schema.

## Prerequisiti per lo sviluppo degli schemi

Per sviluppare i progetti, è bene avere familiarità con AWS Glue e con la scrittura di script per i processi Apache Spark ETL o di shell Python. È inoltre necessario completare le seguenti attività di configurazione.

- Scarica quattro librerie AWS Python da usare negli script di layout dei tuoi blueprint.
- Configura il. AWS SDKs
- Configura il AWS CLI.

### Scaricare le librerie Python

Scarica le seguenti librerie da GitHub e installale nel tuo progetto:

- https://github.com/awslabs/aws-glue-blueprint-libs/tree/master/awsglue/blueprint/base_resource.py
- https://github.com/awslabs/aws-glue-blueprint-libs/tree/master/awsglue/blueprint/workflow.py
- https://github.com/awslabs/aws-glue-blueprint-libs/.py tree/master/awsglue/blueprint/crawler
- https://github.com/awslabs/aws-glue-blueprint-libs/.py tree/master/awsglue/blueprint/job

### Configura AWS Java SDK

Per AWS Java SDK, è necessario aggiungere un `jar` file che includa l'API per i blueprint.

1. Se non l'hai già fatto, configura l' AWS SDK for Java.

- Per Java 1.x, segui le istruzioni in Impostare AWS SDK per Java nella Guida per gli sviluppatori di AWS SDK per Java .
- Per Java 2.x, segui le istruzioni in Configurazione di AWS SDK for Java 2.x nella AWS SDK for Java 2.x Guida per gli sviluppatori di .

2. Scarica il `jar` file client che ha accesso ai progetti APIs for.

   - Per Java 1.x: s3://- -1.11.x.jar awsglue-custom-blueprints-preview artifacts/awsglue-java-sdk-preview/AWSGlueJavaClient
   - Per Java 2.x: s3://awsglue-custom-blueprints-preview- -Glue-2.0.jar artifacts/awsglue-java-sdk-v2-preview/AwsJavaSdk

3. Aggiungi il client `jar` all'inizio del classpath Java per sovrascrivere il client AWS Glue fornito da Java SDK. AWS

   ```
   export CLASSPATH=<path-to-preview-client-jar>:$CLASSPATH
   ```

4. (Facoltativo) Testa l'SDK con la seguente applicazione Java. L'applicazione dovrebbe produrre un elenco vuoto.

   Sostituisci `accessKey` e `secretKey` con le tue credenziali e sostituisci `us-east-1` con la tua regione.

   ```
   import com.amazonaws.auth.AWSCredentials;
   import com.amazonaws.auth.AWSCredentialsProvider;
   import com.amazonaws.auth.AWSStaticCredentialsProvider;
   import com.amazonaws.auth.BasicAWSCredentials;
   import com.amazonaws.services.glue.AWSGlue;
   import com.amazonaws.services.glue.AWSGlueClientBuilder;
   import com.amazonaws.services.glue.model.ListBlueprintsRequest;

   public class App{
       public static void main(String[] args) {
           AWSCredentials credentials = new BasicAWSCredentials("accessKey",
    "secretKey");
           AWSCredentialsProvider provider = new
    AWSStaticCredentialsProvider(credentials);
           AWSGlue glue = AWSGlueClientBuilder.standard().withCredentials(provider)
                   .withRegion("us-east-1").build();
           ListBlueprintsRequest request = new
    ListBlueprintsRequest().withMaxResults(2);
   ```

```
        System.out.println(glue.listBlueprints(request));
    }
}
```

Configura l' AWS SDK Python

I passaggi seguenti presuppongono che sul computer sia installato Python versione 2.7 o successiva oppure versione 3.9 o successiva.

1. Scarica il seguente file boto3 wheel. Se viene richiesto di aprire o salvare, salvate il file. s3://-3-1.17.31-py2.py3-none-any.whl awsglue-custom-blueprints-preview artifacts/aws-python-sdk-preview/boto
2. Scaricate il seguente file awsglue-custom-blueprints-preview botocore wheel: artifacts/aws-python-sdk-preview/botocore s3://- -1.20.31-py2.py3-none-any.whl
3. Controlla la tua versione Python.

   ```
   python --version
   ```

4. A seconda della versione di Python, immetti seguenti comandi (per Linux):

   - Per Python 2.7 o versioni successive.

     ```
     python3 -m pip install --user virtualenv
     source env/bin/activate
     ```

   - Per Python 3.9 o successivo.

     ```
     python3 -m venv python-sdk-test
     source python-sdk-test/bin/activate
     ```

5. Installa il file botocore wheel.

   ```
   python3 -m pip install <download-directory>/botocore-1.20.31-py2.py3-none-any.whl
   ```

6. Installa il file boto3 wheel.

   ```
   python3 -m pip install <download-directory>/boto3-1.17.31-py2.py3-none-any.whl
   ```

7. Configura le credenziali e la regione predefinita nei file `~/.aws/credentials` e `~/.aws/config`. Per ulteriori informazioni, consulta Configurazione della AWS CLI nella Guida per l'utente di AWS Command Line Interface .
8. (Facoltativo) Esegui il test della configurazione. I seguenti comandi devono restituire un elenco vuoto.

   Sostituisci `us-east-1` con la tua regione.

   ```
   $ python
   >>> import boto3
   >>> glue = boto3.client('glue', 'us-east-1')
   >>> glue.list_blueprints()
   ```

Imposta l'anteprima AWS CLI

1. Se non l'hai già fatto, installa l' and/or aggiornamento AWS Command Line Interface (AWS CLI) sul tuo computer. Il modo più semplice per eseguire questa operazione è utilizzare `pip`, l'utility di installazione Python:

   ```
   pip install awscli --upgrade --user
   ```

   Puoi trovare le istruzioni di installazione complete per AWS CLI qui: Installazione di AWS Command Line Interface.
2. Scarica il file AWS CLI wheel da: s3://awsglue-custom-blueprints-preview- artifacts/awscli-preview-build/awscli -1.19.31-py2.py3-none-any.whl
3. Installa il file AWS CLI wheel.

   ```
   python3 -m pip install awscli-1.19.31-py2.py3-none-any.whl
   ```
4. Esegui il comando `aws configure`. Configura AWS le tue credenziali (inclusa la chiave di accesso e la chiave segreta) e la AWS regione. Puoi trovare informazioni sulla configurazione AWS CLI qui: Configurazione di. AWS CLI
5. Prova il. AWS CLI Il seguente comando dovrebbe restituire un elenco vuoto.

   Sostituisci `us-east-1` con la tua regione.

   ```
   aws glue list-blueprints --region us-east-1
   ```

## Scrittura del codice dello schema

Ogni piano creato deve contenere almeno i seguenti file:

- Uno script di layout Python che definisce il flusso di lavoro. Lo script contiene una funzione che definisce le entità (processi e crawler) in un flusso di lavoro e le dipendenze tra di esse.
- Un file di configurazione, `blueprint.cfg`, che definisce:
  - Il percorso completo della funzione di definizione del layout del flusso di lavoro.
  - I parametri accettati dal piano.

Argomenti


- Creazione dello script di layout dello schema
- Creare il file di configurazione
- Specifica dei parametri dello schema


### Creazione dello script di layout dello schema

Lo script di layout del piano deve includere una funzione che genera le entità nel flusso di lavoro. Puoi nominare questa funzione come preferisci. AWS Glue utilizza il file di configurazione per determinare il nome completo della funzione.

La funzione di layout svolge le operazioni seguenti:

- (Facoltativo) Crea un'istanza della classe di `Job` per creare oggetti `Job` e passa argomenti come `Command` e `Role`. Si tratta di proprietà del processo da specificare se lo stai creando tramite la console o l'API AWS Glue.
- (Facoltativo) Crea un'istanza della classe di `Crawler` per creare oggetti `Crawler` e passa argomenti come il nome, il ruolo e la destinazione.
- Per indicare le dipendenze tra gli oggetti (entità del flusso di lavoro), passa gli argomenti aggiuntivi `DependsOn` e `WaitForDependencies` al `Job()` e al `Crawler()`. Questi argomenti sono descritti più avanti in questa sezione.
- Crea un'istanza della classe di `Workflow` per creare l'oggetto di flusso di lavoro che viene restituito ad AWS Glue, passando un argomento `Name`, un argomento `Entities` e un argomento facoltativo `OnSchedule`. L'argomento `Entities` specifica tutti i processi e i crawler da includere nel flusso di lavoro. Per sapere come costruire un oggetto `Entities`, vedi il progetto di esempio più avanti in questa sezione.

- Restituisce l'oggetto `Workflow`.

Per le definizioni delle classi `Job`, `Crawler` e`Workflow`, consulta Riferimento alle classi di schema AWS Glue.

La funzione di layout di deve accettare i seguenti argomenti.

| Argomento | Description |
| --- | --- |
| `user_params` | Dizionario Python di nomi e valori dei parametri del piano. Per ulteriori informazioni, consulta Specifica dei parametri dello schema. |
| `system_params` | Dizionario Python contenente due proprietà: `region` e `accountId` . |

Ecco uno script generatore di layout di esempio in un file chiamato `Layout.py`:

```
import argparse
import sys
import os
import json
from awsglue.blueprint.workflow import *
from awsglue.blueprint.job import *
from awsglue.blueprint.crawler import *


def generate_layout(user_params, system_params):

    etl_job = Job(Name="{}_etl_job".format(user_params['WorkflowName']),
                  Command={
                      "Name": "glueetl",
                      "ScriptLocation": user_params['ScriptLocation'],
                      "PythonVersion": "2"
                  },
                  Role=user_params['PassRole'])
    post_process_job = Job(Name="{}_post_process".format(user_params['WorkflowName']),
                           Command={
                               "Name": "pythonshell",
                               "ScriptLocation": user_params['ScriptLocation'],
                               "PythonVersion": "2"
                           },
                           Role=user_params['PassRole'],
```

```
                            DependsOn={
                                etl_job: "SUCCEEDED"
                            },
                            WaitForDependencies="AND")
    sample_workflow = Workflow(Name=user_params['WorkflowName'],
                            Entities=Entities(Jobs=[etl_job, post_process_job]))
    return sample_workflow
```

Lo script di esempio importa le librerie del piano richieste e include un `generate_layout` che genera un flusso di lavoro con due processi. Si tratta di uno script molto semplice. Uno script più complesso potrebbe impiegare logica e parametri aggiuntivi per generare un flusso di lavoro con molti processi e crawler, o anche un numero variabile di processi e crawler.

Utilizzo dell' DependsOn argomento

La'argomento `DependsOn` è una rappresentazione dizionario di una dipendenza di questa entità su altre entità all'interno del flusso di lavoro. Presenta il seguente formato.

```
DependsOn = {dependency1 : state, dependency2 : state, ...}
```

Le chiavi in questo dizionario rappresentano il riferimento all'oggetto dell'entità (non il nome), mentre i valori sono stringhe che corrispondono allo stato da controllare. AWS Glue deduce le attivazioni appropriate. Per gli stati validi, consulta Struttura Condition.

Ad esempio, un processo potrebbe dipendere dal completamento corretto di un crawler. Se definisci un oggetto crawler denominato `crawler2` come segue:

```
crawler2 = Crawler(Name="my_crawler", ...)
```

Allora un oggetto dipendente da `crawler2` includerebbe un argomento del costruttore come:

```
DependsOn = {crawler2 : "SUCCEEDED"}
```

Ad esempio:

```
job1 = Job(Name="Job1", ..., DependsOn = {crawler2 : "SUCCEEDED", ...})
```

Se `DependsOn` viene omesso per un'entità, tale entità dipende dal trigger di avvio del flusso di lavoro.

Usando l' WaitForDependencies argomento

L'argomento `WaitForDependencies` definisce se un processo o un'entità crawler deve attendere fino a che tutte le entità da cui dipende sono complete o fino a quando è completa una qualsiasi.

I valori consentiti sono "AND" o "ANY".

Usando l' OnSchedule argomento

L'argomento `OnSchedule` per il costruttore della classe `Workflow` è un'espressione `cron` che indica la definizione del trigger iniziale per un flusso di lavoro.

Se questo argomento è specificato, AWS Glue crea un'attivazione di pianificazione con la pianificazione corrispondente. Se non è specificato, il trigger di attivazione del flusso di lavoro è un trigger on demand.

Creare il file di configurazione

Il file di configurazione del piano è un file obbligatorio che definisce il punto di ingresso dello script per la generazione del flusso di lavoro e i parametri accettati dal piano. Il deve essere denominato `blueprint.cfg`.

Segue una configurazione di esempio.

```
{
    "layoutGenerator": "DemoBlueprintProject.Layout.generate_layout",
    "parameterSpec" : {
           "WorkflowName" : {
                "type": "String",
                "collection": false
           },
           "WorkerType" : {
                "type": "String",
                "collection": false,
                "allowedValues": ["G1.X", "G2.X"],
                "defaultValue": "G1.X"
           },
           "Dpu" : {
                "type" : "Integer",
                "allowedValues" : [2, 4, 6],
                "defaultValue" : 2
           },
           "DynamoDBTableName": {
                "type": "String",
```

```
                "collection" : false
            },
            "ScriptLocation" : {
                "type": "String",
                "collection": false
     }
    }
}
```

La proprietà `layoutGenerator` specifica il nome completo della funzione nello script che genera il layout.

La proprietà `parameterSpec` specifica i parametri accettati da questo piano. Per ulteriori informazioni, consulta Specifica dei parametri dello schema.

> **Important**
>
> Il file di configurazione deve includere il nome del flusso di lavoro come parametro del piano oppure è necessario generare un nome di flusso di lavoro univoco nello script di layout.

### Specifica dei parametri dello schema

Il file di configurazione contiene le specifiche dei parametri del piano in un oggetto JSON `parameterSpec`. `parameterSpec` contiene uno o più oggetti parametro.

```
"parameterSpec": {
    "<parameter_name>": {
      "type": "<parameter-type>",
      "collection": true|false,
      "description": "<parameter-description>",
      "defaultValue": "<default value for the parameter if value not specified>"
      "allowedValues": "<list of allowed values>"
    },
    "<parameter_name>": {
       ...
    }
  }
```

Di seguito sono riportate le regole per la codifica di ogni oggetto parametro:

- Il nome e il `type` del parametro sono obbligatori. Tutte le altre proprietà sono facoltative.

- Se si specifica la proprietà `defaultValue`, il parametro è facoltativo. In caso contrario, il parametro è obbligatorio e l'analista dei dati che sta creando un flusso di lavoro dal piano deve fornire un valore per esso.
- Se si imposta la proprietà `collection` su `true`, il parametro può assumere un insieme di valori. Le raccolte possono essere di qualsiasi tipo di dati.
- Se specifichi `allowedValues`, la console AWS Glue mostra un elenco a discesa di valori tra cui l'analista dei dati può scegliere durante la creazione di un flusso di lavoro dal progetto.

Di seguito sono elencati i valori consentiti per `type`:

| Tipo di dati dei parametri | Note |
|---|---|
| `String` | - |
| `Integer` | - |
| `Double` | - |
| `Boolean` | I valori possibili sono `true` e `false`. Genera una casella di controllo nella pagina Create a workflow from <blueprint> (Crea un flusso di lavoro da <blueprint>), sulla console AWS Glue. |
| `S3Uri` | Completa il percorso Amazon S3, iniziando con `s3://`. Genera un campo di testo e un pulsante Browse (Sfoglia) nella pagina Create a workflow from <blueprint> (Crea un flusso di lavoro da <blueprint>). |
| `S3Bucket` | Solo il nome del bucket Amazon S3. Genera un selettore bucket nella scheda Create a workflow from <blueprint> (Crea un flusso di lavoro da <blueprint>). |
| `IAMRoleArn` | Amazon Resource Name (ARN) di un ruolo AWS Identity and Access Management (IAM). Genera un selettore ruolo nella pagina Create a workflow from <blueprint> (Crea un flusso di lavoro da <blueprint>). |
| `IAMRoleName` | Nome di un ruolo IAM. Genera un selettore ruolo nella pagina Create a workflow from <blueprint> (Crea un flusso di lavoro da <blueprint>). |

## Progetto di schema di esempio

La conversione del formato dei dati è un caso d'uso frequente di estrazione, trasformazione e caricamento (ETL). Nei carichi di lavoro analitici tipici, i formati di file basati su colonne come Parquet o ORC sono preferiti rispetto ai formati di testo come CSV o JSON. Questo modello di esempio consente di convertire i dati da.in Parquet per i file su Amazon S3CSV/JSON/etc.

Questo piano accetta un elenco di percorsi S3 definiti da un parametro del piano, converte i dati in formato Parquet e li scrive nella posizione S3 specificata da un altro parametro del piano. Lo script di layout crea un crawler e un processo per ogni percorso. Lo script di layout carica anche lo script ETL in `Conversion.py` in un bucket S3 specificato da un altro parametro del piano. Lo script di layout specifica quindi lo script caricato come script ETL per ogni processo. L'archivio ZIP del progetto contiene lo script di layout, lo script ETL e il file di configurazione del piano.

Per ulteriori informazioni su altri piani di esempio, consulta Esempi di schema.

Di seguito è riportato lo script di layout, nel file `Layout.py`.

```
from awsglue.blueprint.workflow import *
from awsglue.blueprint.job import *
from awsglue.blueprint.crawler import *
import boto3

s3_client = boto3.client('s3')

# Ingesting all the S3 paths as Glue table in parquet format
def generate_layout(user_params, system_params):
    #Always give the full path for the file
    with open("ConversionBlueprint/Conversion.py", "rb") as f:
        s3_client.upload_fileobj(f, user_params['ScriptsBucket'], "Conversion.py")
    etlScriptLocation = "s3://{}/Conversion.py".format(user_params['ScriptsBucket'])

    crawlers = []
    jobs = []
    workflowName = user_params['WorkflowName']
    for path in user_params['S3Paths']:
      tablePrefix = "source_"
      crawler = Crawler(Name="{}_crawler".format(workflowName),
                        Role=user_params['PassRole'],
                        DatabaseName=user_params['TargetDatabase'],
                        TablePrefix=tablePrefix,
                        Targets= {"S3Targets": [{"Path": path}]})
```

```
      crawlers.append(crawler)
      transform_job = Job(Name="{}_transform_job".format(workflowName),
                          Command={"Name": "glueetl",
                                   "ScriptLocation": etlScriptLocation,
                                   "PythonVersion": "3"},
                          Role=user_params['PassRole'],
                          DefaultArguments={"--database_name":
 user_params['TargetDatabase'],
                                            "--table_prefix": tablePrefix,
                                            "--region_name": system_params['region'],
                                            "--output_path":
 user_params['TargetS3Location']},
                          DependsOn={crawler: "SUCCEEDED"},
                          WaitForDependencies="AND")
      jobs.append(transform_job)
   conversion_workflow = Workflow(Name=workflowName, Entities=Entities(Jobs=jobs,
 Crawlers=crawlers))
   return conversion_workflow
```

Di seguito è riportato il corrispondente file `blueprint.cfg` di configurazione del piano.

```
{
    "layoutGenerator": "ConversionBlueprint.Layout.generate_layout",
    "parameterSpec" : {
        "WorkflowName" : {
            "type": "String",
            "collection": false,
            "description": "Name for the workflow."
        },
        "S3Paths" : {
            "type": "S3Uri",
            "collection": true,
            "description": "List of Amazon S3 paths for data ingestion."
        },
        "PassRole" : {
            "type": "IAMRoleName",
            "collection": false,
            "description": "Choose an IAM role to be used in running the job/crawler"
        },
        "TargetDatabase": {
            "type": "String",
            "collection" : false,
            "description": "Choose a database in the Data Catalog."
```

```
        },
        "TargetS3Location": {
            "type": "S3Uri",
            "collection" : false,
            "description": "Choose an Amazon S3 output path: ex:s3://<target_path>/."
        },
        "ScriptsBucket": {
            "type": "S3Bucket",
            "collection": false,
            "description": "Provide an S3 bucket name(in the same AWS Region) to store
 the scripts."
        }
    }
}
```

Il seguente script nel file `Conversion.py` è lo script ETL caricato. Nota che durante la conversione mantiene lo schema di partizionamento.

```
import sys
from pyspark.sql.functions import *
from pyspark.context import SparkContext
from awsglue.transforms import *
from awsglue.context import GlueContext
from awsglue.job import Job
from awsglue.utils import getResolvedOptions
import boto3

args = getResolvedOptions(sys.argv, [
    'JOB_NAME',
    'region_name',
    'database_name',
    'table_prefix',
    'output_path'])
databaseName = args['database_name']
tablePrefix = args['table_prefix']
outputPath = args['output_path']

glue = boto3.client('glue', region_name=args['region_name'])

glue_context = GlueContext(SparkContext.getOrCreate())
spark = glue_context.spark_session
job = Job(glue_context)
job.init(args['JOB_NAME'], args)
```

```
def get_tables(database_name, table_prefix):
    tables = []
    paginator = glue.get_paginator('get_tables')
    for page in paginator.paginate(DatabaseName=database_name, Expression=table_prefix
+"*"):
        tables.extend(page['TableList'])
    return tables

for table in get_tables(databaseName, tablePrefix):
    tableName = table['Name']
    partitionList = table['PartitionKeys']
    partitionKeys = []
    for partition in partitionList:
        partitionKeys.append(partition['Name'])

    # Create DynamicFrame from Catalog
    dyf = glue_context.create_dynamic_frame.from_catalog(
        name_space=databaseName,
        table_name=tableName,
        additional_options={
            'useS3ListImplementation': True
        },
        transformation_ctx='dyf'
    )

    # Resolve choice type with make_struct
    dyf = ResolveChoice.apply(
        frame=dyf,
        choice='make_struct',
        transformation_ctx='resolvechoice_' + tableName
    )

    # Drop null fields
    dyf = DropNullFields.apply(
        frame=dyf,
        transformation_ctx="dropnullfields_" + tableName
    )

    # Write DynamicFrame to S3 in glueparquet
    sink = glue_context.getSink(
        connection_type="s3",
        path=outputPath,
        enableUpdateCatalog=True,
```

```
        partitionKeys=partitionKeys
    )
    sink.setFormat("glueparquet")

    sink.setCatalogInfo(
        catalogDatabase=databaseName,
        catalogTableName=tableName[len(tablePrefix):]
    )
    sink.writeFrame(dyf)

job.commit()
```

> **Note**
>
> Solo due percorsi Amazon S3 possono essere forniti come input per il piano di esempio. Questo perché le attivazioni di AWS Glue possono richiamare solo due operazioni crawler.

## Test di uno schema

Durante lo sviluppo del codice, è necessario eseguire test locali per verificare che il layout del flusso di lavoro sia corretto.

Il test locale non genera processi, crawler o attivazioni AWS Glue. Invece, esegui lo script di layout localmente e utilizzi i metodi `to_json()` e `validate()` per stampare gli oggetti e trovare gli errori. Questi metodi sono disponibili in tutte e tre le classi definite nelle librerie.

Sono disponibili due modi per gestire gli argomenti `user_params` e `system_params` cheAWS Glue passa alla funzione di layout. Il codice test-bench può creare un dizionario di valori di esempio dei parametri del piano e passarli alla funzione di layout come argomento `user_params`. In alternativa, puoi rimuovere i riferimenti a `user_params` e sostituirli con stringhe hardcoded.

Se il tuo codice utilizza proprietà `region` e `accountId` nell'argomento `system_params`, puoi passare nel tuo dizionario per `system_params`.

Per testare un piano

1. Avvia un interprete Python in una directory con le librerie o carica i file del piano e le librerie fornite nel tuo ambiente di sviluppo integrato (IDE) preferito.
2. Assicurati che il tuo codice importi le librerie fornite.

3. Aggiungi codice alla tua funzione di layout per chiamare `validate()` o `to_json()` su qualsiasi entità o sull'oggetto `Workflow`. Ad esempio, se il codice crea un oggetto `Crawler` denominato `mycrawler`, è possibile chiamare `validate()` come segue.

   ```
   mycrawler.validate()
   ```

   Puoi stampare `mycrawler` come segue:

   ```
   print(mycrawler.to_json())
   ```

   Se chiami `to_json` su un oggetto, non è necessario chiamare anche `validate()`, perché `to_json()` chiama `validate()`.

   È molto utile chiamare questi metodi sull'oggetto flusso di lavoro. Supponendo che lo script denomini l'oggetto flusso di lavoro `my_workflow`, convalida e stampa l'oggetto flusso di lavoro come segue.

   ```
   print(my_workflow.to_json())
   ```

   Per ulteriori informazioni su `to_json()` e `validate()`, consulta Metodi di classe.

   Puoi anche importare `pprint` e stampare con precisione l'oggetto flusso di lavoro, come illustrato nell'esempio più avanti in questa sezione.

4. Esegui il codice, correggi gli errori e infine rimuovi tutte le chiamate a `validate()` o `to_json()`.

Example

L'esempio seguente mostra come costruire un dizionario di parametri di esempio del piano e passarli come argomento `user_params` alla funzione di layout `generate_compaction_workflow`. Viene inoltre illustrato come stampare con precisione l'oggetto flusso di lavoro generato.

```
from pprint import pprint
from awsglue.blueprint.workflow import *
from awsglue.blueprint.job import *
from awsglue.blueprint.crawler import *

USER_PARAMS = {"WorkflowName": "compaction_workflow",
```

```
                "ScriptLocation": "s3://amzn-s3-demo-bucket/scripts/threaded-
compaction.py",
                "PassRole": "arn:aws:iam::111122223333:role/GlueRole-ETL",
                "DatabaseName": "cloudtrial",
                "TableName": "ct_cloudtrail",
                "CoalesceFactor": 4,
                "MaxThreadWorkers": 200}


def generate_compaction_workflow(user_params: dict, system_params: dict) -> Workflow:
    compaction_job = Job(Name=f"{user_params['WorkflowName']}_etl_job",
                         Command={"Name": "glueetl",
                                  "ScriptLocation": user_params['ScriptLocation'],
                                  "PythonVersion": "3"},
                         Role="arn:aws:iam::111122223333:role/
AWSGlueServiceRoleDefault",
                         DefaultArguments={"DatabaseName": user_params['DatabaseName'],
                                           "TableName": user_params['TableName'],
                                           "CoalesceFactor":
 user_params['CoalesceFactor'],
                                           "max_thread_workers":
 user_params['MaxThreadWorkers']})

    catalog_target = {"CatalogTargets": [{"DatabaseName": user_params['DatabaseName'],
 "Tables": [user_params['TableName']]}]}

    compacted_files_crawler = Crawler(Name=f"{user_params['WorkflowName']}_post_crawl",
                                      Targets = catalog_target,
                                      Role=user_params['PassRole'],
                                      DependsOn={compaction_job: "SUCCEEDED"},
                                      WaitForDependencies="AND",
                                      SchemaChangePolicy={"DeleteBehavior": "LOG"})

    compaction_workflow = Workflow(Name=user_params['WorkflowName'],
                                   Entities=Entities(Jobs=[compaction_job],

 Crawlers=[compacted_files_crawler]))
    return compaction_workflow

generated = generate_compaction_workflow(user_params=USER_PARAMS, system_params={})
gen_dict = generated.to_json()

pprint(gen_dict)
```

## Pubblicazione di uno schema

Dopo aver sviluppato un piano, devi caricarlo su Amazon S3. Devi disporre delle autorizzazioni di scrittura sul bucket Amazon S3 utilizzato per pubblicare il piano. Devi inoltre assicurarti che l'amministratore AWS Glue, che registrerà il progetto disponga dell'accesso in lettura al bucket Amazon S3. Per le politiche di autorizzazione suggerite AWS Identity and Access Management (IAM) per i personaggi e i ruoli per i AWS Glue blueprint, vedere. Autorizzazioni per utenti e ruoli per gli schemi AWS Glue

Per pubblicare un piano

1. Crea gli script, le risorse e il file di configurazione del piano necessari.
2. Aggiungi tutti i file a un archivio ZIP e carica il file ZIP su Amazon S3. Utilizza un bucket S3 che si trova nella regione in cui gli utenti registreranno ed eseguiranno il piano.

   È possibile creare un file ZIP dalla riga di comando utilizzando il comando seguente.

   ```
   zip -r folder.zip folder
   ```

3. Aggiungi una policy bucket che conceda le autorizzazioni di lettura all'account desiderato. AWS Di seguito è riportata una policy di esempio.

   JSON

   ```
   {
     "Version":"2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Principal": {
           "AWS": "arn:aws:iam::111122223333:root"
         },
         "Action": "s3:GetObject",
         "Resource": "arn:aws:s3:::my-blueprints/*"
       }
     ]
   }
   ```

4. Concedi l'autorizzazione `s3:GetObject` IAM sul bucket Amazon S3 all'amministratore AWS Glue o a chiunque registrerà i progetti. Per un esempio di policy da concedere agli

amministratori, consulta Autorizzazioni per i progetti per l'amministratore di AWS Glue per gli schemi.

Dopo aver completato il test locale del progetto, potresti anche voler testare un progetto su AWS Glue. Per testare un progetto su AWS Glue, questo deve essere registrato. È possibile limitare chi vede il piano registrato utilizzando l'autorizzazione IAM o account di test separati.

Consulta anche:

- Registrazione di uno schema in AWS Glue

## Riferimento alle classi di schema AWS Glue

Le librerie per i piani AWS Glue definiscono tre classi da utilizzare nello script di layout del flusso di lavoro: `Job`, `Crawler` e `Workflow`.

Argomenti

- Classe di processo
- Classe di crawler
- Classe di flusso di lavoro
- Metodi di classe

### Classe di processo

La classe `Job` rappresenta un processo ETL AWS Glue.

Argomenti dei costruttori obbligatori

Di seguito sono illustrati gli argomenti dei costruttori obbligatori per la classe di `Job`.

| Nome argomento | Tipo | Description |
|---|---|---|
| `Name` | `str` | Nome da assegnare al processo. AWS Glue aggiunge un suffisso generato casualmente al nome per distinguere il processo da quelli creati da altre esecuzioni del progetto. |

| Nome argomento | Tipo | Description |
| --- | --- | --- |
| `Role` | `str` | L'Amazon Resource Name (ARN) del ruolo che deve assumere il processo durante l'esecuzione. |
| `Command` | `dict` | Comando del processo, come specificato nella documentazione API in JobCommand struttura. |

Argomenti dei costruttori facoltativi

Di seguito sono illustrati gli argomenti dei costruttori facoltativi per la classe di `Job`.

| Nome argomento | Tipo | Description |
| --- | --- | --- |
| `DependsOn` | `dict` | Elenco delle entità del flusso di lavoro da cui dipende il processo. Per ulteriori informazioni, consulta Utilizzo dell' DependsOn argomento. |
| `WaitForDependencies` | `str` | Indica se il processo deve attendere fino a che tutte le entità da cui dipende sono complete prima dell'esecuzione o fino a quando è completa una qualsiasi. Per ulteriori informazioni, consulta Usando l' WaitForDependencies argomento. Ometti se il processo dipende da una sola entità. |
| (Proprietà processo) | - | Qualsiasi proprietà del processo elencate in Struttura del processo nella documentazione API AWS Glue (eccetto `CreatedOn` e `LastModifiedOn`). |

Classe di crawler

La classe `Crawler` rappresenta un crawler AWS Glue.

Argomenti dei costruttori obbligatori

Di seguito sono illustrati gli argomenti dei costruttori obbligatori per la classe di `Crawler`.

| Nome argomento | Tipo | Description |
| --- | --- | --- |
| `Name` | `str` | Nome da assegnare al crawler. AWS Glue aggiunge un suffisso generato casualmente al nome per distinguere il crawler da quelli creati da altre esecuzioni del progetto. |
| `Role` | `str` | ARN del ruolo che il crawler deve assumere durante l'esecuzione. |
| `Targets` | `dict` | Raccolta di destinazioni da sottoporre al crawling. Gli argomenti dei costruttori della classe `Targets` sono definiti in CrawlerTargets struttura nella documentazione API. Tutti gli argomenti dei costruttori `Targets` sono facoltativi, ma è necessario passarne almeno uno. |

Argomenti dei costruttori facoltativi

Di seguito sono illustrati gli argomenti dei costruttori facoltativi per la classe di `Crawler`.

| Nome argomento | Tipo | Description |
| --- | --- | --- |
| `DependsOn` | `dict` | Elenco delle entità del flusso di lavoro da cui dipende il crawler. Per ulteriori informazioni, consulta Utilizzo dell' DependsOn argomento. |
| `WaitForDependencies` | `str` | Indica se il crawler deve attendere fino a che tutte le entità da cui dipende sono complete prima dell'esecuzione o fino a quando è completa una qualsiasi. Per ulteriori informazioni, consulta Usando l' WaitForDependencies argomento. Ometti se il crawler dipende da una sola entità. |

| Nome argomento | Tipo | Description |
| --- | --- | --- |
| (Proprietà dei crawler) | - | Qualsiasi proprietà del crawler elencata in Struttura dei crawler nella documentazione API AWS Glue, con le seguenti eccezioni:<br>• `State`<br>• `CrawlElapsedTime`<br>• `CreationTime`<br>• `LastUpdated`<br>• `LastCrawl`<br>• `Version` |

Classe di flusso di lavoro

La classe `Workflow` rappresenta un flusso di lavoro AWS Glue. Lo script layout del flusso di lavoro restituisce un `Workflow`. AWS Glue crea un flusso di lavoro basato su questo oggetto.

Argomenti dei costruttori obbligatori

Di seguito sono illustrati gli argomenti dei costruttori obbligatori per la classe di `Workflow`.

| Nome argomento | Tipo | Description |
| --- | --- | --- |
| `Name` | `str` | Nome da assegnare al flusso di lavoro. |
| `Entities` | `Entities` | Insieme di entità (processi e crawler) da includere nel flusso di lavoro. Il costruttore di classi `Entities` accetta un argomento `Jobs`, che è un elenco di oggetti `Job` e un argomento `Crawlers`, che è un elenco di oggetti `Crawler`. |

Argomenti dei costruttori facoltativi

Di seguito sono illustrati gli argomenti dei costruttori facoltativi per la classe di `Workflow`.

| Nome argomento | Tipo | Description |
| --- | --- | --- |
| `Description` | `str` | Per informazioni, consulta Struttura flusso di lavoro. |
| `DefaultRunProperties` | `dict` | Per informazioni, consulta Struttura flusso di lavoro. |
| `OnSchedule` | `str` | Un'espressione `cron`. |

Metodi di classe

Tutte e tre le classi includono i seguenti metodi.

validate()

Convalida le proprietà dell'oggetto e, se vengono rilevati errori, genera un messaggio ed esce. Non genera alcun output se non ci sono errori. Per la classe di `Workflow`, si richiama su ogni entità nel flusso di lavoro.

to_json()

Serializza l'oggetto in JSON. Chiama anche `validate()`. Per la classe di `Workflow`, l'oggetto JSON include elenchi di processi e crawler e un elenco di trigger generati dalle specifiche di dipendenza del processo e del crawler.

## Esempi di schema

Nel Repository Github per i progetti AWS Glue, sono disponibili diversi progetti di esempio . Questi esempi sono solo di riferimento e non sono destinati all'utilizzo.

I titoli dei progetti di esempio sono:

- Compattazione: questo piano crea un lavoro che compatta i file di input in blocchi più grandi in base alla dimensione di file desiderata.
- Conversione: questo piano converte i file di input in vari formati di file standard in formato Apache Parquet, ottimizzato per i carichi di lavoro analitici.
- Crawling di posizioni Amazon S3: questo piano esegue il crawling di più posizioni Amazon S3 per aggiungere tabelle di metadati al catalogo dati.

- Connessione personalizzata a Data Catalog: questo blueprint accede agli archivi dati utilizzando connettori AWS Glue personalizzati, legge i record e compila le definizioni delle tabelle nel AWS Glue Data Catalog in base allo schema dei record.
- Codifica: questo piano converte i file non UTF in file codificati UTF.
- Partizionamento: questo piano crea un processo di partizionamento che inserisce i file di output in partizioni basate su chiavi di partizione specifiche.
- Importazione di dati Amazon S3 in una tabella DynamoDB: questo progetto importa i dati da Amazon S3 in una tabella DynamoDB.
- Da tabella standard a governata: questo progetto importa una tabella AWS Glue Data Catalog in una tabella Lake Formation.

## Registrazione di uno schema in AWS Glue

Dopo che lo sviluppatore di AWS Glue ha codificato il progetto e caricato un archivio in formato ZIP su Amazon Simple Storage Service (Amazon S3), un amministratore di AWS Glue dovrà registrarlo. La registrazione del piano lo rende disponibile per l'uso.

Quando si registra un progetto, AWS Glue ne copia l'archivio in una posizione Amazon S3 riservata. È quindi possibile eliminare l'archivio dalla posizione di caricamento.

Per registrare un piano, hai bisogno delle autorizzazioni di lettura per la posizione Amazon S3 che contiene l'archivio caricato. È inoltre necessaria l'autorizzazione AWS Identity and Access Management (IAM)`glue:CreateBlueprint`. Per le autorizzazioni suggerite da concedere a un amministratore di AWS Glue che deve registrare, visualizzare e gestire i progetti, consulta Autorizzazioni per i progetti per l'amministratore di AWS Glue per gli schemi.

Puoi registrare un progetto utilizzando la console AWS Glue, l'API AWS Glue o la AWS Command Line Interface (AWS CLI).

Per registrare un piano (console)

1. Accertati di disporre delle autorizzazioni di lettura (`s3:GetObject`) per l'archivio ZIP del piano in Amazon S3.
2. Apri la console AWS Glue all'indirizzo https://console.aws.amazon.com/glue/.

   Accedi come un utente che dispone delle autorizzazioni per registrare un piano. Passa alla stessa regione AWS del bucket Amazon S3 che contiene l'archivio ZIP del piano.

3. Nel pannello di navigazione seleziona schemi. Quindi, nella pagina schemi, seleziona Add blueprint (aggiungi schema).
4. Immetti un nome e, facoltativamente, una descrizione.
5. Per ZIP archive location (S3) (Posizione archivio ZIP [S3]), inserisci il percorso Amazon S3 dell'archivio ZIP del piano caricato. Includi il nome del file di archivio nel percorso e inizia il percorso con `s3://`.
6. (Facoltativo) Aggiungi uno o più tag.
7. Scegli Add blueprint (Aggiungi piano).

   La pagina schemi restituisce e mostra che lo stato del piano è `CREATING`. Seleziona il pulsante di aggiornamento fino a quando lo stato non cambia in `ACTIVE` o `FAILED`.
8. Se lo stato è `FAILED`, seleziona il piano e nella scheda Actions (Operazioni), scegli View (Visualizza).

   La pagina dei dettagli mostra il motivo dell'errore. Se il messaggio dell'errore indica che è impossibile accedere all'oggetto nella posizione... o che è negato l'accesso sull'oggetto nella posizione..., verifica i requisiti seguenti:

   - L'utente con cui hai effettuato l'accesso deve disporre dell'autorizzazione di lettura per l'archivio ZIP del piano in Amazon S3.
   - Il bucket Amazon S3 che contiene l'archivio ZIP deve avere una policy sui bucket che conceda l'autorizzazione di lettura sull'oggetto all'ID del tuo account. AWS Per ulteriori informazioni, consulta Sviluppo di schemi in AWS Glue.
   - Il bucket Amazon S3 che stai utilizzando deve trovarsi nella stessa regione di quella alla quale hai eseguito l'accesso sulla console.
9. Assicurati che gli analisti dei dati dispongano delle autorizzazioni per il piano.

   La policy IAM suggerita per gli analisti di dati è mostrata in Autorizzazioni per gli schemi per l'analista dati. Questa policy concede `glue:GetBlueprint` su qualsiasi risorsa. Se i criteri sono più granulari a livello di risorsa, concedi agli analisti di dati le autorizzazioni per questa risorsa appena creata.

Per registrare un blueprint (AWS CLI)

1. Inserisci il comando seguente.

```
aws glue create-blueprint --name <blueprint-name> [--description <description>] --
blueprint-location s3://<s3-path>/<archive-filename>
```

2. Immetti il seguente comando per verificare lo stato del piano. Ripeti il comando fino a quando lo stato non diventa `ACTIVE` o `FAILED`.

```
aws glue get-blueprint --name <blueprint-name>
```

Se lo stato è `FAILED` e il messaggio dell'errore indica che è impossibile accedere all'oggetto nella posizione... o che è negato l'accesso sull'oggetto nella posizione..., verifica i requisiti seguenti:

- L'utente con cui hai effettuato l'accesso deve disporre dell'autorizzazione di lettura per l'archivio ZIP del piano in Amazon S3.
- Il bucket Amazon S3 contenente l'archivio ZIP deve avere una policy sui bucket che conceda l'autorizzazione di lettura sull'oggetto all'ID del tuo account. AWS Per ulteriori informazioni, consulta Pubblicazione di uno schema.
- Il bucket Amazon S3 che stai utilizzando deve trovarsi nella stessa regione di quella alla quale hai eseguito l'accesso sulla console.

Consulta anche:

- Panoramica degli schemi in AWS Glue

## Visualizzazione degli schemi in AWS Glue

Visualizza un piano per esaminare la descrizione, lo stato e le specifiche dei parametri del piano e scaricare l'archivio ZIP del piano.

Puoi visualizzare un progetto utilizzando la console AWS Glue, l'API AWS Glue o la AWS Command Line Interface (AWS CLI).

Per visualizzare un piano (console)

1. Apri la console AWS Glue all'indirizzo https://console.aws.amazon.com/glue/.

2. Nel pannello di navigazione scegli schemi.
3. Nella pagina schemi seleziona uno schema. Quindi nel menu Actions (Operazioni), scegli View (Visualizza).

Per visualizzare un blueprint (AWS CLI)

- Immetti il comando seguente per visualizzare solo il nome, la descrizione e lo stato del piano. Sostituisci *<blueprint-name>* con il nome del blueprint da visualizzare.

```
aws glue get-blueprint --name <blueprint-name>
```

L'output dei comandi è simile al seguente.

```
{
    "Blueprint": {
        "Name": "myDemoBP",
        "CreatedOn": 1587414516.92,
        "LastModifiedOn": 1587428838.671,
        "BlueprintLocation": "s3://amzn-s3-demo-bucket1/demo/
DemoBlueprintProject.zip",
        "Status": "ACTIVE"
    }
}
```

Immetti il seguente comando per visualizzare anche le specifiche di parametro.

```
aws glue get-blueprint --name <blueprint-name>  --include-parameter-spec
```

L'output dei comandi è simile al seguente.

```
{
    "Blueprint": {
        "Name": "myDemoBP",
        "CreatedOn": 1587414516.92,
        "LastModifiedOn": 1587428838.671,
        "ParameterSpec": "{\"WorkflowName\":{\"type\":\"String\",\"collection
\":false,\"description\":null,\"defaultValue\":null,\"allowedValues\":null},
\"PassRole\":{\"type\":\"String\",\"collection\":false,\"description\":null,
\"defaultValue\":null,\"allowedValues\":null},\"DynamoDBTableName\":{\"type
```

```
\":\"String\",\"collection\":false,\"description\":null,\"defaultValue\":null,
\"allowedValues\":null},\"ScriptLocation\":{\"type\":\"String\",\"collection
\":false,\"description\":null,\"defaultValue\":null,\"allowedValues\":null}}",
        "BlueprintLocation": "s3://awsexamplebucket1/demo/
DemoBlueprintProject.zip",
        "Status": "ACTIVE"
    }
}
```

Aggiungi l'argomento `--include-blueprint` per includere nell'output un URL che è possibile incollare nel browser per scaricare l'archivio ZIP del progetto archiviato in AWS Glue.

Consulta anche:

- Panoramica degli schemi in AWS Glue

## Aggiornamento di uno schema in AWS Glue

Puoi aggiornare un piano se si hai uno script di layout revisionato, un set di parametri del piano revisionato o file di supporto revisionati. L'aggiornamento di un piano crea una nuova versione.

L'aggiornamento di un piano non influisce sui flussi di lavoro esistenti creati dal piano.

Puoi aggiornare un progetto utilizzando la console AWS Glue, l'API AWS Glue o la AWS Command Line Interface (AWS CLI).

La procedura seguente presuppone che lo sviluppatore di AWS Glue abbia creato e caricato un archivio in formato ZIP del progetto aggiornato su Amazon S3.

Per aggiornare un piano (console)

1. Accertati di disporre delle autorizzazioni di lettura (`s3:GetObject`) per l'archivio ZIP del piano in Amazon S3.
2. Apri la console AWS Glue all'indirizzo https://console.aws.amazon.com/glue/.

   Accedi come utente che dispone delle autorizzazioni per aggiornare un piano. Passa alla stessa regione AWS del bucket Amazon S3 che contiene l'archivio ZIP del piano.
3. Nel pannello di navigazione scegli schemi.

4. Nella pagina schemi, seleziona un piano e nella scheda Actions (operazioni) scegli Edit (modifica).
5. Nella pagina Edit a blueprint (Modifica un piano), aggiorna la Description (Descrizione) del piano o la ZIP archive location (S3) (Posizione dell'archivio ZIP [S3]). Assicurati di includere il nome del file di archivio nel percorso.
6. Seleziona Save (Salva).

   La pagina schemi restituisce e mostra che lo stato dello schema è `UPDATING`. Seleziona il pulsante di aggiornamento fino a quando lo stato non cambia in `ACTIVE` o `FAILED`.
7. Se lo stato è `FAILED`, seleziona il piano e nella scheda Actions (Operazioni), scegli View (Visualizza).

   La pagina dei dettagli mostra il motivo dell'errore. Se il messaggio dell'errore indica che è impossibile accedere all'oggetto nella posizione... o che è negato l'accesso sull'oggetto nella posizione..., verifica i requisiti seguenti:

   - L'utente con cui hai effettuato l'accesso deve disporre dell'autorizzazione di lettura per l'archivio ZIP del piano in Amazon S3.
   - Il bucket Amazon S3 che contiene l'archivio ZIP deve avere una policy sui bucket che conceda l'autorizzazione di lettura sull'oggetto all'ID del tuo account. AWS Per ulteriori informazioni, consulta Pubblicazione di uno schema.
   - Il bucket Amazon S3 che stai utilizzando deve trovarsi nella stessa regione di quella alla quale hai eseguito l'accesso sulla console.

   > **Note**
   >
   > Se l'aggiornamento non riesce, l'esecuzione successiva del piano utilizza la versione più recente del piano correttamente registrata o aggiornata.

Per aggiornare un piano (AWS CLI)

1. Inserisci il comando seguente.

```
aws glue update-blueprint --name <blueprint-name> [--description <description>] --
blueprint-location s3://<s3-path>/<archive-filename>
```

2. Immetti il seguente comando per verificare lo stato del piano. Ripeti il comando fino a quando lo stato non diventa `ACTIVE` o `FAILED`.

   ```
   aws glue get-blueprint --name <blueprint-name>
   ```

   Se lo stato è `FAILED` e il messaggio dell'errore indica che è impossibile accedere all'oggetto nella posizione... o che è negato l'accesso sull'oggetto nella posizione..., verifica i requisiti seguenti:

   - L'utente con cui hai effettuato l'accesso deve disporre dell'autorizzazione di lettura per l'archivio ZIP del piano in Amazon S3.
   - Il bucket Amazon S3 contenente l'archivio ZIP deve avere una policy sui bucket che conceda l'autorizzazione di lettura sull'oggetto all'ID del tuo account. AWS Per ulteriori informazioni, consulta Pubblicazione di uno schema.
   - Il bucket Amazon S3 che stai utilizzando deve trovarsi nella stessa regione di quella alla quale hai eseguito l'accesso sulla console.

> **Consulta anche**
>
> - Panoramica degli schemi in AWS Glue

## Creazione di un flusso di lavoro da uno schema in AWS Glue

Puoi creare un flusso di lavoro AWS Glue manualmente, aggiungendo un componente alla volta, oppure puoi creare un flusso di lavoro da un progetto AWS Glue. AWS Glue include progetti per casi d'uso comuni. Gli sviluppatori di AWS Glue possono creare progetti aggiuntivi.

> **Important**
>
> Limita il numero totale di processi, crawler e attivazioni all'interno di un flusso di lavoro a 100 o meno. Se includi più di 100, potresti riscontrare errori durante il tentativo di riprendere o interrompere l'esecuzione del flusso di lavoro.

Quando utilizzi un progetto, puoi generare rapidamente un flusso di lavoro per uno specifico caso d'uso basato sul caso d'uso generalizzato definito dal progetto. Puoi definire il caso d'uso specifico

fornendo valori per i parametri del progetto. Ad esempio, un progetto che partiziona un set di dati potrebbe avere i percorsi di origine e destinazione di Amazon S3 come parametri.

AWS Glue crea un flusso di lavoro da un progetto eseguendo il progetto. L'esecuzione del progetto salva i valori dei parametri forniti e viene utilizzata per tenere traccia dell'avanzamento e dell'esito della creazione del flusso di lavoro e dei relativi componenti. Durante la risoluzione dei problemi di un flusso di lavoro, puoi sempre visualizzare l'esecuzione del progetto per determinare i valori dei parametri del progetto utilizzati per creare un flusso di lavoro.

Per creare e visualizzare i flussi di lavoro, è necessario disporre di determinate autorizzazioni IAM. Per la policy IAM suggerita, consulta Autorizzazioni per gli schemi per l'analista dati.

Puoi creare un piano utilizzando la console AWS Glue, l'API AWS Glue o la AWS Command Line Interface (AWS CLI).

Per creare un flusso di lavoro da un progetto (console)

1. Apri la console AWS Glue all'indirizzo https://console.aws.amazon.com/glue/.

   Accedi come un utente che dispone delle autorizzazioni per creare un flusso di lavoro.
2. Nel pannello di navigazione seleziona schemi.
3. Seleziona un progetto e nel menu Actions (Operazioni), scegli Create workflow (Crea flusso di lavoro).
4. Nella pagina Create a workflow from <blueprint-name> (Crea un flusso di lavoro da <blueprint-name>), inserisci le seguenti informazioni:

   Parametri del progetto

   Questi variano in base alla progettazione del progetto. Per domande sui parametri, consulta lo sviluppatore. Gli schemi in genere includono un parametro per il nome del flusso di lavoro.

   Ruolo IAM

   Il ruolo che AWS Glue assume per creare il flusso di lavoro e i relativi componenti. Il ruolo deve disporre delle autorizzazioni per creare ed eliminare flussi di lavoro, processi, crawler e trigger. Per una policy suggerita per il ruolo, consulta Autorizzazioni per i ruoli degli schemi.
5. Seleziona Invia.

   Viene visualizzata la pagina Blueprint Details (Dettagli progetto), che mostra un elenco di esecuzioni del piano nella parte inferiore.

6. Nell'elenco delle esecuzioni del progetto, controlla lo stato della creazione del flusso di lavoro nell'esecuzione del progetto che si trova più in alto.

   Lo stato iniziale è `RUNNING`. Seleziona il pulsante di aggiornamento fino a quando lo stato non diventa `SUCCEEDED` o `FAILED`.

7. Scegli una delle seguenti operazioni:

   - Se lo stato di completamento è `SUCCEEDED`, puoi passare alla pagina Workflows (Flussi di lavoro), selezionare il flusso di lavoro appena creato ed eseguirlo. Prima di eseguire il flusso di lavoro, è possibile esaminare il grafico di progettazione.
   - Se lo stato di completamento è `FAILED`, seleziona l'esecuzione del progetto e nel menu Actions (Operazioni), scegli View (Visualizza) per vedere il messaggio di errore.

Per ulteriori informazioni sui flussi di lavoro e sui progetti, consulta i seguenti argomenti.

- Panoramica di flussi di lavoro in AWS Glue
- Aggiornamento di uno schema in AWS Glue
- Creazione e costruzione manuale di un flusso di lavoro in AWS Glue

## Visualizzazione delle esecuzioni dello schema in AWS Glue

Visualizza l'esecuzione di un piano per vedere le seguenti informazioni:

- Nome del flusso di lavoro che è stato creato.
- Valori dei parametri dello schema utilizzati per creare il flusso di lavoro.
- Stato dell'operazione di creazione del flusso di lavoro.

Puoi visualizzare l'esecuzione di un progetto utilizzando la console AWS Glue, l'API AWS Glue o la AWS Command Line Interface (AWS CLI).

Per visualizzare l'esecuzione di un piano (console)

1. Apri la console AWS Glue all'indirizzo https://console.aws.amazon.com/glue/.
2. Nel pannello di navigazione scegli schemi.
3. Nella pagina schemi seleziona uno schema. Quindi nel menu Actions (Operazioni), scegli View (Visualizza).

4. Nella parte inferiore della finestra Blueprint Details (Dettagli piano), seleziona un'esecuzione del piano e nel menu Actions (Operazioni), scegli View (Visualizza).

Per visualizzare l'esecuzione di un piano (AWS CLI)

- Inserire il seguente comando. Sostituisci *<blueprint-name>* con il nome del blueprint. Sostituisci *<blueprint-run-id>* con l'ID di esecuzione del blueprint.

```
aws glue get-blueprint-run --blueprint-name <blueprint-name> --run-id <blueprint-run-id>
```

Consulta anche:

- Panoramica degli schemi in AWS Glue

# AWS CloudFormation per AWS Glue

CloudFormation è un servizio in grado di creare molte AWS risorse. AWS Gluefornisce operazioni API per creare oggetti in AWS Glue Data Catalog. Tuttavia, potrebbe essere più comodo definire e creare AWS Glue oggetti e altri oggetti di AWS risorsa correlati in un file CloudFormation modello. È quindi possibile automatizzare il processo di creazione degli oggetti.

CloudFormation fornisce una sintassi semplificata, JSON (JavaScript Object Notation) o YAML (YAML Ain't Markup Language), per esprimere la creazione di risorse. AWS Puoi usare modelli CloudFormation per definire oggetti del catalogo dati come database, tabelle, partizioni, crawler, classificatori e connessioni. È anche possibile definire oggetti ETL, come processi, trigger ed endpoint di sviluppo. Crei un modello che descrive tutte le risorse che desideri e si occupa del provisioning e della AWS configurazione di tali risorse per te. CloudFormation

Per ulteriori informazioni, consulta What Is? AWS CloudFormation e Utilizzo dei AWS CloudFormation modelli nella Guida per l'AWS CloudFormation utente.

Se intendi utilizzare CloudFormation modelli compatibili conAWS Glue, in qualità di amministratore, devi concedere l'accesso CloudFormation e ai AWS servizi e alle azioni da cui dipende. Per concedere le autorizzazioni alla creazione di CloudFormation risorse, allega la seguente politica agli utenti che lavorano con CloudFormation:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudformation:*"
      ],
      "Resource": "*"
    }
  ]
}
```

La tabella seguente contiene le azioni che un CloudFormation modello può eseguire per tuo conto. Include collegamenti a informazioni sui tipi di AWS risorse e sui relativi tipi di proprietà che è possibile aggiungere a un CloudFormation modello.

| Risorsa AWS Glue | CloudFormation modello | Esempi di AWS Glue |
|---|---|---|
| Classificatore | AWS::Glue::Classifier | Classificatore Grok, classificatore JSON, classificatore XML |
| Connessione | AWS::Glue::Connection | Connessione MySQL |
| Crawler | AWS::Glue::Crawler | Crawler Amazon S3, crawler MySQL |
| Database | AWS::Glue::Database | Database vuoto, database con tabelle |
| Endpoint di sviluppo | AWS::Glue::DevEndpoint | Endpoint di sviluppo |
| Integrazione | AWS::Glue::Integration | Integrazione zero-ETL |
| Proprietà delle risorse di integrazione | AWS::Glue::IntegrationResourceProprietà | Integrazione zero-ETL con proprietà delle risorse di integrazione |
| Processo | AWS::Glue::Job | Processo Amazon S3, Processo JDBC |
| Trasformazione basata su machine learning | AWS::Glue::MLTransform | Trasformazione basata su machine learning |
| Set di regole sulla qualità dei dati | AWS::Glue::DataQualitySet di regole | Set di regole sulla qualità dei dati, set di regole sulla qualità dei dati con scheduler EventBridge |
| Partizione | AWS::Glue::Partition | Partizioni di una tabella |
| Tabella | AWS::Glue::Table | Tabella in un database |

| Risorsa AWS Glue | CloudFormation modello | Esempi di AWS Glue |
| --- | --- | --- |
| Trigger | AWS::Glue::Trigger | Trigger on demand, trigger pianificato, trigger condizionale |

Per iniziare, usa i modelli di esempio seguenti e personalizzali con i tuoi metadati. Quindi utilizza la CloudFormation console per creare uno CloudFormation stack a cui aggiungere oggetti e tutti i servizi associati. AWS Glue Molti campi di un oggetto AWS Glue sono facoltativi. Questi modelli indicano i campi obbligatori o necessari per un oggetto AWS Glue funzionante e funzionale.

Un CloudFormation modello può essere in formato JSON o YAML. In questi esempi viene usato il formato YAML per semplificare la lettura. Gli esempi contengono commenti (#) per descrivere i valori definiti nei modelli.

CloudFormation i modelli possono includere una sezione. `Parameters` Questa sezione può essere modificata nel testo di esempio o quando il file YAML viene inviato alla CloudFormation console per creare uno stack. La `Resources` sezione del modello contiene la definizione AWS Glue e gli oggetti correlati. CloudFormation le definizioni della sintassi del modello potrebbero contenere proprietà che includono una sintassi delle proprietà più dettagliata. Non tutte le proprietà potrebbero essere necessarie per creare un oggetto AWS Glue. Questi esempi mostrano valori di esempio per alcune proprietà comuni per la creazione di un oggetto AWS Glue.

## CloudFormation Modello di esempio per un database AWS Glue

Un database AWS Glue nel catalogo dati contiene tabelle di metadati. Il database è composto da pochissime proprietà e può essere creato nel Data Catalog con un CloudFormation modello. Il seguente modello di esempio viene fornito per iniziare e per illustrare l'uso degli CloudFormation stack. AWS Glue L'unica risorsa creata dal modello di esempio è un database denominato `cfn-mysampledatabase`. Puoi cambiarlo modificando il testo dell'esempio o cambiando il valore sulla CloudFormation console quando invii il file YAML.

L'esempio seguente mostra valori di esempio per alcune proprietà comuni per la creazione di un database AWS Glue. Per ulteriori informazioni sul modello di CloudFormation database perAWS Glue, vedere. AWS::Glue::Database

```
---
AWSTemplateFormatVersion: '2010-09-09'
# Sample CloudFormation template in YAML to demonstrate creating a database named
 mysampledatabase
# The metadata created in the Data Catalog points to the flights public S3 bucket
#
# Parameters section contains names that are substituted in the Resources section
# These parameters are the names the resources created in the Data Catalog
Parameters:
  CFNDatabaseName:
    Type: String
    Default: cfn-mysampledatabse

# Resources section defines metadata for the Data Catalog
Resources:
# Create an AWS Glue database
  CFNDatabaseFlights:
    Type: AWS::Glue::Database
    Properties:
      # The database is created in the Data Catalog for your account
      CatalogId: !Ref AWS::AccountId
      DatabaseInput:
        # The name of the database is defined in the Parameters section above
        Name: !Ref CFNDatabaseName
        Description: Database to hold tables for flights data
        LocationUri: s3://crawler-public-us-east-1/flight/2016/csv/
        #Parameters: Leave AWS database parameters blank
```

# CloudFormation Modello di esempio per un AWS Glue database, una tabella e una partizione

Una tabella AWS Glue contiene i metadati che definiscono la struttura e la posizione dei dati che vuoi elaborare con gli script ETL. In una tabella è possibile definire partizioni per parallelizzare l'elaborazione dei dati. Una partizione è un blocco di dati definito con una chiave. Se, ad esempio, usi il mese come chiave, tutti i dati per gennaio vengono inclusi nella stessa partizione. In AWS Glue i database possono contenere tabelle e le tabelle possono contenere partizioni.

L'esempio seguente mostra come popolare un database, una tabella e le partizioni usando un modello CloudFormation . Il formato dei dati di base è `csv`, con valori delimitati da una virgola (,). Poiché un database deve esistere per poter contenere una tabella e una tabella deve esistere per

poter creare le partizioni, il modello usa l'istruzione `DependsOn` per definire la dipendenza di questi oggetti quando vengono creati.

I valori in questo esempio definiscono una tabella che contiene dati di voli da un bucket Amazon S3 disponibile pubblicamente. A scopo illustrativo, sono definite solo alcune colonne di dati e una chiave di partizionamento. Vengono definite anche quattro partizioni nel catalogo dati. Nei campi `StorageDescriptor` sono mostrati anche alcuni campi per descrivere lo storage dei dati di base.

```
---
AWSTemplateFormatVersion: '2010-09-09'
# Sample CloudFormation template in YAML to demonstrate creating a database, a table,
 and partitions
# The metadata created in the Data Catalog points to the flights public S3 bucket
#
# Parameters substituted in the Resources section
# These parameters are names of the resources created in the Data Catalog
Parameters:
  CFNDatabaseName:
    Type: String
    Default: cfn-database-flights-1
  CFNTableName1:
    Type: String
    Default: cfn-manual-table-flights-1
# Resources to create metadata in the Data Catalog
Resources:
###
# Create an AWS Glue database
  CFNDatabaseFlights:
    Type: AWS::Glue::Database
    Properties:
      CatalogId: !Ref AWS::AccountId
      DatabaseInput:
        Name: !Ref CFNDatabaseName
        Description: Database to hold tables for flights data
###
# Create an AWS Glue table
  CFNTableFlights:
    # Creating the table waits for the database to be created
    DependsOn: CFNDatabaseFlights
    Type: AWS::Glue::Table
    Properties:
      CatalogId: !Ref AWS::AccountId
```

```
      DatabaseName: !Ref CFNDatabaseName
      TableInput:
        Name: !Ref CFNTableName1
        Description: Define the first few columns of the flights table
        TableType: EXTERNAL_TABLE
        Parameters: {
    "classification": "csv"
  }
#       ViewExpandedText: String
        PartitionKeys:
        # Data is partitioned by month
        - Name: mon
          Type: bigint
        StorageDescriptor:
          OutputFormat: org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat
          Columns:
          - Name: year
            Type: bigint
          - Name: quarter
            Type: bigint
          - Name: month
            Type: bigint
          - Name: day_of_month
            Type: bigint
          InputFormat: org.apache.hadoop.mapred.TextInputFormat
          Location: s3://crawler-public-us-east-1/flight/2016/csv/
          SerdeInfo:
            Parameters:
              field.delim: ","
            SerializationLibrary: org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe
# Partition 1
# Create an AWS Glue partition
  CFNPartitionMon1:
    DependsOn: CFNTableFlights
    Type: AWS::Glue::Partition
    Properties:
      CatalogId: !Ref AWS::AccountId
      DatabaseName: !Ref CFNDatabaseName
      TableName: !Ref CFNTableName1
      PartitionInput:
        Values:
        - 1
        StorageDescriptor:
          OutputFormat: org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat
```

```
          Columns:
          - Name: mon
            Type: bigint
          InputFormat: org.apache.hadoop.mapred.TextInputFormat
          Location: s3://crawler-public-us-east-1/flight/2016/csv/mon=1/
          SerdeInfo:
            Parameters:
              field.delim: ","
            SerializationLibrary: org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe
# Partition 2
# Create an AWS Glue partition
  CFNPartitionMon2:
    DependsOn: CFNTableFlights
    Type: AWS::Glue::Partition
    Properties:
      CatalogId: !Ref AWS::AccountId
      DatabaseName: !Ref CFNDatabaseName
      TableName: !Ref CFNTableName1
      PartitionInput:
        Values:
        - 2
        StorageDescriptor:
          OutputFormat: org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat
          Columns:
          - Name: mon
            Type: bigint
          InputFormat: org.apache.hadoop.mapred.TextInputFormat
          Location: s3://crawler-public-us-east-1/flight/2016/csv/mon=2/
          SerdeInfo:
            Parameters:
              field.delim: ","
            SerializationLibrary: org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe
# Partition 3
# Create an AWS Glue partition
  CFNPartitionMon3:
    DependsOn: CFNTableFlights
    Type: AWS::Glue::Partition
    Properties:
      CatalogId: !Ref AWS::AccountId
      DatabaseName: !Ref CFNDatabaseName
      TableName: !Ref CFNTableName1
      PartitionInput:
        Values:
        - 3
```

```
        StorageDescriptor:
          OutputFormat: org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat
          Columns:
          - Name: mon
            Type: bigint
          InputFormat: org.apache.hadoop.mapred.TextInputFormat
          Location: s3://crawler-public-us-east-1/flight/2016/csv/mon=3/
          SerdeInfo:
            Parameters:
              field.delim: ","
            SerializationLibrary: org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe
# Partition 4
# Create an AWS Glue partition
  CFNPartitionMon4:
    DependsOn: CFNTableFlights
    Type: AWS::Glue::Partition
    Properties:
      CatalogId: !Ref AWS::AccountId
      DatabaseName: !Ref CFNDatabaseName
      TableName: !Ref CFNTableName1
      PartitionInput:
        Values:
        - 4
        StorageDescriptor:
          OutputFormat: org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat
          Columns:
          - Name: mon
            Type: bigint
          InputFormat: org.apache.hadoop.mapred.TextInputFormat
          Location: s3://crawler-public-us-east-1/flight/2016/csv/mon=4/
          SerdeInfo:
            Parameters:
              field.delim: ","
            SerializationLibrary: org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe
```

# CloudFormation Modello di esempio per un classificatore AWS Glue grok

Un classificatore AWS Glue determina lo schema dei dati. Un tipo di classificatore personalizzato usa un pattern grok per trovare la corrispondenza con i dati. Se il pattern corrisponde, il classificatore

personalizzato viene usato per creare lo schema della tabella e impostare `classification` sul valore impostato nella definizione del classificatore.

Questo esempio crea un classificatore che crea a sua volta uno schema con una colonna denominata `message` e imposta la classificazione su `greedy`.

```
---
AWSTemplateFormatVersion: '2010-09-09'
# Sample CFN YAML to demonstrate creating a classifier
#
# Parameters section contains names that are substituted in the Resources section
# These parameters are the names the resources created in the Data Catalog
Parameters:

# The name of the classifier to be created
  CFNClassifierName:
    Type: String
    Default: cfn-classifier-grok-one-column-1

#
#
# Resources section defines metadata for the Data Catalog
Resources:
# Create classifier that uses grok pattern to put all data in one column and classifies
 it as "greedy".
  CFNClassifierFlights:
    Type: AWS::Glue::Classifier
    Properties:
      GrokClassifier:
        #Grok classifier that puts all data in one column
        Name: !Ref CFNClassifierName
        Classification: greedy

        GrokPattern: "%{GREEDYDATA:message}"
        #CustomPatterns: none
```

# CloudFormation Modello di esempio per un classificatore JSON AWS Glue

Un classificatore AWS Glue determina lo schema dei dati. Un tipo di classificatore personalizzato utilizza una `JsonPath` stringa che definisce i dati JSON che il classificatore deve classificare. AWS Gluesupporta un sottoinsieme degli operatori per`JsonPath`, come descritto in Writing Custom Classifiers. JsonPath

Se il modello corrisponde, il classificatore personalizzato viene utilizzato per creare il tuo schema della tabella.

Questo esempio crea un classificatore che a sua volta crea uno schema con ogni record nella matrice `Records3` in un oggetto.

```
---
AWSTemplateFormatVersion: '2010-09-09'
# Sample CFN YAML to demonstrate creating a JSON classifier
#
# Parameters section contains names that are substituted in the Resources section
# These parameters are the names the resources created in the Data Catalog
Parameters:

# The name of the classifier to be created
  CFNClassifierName:
    Type: String
    Default: cfn-classifier-json-one-column-1

#
#
# Resources section defines metadata for the Data Catalog
Resources:
# Create classifier that uses a JSON pattern.
  CFNClassifierFlights:
    Type: AWS::Glue::Classifier
    Properties:
      JSONClassifier:
        #JSON classifier
        Name: !Ref CFNClassifierName
        JsonPath: $.Records3[*]
```

# CloudFormation Modello di esempio per un classificatore XML AWS Glue

Un classificatore AWS Glue determina lo schema dei dati. Un tipo di classificatore personalizzato specifica un tag XML per designare l'elemento che contiene ogni record in un documento XML sottoposto ad analisi. Se il pattern corrisponde, il classificatore personalizzato viene usato per creare lo schema della tabella e impostare `classification` sul valore impostato nella definizione del classificatore.

Questo esempio crea un classificatore che crea a sua volta uno schema con ciascun record nel tag `Record` e imposta la classificazione su `XML`.

```
---
AWSTemplateFormatVersion: '2010-09-09'
# Sample CFN YAML to demonstrate creating an XML classifier
#
# Parameters section contains names that are substituted in the Resources section
# These parameters are the names the resources created in the Data Catalog
Parameters:

# The name of the classifier to be created
  CFNClassifierName:
    Type: String
    Default: cfn-classifier-xml-one-column-1

#
#
# Resources section defines metadata for the Data Catalog
Resources:
# Create classifier that uses the XML pattern and classifies it as "XML".
  CFNClassifierFlights:
    Type: AWS::Glue::Classifier
    Properties:
      XMLClassifier:
        #XML classifier
        Name: !Ref CFNClassifierName
        Classification: XML
        RowTag: <Records>
```

# CloudFormation Modello di esempio per un AWS Glue crawler per Amazon S3

Un crawler AWS Glue crea nel catalogo dati tabelle di metadati che corrispondono ai dati. Puoi quindi usare queste definizioni di tabella come origini e target nei processi ETL.

Questo esempio crea un crawler, il ruolo IAM necessario e un database AWS Glue nel catalogo dati. Quando il crawler viene eseguito, assume il ruolo IAM e crea una tabella del database per i dati dei voli pubblici. La tabella viene creata con il prefisso "`cfn_sample_1_`". Il ruolo IAM creato da questo modello concede autorizzazioni globali. Potresti voler creare un ruolo personalizzato. Tramite questo classificatore non vengono definiti classificatori personalizzati. Per impostazione predefinita, vengono usati classificatori AWS Glue predefiniti.

Quando invii questo esempio alla CloudFormation console, devi confermare di voler creare il ruolo IAM.

```
---
AWSTemplateFormatVersion: '2010-09-09'
# Sample CFN YAML to demonstrate creating a crawler
#
# Parameters section contains names that are substituted in the Resources section
# These parameters are the names the resources created in the Data Catalog
Parameters:

# The name of the crawler to be created
  CFNCrawlerName:
    Type: String
    Default: cfn-crawler-flights-1
  CFNDatabaseName:
    Type: String
    Default: cfn-database-flights-1
  CFNTablePrefixName:
    Type: String
    Default: cfn_sample_1_
#
#
# Resources section defines metadata for the Data Catalog
Resources:
#Create IAM Role assumed by the crawler. For demonstration, this role is given all
 permissions.
```

```
 CFNRoleFlights:
   Type: AWS::IAM::Role
   Properties:
     AssumeRolePolicyDocument:
       Version: "2012-10-17"
       Statement:
         -
           Effect: "Allow"
           Principal:
             Service:
               - "glue.amazonaws.com"
           Action:
             - "sts:AssumeRole"
     Path: "/"
     Policies:
       -
         PolicyName: "root"
         PolicyDocument:
           Version: "2012-10-17"
           Statement:
             -
               Effect: "Allow"
               Action: "*"
               Resource: "*"
# Create a database to contain tables created by the crawler
 CFNDatabaseFlights:
   Type: AWS::Glue::Database
   Properties:
     CatalogId: !Ref AWS::AccountId
     DatabaseInput:
       Name: !Ref CFNDatabaseName
       Description: "AWS Glue container to hold metadata tables for the flights
crawler"
#Create a crawler to crawl the flights data on a public S3 bucket
 CFNCrawlerFlights:
   Type: AWS::Glue::Crawler
   Properties:
     Name: !Ref CFNCrawlerName
     Role: !GetAtt CFNRoleFlights.Arn
     #Classifiers: none, use the default classifier
     Description: AWS Glue crawler to crawl flights data
     #Schedule: none, use default run-on-demand
     DatabaseName: !Ref CFNDatabaseName
     Targets:
```

```
        S3Targets:
          # Public S3 bucket with the flights data
          - Path: "s3://crawler-public-us-east-1/flight/2016/csv"
      TablePrefix: !Ref CFNTablePrefixName
      SchemaChangePolicy:
        UpdateBehavior: "UPDATE_IN_DATABASE"
        DeleteBehavior: "LOG"
      Configuration: "{\"Version\":1.0,\"CrawlerOutput\":{\"Partitions\":
{\"AddOrUpdateBehavior\":\"InheritFromTable\"},\"Tables\":{\"AddOrUpdateBehavior\":
\"MergeNewColumns\"}}}"
```

# CloudFormation Modello di esempio per una AWS Glue connessione

Una connessione AWS Glue nel catalogo dati contiene le informazioni di rete e JDBC necessarie per la connessione a un database JDBC. Queste informazioni vengono usate per la connessione a un database JDBC per il crawling o l'esecuzione di processi ETL.

In questo esempio viene creata una connessione a un database Amazon RDS MySQL denominato `devdb`. Quando la connessione viene usata, è necessario fornire anche un ruolo IAM, le credenziali del database e i valori per la connessione di rete. Consulta i dettagli dei campi necessari nel modello.

```
---
AWSTemplateFormatVersion: '2010-09-09'
# Sample CFN YAML to demonstrate creating a connection
#
# Parameters section contains names that are substituted in the Resources section
# These parameters are the names the resources created in the Data Catalog
Parameters:

# The name of the connection to be created
  CFNConnectionName:
    Type: String
    Default: cfn-connection-mysql-flights-1
  CFNJDBCString:
    Type: String
    Default: "jdbc:mysql://xxx-mysql.yyyyyyyyyyyyyy.us-east-1.rds.amazonaws.com:3306/
devdb"
  CFNJDBCUser:
    Type: String
```

```
    Default: "master"
  CFNJDBCPassword:
    Type: String
    Default: "12345678"
    NoEcho: true
#
#
# Resources section defines metadata for the Data Catalog
Resources:
  CFNConnectionMySQL:
    Type: AWS::Glue::Connection
    Properties:
      CatalogId: !Ref AWS::AccountId
      ConnectionInput:
        Description: "Connect to MySQL database."
        ConnectionType: "JDBC"
        #MatchCriteria: none
        PhysicalConnectionRequirements:
          AvailabilityZone: "us-east-1d"
          SecurityGroupIdList:
           - "sg-7d52b812"
          SubnetId: "subnet-84f326ee"
        ConnectionProperties: {
          "JDBC_CONNECTION_URL": !Ref CFNJDBCString,
          "USERNAME": !Ref CFNJDBCUser,
          "PASSWORD": !Ref CFNJDBCPassword
        }
        Name: !Ref CFNConnectionName
```

# CloudFormation Modello di esempio per un'integrazione AWS Glue zero-ETL

AWSZero-ETL è un set di integrazioni completamente gestite che riducono al minimo la necessità di creare pipeline di dati ETL per casi d'uso comuni di acquisizione e replica.

Questo esempio crea un'integrazione zero-ETL dall'origine specificata alla destinazione.

```
---
AWSTemplateFormatVersion: '2010-09-09'
# Sample CFN YAML to demonstrate creating a zero-ETL integration in AWS Glue
```

```
#
# Parameters section contains names that are substituted in the Resources section
#
Parameters:

  # The name of the zero-ETL integration to be created
  IntegrationName:
    Type: String
  # The ARN for the source of the zero-ETL integration
  SourceArn:
    Type: String
  # The ARN for the target of the zero-ETL integration
  TargetArn:
    Type: String
#
#
Resources:
# Create an AWS Glue zero-ETL integration
  GlueIntegration:
    Type: AWS::Glue::Integration
    Properties:
      IntegrationName: !Ref IntegrationName
      Description: "AWS Glue zero-ETL integration"
      SourceArn: !Ref SourceArn
      TargetArn: !Ref TargetArn
      DataFilter: "include:table1"
      Tags:
        - Key: Purpose
          Value: GlueZeroETLIntegration
```

# CloudFormation Modello di esempio per un'integrazione AWS Glue zero-ETL con le proprietà delle risorse di integrazione

Un'integrazione AWS Glue zero-ETL richiede la definizione delle proprietà delle risorse per l'origine e la destinazione. Per l'origine, l'unica proprietà che deve essere definita è il ruolo IAM che l'integrazione utilizzerà per accedere alla AWS Glue connessione o al database DynamoDB. Per la destinazione, le proprietà che possono essere configurate includono il ruolo IAM che verrà utilizzato per accedere alla destinazione, la rete VPC in cui deve essere creata l'integrazione, il bus degli eventi che verrà utilizzato per configurare le notifiche degli eventi per l'integrazione e la chiave KMS che verrà utilizzata per la crittografia dei dati.

L'esempio seguente definisce le proprietà delle risorse di origine e di destinazione e quindi crea un'integrazione zero-ETL dall'origine alla destinazione.

```
---
AWSTemplateFormatVersion: '2010-09-09'
# Sample CFN YAML to demonstrate defining the integration resource properties and then
 creating a zero-ETL integration in AWS Glue
#
# Parameters section contains names that are substituted in the Resources section
#
Parameters:
  #The name of the zero-ETL integration to be created
  IntegrationName:
    Type: String
  # The ARN for the target of the zero-ETL integration
  TargetArn:
    Type: String
  # The ARN for the IAM role that will be used to access the target
  TargetRoleArn:
    Type: String
  # The ARN for the source of the zero-ETL integration
  SourceArn:
    Type: String
  # The ARN for the IAM role that will be used to access thesource
  SourceRoleArn:
    Type: String
#
#
Resources:
  # Integration Resource Property for zero-ETL target
  TargetIntegrationResourceProperty:
    Type: AWS::Glue::IntegrationResourceProperty
    Properties:
      ResourceArn: !Ref TargetArn
      TargetProcessingProperties:
        RoleArn: !Ref TargetRoleArn
      Tags:
        - Key: Purpose
          Value: TargetIrpTag

  # Integration Resource Property for zero-ETL source
  SourceIntegrationResourceProperty:
```

```
    Type: AWS::Glue::IntegrationResourceProperty
    Properties:
      ResourceArn: !Ref SourceArn
      SourceProcessingProperties:
        RoleArn: !Ref SourceRoleArn
      Tags:
        - Key: Purpose
          Value: SourceIRPTag

  # Create an AWS Glue zero-ETL integration
  GlueIntegration:
    Type: AWS::Glue::Integration
    Properties:
      IntegrationName: !Ref IntegrationName
      Description: "AWS Glue zero-ETL integration"
      SourceArn: !Ref SourceArn
      TargetArn: !Ref TargetArn
      DataFilter: "include:table1"
      Tags:
        - Key: Purpose
          Value: GlueZeroETLIntegration
```

# CloudFormation Modello di esempio per un crawler per JDBC AWS Glue

Un crawler AWS Glue crea nel catalogo dati tabelle di metadati che corrispondono ai dati. Puoi quindi usare queste definizioni di tabella come origini e target nei processi ETL.

Questo esempio crea un crawler, il ruolo IAM necessario e un database AWS Glue nel catalogo dati. Quando il crawler viene eseguito, assume il ruolo IAM e crea una tabella nel database per i dati dei voli pubblici archiviati in un database MySQL. La tabella viene creata con il prefisso "`cfn_jdbc_1_`". Il ruolo IAM creato da questo modello concede autorizzazioni globali. Potresti voler creare un ruolo personalizzato. Per i dati JDBC non è possibile definire classificatori personalizzati. Per impostazione predefinita, vengono usati classificatori AWS Glue predefiniti.

Quando invii questo esempio alla CloudFormation console, devi confermare di voler creare il ruolo IAM.

```
---
AWSTemplateFormatVersion: '2010-09-09'
```

```
# Sample CFN YAML to demonstrate creating a crawler
#
# Parameters section contains names that are substituted in the Resources section
# These parameters are the names the resources created in the Data Catalog
Parameters:

# The name of the crawler to be created
  CFNCrawlerName:
    Type: String
    Default: cfn-crawler-jdbc-flights-1
# The name of the database to be created to contain tables
  CFNDatabaseName:
    Type: String
    Default: cfn-database-jdbc-flights-1
# The prefix for all tables crawled and created
  CFNTablePrefixName:
    Type: String
    Default: cfn_jdbc_1_
# The name of the existing connection to the MySQL database
  CFNConnectionName:
    Type: String
    Default: cfn-connection-mysql-flights-1
# The name of the JDBC path (database/schema/table) with wildcard (%) to crawl
  CFNJDBCPath:
    Type: String
    Default: saldev/%
#
#
# Resources section defines metadata for the Data Catalog
Resources:
#Create IAM Role assumed by the crawler. For demonstration, this role is given all
 permissions.
  CFNRoleFlights:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: "2012-10-17"
        Statement:
          -
            Effect: "Allow"
            Principal:
              Service:
                - "glue.amazonaws.com"
            Action:
```

```
              - "sts:AssumeRole"
      Path: "/"
      Policies:
        -
          PolicyName: "root"
          PolicyDocument:
            Version: "2012-10-17"
            Statement:
              -
                Effect: "Allow"
                Action: "*"
                Resource: "*"
# Create a database to contain tables created by the crawler
 CFNDatabaseFlights:
   Type: AWS::Glue::Database
   Properties:
     CatalogId: !Ref AWS::AccountId
     DatabaseInput:
       Name: !Ref CFNDatabaseName
       Description: "AWS Glue container to hold metadata tables for the flights
crawler"
#Create a crawler to crawl the flights data in MySQL database
 CFNCrawlerFlights:
   Type: AWS::Glue::Crawler
   Properties:
     Name: !Ref CFNCrawlerName
     Role: !GetAtt CFNRoleFlights.Arn
     #Classifiers: none, use the default classifier
     Description: AWS Glue crawler to crawl flights data
     #Schedule: none, use default run-on-demand
     DatabaseName: !Ref CFNDatabaseName
     Targets:
       JdbcTargets:
         # JDBC MySQL database with the flights data
         - ConnectionName: !Ref CFNConnectionName
           Path: !Ref CFNJDBCPath
         #Exclusions: none
     TablePrefix: !Ref CFNTablePrefixName
     SchemaChangePolicy:
       UpdateBehavior: "UPDATE_IN_DATABASE"
       DeleteBehavior: "LOG"
```

```
   Configuration: "{\"Version\":1.0,\"CrawlerOutput\":{\"Partitions\":
{\"AddOrUpdateBehavior\":\"InheritFromTable\"},\"Tables\":{\"AddOrUpdateBehavior\":
\"MergeNewColumns\"}}}"
```

# CloudFormation Modello di esempio per un AWS Glue lavoro da Amazon S3 ad Amazon S3

Un processo AWS Glue nel catalogo dati contiene i valori dei parametri necessari per eseguire uno script in AWS Glue.

Questo esempio crea un processo che legge i dati dei voli da un bucket Amazon S3 in formato `csv` e li scrive in un file Parquet in Amazon S3. Lo script eseguito da questo processo deve esistere già. Puoi generare uno script ETL per l'ambiente con la console AWS Glue. Quando questo processo viene eseguito, è necessario fornire anche un ruolo IAM con le autorizzazioni appropriate.

I valori dei parametri comuni sono mostrati nel modello. Ad esempio, il valore predefinito di `AllocatedCapacity` (DPUs) è 5.

```
---
AWSTemplateFormatVersion: '2010-09-09'
# Sample CFN YAML to demonstrate creating a job using the public flights S3 table in a
 public bucket
#
# Parameters section contains names that are substituted in the Resources section
# These parameters are the names the resources created in the Data Catalog
Parameters:

# The name of the job to be created
  CFNJobName:
    Type: String
    Default: cfn-job-S3-to-S3-2
# The name of the IAM role that the job assumes. It must have access to data, script,
 temporary directory
  CFNIAMRoleName:
    Type: String
    Default: AWSGlueServiceRoleGA
# The S3 path where the script for this job is located
  CFNScriptLocation:
    Type: String
    Default: s3://aws-glue-scripts-123456789012-us-east-1/myid/sal-job-test2
```

```
#
#
# Resources section defines metadata for the Data Catalog
Resources:
# Create job to run script which accesses flightscsv table and write to S3 file as
 parquet.
# The script already exists and is called by this job
  CFNJobFlights:
    Type: AWS::Glue::Job
    Properties:
      Role: !Ref CFNIAMRoleName
      #DefaultArguments: JSON object
      # If script written in Scala, then set DefaultArguments={'--job-language';
 'scala', '--class': 'your scala class'}
      #Connections:  No connection needed for S3 to S3 job
      #  ConnectionsList
      #MaxRetries: Double
      Description: Job created with CloudFormation
      #LogUri: String
      Command:
        Name: glueetl
        ScriptLocation: !Ref CFNScriptLocation
             # for access to directories use proper IAM role with permission to buckets
 and folders that begin with "aws-glue-"
             # script uses temp directory from job definition if required (temp
 directory not used S3 to S3)
             # script defines target for output as s3://aws-glue-target/sal
      AllocatedCapacity: 5
      ExecutionProperty:
        MaxConcurrentRuns: 1
      Name: !Ref CFNJobName
```

# CloudFormation Modello di esempio per un AWS Glue job da JDBC ad Amazon S3

Un processo AWS Glue nel catalogo dati contiene i valori dei parametri necessari per eseguire uno script in AWS Glue.

Questo esempio crea un processo che legge i dati dei voli da un database JDBC MySQL in base a quanto definito dalla connessione denominata `cfn-connection-mysql-flights-1` e li scrive in un file Parquet in Amazon S3. Lo script eseguito da questo processo deve esistere già. Puoi generare

uno script ETL per l'ambiente con la console AWS Glue. Quando questo processo viene eseguito, è necessario fornire anche un ruolo IAM con le autorizzazioni appropriate.

I valori dei parametri comuni sono mostrati nel modello. Ad esempio, il valore predefinito di `AllocatedCapacity` (DPUs) è 5.

```
---
AWSTemplateFormatVersion: '2010-09-09'
# Sample CFN YAML to demonstrate creating a job using a MySQL JDBC DB with the flights
 data to an S3 file
#
# Parameters section contains names that are substituted in the Resources section
# These parameters are the names the resources created in the Data Catalog
Parameters:

# The name of the job to be created
  CFNJobName:
    Type: String
    Default: cfn-job-JDBC-to-S3-1
# The name of the IAM role that the job assumes. It must have access to data, script,
 temporary directory
  CFNIAMRoleName:
    Type: String
    Default: AWSGlueServiceRoleGA
# The S3 path where the script for this job is located
  CFNScriptLocation:
    Type: String
    Default: s3://aws-glue-scripts-123456789012-us-east-1/myid/sal-job-dec4a
# The name of the connection used for JDBC data source
  CFNConnectionName:
    Type: String
    Default: cfn-connection-mysql-flights-1
#
#
# Resources section defines metadata for the Data Catalog
Resources:
# Create job to run script which accesses JDBC flights table via a connection and write
 to S3 file as parquet.
# The script already exists and is called by this job
  CFNJobFlights:
    Type: AWS::Glue::Job
    Properties:
```

```
      Role: !Ref CFNIAMRoleName
      #DefaultArguments: JSON object
      # For example, if required by script, set temporary directory as
 DefaultArguments={'--TempDir'; 's3://aws-glue-temporary-xyc/sal'}
      Connections:
        Connections:
        - !Ref CFNConnectionName
      #MaxRetries: Double
      Description: Job created with CloudFormation using existing script
      #LogUri: String
      Command:
        Name: glueetl
        ScriptLocation: !Ref CFNScriptLocation
             # for access to directories use proper IAM role with permission to buckets
 and folders that begin with "aws-glue-"
             # if required, script defines temp directory as argument TempDir and used
 in script like redshift_tmp_dir = args["TempDir"]
             # script defines target for output as s3://aws-glue-target/sal
      AllocatedCapacity: 5
      ExecutionProperty:
        MaxConcurrentRuns: 1
      Name: !Ref CFNJobName
```

# CloudFormation Modello di esempio per un trigger su richiesta AWS Glue

Un trigger AWS Glue nel catalogo dati contiene i valori dei parametri necessari per avviare l'esecuzione di un processo quando viene attivato il trigger. Un trigger on demand viene attivato quando lo si abilita.

In questo esempio viene creato un trigger on demand che avvia un processo denominato `cfn-job-S3-to-S3-1`.

```
---
AWSTemplateFormatVersion: '2010-09-09'
# Sample CFN YAML to demonstrate creating an on-demand trigger
#
# Parameters section contains names that are substituted in the Resources section
# These parameters are the names the resources created in the Data Catalog
Parameters:
```

```
  # The existing job to be started by this trigger
  CFNJobName:
    Type: String
    Default: cfn-job-S3-to-S3-1
  # The name of the trigger to be created
  CFNTriggerName:
    Type: String
    Default: cfn-trigger-ondemand-flights-1
#
# Resources section defines metadata for the Data Catalog
# Sample CFN YAML to demonstrate creating an on-demand trigger for a job
Resources:
# Create trigger to run an existing job (CFNJobName) on an on-demand schedule.
  CFNTriggerSample:
    Type: AWS::Glue::Trigger
    Properties:
      Name:
        Ref: CFNTriggerName
      Description: Trigger created with CloudFormation
      Type: ON_DEMAND
      Actions:
        - JobName: !Ref CFNJobName
        # Arguments: JSON object
      #Schedule:
      #Predicate:
```

# CloudFormation Modello di esempio per un trigger AWS Glue pianificato

Un trigger AWS Glue nel catalogo dati contiene i valori dei parametri necessari per avviare l'esecuzione di un processo quando viene attivato il trigger. Un trigger pianificato viene attivato quando è abilitato e il timer cron raggiunge il valore definito.

In questo esempio viene creato un trigger pianificato che avvia un processo denominato `cfn-job-S3-to-S3-1`. Il timer è un'espressione cron per l'esecuzione del processo ogni 10 minuti nei giorni feriali.

```
---
AWSTemplateFormatVersion: '2010-09-09'
# Sample CFN YAML to demonstrate creating a scheduled trigger
```

```
#
# Parameters section contains names that are substituted in the Resources section
# These parameters are the names the resources created in the Data Catalog
Parameters:
  # The existing job to be started by this trigger
  CFNJobName:
    Type: String
    Default: cfn-job-S3-to-S3-1
  # The name of the trigger to be created
  CFNTriggerName:
    Type: String
    Default: cfn-trigger-scheduled-flights-1
#
# Resources section defines metadata for the Data Catalog
# Sample CFN YAML to demonstrate creating a scheduled trigger for a job
#
Resources:
# Create trigger to run an existing job (CFNJobName) on a cron schedule.
  TriggerSample1CFN:
    Type: AWS::Glue::Trigger
    Properties:
      Name:
        Ref: CFNTriggerName
      Description: Trigger created with CloudFormation
      Type: SCHEDULED
      Actions:
        - JobName: !Ref CFNJobName
        # Arguments: JSON object
      # # Run the trigger every 10 minutes on Monday to Friday
      Schedule: cron(0/10 * ? * MON-FRI *)
      #Predicate:
```

# CloudFormation Modello di esempio per un trigger AWS Glue condizionale

Un trigger AWS Glue nel catalogo dati contiene i valori dei parametri necessari per avviare l'esecuzione di un processo quando viene attivato il trigger. Un trigger condizionale viene attivato quando è abilitato e le relative condizioni vengono soddisfatte, ad esempio un processo viene completato correttamente.

In questo esempio viene creato un trigger condizionale che avvia un processo denominato `cfn-job-S3-to-S3-1`. Questo processo viene avviato quando il processo denominato `cfn-job-S3-to-S3-2` viene completato correttamente.

```
---
AWSTemplateFormatVersion: '2010-09-09'
# Sample CFN YAML to demonstrate creating a conditional trigger for a job, which starts
 when another job completes
#
# Parameters section contains names that are substituted in the Resources section
# These parameters are the names the resources created in the Data Catalog
Parameters:
  # The existing job to be started by this trigger
  CFNJobName:
    Type: String
    Default: cfn-job-S3-to-S3-1
  # The existing job that when it finishes causes trigger to fire
  CFNJobName2:
    Type: String
    Default: cfn-job-S3-to-S3-2
  # The name of the trigger to be created
  CFNTriggerName:
    Type: String
    Default: cfn-trigger-conditional-1
#
Resources:
# Create trigger to run an existing job (CFNJobName) when another job completes
 (CFNJobName2).
  CFNTriggerSample:
    Type: AWS::Glue::Trigger
    Properties:
      Name:
        Ref: CFNTriggerName
      Description: Trigger created with CloudFormation
      Type: CONDITIONAL
      Actions:
        - JobName: !Ref CFNJobName
        # Arguments: JSON object
      #Schedule: none
      Predicate:
        #Value for Logical is required if more than 1 job listed in Conditions
        Logical: AND
```

```
        Conditions:
          - LogicalOperator: EQUALS
            JobName: !Ref CFNJobName2
            State: SUCCEEDED
```

# CloudFormation Modello di esempio per un endpoint di AWS Glue sviluppo

Una trasformazione basata su machine learning di AWS Glue è una trasformazione personalizzata per ripulire i dati. Attualmente è disponibile una trasformazione denominata FindMatches. La FindMatches trasformazione consente di identificare i record duplicati o corrispondenti nel set di dati, anche quando i record non hanno un identificatore univoco comune e nessun campo corrisponde esattamente.

Questo esempio mostra come creare una trasformazione basata su machine learning. Per ulteriori informazioni sui parametri necessari per creare una trasformazione basata su machine learning, consulta Record di abbinamento con AWS Lake Formation FindMatches.

```
---
AWSTemplateFormatVersion: '2010-09-09'
# Sample CFN YAML to demonstrate creating a machine learning transform
#
# Resources section defines metadata for the machine learning transform
Resources:
  MyMLTransform:
    Type: "AWS::Glue::MLTransform"
    Condition: "isGlueMLGARegion"
    Properties:
      Name: !Sub "MyTransform"
      Description: "The bestest transform ever"
      Role: !ImportValue MyMLTransformUserRole
      GlueVersion: "1.0"
      WorkerType: "Standard"
      NumberOfWorkers: 5
      Timeout: 120
      MaxRetries: 1
      InputRecordTables:
        GlueTables:
          - DatabaseName: !ImportValue MyMLTransformDatabase
            TableName: !ImportValue MyMLTransformTable
```

```
      TransformParameters:
        TransformType: "FIND_MATCHES"
        FindMatchesParameters:
          PrimaryKeyColumnName: "testcolumn"
          PrecisionRecallTradeoff: 0.5
          AccuracyCostTradeoff: 0.5
          EnforceProvidedLabels: True
      Tags:
        key1: "value1"
        key2: "value2"
      TransformEncryption:
        TaskRunSecurityConfigurationName: !ImportValue
 MyMLTransformSecurityConfiguration
        MLUserDataEncryption:
          MLUserDataEncryptionMode: "SSE-KMS"
          KmsKeyId: !ImportValue MyMLTransformEncryptionKey
```

# CloudFormation Modello di esempio per un set di regole AWS Glue Data Quality

Un set di regole per la qualità AWS Glue dei dati contiene regole che possono essere valutate su una tabella all'interno del Data Catalog. Una volta che il set di regole è stato inserito nella tabella di destinazione, è possibile accedere a Catalogo dati ed eseguire una valutazione che esamina i dati in base a tali regole all'interno del set di regole. Queste regole spaziano dalla valutazione del conteggio delle righe alla valutazione dell'integrità referenziale dei dati.

L'esempio seguente è un CloudFormation modello che crea un set di regole con una varietà di regole sulla tabella di destinazione specificata.

```
AWSTemplateFormatVersion: '2010-09-09'
# Sample CFN YAML to demonstrate creating a DataQualityRuleset
#
# Parameters section contains names that are substituted in the Resources section
# These parameters are the names the resources created in the Data Catalog
Parameters:

  # The name of the ruleset to be created
  RulesetName:
    Type: String
    Default: "CFNRulesetName"
  RulesetDescription:
```

```
    Type: String
    Default: "CFN DataQualityRuleset"
  # Rules that will be associated with this ruleset
  Rules:
    Type: String
    Default: 'Rules = [
        RowCount > 100,
        IsUnique "id",
        IsComplete "nametype"
        ]'
  # Name of database and table within Data Catalog which the ruleset will
  # be applied too
  DatabaseName:
    Type: String
    Default: "ExampleDatabaseName"
  TableName:
    Type: String
    Default: "ExampleTableName"

# Resources section defines metadata for the Data Catalog
Resources:
  # Creates a Data Quality ruleset under specified rules
  DQRuleset:
    Type: AWS::Glue::DataQualityRuleset
    Properties:
      Name: !Ref RulesetName
      Description: !Ref RulesetDescription
      # The String within rules must be formatted in DQDL, a language
      # used specifically to make rules
      Ruleset: !Ref Rules
      # The targeted table must exist within Data Catalog alongside
      # the correct database
      TargetTable:
        DatabaseName: !Ref DatabaseName
        TableName: !Ref TableName
```

# CloudFormation Modello di esempio per un AWS Glue Data Quality set di regole con scheduler EventBridge

Un set di regole per la qualità AWS Glue dei dati contiene regole che possono essere valutate su una tabella all'interno del Data Catalog. Una volta che il set di regole è stato inserito nella tabella di destinazione, è possibile accedere a Catalogo dati ed eseguire una valutazione che esamina i

dati in base a tali regole all'interno del set di regole. Invece di dover accedere manualmente al Data Catalog per valutare il set di regole, puoi anche aggiungere uno EventBridge Scheduler all'interno del nostro CloudFormation modello per pianificare queste valutazioni del set di regole per te in base a un intervallo di tempo.

L'esempio seguente è un CloudFormation modello che crea un set di regole di Data Quality e uno EventBridge Scheduler per valutare il suddetto set di regole ogni cinque minuti.

```
AWSTemplateFormatVersion: '2010-09-09'
# Sample CFN YAML to demonstrate creating a DataQualityRuleset
#
# Parameters section contains names that are substituted in the Resources section
# These parameters are the names the resources created in the Data Catalog
Parameters:

  # The name of the ruleset to be created
  RulesetName:
    Type: String
    Default: "CFNRulesetName"
  # Rules that will be associated with this Ruleset
  Rules:
    Type: String
    Default: 'Rules = [
        RowCount > 100,
        IsUnique "id",
        IsComplete "nametype"
        ]'
  # The name of the Schedule to be created
  ScheduleName:
    Type: String
    Default: "ScheduleDQRulsetEvaluation"
  # This expression determines the rate at which the Schedule will evaluate
  # your data using the above ruleset
  ScheduleRate:
    Type: String
    Default: "rate(5 minutes)"
  # The Request that being sent must match the details of the Data Quality Ruleset
  ScheduleRequest:
    Type: String
    Default: '
        { "DataSource": { "GlueTable": { "DatabaseName": "ExampleDatabaseName",
         "TableName": "ExampleTableName" } },
         "Role": "role/AWSGlueServiceRoleDefault",
```

```
          "RulesetNames": [ ""CFNRulesetName"" ] }
        '

# Resources section defines metadata for the Data Catalog
Resources:
  # Creates a Data Quality ruleset under specified rules
  DQRuleset:
    Type: AWS::Glue::DataQualityRuleset
    Properties:
      Name: !Ref RulesetName
      Description: "CFN DataQualityRuleset"
      # The String within rules must be formatted in DQDL, a language
      # used specifically to make rules
      Ruleset: !Ref Rules
      # The targeted table must exist within Data Catalog alongside
      # the correct database
      TargetTable:
        DatabaseName: "ExampleDatabaseName"
        TableName: "ExampleTableName"
  # Create a Scheduler to schedule evaluation runs on the above ruleset
  ScheduleDQEval:
    Type: AWS::Scheduler::Schedule
    Properties:
      Name: !Ref ScheduleName
      Description: "Schedule DataQualityRuleset Evaluations"
      FlexibleTimeWindow:
        Mode: "OFF"
      ScheduleExpression: !Ref ScheduleRate
      ScheduleExpressionTimezone: "America/New_York"
      State: "ENABLED"
      Target:
        # The ARN is the API that will be run, since we want to evaluate our ruleset
        # we want this specific ARN
        Arn: "arn:aws:scheduler:::aws-sdk:glue:startDataQualityRulesetEvaluationRun"
        # Your RoleArn must have approval to schedule
        RoleArn: "arn:aws:iam::123456789012:role/AWSGlueServiceRoleDefault"
        # This is the Request that is being sent to the Arn
        Input: '
        { "DataSource": { "GlueTable": { "DatabaseName": "sampledb", "TableName":
 "meteorite" } },
         "Role": "role/AWSGlueServiceRoleDefault",
          "RulesetNames": [ "TestCFN" ] }
        '
```

# Modello AWS Glue di esempio per un endpoint di sviluppo CloudFormation

Un endpoint di sviluppo AWS Glue è un ambiente che puoi usare per sviluppare e testare gli script AWS Glue.

In questo esempio viene creato un endpoint di sviluppo con i valori dei parametri di rete minimi necessari per la creazione. Per ulteriori informazioni sui parametri necessari per configurare un endpoint di sviluppo, consulta Creazione di reti per lo sviluppo per AWS Glue.

Per creare l'endpoint di sviluppo, puoi fornire un ARN (Amazon Resource Name) di un ruolo IAM esistente. Fornisci una chiave pubblica RSA valida e tieni a disposizione la chiave privata corrispondente se prevedi di creare un server notebook nell'endpoint di sviluppo.

> **Note**
>
> Effettui la gestione di qualsiasi server notebook che hai creato e che è associato a un endpoint di sviluppo. Pertanto, se si elimina l'endpoint di sviluppo, per eliminare il server notebook, è necessario eliminare lo CloudFormation stack sulla console. CloudFormation

```
---
AWSTemplateFormatVersion: '2010-09-09'
# Sample CFN YAML to demonstrate creating a development endpoint
#
# Parameters section contains names that are substituted in the Resources section
# These parameters are the names the resources created in the Data Catalog
Parameters:

# The name of the crawler to be created
  CFNEndpointName:
    Type: String
    Default: cfn-devendpoint-1
  CFNIAMRoleArn:
    Type: String
    Default: arn:aws:iam::123456789012/role/AWSGlueServiceRoleGA
#
#
# Resources section defines metadata for the Data Catalog
```

```
Resources:
  CFNDevEndpoint:
    Type: AWS::Glue::DevEndpoint
    Properties:
      EndpointName: !Ref CFNEndpointName
      #ExtraJarsS3Path: String
      #ExtraPythonLibsS3Path: String
      NumberOfNodes: 5
      PublicKey: ssh-rsa public.....key myuserid-key
      RoleArn: !Ref CFNIAMRoleArn
      SecurityGroupIds:
        - sg-64986c0b
      SubnetId: subnet-c67cccac
```

# AWS Glue guida alla programmazione

Uno script contiene il codice che estrae i dati dalle fonti, li trasforma e li carica in obiettivi. AWS Glue esegue uno script quando avvia un processo.

AWS Glue Gli script ETL sono codificati in Python o Scala. Sebbene tutti i tipi di processo possano essere scritti in Python, in AWS Glue per Spark i processi possono essere scritti anche in Scala. Quando generi automaticamente la logica del codice sorgente per il tuo job in AWS Glue Studio, viene creato uno script. Puoi modificare questo script oppure puoi fornire il tuo script per elaborare il lavoro ETL.

## Fornire i propri script personalizzati

Gli script eseguono le operazioni di estrazione, trasformazione e caricamento (ETL). AWS Glue Uno script viene creato quando si genera automaticamente la logica del codice origine per un processo. Puoi modificare questo script generato oppure fornire il tuo script personalizzato.

Per fornire uno script personalizzato AWS Glue, segui questi passaggi generali:

1. Accedi Console di gestione AWS e apri la AWS Glue console all'indirizzo https://console.aws.amazon.com/glue/.
2. Scegli la scheda Processi ETL, quindi visualizza la sezione Crea processo. Scegli un'opzione per l'editor di script.
3. In This job runs (Questo processo viene eseguito), seleziona una delle opzioni seguenti:
   - Creazione di un nuovo script con codice boilerplate
   - Caricamento e modifica di uno script esistente
4. Nella schermata Dettagli del processo, seleziona il Ruolo IAM richiesto per l'esecuzione dello script personalizzato. Per ulteriori informazioni, consulta Gestione delle identità e degli accessi per AWS Glue.
5. Scegli qualsiasi connessione a cui fa riferimento lo script. Questi oggetti sono richiesti per connettersi ai datastore JDBC necessari.

   Un'interfaccia di rete elastica è un'interfaccia di rete virtuale che è possibile collegare a un'istanza in un Virtual Private Cloud (VPC). Scegli l'interfaccia di rete elastica necessaria per connetterti al datastore utilizzato nello script.

6. Fornisci i dettagli di configurazione aggiuntivi, inclusi i parametri, specifici per il tuo tipo di processo. Per ulteriori informazioni sulla configurazione in base al tipo di processo, consulta la sezione Creazione di processi ETL visivi.
7. Nella scheda Script, incolla o scrivi lo script personalizzato.

Utilizza il contenuto di questa sezione per guidare il processo di scrittura dello script personalizzato.

Per ulteriori informazioni sull'aggiunta di lavori in AWS Glue, consultaCreazione di processi ETL visivi.

Per step-by-step ulteriori informazioni, consulta il tutorial Aggiungi lavoro nella AWS Glue console.

# Script di programmazione Spark

AWS Glue semplifica la scrittura o la generazione automatica di script di estrazione, trasformazione e caricamento (ETL), oltre a testarli ed eseguirli. Questa sezione descrive le estensioni di Apache Spark introdotte da AWS Glue e fornisce esempi di come codificare ed eseguire gli script ETL in Python e Scala.

> **Important**
>
> Diverse versioni di AWS Glue supportano diverse versioni di Apache Spark. Lo script personalizzato deve essere compatibile con la versione di Apache Spark supportata. Per ulteriori informazioni sulle versioni di AWS Glue, consultare Glue version job property.

Argomenti

- Tutorial: Scrivere uno script di AWS Glue for Spark
- Programma gli script ETL di AWS Glue in PySpark
- Programmazione di script ETL AWS Glue in Scala
- Funzionalità e ottimizzazioni per la programmazione degli script ETL di AWS Glue per Spark

## Tutorial: Scrivere uno script di AWS Glue for Spark

Questo tutorial ti introduce al processo di scrittura degli script AWS Glue. È possibile eseguire script in base a una pianificazione con processi o in modo interattivo con sessioni. Per ulteriori informazioni

sui processi, consultare Creazione di processi ETL visivi. Per ulteriori informazioni, sulle sessioni interattive, consultare the section called "Panoramica delle sessioni interattive in AWS Glue".

L'editor visivo di AWS Glue Studio offre un'interfaccia grafica senza codice per la creazione di lavori AWS Glue. AWS Glue gli script ai lavori visivi. Danno accesso al set esteso di strumenti disponibili per lavorare con i programmi Apache Spark. Puoi accedere a Spark APIs native e alle librerie AWS Glue che facilitano i flussi di lavoro di estrazione, trasformazione e caricamento (ETL) dall'interno di uno script AWS Glue.

In questo tutorial, si estrae, trasforma e carica un set di dati di multe per il parcheggio. Lo script che esegue questo lavoro è identico nella forma e nella funzione a quello generato in Making ETL easy with AWS Glue Studio sul AWS Big Data Blog, che introduce l'editor visivo di AWS Glue Studio. Eseguendo questo script in un job, è possibile confrontarlo con i lavori visivi e vedere come funzionano gli script AWS Glue ETL. Questo ti prepara a utilizzare funzionalità aggiuntive che non sono ancora disponibili nei processi visivi.

Questo tutorial utilizza il linguaggio e le librerie Python. Funzionalità simili sono disponibili in Scala. Dopo aver seguito questo tutorial, dovresti essere in grado di generare e ispezionare uno script Scala di esempio per capire come eseguire il processo di scrittura dello script Scala AWS Glue ETL.

## Prerequisiti

Di seguito sono elencati i requisiti per questo tutorial:

- Gli stessi prerequisiti del post del blog di AWS Glue Studio, che ti spiega come eseguire un CloudFormation modello.

  Questo modello utilizza il AWS Glue Data Catalog per gestire il set di dati dei biglietti di parcheggio disponibile in`s3://aws-bigdata-blog/artifacts/gluestudio/`. Questa operazione crea le risorse seguenti, a cui verrà fatto riferimento:
- AWS Glue StudioRuolo: il ruolo IAM per eseguire i processi AWS Glue
- AWS Glue StudioBucket Amazon S3: il nome del bucket Amazon S3 per archiviare i file relativi al blog
- AWS Glue StudioTicketsYYZDB: il database del Catalogo dati di AWS Glue
- AWS Glue StudioTableTickets— Tabella Data Catalog da utilizzare come fonte
- AWS Glue StudioTableTrials— Tabella Data Catalog da utilizzare come fonte
- AWS Glue StudioParkingTicketCount — Tabella Data Catalog da utilizzare come destinazione

- Lo script generato nel post del blog di AWS Glue Studio. Nel caso in cui il post venga modificato, è possibile trovare lo script anche nel testo seguente.

### Generazione di uno script di esempio

Puoi usare l'editor visivo di AWS Glue Studio come potente strumento di generazione di codice per creare uno scaffold per lo script che desideri scrivere. Questo strumento verrà utilizzato per creare uno script di esempio.

Se si preferisce saltare queste fasi, è possibile utilizzare lo script seguente.

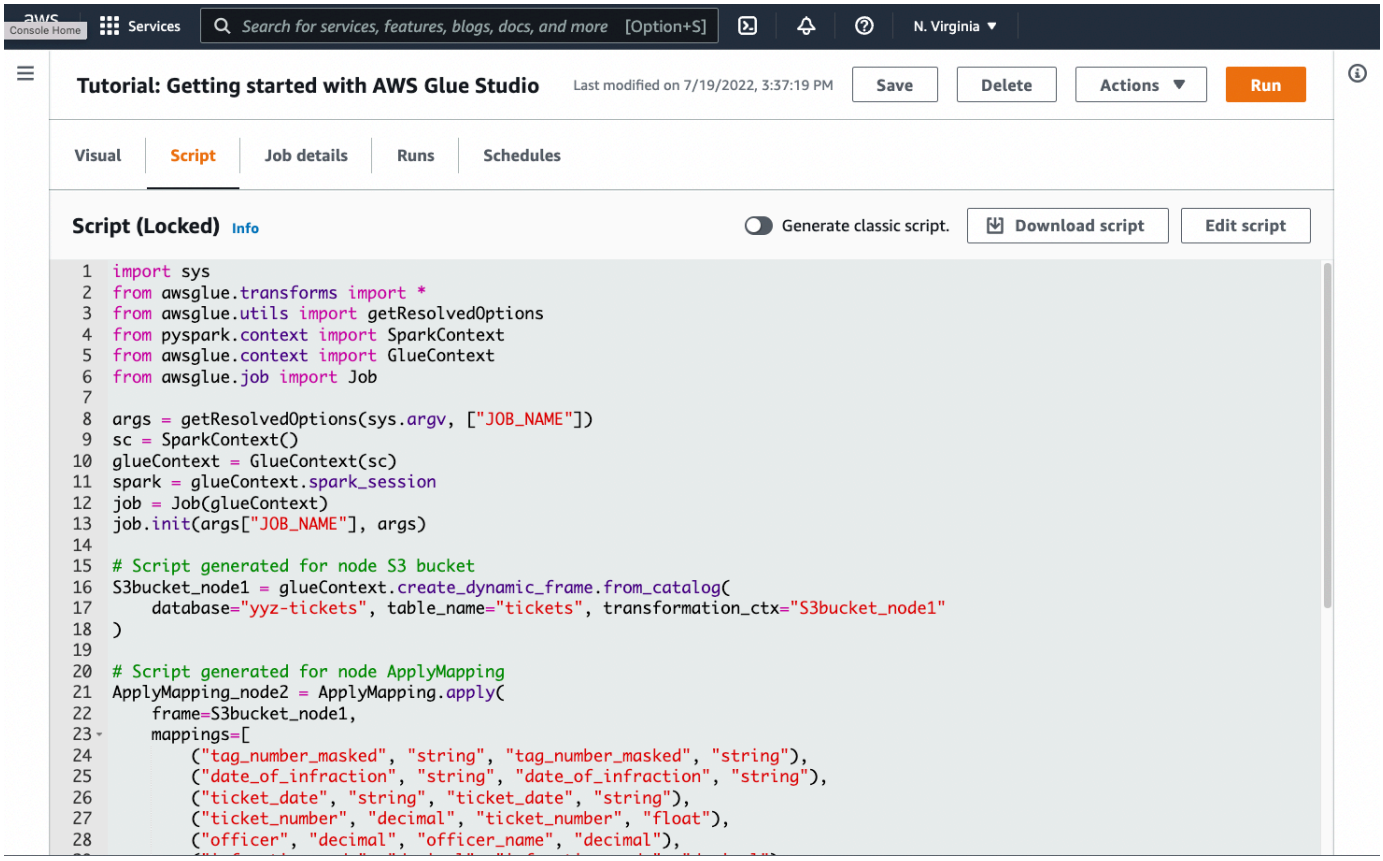
### Script di esempio per il tutorial

```
import sys
from awsglue.transforms import *
from awsglue.utils import getResolvedOptions
from pyspark.context import SparkContext
from awsglue.context import GlueContext
from awsglue.job import Job

args = getResolvedOptions(sys.argv, ["JOB_NAME"])
sc = SparkContext()
glueContext = GlueContext(sc)
spark = glueContext.spark_session
job = Job(glueContext)
job.init(args["JOB_NAME"], args)

# Script generated for node S3 bucket
S3bucket_node1 = glueContext.create_dynamic_frame.from_catalog(
    database="yyz-tickets", table_name="tickets", transformation_ctx="S3bucket_node1"
)

# Script generated for node ApplyMapping
ApplyMapping_node2 = ApplyMapping.apply(
    frame=S3bucket_node1,
    mappings=[
        ("tag_number_masked", "string", "tag_number_masked", "string"),
        ("date_of_infraction", "string", "date_of_infraction", "string"),
        ("ticket_date", "string", "ticket_date", "string"),
        ("ticket_number", "decimal", "ticket_number", "float"),
        ("officer", "decimal", "officer_name", "decimal"),
        ("infraction_code", "decimal", "infraction_code", "decimal"),
```

```
        ("infraction_description", "string", "infraction_description", "string"),
        ("set_fine_amount", "decimal", "set_fine_amount", "float"),
        ("time_of_infraction", "decimal", "time_of_infraction", "decimal"),
    ],
    transformation_ctx="ApplyMapping_node2",
)

# Script generated for node S3 bucket
S3bucket_node3 = glueContext.write_dynamic_frame.from_options(
    frame=ApplyMapping_node2,
    connection_type="s3",
    format="glueparquet",
    connection_options={"path": "s3://DOC-EXAMPLE-BUCKET", "partitionKeys": []},
    format_options={"compression": "gzip"},
    transformation_ctx="S3bucket_node3",
)

job.commit()
```

Generazione di uno script di esempio

1. Completa il tutorial di AWS Glue Studio. Per completare questo tutorial, vedi Creazione di un lavoro in AWS Glue Studio da un lavoro di esempio.
2. Passare alla scheda Script nella pagina del processo, come mostrato nello screenshot seguente:




3. Copiare tutto il contenuto della scheda Script. Impostando il linguaggio dello script nella sezione Job details (Dettagli del processo), è possibile passare dalla generazione di codice Python a Scala e viceversa.

## Passaggio 1. Creare un processo e incollare lo script

In questo passaggio, si crea un lavoro AWS Glue in Console di gestione AWS. Questo imposta una configurazione che consente a AWS Glue di eseguire lo script. e si crea contemporaneamente uno spazio dove archiviarlo e modificarlo.

Come creare un processo

1. Nel Console di gestione AWS, vai alla landing page di AWS Glue.
2. Nel riquadro di navigazione laterale, scegliere Jobs (Processi).
3. Scegliere Editor di script Spark in Creazione di processo, quindi scegli Crea.
4. Facoltativo: incollare il testo completo dello script nel riquadro Script. In alternativa, è possibile seguire il tutorial.

## Passaggio 2. Importa librerie AWS Glue

È necessario impostare lo script in modo che interagisca con il codice e la configurazione che sono stati definiti all'esterno dello script. Questo lavoro viene svolto dietro le quinte di AWS Glue Studio.

In questa fase, procedere secondo quanto descritto di seguito.

- Importare e inizializzare un oggetto `GlueContext`. Questa è l'importazione più importante dal punto di vista della scrittura dello script Ciò espone i metodi standard per la definizione dei set di dati di origine e di destinazione, ossia il punto di partenza per qualsiasi script ETL. Per ulteriori informazioni sulla classe `GlueContext`, consultare la pagina GlueContext classe.
- Inizializza `SparkContext` e `SparkSession`. Questi consentono di configurare il motore Spark disponibile all'interno del lavoro AWS Glue. Non sarà necessario utilizzarli direttamente negli script introduttivi di AWS Glue.
- Richiamare `getResolvedOptions` per preparare gli argomenti del processo da utilizzare all'interno dello script. Per ulteriori informazioni sulla risoluzione dei parametri di processo, consultare the section called "getResolvedOptions".
- Inizializzare un `Job`. L'`Job`oggetto imposta la configurazione e tiene traccia dello stato di varie funzioni opzionali di AWS Glue. Lo script può essere eseguito senza un oggetto `Job`, tuttavia la procedura consigliata consiste nell'inizializzarlo in modo da non essere confusi da un'eventuale integrazione successiva di queste funzionalità.

  Una di queste è rappresentata dai segnalibri di processo, che puoi configurare facoltativamente in questo tutorial. Per informazioni sui segnalibri di processo, consulta la sezione the section called "Facoltativo: abilita i segnalibri di processo".

In questa procedura si scriverà il codice seguente. Questo codice è una parte dello script di esempio generato.

```
from awsglue.transforms import *
from awsglue.utils import getResolvedOptions
from pyspark.context import SparkContext
from awsglue.context import GlueContext
from awsglue.job import Job

args = getResolvedOptions(sys.argv, ["JOB_NAME"])
sc = SparkContext()
glueContext = GlueContext(sc)
spark = glueContext.spark_session
```

```
job = Job(glueContext)
job.init(args["JOB_NAME"], args)
```

Per importare le librerie AWS Glue

- Copia questa sezione del codice e incollala nell'editor Script.

> **Note**
>
> La copia del codice potrebbe essere considerata una pratica di ingegneria non consigliata. In questo tutorial, ti suggeriamo questo per incoraggiarti a denominare in modo coerente le tue variabili principali in tutti gli script ETL di AWS Glue.

## Fase 3. Estrazione di dati da un'origine

In qualsiasi processo ETL, è innanzitutto necessario definire un set di dati di origine che si vuole modificare. Nell'editor visivo di AWS Glue Studio, fornisci queste informazioni creando un nodo Source.

In questa fase, fornire al metodo `create_dynamic_frame.from_catalog` i parametri `database` e `table_name` per estrarre i dati da un'origine configurata nel Data Catalog AWS Glue.

Nella fase precedente è stato inizializzato un oggetto `GlueContext` Utilizzare questo oggetto per trovare i metodi usati per configurare le origini, ad esempio `create_dynamic_frame.from_catalog`.

In questa procedura si scriverà il codice seguente utilizzando `create_dynamic_frame.from_catalog`. Questo codice è una parte dello script di esempio generato.

```
S3bucket_node1 = glueContext.create_dynamic_frame.from_catalog(
    database="yyz-tickets", table_name="tickets", transformation_ctx="S3bucket_node1"
    )
```

Estrazione di dati da un'origine

1. Esamina la documentazione per trovare un metodo `GlueContext` per estrarre dati da una fonte definita nel AWS Glue Data Catalog. Questi metodi sono documentati in the section called

"GlueContext". Scegliere il metodo create_dynamic_frame.from_catalog. Richiamare questo metodo in `glueContext`.

2. Esaminare la documentazione relativa a `create_dynamic_frame.from_catalog`. Questo metodo richiede i parametri `database` e `table_name`. Fornire i parametri necessari per `create_dynamic_frame.from_catalog`.

   Il AWS Glue Data Catalog memorizza le informazioni sulla posizione e il formato dei dati di origine ed è stato impostato nella sezione dei prerequisiti. Non è necessario fornire direttamente allo script tali informazioni.

3. Facoltativo: fornisci al metodo il parametro `transformation_ctx` per supportare i segnalibri di processo. Per informazioni sui segnalibri di processo, consultare la sezione the section called "Facoltativo: abilita i segnalibri di processo".

> **Note**
>
> Metodi comuni per l'estrazione dei dati
> the section called "create_dynamic_frame_from_catalog"viene utilizzato per connettersi alle tabelle nel AWS Glue Data Catalog.
> Se è necessario fornire direttamente al processo una configurazione che descriva la struttura e la posizione dell'origine, consulta il metodo the section called "create_dynamic_frame_from_options". È necessario fornire parametri più dettagliati per la descrizione dei dati rispetto a quando utilizzi `create_dynamic_frame.from_catalog`. Consultare la documentazione supplementare su `format_options` e `connection_parameters` per identificare i parametri obbligatori. Per una spiegazione su come fornire allo script informazioni sul formato dei dati di origine, consultare la sezione the section called "Opzioni del formato dei dati". Per una spiegazione su come fornire allo script informazioni sulla posizione dei dati di origine, consultare la sezione the section called "Parametri di connessione".
> Se si sta leggendo le informazioni da un'origine di streaming, fornire al processo le informazioni di origine tramite i metodi the section called "create_data_frame_from_catalog" o the section called "create_data_frame_from_options". Nota: questi metodi restituiscono `DataFrames` di Apache Spark.
> Il codice generato effettua una chiamata a `create_dynamic_frame.from_catalog`, mentre la documentazione fa riferimento a `create_dynamic_frame_from_catalog`. Questi metodi richiamano in definitiva lo stesso codice e sono stati inclusi per consentire di

scrivere un codice più pulito. Puoi verificarlo visualizzando il sorgente del nostro wrapper Python, disponibile all'indirizzo aws-glue-libs.

## Passaggio 4. Trasforma i dati con AWS Glue

Dopo aver estratto i dati di origine in un processo ETL, è necessario specificare il modo in cui modificare i dati. Fornisci queste informazioni creando un nodo Transform nell'editor visivo di AWS Glue Studio.

In questa fase, fornisci al metodo `ApplyMapping` una mappa dei nomi e dei tipi di campo attuali e desiderati per trasformare il `DynamicFrame`.

Eseguire le trasformazioni seguenti.

- Rilasciare le quattro chiavi `location` e `province`.
- Modificare il nome di `officer` in `officer_name`.
- Modificare il tipo di `ticket_number` e `set_fine_amount` in `float`.

`create_dynamic_frame.from_catalog` fornisce un oggetto `DynamicFrame`. A `DynamicFrame` rappresenta un set di dati in AWS Glue. AWS Le Glue Transform sono operazioni che cambiano`DynamicFrames`.

> **Note**
>
> Che cos'è una `DynamicFrame`?
> Un `DynamicFrame` è un'astrazione che consente di collegare un set di dati con una descrizione dei nomi e dei tipi di voci presenti nei dati. In Apache Spark esiste un'astrazione simile chiamata a. DataFrame Per una spiegazione di DataFrames, consulta Spark SQL Guide.
> I `DynamicFrames`, consentono di descrivere gli schemi del set di dati in modo dinamico. Prendiamo in considerazione un set di dati con una colonna dei prezzi, in cui alcune voci memorizzano il prezzo come stringa e altre il prezzo come doppio. AWS Glue calcola uno schema on-the-fly: crea un record autodescrittivo per ogni riga.
> I campi non coerenti (come il prezzo) sono rappresentati esplicitamente con un tipo (`ChoiceType`) nello schema del riquadro. Puoi affrontare il problema dei campi non coerenti eliminandoli con `DropFields` o risolvendoli con `ResolveChoice`. Queste trasformazioni

che sono disponibili su `DynamicFrame`. Puoi quindi riscrivere i dati sul data lake con `writeDynamicFrame`.
È possibile richiamare la maggior parte di queste trasformazioni dai metodi della classe `DynamicFrame`, ottenendo così script più leggibili. Per ulteriori informazioni su `DynamicFrame`, consultare the section called "DynamicFrame".

In questa procedura si scriverà il codice seguente utilizzando `ApplyMapping`. Questo codice è una parte dello script di esempio generato.

```
ApplyMapping_node2 = ApplyMapping.apply(
    frame=S3bucket_node1,
    mappings=[
        ("tag_number_masked", "string", "tag_number_masked", "string"),
        ("date_of_infraction", "string", "date_of_infraction", "string"),
        ("ticket_date", "string", "ticket_date", "string"),
        ("ticket_number", "decimal", "ticket_number", "float"),
        ("officer", "decimal", "officer_name", "decimal"),
        ("infraction_code", "decimal", "infraction_code", "decimal"),
        ("infraction_description", "string", "infraction_description", "string"),
        ("set_fine_amount", "decimal", "set_fine_amount", "float"),
        ("time_of_infraction", "decimal", "time_of_infraction", "decimal"),
    ],
    transformation_ctx="ApplyMapping_node2",
)
```

Per trasformare i dati con AWS Glue

1. Esaminare la documentazione per identificare una trasformazione volta a modificare ed eliminare i campi. Per informazioni dettagliate, vedi the section called "GlueTransform". Selezionare la trasformazione `ApplyMapping`. Per ulteriori informazioni su `ApplyMapping`, consultare the section called "ApplyMapping". Richiamare `apply` nell'oggetto di trasformazione `ApplyMapping`.

   > **Note**
   >
   > Cos'è `ApplyMapping`?
   > `ApplyMapping` prende un `DynamicFrame` e lo trasforma. Prende un elenco di tuple che rappresentano le trasformazioni sui campi, una "mappatura". I primi due elementi della tupla, ovvero un nome e un tipo di campo, vengono utilizzati per identificare un

campo nel riquadro. Anche gli altri due parametri rappresentano un nome e un tipo di campo.
ApplyMapping converte il campo di origine nel nome di destinazione e ne digita uno nuovo`DynamicFrame`, che restituisce. I campi non forniti verranno eliminati dal valore restituito.
Invece di effettuare una chiamata ad `apply`, è possibile richiamare la stessa trasformazione con il metodo `apply_mapping` nell'oggetto `DynamicFrame`, creando così un codice più fluido e leggibile. Per ulteriori informazioni, consultare the section called "apply_mapping".

2. Esaminare la documentazione relativa ad `ApplyMapping` per identificare i parametri obbligatori. Per informazioni, consulta the section called "ApplyMapping". Si noterà che questo metodo richiede i parametri `frame` e `mappings`. Fornire i parametri necessari per `ApplyMapping`.
3. Facoltativo: fornire `transformation_ctx` al metodo per supportare i segnalibri di processo. Per informazioni sui segnalibri di processo, consulta la sezione the section called "Facoltativo: abilita i segnalibri di processo".

> **Note**
>
> Funzionalità di Apache Spark
> Forniamo trasformazioni per semplificare i flussi di lavoro ETL all'interno del tuo processo. Si avrà inoltre accesso alle librerie che sono state create per scopi più generali e disponibili in un programma Spark nel processo. Per utilizzarle, è necessario eseguire la conversione tra `DynamicFrame` e `DataFrame`.
> È possibile creare un `DataFrame` con the section called "toDF". Quindi, puoi utilizzare i metodi disponibili su DataFrame per trasformare il tuo set di dati. Per ulteriori informazioni su questi metodi, vedere DataFrame. È quindi possibile eseguire la conversione all'indietro con the section called "fromDF" per utilizzare le operazioni AWS Glue per caricare la cornice su un bersaglio.

## Fase 5. Caricare i dati in una destinazione

In genere, i dati trasformati vengono archiviati in una posizione diversa da quella di origine. Questa operazione viene eseguita creando un nodo di destinazione nell'editor visivo di AWS Glue Studio.

In questa fase si forniscono al metodo `write_dynamic_frame.from_options` i parametri `connection_type`, `connection_options`, `format` e `format_options` per caricare i dati in un bucket di destinazione di Amazon S3.

Nella fase 1 hai inizializzato un oggetto `GlueContext`. In AWS Glue, è qui che troverai i metodi utilizzati per configurare gli obiettivi, proprio come i sorgenti.

In questa procedura si scriverà il codice seguente utilizzando `write_dynamic_frame.from_options`. Questo codice è una parte dello script di esempio generato.

```
S3bucket_node3 = glueContext.write_dynamic_frame.from_options(
    frame=ApplyMapping_node2,
    connection_type="s3",
    format="glueparquet",
    connection_options={"path": "s3://amzn-s3-demo-bucket", "partitionKeys": []},
    format_options={"compression": "gzip"},
    transformation_ctx="S3bucket_node3",
    )
```

Caricare i dati in una destinazione

1. Esaminare la documentazione per trovare un metodo adatto a caricare i dati in un bucket Amazon S3 di destinazione. Questi metodi sono documentati in the section called "GlueContext". Scegliere il metodo the section called "write_dynamic_frame_from_options". Richiamare questo metodo in `glueContext`.

   > **Note**
   >
   > Metodi comuni per il caricamento dei dati
   > `write_dynamic_frame.from_options` è il metodo più comunemente usato per caricare i dati. Supporta tutti i target disponibili in AWS Glue.
   > Se stai scrivendo su un target JDBC definito in una connessione AWS Glue, usa il the section called "write_dynamic_frame_from_jdbc_conf" metodo. AWS Le connessioni Glue memorizzano informazioni su come connettersi a un'origine dati. Ciò elimina la necessità di fornire tali informazioni in `connection_options`. Tuttavia, + necessario comunque utilizzare `connection_options` per fornire `dbtable`.
   > `write_dynamic_frame.from_catalog` non è un metodo comune per caricare i dati. Questo metodo aggiorna il AWS Glue Data Catalog senza aggiornare il set di dati

sottostante e viene utilizzato in combinazione con altri processi che modificano il set di dati sottostante. Per ulteriori informazioni, consulta the section called "Aggiornamento dello schema e aggiunta di nuove partizioni".

2. Esaminare la documentazione relativa a the section called "write_dynamic_frame_from_options". Questo metodo richiede `frame`, `connection_type`, `format`, `connection_options`, `format_options`. Richiamare questo metodo in `glueContext`.

   a. Consultare la documentazione aggiuntiva relativa a `format_options` e `format` per identificare i parametri necessari. Per una spiegazione dei formati di dati, consultare la sezione the section called "Opzioni del formato dei dati".

   b. Consultare la documentazione aggiuntiva relativa a `connection_type` e `connection_options` per identificare i parametri necessari. Per una spiegazione delle connessioni, consulta la sezione the section called "Parametri di connessione".

   c. Fornire i parametri necessari per `write_dynamic_frame.from_options`. Questo metodo ha una configurazione simile a `create_dynamic_frame.from_options`.

3. Facoltativo: fornire `transformation_ctx` al metodo `write_dynamic_frame.from_options` per supportare i segnalibri di processo. Per informazioni sui segnalibri di processo, consultare la sezione the section called "Facoltativo: abilita i segnalibri di processo".

## Fase 6. Commit dell'oggetto `Job`

Nella fase 1 è stato inizializzato un oggetto `Job`. Potrebbe essere necessario terminarne manualmente il ciclo di vita alla fine dello script nel caso alcune funzionalità facoltative lo richiedano per funzionare correttamente, ad esempio quando si utilizzano i segnalibri di processo. Questo lavoro viene svolto dietro le quinte di AWS Glue Studio.

In questa fase, effettuare una chiamata al metodo `commit` nell'oggetto `Job`.

In questa procedura si scriverà il codice seguente. Questo codice è una parte dello script di esempio generato.

```
job.commit()
```

Commit dell'oggetto **`Job`**

1. Se non è stato già fatto, eseguire le fasi facoltative descritte nelle sezioni precedenti per includere `transformation_ctx`.
2. Chiama `commit`.

## Facoltativo: abilita i segnalibri di processo

In ogni fase precedente, ti è stato chiesto di impostare i parametri `transformation_ctx`. Questa operazione è correlata a una funzionalità denominata segnalibri di processo.

Con i segnalibri di processo risparmi tempo e denaro grazie a processi eseguiti su base ricorrente rispetto a set di dati in cui è possibile tracciare facilmente il lavoro precedente. I segnalibri Job tengono traccia dell'avanzamento di una trasformazione AWS Glue su un set di dati delle esecuzioni precedenti. Tracciando dove sono terminate le esecuzioni precedenti, AWS Glue può limitare il lavoro alle righe che non ha mai elaborato prima. Per ulteriori informazioni sui segnalibri di processo, consultare the section called “Monitoraggio dei dati elaborati mediante segnalibri di processo”.

Per abilitare i segnalibri di processo, aggiungere prima le informazioni `transformation_ctx` nelle funzioni fornite, come descritto negli esempi precedenti. Lo stato del segnalibro di processo viene mantenuto tra le esecuzioni. I parametri `transformation_ctx` sono chiavi utilizzate per accedere a tale stato. Da sole, queste istruzioni non servono a nulla. È necessario attivare la funzionalità nella configurazione del processo.

In questa procedura attivare i segnalibri di processo utilizzando la Console di gestione AWS.

Abilitazione dei segnalibri di processo

1. Passare alla sezione Job details (Dettagli del processo) del processo corrispondente.
2. Impostare Job bookmark (Segnalibro di processo) su Enable (Abilita).

## Fase 7. Esecuzione del codice come processo

In questa fase, esegui il processo per verificare di aver completato correttamente questo tutorial. Questo viene fatto con il clic di un pulsante, come nell'editor visivo di AWS Glue Studio.

Esecuzione del codice come processo

1. Scegliere Untitled job (Processo senza titolo) sulla barra del titolo per modificare e impostare il nome del processo.
2. Passare alla scheda Job details (Dettagli del lavoro). Assegna al processo un Ruolo IAM. Puoi usare quello creato dal CloudFormation modello nei prerequisiti per il tutorial di AWS Glue Studio. Se è stato completato quel tutorial, dovrebbe essere disponibile come `AWS Glue StudioRole`.
3. Scegliere Save (Salva) per salvare lo script.
4. Scegliere Run (Esegui) per eseguire il processo.
5. Passare alla scheda Runs (Esecuzioni) per verificare il completamento del processo.
6. Vai a*`amzn-s3-demo-bucket`*, l'obiettivo di`write_dynamic_frame.from_options`. Verificare che l'output corrisponda alle proprie aspettative.

Per ulteriori informazioni sulla configurazione e la gestione dei processi, consultare la sezione the section called "Fornire i propri script personalizzati".

### Ulteriori informazioni

Le librerie e i metodi Apache Spark sono disponibili negli script AWS Glue. Per comprendere quali operazioni è possibile eseguire con le librerie incluse, consulta la documentazione di Spark. Per ulteriori informazioni, consultare la sezione esempi del repository di orgine di Spark.

AWS Glue 2.0+ include diverse librerie Python comuni per impostazione predefinita. Esistono anche meccanismi per caricare le proprie dipendenze in un lavoro AWS Glue in un ambiente Scala o Python. Per ulteriori informazioni sulle dipendenze Python, consultare the section called "Librerie Python".

Per altri esempi di come usare le funzionalità di AWS Glue in Python, vedi. the section called "Esempi Python" I processi in Scala e Python presentano una condizione di parità di funzioni, per cui gli esempi relativi a Python consentono di comprendere l'esecuzione di un lavoro simile in Scala.

## Programma gli script ETL di AWS Glue in PySpark

Puoi trovare esempi di codice Python e utilità per AWS Glue nel repository degli esempi di AWS Glue sul sito web. GitHub

## Usare Python con Glue AWS

AWS Glue supporta un'estensione del dialetto PySpark Python per lo scripting di lavori di estrazione, trasformazione e caricamento (ETL). Questa sezione descrive come usare Python negli script ETL e con l'API Glue. AWS

- Configurazione per l'uso di Python con AWS Glue
- Chiamate AWS Glue APIs in Python
- Usare le librerie Python con Glue AWS
- Esempi di codice Python in AWS Glue

## AWS PySpark Estensioni Glue

AWS Glue ha creato le seguenti estensioni al PySpark dialetto Python.

- Accesso ai parametri utilizzando `getResolvedOptions`
- PySpark tipi di estensione
- DynamicFrame classe
- DynamicFrameCollection classe
- DynamicFrameWriter classe
- DynamicFrameReader classe
- GlueContext classe

## AWS Glue si PySpark trasforma

AWS Glue ha creato le seguenti classi di trasformazione da utilizzare nelle operazioni PySpark ETL.

- GlueTransform classe base
- ApplyMapping classe
- DropFields classe
- DropNullFields classe
- ErrorsAsDynamicFrame classe
- FillMissingValues classe
- Classe filtro

- FindIncrementalMatches classe
- FindMatches classe
- FlatMap classe
- Classe join
- Classe mappatura
- MapToCollection classe
- mergeDynamicFrame
- Classe relazionalizzazione
- RenameField classe
- ResolveChoice classe
- SelectFields classe
- SelectFromCollection classe
- Classe Spigot
- SplitFields classe
- SplitRows classe
- Classe unbox
- UnnestFrame classe

## Configurazione per l'uso di Python con AWS Glue

Utilizza Python per sviluppare script ETL per processi Spark. Le versioni Python supportate per i processi ETL dipendono dalla versione AWS Glue del processo. Per ulteriori informazioni sulle versioni di AWS Glue, consulta Glue version job property.

Per configurare il sistema per l'uso di Python con AWS Glue

Segui questi passaggi per installare Python e poter invocare il. AWS Glue APIs

1. Se non disponi ancora di Python installato, scaricalo e installalo dalla Python.org download page.
2. Installa AWS Command Line Interface (AWS CLI) come documentato nella documentazione AWS CLI.

   Non AWS CLI è direttamente necessario per usare Python. Tuttavia, installarlo e configurarlo è un modo conveniente per impostare AWS le credenziali dell'account e verificarne il funzionamento.

3. Installa l' AWS SDK per Python (Boto 3), come documentato in Boto3 Quickstart.

   Le risorse Boto 3 non sono ancora disponibili per. APIs AWS Glue Attualmente è APIs possibile utilizzare solo il client Boto 3.

   Per ulteriori informazioni su Boto 3, consulta AWS Guida introduttiva di SDK for Python (Boto3) .

Puoi trovare esempi di codice e utilità per Python AWS Glue nel repository degli AWS Glue esempi sul sito web. GitHub

## Chiamate AWS Glue APIs in Python

Nota che le risorse Boto 3 non APIs sono ancora disponibili per. AWS Glue Attualmente è APIs possibile utilizzare solo il client Boto 3.

Nomi delle API AWS Glue in Python

AWS I nomi delle API Glue in Java e in altri linguaggi di programmazione sono generalmente CamelCased. Tuttavia, quando vengono chiamati da Python, questi nomi generici vengono modificati in minuscolo, con le parti del nome separate da caratteri di sottolineatura per renderli più adatti a Python. Nella documentazione AWS Glue API di riferimento, questi nomi Python sono elencati tra parentesi dopo i nomi generici. CamelCased

Tuttavia, anche se i nomi delle API AWS Glue vengono trasformati in lettere minuscole, i nomi dei relativi parametri rimangono in maiuscolo. È importante ricordarlo, perché i parametri devono essere passati per nome durante la chiamata AWS Glue APIs, come descritto nella sezione seguente.

Passaggio di parametri Python in AWS Glue e accesso ai parametri

Nelle chiamate Python a AWS Glue APIs, è meglio passare i parametri in modo esplicito per nome. Esempio:

```
job = glue.create_job(Name='sample', Role='Glue_DefaultRole',
                      Command={'Name': 'glueetl',
                               'ScriptLocation': 's3://my_script_bucket/scripts/
my_etl_script.py'})
```

È utile capire che Python crea un dizionario delle name/value tuple specificate come argomenti di uno script ETL in un or. Struttura del processo JobRun struttura Boto 3 esegue quindi il passaggio ad AWS Glue in formato JSON mediante una chiamata API REST. Questo significa che non puoi fare affidamento sull'ordine degli argomenti al momento dell'accesso nello script.

Ad esempio, supponiamo che stai avviando un `JobRun` in una funzione del gestore Lambda di Python e che desideri specificare più parametri. Il tuo codice potrebbe essere simile a quanto segue:

```
from datetime import datetime, timedelta

client = boto3.client('glue')

def lambda_handler(event, context):
  last_hour_date_time = datetime.now() - timedelta(hours = 1)
  day_partition_value = last_hour_date_time.strftime("%Y-%m-%d")
  hour_partition_value = last_hour_date_time.strftime("%-H")

  response = client.start_job_run(
              JobName = 'my_test_Job',
              Arguments = {
                '--day_partition_key':   'partition_0',
                '--hour_partition_key':  'partition_1',
                '--day_partition_value':  day_partition_value,
                '--hour_partition_value': hour_partition_value } )
```

Per accedere a questi parametri in modo affidabile nello script ETL, specificali per nome usando la funzione AWS Glue di `getResolvedOptions` e quindi esegui l'accesso dal dizionario risultante:

```
import sys
from awsglue.utils import getResolvedOptions

args = getResolvedOptions(sys.argv,
                          ['JOB_NAME',
                           'day_partition_key',
                           'hour_partition_key',
                           'day_partition_value',
                           'hour_partition_value'])
print "The day partition key is: ", args['day_partition_key']
print "and the day partition value is: ", args['day_partition_value']
```

Se desideri passare un argomento che è una stringa JSON annidata, per preservare il valore del parametro man mano che viene passato al processo ETL AWS Glue, devi codificare la stringa del parametro prima di avviare l'esecuzione del processo, quindi decodificare la stringa del parametro prima di riferirla allo script di processo. Ad esempio, considera la seguente stringa di argomento:

```
glue_client.start_job_run(JobName = "gluejobname", Arguments={
"--my_curly_braces_string": '{"a": {"b": {"c": [{"d": {"e": 42}}]}}}'
```

```
})
```

Per passare correttamente questo parametro, devi codificare l'argomento come una stringa codificata Base64.

```
import base64
...
sample_string='{"a": {"b": {"c": [{"d": {"e": 42}}]}}}'
sample_string_bytes = sample_string.encode("ascii")

base64_bytes = base64.b64encode(sample_string_bytes)
base64_string = base64_bytes.decode("ascii")
...
glue_client.start_job_run(JobName = "gluejobname", Arguments={
"--my_curly_braces_string": base64_bytes})
...
sample_string_bytes = base64.b64decode(base64_bytes)
sample_string = sample_string_bytes.decode("ascii")
print(f"Decoded string: {sample_string}")
...
```

Esempio: creazione ed esecuzione di un processo

L'esempio seguente mostra come chiamare l'AWS Glue APIsusing Python, per creare ed eseguire un processo ETL.

Per creare ed eseguire un processo

1. Crea un'istanza del client AWS Glue:

   ```
   import boto3
   glue = boto3.client(service_name='glue', region_name='us-east-1',
                endpoint_url='https://glue.us-east-1.amazonaws.com')
   ```

2. Crea un processo. Devi utilizzare `glueetl` come nome per il comando ETL, come mostrato nel codice seguente:

   ```
   myJob = glue.create_job(Name='sample', Role='Glue_DefaultRole',
                           Command={'Name': 'glueetl',
                                    'ScriptLocation': 's3://my_script_bucket/
   scripts/my_etl_script.py'})
   ```

3. Avvia una nuova esecuzione del processo creato nella fase precedente:

```
myNewJobRun = glue.start_job_run(JobName=myJob['Name'])
```

4. Ottieni lo stato del processo:

```
status = glue.get_job_run(JobName=myJob['Name'], RunId=myNewJobRun['JobRunId'])
```

5. Stampa lo stato attuale del processo eseguito:

```
print(status['JobRun']['JobRunState'])
```

## Usare le librerie Python con Glue AWS

È possibile installare moduli e librerie Python aggiuntivi da utilizzare con AWS Glue ETL. Per AWS Glue 2.0 e versioni successive, AWS Glue utilizza Python Package Installer (pip3) per installare moduli aggiuntivi utilizzati da Glue ETL. AWS AWS Glue offre diverse opzioni per portare i moduli Python aggiuntivi nell'ambiente di lavoro AWS Glue. Puoi usare il `--additional-python-modules` parametro per inserire nuovi moduli usando file zip contenenti ruote Python in bundle (note anche come «zip of wheels», disponibili per AWS Glue 5.0 e versioni successive), singoli file wheel Python, file dei requisiti (requirements.txt, disponibile per AWS Glue 5.0 e versioni successive) o un elenco di moduli Python separati da virgole. Potrebbe anche essere usato per cambiare la versione dei moduli python forniti nell'ambiente AWS Glue (vedi Moduli Python già forniti in Glue AWS per maggiori dettagli).

Argomenti

- Installazione di moduli Python aggiuntivi con pip in AWS Glue 2.0 o versioni successive
- Inclusione di file Python con funzionalità native PySpark
- Script di programmazione che utilizzano trasformazioni visive
- Compressione delle librerie per l'inclusione
- Caricamento delle librerie Python nei notebook AWS Glue Studio
- Caricamento delle librerie Python in un endpoint di sviluppo in Glue 0.9/1.0 AWS
- Usare le librerie Python in un lavoro o JobRun
- Analizza in modo proattivo le dipendenze di Python
- Moduli Python già forniti in Glue AWS

- Appendice A: Creazione di un artefatto Zip of Wheels
- Appendice B: Dettagli sull'ambiente AWS Glue

Installazione di moduli Python aggiuntivi con pip in AWS Glue 2.0 o versioni successive

AWS Glue utilizza Python Package Installer (pip3) per installare moduli aggiuntivi che devono essere utilizzati da Glue ETL. AWS Puoi utilizzare il parametro `--additional-python-modules` con un elenco di moduli Python separati da virgole per aggiungere un nuovo modulo o modificare la versione di un modulo esistente. Puoi installare gli artefatti delle ruote predefiniti tramite una cerniera di ruote o un elemento ruota indipendente caricando il file su Amazon S3, quindi includendo il percorso dell'oggetto Amazon S3 nell'elenco dei moduli. Per ulteriori informazioni sull'impostazione dei parametri del lavoro, vedere Utilizzo dei parametri del lavoro nei lavori AWS Glue.

Puoi passare opzioni aggiuntive a pip3 tramite il parametro `--python-modules-installer-option`. Ad esempio, potresti passare `--only-binary` a force pip per installare solo artefatti precompilati per i pacchetti specificati da. `--additional-python-modules` Per altri esempi, consulta Creazione di moduli Python da una ruota per carichi di lavoro Spark ETL con Glue 2.0. AWS

Le migliori pratiche per la gestione delle dipendenze in Python

Per i carichi di lavoro di produzione, AWS Glue consiglia di impacchettare tutte le dipendenze Python come file wheel in un unico elemento zip. Questo approccio fornisce:

- Esecuzione deterministica: controllo esatto su quali versioni del pacchetto sono installate
- Affidabilità: nessuna dipendenza da archivi di pacchetti esterni durante l'esecuzione del lavoro
- Prestazioni: operazione di download singola anziché più chiamate di rete
- Installazione offline: funziona in ambienti VPC privati senza accesso a Internet

Considerazioni importanti

Secondo il modello di responsabilitàAWS condivisa, sei responsabile della gestione di moduli Python aggiuntivi, librerie e relative dipendenze. Questo include:

- Aggiornamenti di sicurezza: aggiornamento regolare dei pacchetti per risolvere le vulnerabilità di sicurezza
- Compatibilità delle versioni: Garantire che i pacchetti siano compatibili con la versione di AWS Glue in uso

- Test: verifica del corretto funzionamento delle dipendenze impacchettate nell'ambiente Glue

Se hai dipendenze minime, potresti prendere in considerazione l'utilizzo di singoli file wheel.

(Consigliato) Installazione di librerie Python aggiuntive in AWS Glue 5.0 o versioni successive utilizzando Zip of Wheels

AWS Glue 5.0 e versioni successive supportano il pacchetto di più file wheel in un unico artefatto zip contenente ruote Python in bundle per una gestione delle dipendenze più affidabile e deterministica. Per utilizzare questo approccio, crea un file zip contenente tutte le dipendenze della ruota e le relative dipendenze transitive con il `.gluewheels.zip` suffisso, caricalo su Amazon S3 e fai riferimento ad esso utilizzando il parametro. `--additional-python-modules` Assicurati di `--no-index` aggiungerlo al parametro job. `--python-modules-installer-option` Con questa configurazione, lo zip del file wheels funge essenzialmente da indice locale da cui pip risolve le dipendenze in fase di esecuzione. Ciò elimina le dipendenze da repository di pacchetti esterni come PyPI durante l'esecuzione del lavoro, fornendo maggiore stabilità e coerenza per i carichi di lavoro di produzione. Esempio:

```
--additional-python-modules s3://amzn-s3-demo-bucket/path/to/zip-of-
wheels-1.0.0.gluewheels.zip
--python-modules-installer-option --no-index
```

Per istruzioni su come creare un file zip of wheels, consulta. Appendice A: Creazione di un artefatto Zip of Wheels

Installazione di librerie Python aggiuntive utilizzando Wheel

AWS Glue supporta l'installazione di pacchetti Python personalizzati utilizzando file wheel (.whl) archiviati in Amazon S3. Per includere i file wheel nei tuoi lavori AWS Glue, fornisci un elenco separato da virgole dei tuoi file wheel memorizzati in s3 al parametro job. `--additional-python-modules` Esempio:

```
--additional-python-modules s3://amzn-s3-demo-bucket/path/to/package-1.0.0-py3-none-
any.whl,s3://your-bucket/path/to/another-package-2.1.0-cp311-cp311-linux_x86_64.whl
```

Questo approccio supporta anche le casistiche in cui sono necessarie distribuzioni personalizzate o pacchetti con dipendenze native precompilati per il sistema operativo corretto. Per altri esempi, consulta Creazione di moduli Python da una ruota per carichi di lavoro Spark ETL con Glue 2.0. AWS

Installazione di librerie Python aggiuntive in AWS Glue 5.0 o versioni successive utilizzando requirements.txt

In AWS Glue 5.0+, puoi fornire lo standard di fatto per gestire le dipendenze della libreria `requirements.txt` Python. A tale scopo, fornire i seguenti due parametri di lavoro:

- Chiave: `--python-modules-installer-option`

  Valore: `-r`
- Chiave: `--additional-python-modules`

  Valore: `s3://path_to_requirements.txt`

AWS I nodi Glue 5.0 caricano inizialmente le librerie python specificate in`requirements.txt`.

Segue un file di esempio requirements.txt:

```
awswrangler==3.9.1
elasticsearch==8.15.1
PyAthena==3.9.0
PyMySQL==1.1.1
PyYAML==6.0.2
pyodbc==5.2.0
pyorc==0.9.0
redshift-connector==2.1.3
scipy==1.14.1
scikit-learn==1.5.2
SQLAlchemy==2.0.36
```

**Important**

Utilizzate questa opzione con cautela, specialmente nei carichi di lavoro di produzione. Estrarre le dipendenze da PyPI in fase di esecuzione è altamente rischioso perché non si può essere sicuri a quale artefatto si risolva pip. L'uso di versioni di librerie non bloccate è particolarmente rischioso poiché richiama l'ultima versione dei moduli python, che può introdurre modifiche sostanziali o introdurre moduli python incompatibili. Ciò potrebbe causare un errore del lavoro dovuto a un errore di installazione di Python nell'ambiente di lavoro AWS Glue. Sebbene il blocco della versione della libreria aumenti la stabilità, la risoluzione pip non è ancora completamente deterministica, quindi possono sorgere problemi simili. Come buona pratica, AWS Glue consiglia di utilizzare artefatti congelati come zip

di ruote o singoli file di ruote (vedi the section called "(Consigliato) Installazione di librerie Python aggiuntive in AWS Glue 5.0 o versioni successive utilizzando Zip of Wheels" per maggiori dettagli).

> **Important**
>
> Se non inserite le versioni delle vostre dipendenze transitive, una dipendenza primaria può generare versioni di dipendenza transitive incompatibili. Come best practice, tutte le versioni della libreria devono essere bloccate per una maggiore coerenza nei job AWS Glue. Ancora meglio, AWS Glue consiglia di impacchettare le dipendenze in un file zip o wheels per garantire la massima coerenza e affidabilità per i carichi di lavoro di produzione.

Installazione di librerie Python aggiuntive configurando direttamente come elenco separato da virgole

Per aggiornare o aggiungere un nuovo modulo Python, AWS Glue consente di passare `--additional-python-modules` parametri con un elenco di moduli Python separati da virgole come valori. Ad esempio, per aggiornare o aggiungere un modulo scikit-learn, usa la seguente coppia chiave-valore: `"--additional-python-modules", "scikit-learn==0.21.3"`. Sono disponibili due opzioni per configurare direttamente i moduli Python.

- Modulo Python bloccato

  ```
  "--additional-python-modules", "scikit-learn==0.21.3,ephem==4.1.6"
  ```

- Modulo Python non bloccato: (non consigliato per carichi di lavoro di produzione)

  ```
  "--additional-python-modules", "scikit-learn>==0.20.0,ephem>=4.0.0"
  ```

  O

  ```
  "--additional-python-modules", "scikit-learn,ephem"
  ```

> **Important**
>
> Utilizzate questa opzione con cautela, specialmente nei carichi di lavoro di produzione. Estrarre le dipendenze da PyPI in fase di esecuzione è altamente rischioso perché non si può essere sicuri a quale artefatto si risolva pip. L'uso di versioni di librerie non bloccate

è particolarmente rischioso poiché richiama l'ultima versione dei moduli python, che può introdurre modifiche sostanziali o introdurre moduli python incompatibili. Ciò potrebbe causare un errore del lavoro dovuto a un errore di installazione di Python nell'ambiente di lavoro AWS Glue. Sebbene il blocco della versione della libreria aumenti la stabilità, la risoluzione pip non è ancora completamente deterministica, quindi possono sorgere problemi simili. Come buona pratica, AWS Glue consiglia di utilizzare artefatti congelati come zip di ruote o singoli file di ruote (vedi the section called "(Consigliato) Installazione di librerie Python aggiuntive in AWS Glue 5.0 o versioni successive utilizzando Zip of Wheels" per maggiori dettagli).

**Important**

Se non inserite le versioni delle vostre dipendenze transitive, una dipendenza primaria può generare versioni di dipendenza transitive incompatibili. Come best practice, tutte le versioni della libreria devono essere bloccate per una maggiore coerenza nei job AWS Glue. Ancora meglio, AWS Glue consiglia di impacchettare le dipendenze in un file zip o wheels per garantire la massima coerenza e affidabilità per i carichi di lavoro di produzione.

Inclusione di file Python con funzionalità native PySpark

AWS Glue utilizza PySpark per includere file Python nei job AWS Glue ETL. Quando possibile, ti consigliamo di usare `--additional-python-modules` per gestire le dipendenze. Puoi utilizzare il parametro del processo `--extra-py-files` per includere i file Python. Le dipendenze devono essere ospitate in Amazon S3 e il valore dell'argomento deve essere un elenco delimitato da virgole di percorsi Amazon S3 senza spazi. Questa funzionalità si comporta come la gestione delle dipendenze Python che useresti con Spark. Per ulteriori informazioni sulla gestione delle dipendenze di Python in Spark, consulta la pagina Utilizzo delle funzionalità PySpark native nella documentazione di Apache Spark. `--extra-py-files`è utile nei casi in cui il codice aggiuntivo non è incluso nel pacchetto o quando si sta migrando un programma Spark con una toolchain esistente per la gestione delle dipendenze. Affinché gli strumenti di dipendenza siano gestibili, sarà necessario raggruppare le dipendenze prima di inviarle.

Script di programmazione che utilizzano trasformazioni visive

Quando crei un lavoro AWS Glue utilizzando l'interfaccia visiva di AWS Glue Studio, puoi trasformare i tuoi dati con nodi di trasformazione dati gestiti e trasformazioni visive personalizzate. Per ulteriori

informazioni sui nodi di trasformazione dei dati, consultare the section called "Trasformare i dati con AWS Glue trasformazioni gestite". Per ulteriori informazioni sulle trasformazioni visive, consultare the section called " Trasforma i dati con le trasformazioni visive personalizzate ". Gli script che utilizzano le trasformazioni visive possono essere generati solo quando il linguaggio del processo è impostato per utilizzare Python.

Quando si genera un lavoro AWS Glue utilizzando trasformazioni visive, AWS Glue Studio includerà queste trasformazioni nell'ambiente di runtime utilizzando il `--extra-py-files` parametro nella configurazione del lavoro. Per ulteriori informazioni sui parametri di processo, consulta the section called "Parametri Job". Quando a si apportano modifiche a uno script o a un ambiente di runtime generato, è necessario mantenere questa configurazione del lavoro affinché lo script venga eseguito correttamente.

Compressione delle librerie per l'inclusione

A meno che una libreria non sia contenuta in un singolo file `.py`, deve essere compressa in un archivio `.zip`. La directory del pacchetto deve trovarsi al livello radice dell'archivio e deve contenere un file `__init__.py` per il pacchetto. Python sarà in grado di importare il pacchetto nel modo normale.

Se la tua libreria è composta da un singolo modulo Python in un file `.py`, non è necessario trasferirla in un file `.zip`.

Caricamento delle librerie Python nei notebook AWS Glue Studio

Per specificare le librerie Python nei notebook AWS Glue Studio, vedi Installazione di moduli Python aggiuntivi.

Caricamento delle librerie Python in un endpoint di sviluppo in Glue 0.9/1.0 AWS

Se utilizzi diversi set di librerie per diversi script ETL, puoi impostare un endpoint di sviluppo separato per ciascun set oppure sovrascrivere i file `.zip` della libreria che l'endpoint di sviluppo carica ogni volta che si cambia script.

Puoi utilizzare la console per specificare uno o più file .zip di libreria per un endpoint di sviluppo al momento della creazione. Dopo l'assegnazione di un nome e un ruolo IAM, scegli Script Libraries and job parameters (optional) (Librerie di Script e parametri di processo -opzionale) e immetti il percorso Amazon S3 completo per i tuoi file `.zip` della libreria nella casella Python library path (Percorso libreria Python). Ad esempio:

```
s3://bucket/prefix/site-packages.zip
```

Se lo desideri, puoi specificare più percorsi completi per i file, separandoli con virgole ma non spazi, in questo modo:

```
s3://bucket/prefix/lib_A.zip,s3://bucket_B/prefix/lib_X.zip
```

Se aggiorni questi file `.zip` in un secondo momento, puoi utilizzare la console per importarli nuovamente nell'endpoint di sviluppo. Individua l'endpoint dello sviluppatore in questione, verifica la casella a esso corrispondente e scegli Update ETL libraries (Aggiorna librerie ETL) dal menu Action (Operazione).

Allo stesso modo, potete specificare i file di libreria usando AWS Glue APIs. Quando crei un endpoint di sviluppo chiamando CreateDevEndpoint azione (Python: create_dev_endpoint), puoi specificare uno o più percorsi completi per le librerie nel parametro `ExtraPythonLibsS3Path` in una chiamata come la seguente:

```
dep = glue.create_dev_endpoint(
             EndpointName="testDevEndpoint",
             RoleArn="arn:aws:iam::123456789012",
             SecurityGroupIds="sg-7f5ad1ff",
             SubnetId="subnet-c12fdba4",
             PublicKey="ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCtp04H/y...",
             NumberOfNodes=3,
             ExtraPythonLibsS3Path="s3://bucket/prefix/lib_A.zip,s3://bucket_B/prefix/
lib_X.zip")
```

Quando aggiorni un endpoint di sviluppo, puoi anche aggiornare le librerie caricate utilizzando un oggetto DevEndpointCustomLibraries e impostare il parametro `UpdateEtlLibraries` su `True` durante la chiamata UpdateDevEndpoint (update_dev_endpoint).

Usare le librerie Python in un lavoro o JobRun

Quando si crea un nuovo processo nella console, è possibile specificare uno o più file ZIP di libreria scegliendo Script Libraries and job parameters (optional) (Librerie di script e parametri di processo - opzionale) e immettendo percorsi di librerie Amazon S3 completi, analogamente a come si farebbe quando si crea un endpoint di sviluppo:

```
s3://bucket/prefix/lib_A.zip,s3://bucket_B/prefix/lib_X.zip
```

Se si sta chiamando CreateJob (create_job), è possibile specificare uno o più percorsi completi alle librerie predefinite utilizzando il parametro predefinito `--extra-py-files`, come segue:

```
job = glue.create_job(Name='sampleJob',
                      Role='Glue_DefaultRole',
                      Command={'Name': 'glueetl',
                               'ScriptLocation': 's3://my_script_bucket/scripts/
my_etl_script.py'},
                      DefaultArguments={'--extra-py-files': 's3://bucket/prefix/
lib_A.zip,s3://bucket_B/prefix/lib_X.zip'})
```

Quindi, quando avvii un JobRun, puoi sovrascrivere l'impostazione della libreria predefinita con una diversa:

```
runId = glue.start_job_run(JobName='sampleJob',
                           Arguments={'--extra-py-files': 's3://bucket/prefix/
lib_B.zip'})
```

Analizza in modo proattivo le dipendenze di Python

Per identificare in modo proattivo potenziali problemi di dipendenza prima della distribuzione su AWS Glue, puoi utilizzare lo strumento di analisi delle dipendenze per convalidare i tuoi pacchetti Python rispetto all'ambiente Glue di destinazione. AWS

AWS fornisce uno strumento di analisi delle dipendenze Python open source progettato specificamente per gli ambienti Glue. AWS Questo strumento è disponibile nell'archivio degli esempi AWS Glue e può essere utilizzato localmente per convalidare le dipendenze prima della distribuzione.

Questa analisi aiuta a garantire che le dipendenze seguano la pratica consigliata del pinning di tutte le versioni della libreria per implementazioni di produzione coerenti. Per maggiori dettagli, consultare la sezione README dello strumento.

Utilizzo del AWS Glue Dependency Analyzer

Il AWS Glue Python Dependency Analyzer aiuta a identificare le dipendenze non bloccate e i conflitti di versione simulando l'installazione di pip con vincoli specifici della piattaforma che corrispondono all'ambiente Glue di destinazione. AWS

```
# Analyze a single Glue job
```

```
python glue_dependency_analyzer.py -j my-glue-job

# Analyze multiple jobs with specific AWS configuration
python glue_dependency_analyzer.py -j job1 -j job2 --aws-profile production --aws-
region us-west-2
```

Lo strumento contrassegnerà:

- Dipendenze non bloccate che potrebbero installare versioni diverse tra le esecuzioni dei processi
- Conflitti delle versioni tra pacchetti
- Dipendenze non disponibili per l'ambiente AWS Glue di destinazione

Analizzare e correggere gli errori nei processi dovuti alle dipendenze di Python con Amazon Q Developer

Amazon Q Developer è un assistente conversazionale generativo basato sull'intelligenza artificiale (AI) che può aiutarti a comprendere, creare, estendere e utilizzare le applicazioni. AWS È possibile scaricarlo seguendo le istruzioni nella Guida alle nozioni di base per Amazon Q.

Amazon Q Developer può essere usato per analizzare e correggere gli errori nei processi dovuti alle dipendenze Python. Suggeriamo di utilizzare il seguente prompt sostituendo il placeholder <Job-Name> con il nome del processo Glue.

```
I have an AWS Glue job named <Job-Name> that has failed due to Python module
 installation conflicts. Please assist in diagnosing and resolving this issue using the
 following systematic approach. Proceed once sufficient information is available.

Objective: Implement a fix that addresses the root cause module while minimizing
 disruption to the existing working environment.

Step 1: Root Cause Analysis
• Retrieve the most recent failed job run ID for the specified Glue job
• Extract error logs from CloudWatch Logs using the job run ID as a log stream prefix
• Analyze the logs to identify:
  • The recently added or modified Python module that triggered the dependency conflict
  • The specific dependency chain causing the installation failure
  • Version compatibility conflicts between required and existing modules

Step 2: Baseline Configuration Identification
• Locate the last successful job run ID prior to the dependency failure
```

```
• Document the Python module versions that were functioning correctly in that baseline
 run
• Establish the compatible version constraints for conflicting dependencies

Step 3: Targeted Resolution Implementation
• Apply pinning by updating the job's additional_python_modules parameter
• Pin only the root cause module and its directly conflicting dependencies to
 compatible versions, and do not remove python modules unless necessary
• Preserve flexibility for non-conflicting modules by avoiding unnecessary version
 constraints
• Deploy the configuration changes with minimal changes to the existing configuration
 and execute a validation test run. Do not change the Glue versions.

Implementation Example:
Scenario: Recently added pandas==2.0.0 to additional_python_modules
Error: numpy version conflict (pandas 2.0.0 requires numpy>=1.21, but existing job code
 requires numpy<1.20)
Resolution: Update additional_python_modules to "pandas==1.5.3,numpy==1.19.5"
Rationale: Use pandas 1.5.3 (compatible with numpy 1.19.5) and pin numpy to last known
 working version

Expected Outcome: Restore job functionality with minimal configuration changes while
 maintaining system stability.
```

Il prompt indica a Q di:

1. Recuperare l'ultimo ID di esecuzione dei processi non riuscito
2. Trovare i log associati e i dettagli
3. Trovare le esecuzioni dei processi riuscite per rilevare eventuali pacchetti Python modificati
4. Apportare eventuali correzioni alla configurazione e avviare un'altra esecuzione di test

### Moduli Python già forniti in Glue AWS

È possibile modificare la versione dei moduli disponibili con il parametro di processo `--additional-python-modules`.

#### AWS Glue version 5.1

AWS La versione 5.1 di Glue include i seguenti moduli Python pronti all'uso:

- aiobotocore==2.25.1

- aiohappy eyebuls==2.6.1
- aiohttp=3.13.2
- aioitertools=0.12.0
- aiosignal=1.4.0
- appdirs==1.4.4
- attrs==25.4.0
- boto3=1.40.61
- botocore=1.40.61
- certifi==2025.10.5
- charset-normalizer==3.4.4
- coreografo ==1.2.0
- contourpy==1.3.3
- ciclo==0.12.1
- distlib==0.4.0
- filelock==3.20.0
- strumenti di carattere ==4.60.1
- lista congelata ==1.8.0
- fsspec==2025.10.0
- idna=3.11
- iniconfig=2.3.0
- jmespath=1.0.1
- caleido==1.2.0
- kiwi ver== 1.4.9
- logistro==2.0.1
- matplotlib==3.10.7
- multidict==6.7.0
- narvali==2.10.2
- numpy==2.3.4

- o json==3.11.4
- imballaggio==25.0
- panda ==2.3.3
- cuscino ==12.0.0
- pip=24,0
- platformdirs ==4.5.0
- plotly==6.4.0
- pluggy==1.6.0
- propcache==0.4.1
- pyarrow==22.0.0
- Pigmenti==2.19.2
- pyparsing==3.2.5
- pytest-timeout==2.4.0
- pytest==8.4.2
- python-dateutil==2.9.0.post0
- pytz==2025.2
- richieste==2.32.5
- s3fs==2025.10.0
- s3transfer==0.14.0
- seaborn==0.13.2
- strumenti di configurazione ==79.0.1
- simplejson==3.20.2
- sei==1.17.0
- tenacia==9.1.2
- typing_extensions==4.15.0
- tzdata==2025,2
- urllib3==2.5.0
- uv=0.9.7
- ambiente virtuale ==20.35.4

- wrapt==1.17.3
- yarl=1.22.0

### AWS Glue version 5.0

AWS La versione 5.0 di Glue include i seguenti moduli Python pronti all'uso:

- aiobotocore==2.13.1
- aiohappyeyeballs==2.3.5
- aiohttp=3.10.1
- aioitertools==0.11.0
- aiosignal==1.3.1
- appdirs==1.4.4
- attrs==24.2.0
- boto3==1.34.131
- botocore==1.34.131
- certifi==2024.7.4
- charset-normalizer==3.3.2
- contourpy==1.2.1
- ciclo==0.12.1
- fonttools==4.53.1
- lista congelata ==1.4.1
- fsspec==2024.6.1
- idna==2.10
- jmespath==0.10.0
- kaleido==0.2.1
- kiwi ver== 1.4.5
- matplotlib==3.9.0
- multidict==6.0.5
- numpy==1.26.4

- imballaggio==24.1
- panda ==2.2.2
- cuscino ==10.4.0
- pip==23.0.1
- plotly==5.23.0
- pyarrow==17.0.0
- pyparsing==3.1.2
- python-dateutil==2.9.0.post0
- pytz==2024.1
- richieste==2.32.2
- s3fs==2024.6.1
- s3transfer ==0.10.2
- seaborn==0.13.2
- strumenti di configurazione ==59.6.0
- six==1.16.0
- tenacity==9.0.0
- tzdata==2024.1
- urllib3==1.25.10
- virtualenv==20.4.0
- wrapt==1.16.0
- yarl=1.9.4

AWS Glue version 4.0

AWS La versione 4.0 di Glue include i seguenti moduli Python pronti all'uso:

- aiobotocore==2.4.1
- aiohttp==3.8.3
- aioitertools==0.11.0
- aiosignal==1.3.1

- async-timeout==4.0.2
- asynctest==0.13.0
- attrs==22.2.0
- avro-python3==1.10.2
- boto3==1.24.70
- botocore==1.27.59
- certifi==2021.5.30
- chardet==3.0.4
- charset-normalizer==2.1.1
- click==8.1.3
- cycler==0.10.0
- Cython==0.29.32
- fsspec==2021.8.1
- idna==2.10
- importlib-metadata==5.0.0
- jmespath==0.10.0
- joblib==1.0.1
- kaleido==0.2.1
- kiwisolver==1.4.4
- matplotlib==3.4.3
- mpmath==1.2.1
- multidict==6.0.4
- nltk==3.7
- numpy==1.23.5
- packaging==23.0
- pandas == 1.5.1
- patsy==0.5.1
- Pillow==9.4.0

- pip==23.0.1
- plotly==5.16.0
- pmdarima==2.0.1
- ptvsd==4.3.2
- pyarrow==10.0.0
- pydevd==2.5.0
- pyhocon==0.3.58
- PyMySQL==1.0.2
- pyparsing==2.4.7
- python-dateutil==2.8.2
- pytz==2021.1
- PyYAML==6.0.1
- regex==2022.10.31
- requests==2.23.0
- s3fs==2022.11.0
- s3transfer==0.6.0
- scikit-learn==1.1.3
- scipy==1.9.3
- setuptools==49.1.3
- six==1.16.0
- statsmodels==0.13.5
- subprocess32==3.5.4
- sympy==1.8
- tbats==1.1.0
- threadpoolctl==3.1.0
- tqdm==4.64.1
- typing_extensions==4.4.0
- urllib3==1.25.11

- wheel==0.37.0
- wrapt==1.14.1
- yarl==1.8.2
- zipp==3.10.0

AWS Glue version 3.0

AWS La versione 3.0 di Glue include i seguenti moduli Python pronti all'uso:,

- aiobotocore==1.4.2
- aiohttp==3.8.3
- aioitertools==0.11.0
- aiosignal==1.3.1
- async-timeout==4.0.2
- asynctest==0.13.0
- attrs==22.2.0
- avro-python3==1.10.2
- boto3==1.18.50
- botocore==1.21.50
- certifi==2021.5.30
- chardet==3.0.4
- charset-normalizer==2.1.1
- click==8.1.3
- cycler==0.10.0
- Cython==0.29.4
- docutils==0.17.1
- enum34==1.1.10
- frozenlist==1.3.3
- fsspec==2021.8.1
- idna==2.10

- importlib-metadata==6.0.0
- jmespath==0.10.0
- joblib==1.0.1
- kiwisolver==1.3.2
- matplotlib==3.4.3
- mpmath==1.2.1
- multidict==6.0.4
- nltk==3.6.3
- numpy==1.19.5
- packaging==23.0
- pandas==1.3.2
- patsy==0.5.1
- Pillow==9.4.0
- pip==23.0
- pmdarima==1.8.2
- ptvsd==4.3.2
- pyarrow==5.0.0
- pydevd==2.5.0
- pyhocon==0.3.58
- PyMySQL==1.0.2
- pyparsing==2.4.7
- python-dateutil==2.8.2
- pytz==2021.1
- PyYAML==5.4.1
- regex==2022.10.31
- requests==2.23.0
- s3fs==2021.8.1
- s3transfer==0.5.0
- scikit-learn==0.24.2

- scipy==1.7.1
- six==1.16.0
- Spark==1.0
- statsmodels==0.12.2
- subprocess32==3.5.4
- sympy==1.8
- tbats==1.1.0
- threadpoolctl==3.1.0
- tqdm==4.64.1
- typing_extensions==4.4.0
- urllib3==1.25.11
- wheel==0.37.0
- wrapt==1.14.1
- yarl==1.8.2
- zipp==3.12.0

AWS Glue version 2.0

AWS La versione 2.0 di Glue include i seguenti moduli Python pronti all'uso:

- avro-python3==1.10.0
- awscli==1.27.60
- boto3==1.12.4
- botocore==1.15.4
- certifi==2019.11.28
- chardet==3.0.4
- click==8.1.3
- colorama==0.4.4
- cycler==0.10.0
- Cython==0.29.15

- docutils==0.15.2
- enum34==1.1.9
- fsspec==0.6.2
- idna==2.9
- importlib-metadata==6.0.0
- jmespath==0.9.4
- joblib==0.14.1
- kiwisolver==1.1.0
- matplotlib==3.1.3
- mpmath==1.1.0
- nltk==3.5
- numpy==1.18.1
- pandas==1.0.1
- patsy==0.5.1
- pmdarima==1.5.3
- ptvsd==4.3.2
- pyarrow==0.16.0
- pyasn1==0.4.8
- pydevd==1.9.0
- pyhocon==0.3.54
- PyMySQL==0.9.3
- pyparsing==2.4.6
- python-dateutil==2.8.1
- pytz==2019.3
- PyYAML==5.3.1
- regex==2022.10.31
- requests==2.23.0
- rsa==4.7.2
- s3fs==0.4.0

- s3transfer==0.3.3
- scikit-learn==0.22.1
- scipy==1.4.1
- setuptools==45.2.0
- six==1.14.0
- Spark==1.0
- statsmodels==0.11.1
- subprocess32==3.5.4
- sympy==1.5.1
- tbats==1.0.9
- tqdm==4.64.1
- typing-extensions==4.4.0
- urllib3==1.25.8
- wheel==0.35.1
- zipp==3.12.0

Appendice A: Creazione di un artefatto Zip of Wheels

Dimostriamo con un esempio come creare un artefatto Zip of Wheels. L'esempio mostrato scarica i pacchetti `cryptography` e `scipy` in un artefatto zip of wheels e copia lo zip di wheels in una posizione Amazon S3.

1. È necessario eseguire i comandi per creare lo zip di wheels in un ambiente Amazon Linux simile all'ambiente di Glue. Per informazioni, consulta Appendice B: Dettagli sull'ambiente AWS Glue. Glue 5.1 utilizza AL2023 la versione 3.11 di Python. Crea Dockerfile che creerà questo ambiente:

```
FROM --platform=linux/amd64 public.ecr.aws/amazonlinux/amazonlinux:2023-minimal

# Install Python 3.11, pip, and zip utility
RUN dnf install -y python3.11 pip zip && \
    dnf clean all

WORKDIR /build
```

2. Crea un file requirements.txt

```
cryptography
scipy
```

3. Crea e avvia un contenitore docker

```
# Build docker image
docker build --platform linux/amd64 -t glue-wheel-builder .

# Spin up container
docker run --platform linux/amd64 -v $(pwd)/requirements.txt:/input/
requirements.txt:ro -v $(pwd):/output -it glue-wheel-builder bash
```

4. Esegui i seguenti comandi nell'immagine docker

```
# Create a directory for the wheels
mkdir wheels

# Copy requirements.txt into wheels directory
cp /input/requirements.txt wheels/

# Download the wheels with the correct platform and Python version
pip3 download \
    -r wheels/requirements.txt \
    --dest wheels/ \
    --platform manylinux2014_x86_64 \
    --python-version 311 \
    --only-binary=:all:

# Package the wheels into a zip archive with the .gluewheels.zip suffix
zip -r mylibraries-1.0.0.gluewheels.zip wheels/

# Copy zip to output
cp mylibraries-1.0.0.gluewheels.zip /output/

# Exit the container
exit
```

5. Carica zip of wheels nella sede di Amazon S3

```
aws s3 cp mylibraries-1.0.0.gluewheels.zip s3://amzn-s3-demo-bucket/example-prefix/
```

6. Pulizia opzionale

```
rm mylibraries-1.0.0.gluewheels.zip
rm Dockerfile
rm requirements.txt
```

7. Esegui il job Glue con i seguenti job args:

```
--additional-python-modules s3://amzn-s3-demo-bucket/example-prefix/
mylibraries-1.0.0.gluewheels.zip
--python-modules-installer-option --no-index
```

Appendice B: Dettagli sull'ambiente AWS Glue

Compatibilità della versione Glue e metodi di installazione

| AWS Versione Glue | Versione di Python | Immagine di base | versione glibc | Tag della piattaforma compatibili |
|---|---|---|---|---|
| 5.1 | 3.11 | Amazon Linux 2023 (AL2023) | 2.34 | manylinux_2_34_x86_64<br>manylinux_2_28_x86_64<br>manylinux2014_x86_64 |
| 5.0 | 3.11 | Amazon Linux 2023 (AL2023) | 2.34 | manylinux_2_34_x86_64<br>manylinux_2_28_x86_64<br>manylinux2014_x86_64 |
| 4.0 | 3,10 | Amazon Linux (2AL2) | 2.26 | manylinux2014_x86_64 |
| 3.0 | 3.7 | Amazon Linux (2AL2) | 2.26 | manylinux2014_x86_64 |
| 2.0 | 3.7 | AMI Amazon Linux (AL1) | 2.17 | manylinux2014_x86_64 |

Secondo il modello di responsabilitàAWS condivisa, sei responsabile della gestione di moduli Python aggiuntivi, librerie e relative dipendenze che usi con i tuoi job AWS Glue ETL. Ciò include l'applicazione di aggiornamenti e patch di sicurezza.

AWS Glue non supporta la compilazione di codice nativo nell'ambiente di lavoro. Tuttavia, i job AWS Glue vengono eseguiti in un ambiente Linux gestito da Amazon. È possibile fornire le dipendenze native in un formato compilato tramite un file wheel Python. Si prega di fare riferimento alla tabella precedente per i dettagli sulla compatibilità della versione di AWS Glue.

> **Important**
>
> L'utilizzo di dipendenze incompatibili può causare problemi di runtime, in particolare per le librerie con estensioni native che devono corrispondere all'architettura e alle librerie di sistema dell'ambiente di destinazione. Ogni versione di AWS Glue funziona su una versione di Python specifica con librerie e configurazioni di sistema preinstallate.

## Esempi di codice Python in AWS Glue

- Esempio di codice: unione e relazioni dei dati
- Esempio di codice: preparazione dei dati utilizzando ResolveChoice, Lambda e ApplyMapping

### Esempio di codice: unione e relazioni dei dati

In questo esempio viene usato un set di dati scaricato da http://everypolitician.org/ nel bucket `sample-dataset` in Amazon Simple Storage Service (Amazon S3): `s3://awsglue-datasets/examples/us-legislators/all`. Il set di dati contiene i dati in formato JSON sui legislatori degli Stati Uniti e sui seggi che hanno occupato nella Camera dei rappresentanti e al Senato che sono stati modificati leggermente e resi disponibili in un bucket Amazon S3 pubblico a fini di questo tutorial.

È possibile trovare il codice sorgente di questo esempio nel `join_and_relationalize.py` file dell'archivio degli AWS Glue esempi sul GitHub sito Web.

L'esercitazione illustra con questi dati come:

- Usa un AWS Glue crawler per classificare gli oggetti archiviati in un bucket Amazon S3 pubblico e salvare i relativi schemi nel Glue Data Catalog. AWS
- Esaminare gli schemi e i metadati della tabella restituiti dal crawling.
- Scrivere uno script di estrazione, trasferimento e caricamento (ETL) Python che usa i metadati del catalogo dati per:
  - Unire insieme i dati dei diversi file di origine in un'unica tabella di dati (ovvero denormalizzare i dati).

- Filtrare la tabella unita in tabelle separate in base al tipo di legislatore.
- Scrivere i dati risultanti per separare i file di Apache Parquet per analisi successive.

Il modo preferito per eseguire il debug di Python PySpark o degli script durante l'esecuzione consiste nell'utilizzare AWS Notebooks su Glue Studio. AWS

Fase 1: esecuzione del crawling sui dati nel bucket Amazon S3

1. Accedi a e apri la console all'indirizzo Console di gestione AWS. AWS Glue https://console.aws.amazon.com/glue/
2. Seguendo i passaggi descrittiConfigurazione di un crawler, crea un nuovo crawler in grado di eseguire la scansione del `s3://awsglue-datasets/examples/us-legislators/all` set di dati in un database denominato nel AWS Glue Data `legislators` Catalog. I dati di esempio sono già in questo bucket Amazon S3 pubblico.
3. Esegui il nuovo crawler e controlla il database `legislators`.

   Il crawler crea le seguenti tabelle di metadati:

   - `persons_json`
   - `memberships_json`
   - `organizations_json`
   - `events_json`
   - `areas_json`
   - `countries_r_json`

   Si tratta di una raccolta di tabelle semi-normalizzata contenenti i legislatori e le relative storie.

Fase 2: aggiunta dello script Boilerplate al notebook degli endpoint di sviluppo

Incolla lo script boilerplate seguente nel notebook degli endpoint di sviluppo per importare le librerie AWS Glue necessarie e configurare un singolo oggetto `GlueContext`:

```
import sys
from awsglue.transforms import *
from awsglue.utils import getResolvedOptions
```

```
from pyspark.context import SparkContext
from awsglue.context import GlueContext
from awsglue.job import Job

glueContext = GlueContext(SparkContext.getOrCreate())
```

Fase 3: esame degli schemi dai dati nel catalogo dati

Successivamente, puoi facilmente creare examine a DynamicFrame dal AWS Glue Data Catalog ed esaminare gli schemi dei dati. Ad esempio, per visualizzare lo schema della tabella `persons_json`, aggiungi quanto segue nel notebook:

```
persons = glueContext.create_dynamic_frame.from_catalog(
             database="legislators",
             table_name="persons_json")
print "Count: ", persons.count()
persons.printSchema()
```

Ecco l'output dalle chiamate di stampa:

```
Count:  1961
root
|-- family_name: string
|-- name: string
|-- links: array
|    |-- element: struct
|    |    |-- note: string
|    |    |-- url: string
|-- gender: string
|-- image: string
|-- identifiers: array
|    |-- element: struct
|    |    |-- scheme: string
|    |    |-- identifier: string
|-- other_names: array
|    |-- element: struct
|    |    |-- note: string
|    |    |-- name: string
|    |    |-- lang: string
|-- sort_name: string
```

```
|-- images: array
|    |-- element: struct
|    |    |-- url: string
|-- given_name: string
|-- birth_date: string
|-- id: string
|-- contact_details: array
|    |-- element: struct
|    |    |-- type: string
|    |    |-- value: string
|-- death_date: string
```

Ogni persona nella tabella è membro di alcuni enti del Congresso degli Stati Uniti.

Per visualizzare lo schema della tabella `memberships_json`, digita quando segue:

```
memberships = glueContext.create_dynamic_frame.from_catalog(
                 database="legislators",
                 table_name="memberships_json")
print "Count: ", memberships.count()
memberships.printSchema()
```

L'output è il seguente:

```
Count:  10439
root
|-- area_id: string
|-- on_behalf_of_id: string
|-- organization_id: string
|-- role: string
|-- person_id: string
|-- legislative_period_id: string
|-- start_date: string
|-- end_date: string
```

Gli elementi `organizations` sono i partiti e le due camere del Congresso, il Senato e la Camera dei rappresentanti. Per visualizzare lo schema della tabella `organizations_json`, digita quando segue:

```
orgs = glueContext.create_dynamic_frame.from_catalog(
           database="legislators",
           table_name="organizations_json")
print "Count: ", orgs.count()
orgs.printSchema()
```

L'output è il seguente:

```
Count:  13
root
|-- classification: string
|-- links: array
|    |-- element: struct
|    |    |-- note: string
|    |    |-- url: string
|-- image: string
|-- identifiers: array
|    |-- element: struct
|    |    |-- scheme: string
|    |    |-- identifier: string
|-- other_names: array
|    |-- element: struct
|    |    |-- lang: string
|    |    |-- note: string
|    |    |-- name: string
|-- id: string
|-- name: string
|-- seats: int
|-- type: string
```

Fase 4: filtrare i dati

A questo punto mantieni solo i campi che desideri e rinomina `id` in `org_id`. Il set di dati è sufficientemente piccolo da poterlo visualizzare tutto insieme.

L'elemento `toDF()` converte un oggetto `DynamicFrame` in un elemento `DataFrame` di Apache Spark in modo da poter applicare le trasformazioni già esistenti in Apache Spark SQL:

```
orgs = orgs.drop_fields(['other_names',
                         'identifiers']).rename_field(
                             'id', 'org_id').rename_field(
```

```
                                'name', 'org_name')
orgs.toDF().show()
```

Di seguito è riportato l'output:

```
+--------------+--------------------+--------------------+--------------------+-----
+-----------+--------------------+
|classification|              org_id|            org_name|               links|seats|
     type|               image|
+--------------+--------------------+--------------------+--------------------+-----
+-----------+--------------------+
|         party|            party/al|                  AL|                null| null|
     null|                null|
|         party|      party/democrat|            Democrat|[[website,http://...| null|
     null|https://upload.wi...|
|         party|party/democrat-li...|    Democrat-Liberal|[[website,http://...| null|
     null|                null|
|   legislature|d56acebe-8fdc-47b...|House of Represen...|                null|  435|
lower house|                null|
|         party|   party/independent|         Independent|                null| null|
     null|                null|
|         party|party/new_progres...|     New Progressive|[[website,http://...| null|
     null|https://upload.wi...|
|         party|party/popular_dem...|    Popular Democrat|[[website,http://...| null|
     null|                null|
|         party|    party/republican|          Republican|[[website,http://...| null|
     null|https://upload.wi...|
|         party|party/republican-...|Republican-Conser...|[[website,http://...| null|
     null|                null|
|         party|      party/democrat|            Democrat|[[website,http://...| null|
     null|https://upload.wi...|
|         party|   party/independent|         Independent|                null| null|
     null|                null|
|         party|    party/republican|          Republican|[[website,http://...| null|
     null|https://upload.wi...|
|   legislature|8fa6c3d2-71dc-478...|              Senate|                null|  100|
upper house|                null|
+--------------+--------------------+--------------------+--------------------+-----
+-----------+--------------------+
```

Digita quanto segue per visualizzare gli elementi `organizations` presenti nell'oggetto `memberships`:

```
memberships.select_fields(['organization_id']).toDF().distinct().show()
```

Di seguito è riportato l'output:

```
+--------------------+
|     organization_id|
+--------------------+
|d56acebe-8fdc-47b...|
|8fa6c3d2-71dc-478...|
+--------------------+
```

Fase 5: unione dei dati

Usa quindi AWS Glue per unire queste tabelle relazionali e creare una tabella con la cronologia completa per l'oggetto `memberships` dei legislatori e i relativi elementi `organizations`.

1. In primo luogo, unisci `persons` e `memberships` in `id` e `person_id`.
2. Quindi, unisci il risultato a `orgs` in `org_id` e `organization_id`.
3. Quindi, rilascia i campi ridondanti, `person_id` e `org_id`.

Puoi eseguire tutte queste operazioni in una sola riga di codice estesa:

```
l_history = Join.apply(orgs,
                       Join.apply(persons, memberships, 'id', 'person_id'),
                       'org_id', 'organization_id').drop_fields(['person_id',
 'org_id'])
print "Count: ", l_history.count()
l_history.printSchema()
```

L'output è il seguente:

```
Count:  10439
root
|-- role: string
|-- seats: int
```

```
|-- org_name: string
|-- links: array
|    |-- element: struct
|    |    |-- note: string
|    |    |-- url: string
|-- type: string
|-- sort_name: string
|-- area_id: string
|-- images: array
|    |-- element: struct
|    |    |-- url: string
|-- on_behalf_of_id: string
|-- other_names: array
|    |-- element: struct
|    |    |-- note: string
|    |    |-- name: string
|    |    |-- lang: string
|-- contact_details: array
|    |-- element: struct
|    |    |-- type: string
|    |    |-- value: string
|-- name: string
|-- birth_date: string
|-- organization_id: string
|-- gender: string
|-- classification: string
|-- death_date: string
|-- legislative_period_id: string
|-- identifiers: array
|    |-- element: struct
|    |    |-- scheme: string
|    |    |-- identifier: string
|-- image: string
|-- given_name: string
|-- family_name: string
|-- id: string
|-- start_date: string
|-- end_date: string
```

Ora hai la tabella finale che puoi utilizzare per l'analisi. Puoi scrivere in un formato compatto ed efficiente per le analisi dei dati, vale a dire Parquet, che puoi usare per eseguire SQL in AWS Glue, Amazon Athena o Amazon Redshift Spectrum.

La seguente chiamata scrive la tabella in più file per supportare le operazioni di lettura parallela veloce nella fase di analisi successiva:

```
glueContext.write_dynamic_frame.from_options(frame = l_history,
          connection_type = "s3",
          connection_options = {"path": "s3://glue-sample-target/output-dir/
legislator_history"},
          format = "parquet")
```

Per inserire tutti i dati cronologici in un singolo file, devi convertirli in un frame di dati, suddividerlo in partizioni e scriverlo:

```
s_history = l_history.toDF().repartition(1)
s_history.write.parquet('s3://glue-sample-target/output-dir/legislator_single')
```

In alternativa, se vuoi separarlo dal Senato e dalla Camera:

```
l_history.toDF().write.parquet('s3://glue-sample-target/output-dir/legislator_part',
                               partitionBy=['org_name'])
```

Fase 6: trasformare i dati per i database relazionali

AWS Glue permette di scrivere in modo semplice i dati, anche semistrutturati, in database relazionali come Amazon Redshift,. Offre una trasformazione di tipo `relationalize`, che appiattisce gli elementi `DynamicFrames` indipendentemente dalla complessità degli oggetti in frame.

Utilizzando `l_history` `DynamicFrame` in questo esempio, passi il nome di una tabella radice (`hist_root`) e un percorso temporaneo a `relationalize`. Viene restituito un elemento `DynamicFrameCollection`. Puoi quindi elencare i nomi degli elementi `DynamicFrames` nella raccolta:

```
dfc = l_history.relationalize("hist_root", "s3://glue-sample-target/temp-dir/")
dfc.keys()
```

Di seguito è riportato l'output della chiamata `keys`:

```
[u'hist_root', u'hist_root_contact_details', u'hist_root_links',
 u'hist_root_other_names', u'hist_root_images', u'hist_root_identifiers']
```

`Relationalize` suddivide la tabella della cronologia in sei nuove tabelle: una tabella radice che contiene un record per ogni oggetto dell'elemento `DynamicFrame` e le tabelle ausiliarie per le matrici. La gestione delle matrici nei database relazionali spesso non è ottimali, soprattutto quando le matrici diventano grandi. Separando le matrici in tabelle diverse velocizza l'esecuzione delle query.

A questo punto, controlla la separazione esaminando `contact_details`:

```
l_history.select_fields('contact_details').printSchema()
dfc.select('hist_root_contact_details').toDF().where("id = 10 or id =
 75").orderBy(['id','index']).show()
```

Di seguito è riportato l'output della chiamata `show`:

```
root
|-- contact_details: array
|    |-- element: struct
|    |    |-- type: string
|    |    |-- value: string
+---+-----+-----------------------+-------------------------+
| id|index|contact_details.val.type|contact_details.val.value|
+---+-----+-----------------------+-------------------------+
| 10|    0|                    fax|                         |
| 10|    1|                       |             202-225-1314|
| 10|    2|                  phone|                         |
| 10|    3|                       |             202-225-3772|
| 10|    4|                twitter|                         |
| 10|    5|                       |          MikeRossUpdates|
| 75|    0|                    fax|                         |
| 75|    1|                       |             202-225-7856|
| 75|    2|                  phone|                         |
| 75|    3|                       |             202-225-2711|
| 75|    4|                twitter|                         |
| 75|    5|                       |                SenCapito|
+---+-----+-----------------------+-------------------------+
```

Il campo `contact_details` era una matrice di strutture nell'elemento `DynamicFrame` originale. Ogni elemento di tali matrici è una riga separata nella tabella ausiliaria, indicizzata da `index`. L'`id` qui è una chiave esterna nella tabella `hist_root` con la chiave `contact_details`:

```
dfc.select('hist_root').toDF().where(
    "contact_details = 10 or contact_details = 75").select(
       ['id', 'given_name', 'family_name', 'contact_details']).show()
```

Di seguito è riportato l'output:

```
+--------------------+----------+-----------+---------------+
|                  id|given_name|family_name|contact_details|
+--------------------+----------+-----------+---------------+
|f4fc30ee-7b42-432...|      Mike|       Ross|             10|
|e3c60f34-7d1b-4c0...|   Shelley|     Capito|             75|
+--------------------+----------+-----------+---------------+
```

In questi comandi vengono utilizzati `toDF()` e un'espressione `where` per filtrare le righe che vuoi vedere.

Quindi, unendo la tabella `hist_root` con le tabelle ausiliarie ti consente di effettuare le operazioni descritte di seguito.

- Caricare i dati nei database senza il supporto di matrici.
- Eseguire la query di ogni singolo elemento in una matrice con SQL.

Archivia e accedi in modo sicuro alle tue credenziali Amazon Redshift con una connessione AWS Glue. Per informazioni su come creare la tua connessione, vedi Connessione ai dati.

Siete ora pronti a scrivere i vostri dati su una connessione, scorrendo uno alla volta i `DynamicFrames`:

```
for df_name in dfc.keys():
  m_df = dfc.select(df_name)
  print "Writing to table: ", df_name
  glueContext.write_dynamic_frame.from_jdbc_conf(frame = m_df, connection settings
 here)
```

Le impostazioni di connessione variano in base al tipo di database relazionale:

- Per istruzioni su come scrivere su Amazon Redshift, consultare the section called "Connessioni Redshift".
- Per altri database, consultare Tipi e opzioni di connessione per ETL in AWS Glue per Spark.

Conclusioni

Complessivamente, AWS Glue è molto flessibile. Con poche righe di codice, ti consente di eseguire ciò che normalmente ti potrebbe richiedere giorni di scrittura. Puoi trovare tutti gli script source-to-target ETL nel file Python `join_and_relationalize.py` negli esempi su. AWS Glue GitHub

Esempio di codice: preparazione dei dati utilizzando ResolveChoice, Lambda e ApplyMapping

Il set di dati utilizzato in questo esempio è costituito dai dati di pagamento di Medicare Provider scaricati da due set di dati Data.cms.gov: «Inpatient Prospective Payment System Provider Summary for the Top 100 Diagnostis-Related Groups» e «Inpatient Charge Data FY 2011". FY2011 Dopo aver scaricato i dati, abbiamo apportato delle modifiche al set di dati al fine di introdurre alcuni record errati nella parte finale del file. Questo file modificato si trova in un bucket pubblico Amazon S3 in `s3://awsglue-datasets/examples/medicare/Medicare_Hospital_Provider.csv`.

È possibile trovare il codice sorgente di questo esempio nel file dell'archivio degli esempi. `data_cleaning_and_lambda.py`AWS Glue GitHub

Il modo preferito per eseguire il debug di Python PySpark o degli script durante l'esecuzione consiste nell'utilizzare AWS Notebooks su Glue Studio. AWS

Fase 1: esecuzione del crawling sui dati nel bucket Amazon S3

1. Accedi a e apri la console all'indirizzo. Console di gestione AWS AWS Glue https://console.aws.amazon.com/glue/
2. Seguendo il processo descritto inConfigurazione di un crawler, create un nuovo crawler in grado di eseguire la scansione del `s3://awsglue-datasets/examples/medicare/Medicare_Hospital_Provider.csv` file e di inserire i metadati risultanti in un database denominato nel AWS Glue Data `payments` Catalog.
3. Esegui il nuovo crawler e controlla il database `payments`. Il crawler dovrebbe aver creato una tabella di metadati denominata `medicare` nel database dopo aver letto l'inizio del file per determinarne il formato e il delimitatore.

Lo schema della nuova tabella `medicare` è il seguente:

```
Column  name                              Data type
================================================
drg definition                              string
provider id                                 bigint
provider name                               string
provider street address                     string
provider city                               string
provider state                              string
provider zip code                           bigint
hospital referral region description        string
total discharges                            bigint
average covered charges                     string
average total payments                      string
average medicare payments                   string
```

Fase 2: aggiunta dello script boilerplate al notebook degli endpoint di sviluppo

Incolla lo script boilerplate seguente nel notebook degli endpoint di sviluppo per importare le librerie AWS Glue necessarie e configurare un singolo oggetto `GlueContext`:

```
import sys
from awsglue.transforms import *
from awsglue.utils import getResolvedOptions
from pyspark.context import SparkContext
from awsglue.context import GlueContext
from awsglue.job import Job

glueContext = GlueContext(SparkContext.getOrCreate())
```

Fase 3: confronta differenti analisi di schema

Puoi quindi verificare se lo schema riconosciuto da un oggetto `DataFrame` Apache Spark è uguale a quello registrato dal crawler AWS Glue. Esegui questo codice:

```

medicare = spark.read.format(
   "com.databricks.spark.csv").option(
   "header", "true").option(
```

```
   "inferSchema", "true").load(
   's3://awsglue-datasets/examples/medicare/Medicare_Hospital_Provider.csv')
medicare.printSchema()
```

Ecco l'output dalla chiamata `printSchema`:

```
root
 |-- DRG Definition: string (nullable = true)
 |-- Provider Id: string (nullable = true)
 |-- Provider Name: string (nullable = true)
 |-- Provider Street Address: string (nullable = true)
 |-- Provider City: string (nullable = true)
 |-- Provider State: string (nullable = true)
 |-- Provider Zip Code: integer (nullable = true)
 |-- Hospital Referral Region Description: string (nullable = true)
 |--  Total Discharges : integer (nullable = true)
 |--  Average Covered Charges : string (nullable = true)
 |--  Average Total Payments : string (nullable = true)
 |-- Average Medicare Payments: string (nullable = true)
```

Controlla quindi lo schema generato da un oggetto AWS Glue `DynamicFrame`:

```
medicare_dynamicframe = glueContext.create_dynamic_frame.from_catalog(
       database = "payments",
       table_name = "medicare")
medicare_dynamicframe.printSchema()
```

L'output `printSchema` è il seguente:

```
root
 |-- drg definition: string
 |-- provider id: choice
 |    |-- long
 |    |-- string
 |-- provider name: string
 |-- provider street address: string
 |-- provider city: string
 |-- provider state: string
 |-- provider zip code: long
```

```
 |-- hospital referral region description: string
 |-- total discharges: long
 |-- average covered charges: string
 |-- average total payments: string
 |-- average medicare payments: string
```

Il DynamicFrame genera uno schema in cui `provider id` potrebbe essere un tipo `long` o `string`. Lo schema DataFrame elenca `Provider Id` come tipo `string` e il catalogo dati elenca `provider id` come tipo `bigint`.

Qual è corretto? Sono disponibili due record alla fine del file (su 160.000 record) con i valori `string` nella colonna. Questi sono i record errati che sono stati introdotti per illustrare un problema.

Per risolvere questo problema, l'oggetto AWS Glue DynamicFrame introduce il concetto di tipo choice. In questo caso, DynamicFrame mostra che entrambi i valori `long` e `string` possono essere visualizzati nella colonna. Il crawler AWS Glue ha saltato i valori `string` perché ha considerato solo un prefisso di 2 MB di dati. L'Apache Spark DataFrame ha considerato l'intero set di dati, ma è stato costretto ad assegnare il tipo più generale alla colonna, ossia `string`. Infatti, Spark spesso ricorre al caso più generale quando non ci sono tipi complessi o variazioni con cui non è familiare.

Per eseguire una query sulla colonna `provider id`, risolvi prima il tipo di scelta. Puoi utilizzare il metodo di trasformazione `resolveChoice` in DynamicFrame per convertire quei valori `string` in valori `long` con un'opzione `cast:long`:

```
medicare_res = medicare_dynamicframe.resolveChoice(specs = [('provider
 id','cast:long')])
medicare_res.printSchema()
```

L'output `printSchema` è ora:

```
root
 |-- drg definition: string
 |-- provider id: long
 |-- provider name: string
 |-- provider street address: string
 |-- provider city: string
 |-- provider state: string
 |-- provider zip code: long
 |-- hospital referral region description: string
```

```
 |-- total discharges: long
 |-- average covered charges: string
 |-- average total payments: string
 |-- average medicare payments: string
```

Nel caso di un valore `string` di cui non è stato possibile eseguire il cast, AWS Glue ha inserito `null`.

Un'altra opzione consiste nel convertire il tipo di scelta in `struct`, che mantiene i valori di entrambi i tipi.

Quindi, esaminare le righe anomale:

```
medicare_res.toDF().where("'provider id' is NULL").show()
```

Verrà visualizzato quanto segue:

```
+--------------------+----------+--------------+----------------------+-------------
+-------------+-----------------+------------------------------------+---------------
+----------------------+---------------------+------------------------+
|      drg definition|provider id|  provider name|provider street address|provider
 city|provider state|provider zip code|hospital referral region description|total
 discharges|average covered charges|average total payments|average medicare payments|
+--------------------+----------+--------------+----------------------+-------------
+-------------+-----------------+------------------------------------+---------------
+----------------------+---------------------+------------------------+
|948 - SIGNS & SYM...|       null|            INC|       1050 DIVISION ST|
 MAUSTON|            WI|            53948|                        WI - Madison|
      12|             $11961.41|             $4619.00|                $3775.33|
|948 - SIGNS & SYM...|       null| INC- ST JOSEPH|     5000 W CHAMBERS ST|
 MILWAUKEE|            WI|            53210|                        WI - Milwaukee|
       14|             $10514.28|             $5562.50|                $4522.78|
+--------------------+----------+--------------+----------------------+-------------
+-------------+-----------------+------------------------------------+---------------
+----------------------+---------------------+------------------------+
```

Ora rimuovi i due record difettosi, come segue:

```
medicare_dataframe = medicare_res.toDF()
```

```
medicare_dataframe = medicare_dataframe.where("'provider id' is NOT NULL")
```

Fase 4: mappatura dei dati e utilizzo di funzioni Lambda Apache Spark

AWS Glue non supporta ancora direttamente le funzioni Lambda, note anche come funzioni definite dall'utente. Tuttavia, puoi sempre convertire un `DynamicFrame` in e da un `DataFrame` Apache Spark per trarre vantaggio dalle funzionalità Spark, oltre alle funzionalità speciali di `DynamicFrames`.

Trasforma quindi i dati di pagamento in numeri, in modo che i motori di analisi come Amazon Redshift o Amazon Athena possano eseguire i calcoli più rapidamente:

```
from pyspark.sql.functions import udf
from pyspark.sql.types import StringType

chop_f = udf(lambda x: x[1:], StringType())
medicare_dataframe = medicare_dataframe.withColumn(
        "ACC", chop_f(
            medicare_dataframe["average covered charges"])).withColumn(
                "ATP", chop_f(
                    medicare_dataframe["average total payments"])).withColumn(
                        "AMP", chop_f(
                            medicare_dataframe["average medicare payments"]))
medicare_dataframe.select(['ACC', 'ATP', 'AMP']).show()
```

L'output dalla chiamata `show` è:

```
+--------+-------+-------+
|     ACC|    ATP|    AMP|
+--------+-------+-------+
|32963.07|5777.24|4763.73|
|15131.85|5787.57|4976.71|
|37560.37|5434.95|4453.79|
|13998.28|5417.56|4129.16|
|31633.27|5658.33|4851.44|
|16920.79|6653.80|5374.14|
|11977.13|5834.74|4761.41|
|35841.09|8031.12|5858.50|
|28523.39|6113.38|5228.40|
|75233.38|5541.05|4386.94|
|67327.92|5461.57|4493.57|
|39607.28|5356.28|4408.20|
|22862.23|5374.65|4186.02|
```

```
|31110.85|5366.23|4376.23|
|25411.33|5282.93|4383.73|
| 9234.51|5676.55|4509.11|
|15895.85|5930.11|3972.85|
|19721.16|6192.54|5179.38|
|10710.88|4968.00|3898.88|
|51343.75|5996.00|4962.45|
+--------+-------+-------+
only showing top 20 rows
```

Questi sono ancora tutte stringhe nei dati. Puoi utilizzare il potente metodo di trasformazione `apply_mapping` per eliminare, rinominare, trasmettere e nidificare i dati in modo che i dati di altri linguaggi di programmazione e sistemi possano accedere facilmente:

```
from awsglue.dynamicframe import DynamicFrame
medicare_tmp_dyf = DynamicFrame.fromDF(medicare_dataframe, glueContext, "nested")
medicare_nest_dyf = medicare_tmp_dyf.apply_mapping([('drg definition', 'string', 'drg',
 'string'),
                ('provider id', 'long', 'provider.id', 'long'),
                ('provider name', 'string', 'provider.name', 'string'),
                ('provider city', 'string', 'provider.city', 'string'),
                ('provider state', 'string', 'provider.state', 'string'),
                ('provider zip code', 'long', 'provider.zip', 'long'),
                ('hospital referral region description', 'string','rr', 'string'),
                ('ACC', 'string', 'charges.covered', 'double'),
                ('ATP', 'string', 'charges.total_pay', 'double'),
                ('AMP', 'string', 'charges.medicare_pay', 'double')])
medicare_nest_dyf.printSchema()
```

L'output `printSchema` è il seguente:

```
root
 |-- drg: string
 |-- provider: struct
 |    |-- id: long
 |    |-- name: string
 |    |-- city: string
 |    |-- state: string
 |    |-- zip: long
 |-- rr: string
 |-- charges: struct
 |    |-- covered: double
```

```
 |    |-- total_pay: double
 |    |-- medicare_pay: double
```

Trasformando i dati in un`DataFrame` Spark, puoi visualizzare quello che appare ora:

```
medicare_nest_dyf.toDF().show()
```

L'output è il seguente:

```
+--------------------+--------------------+--------------+--------------------+
|                 drg|            provider|            rr|             charges|
+--------------------+--------------------+--------------+--------------------+
|039 - EXTRACRANIA...|[10001,SOUTHEAST ...|    AL - Dothan|[32963.07,5777.24...|
|039 - EXTRACRANIA...|[10005,MARSHALL M...|AL - Birmingham|[15131.85,5787.57...|
|039 - EXTRACRANIA...|[10006,ELIZA COFF...|AL - Birmingham|[37560.37,5434.95...|
|039 - EXTRACRANIA...|[10011,ST VINCENT...|AL - Birmingham|[13998.28,5417.56...|
|039 - EXTRACRANIA...|[10016,SHELBY BAP...|AL - Birmingham|[31633.27,5658.33...|
|039 - EXTRACRANIA...|[10023,BAPTIST ME...|AL - Montgomery|[16920.79,6653.8,...|
|039 - EXTRACRANIA...|[10029,EAST ALABA...|AL - Birmingham|[11977.13,5834.74...|
|039 - EXTRACRANIA...|[10033,UNIVERSITY...|AL - Birmingham|[35841.09,8031.12...|
|039 - EXTRACRANIA...|[10039,HUNTSVILLE...|AL - Huntsville|[28523.39,6113.38...|
|039 - EXTRACRANIA...|[10040,GADSDEN RE...|AL - Birmingham|[75233.38,5541.05...|
|039 - EXTRACRANIA...|[10046,RIVERVIEW ...|AL - Birmingham|[67327.92,5461.57...|
|039 - EXTRACRANIA...|[10055,FLOWERS HO...|    AL - Dothan|[39607.28,5356.28...|
|039 - EXTRACRANIA...|[10056,ST VINCENT...|AL - Birmingham|[22862.23,5374.65...|
|039 - EXTRACRANIA...|[10078,NORTHEAST ...|AL - Birmingham|[31110.85,5366.23...|
|039 - EXTRACRANIA...|[10083,SOUTH BALD...|    AL - Mobile|[25411.33,5282.93...|
|039 - EXTRACRANIA...|[10085,DECATUR GE...|AL - Huntsville|[9234.51,5676.55,...|
|039 - EXTRACRANIA...|[10090,PROVIDENCE...|    AL - Mobile|[15895.85,5930.11...|
|039 - EXTRACRANIA...|[10092,D C H REGI...|AL - Tuscaloosa|[19721.16,6192.54...|
|039 - EXTRACRANIA...|[10100,THOMAS HOS...|    AL - Mobile|[10710.88,4968.0,...|
|039 - EXTRACRANIA...|[10103,BAPTIST ME...|AL - Birmingham|[51343.75,5996.0,...|
+--------------------+--------------------+--------------+--------------------+
only showing top 20 rows
```

Fase 5: scrittura dei dati in Apache Parquet

AWS Glue semplifica la scrittura dei dati in un formato come Apache Parquet, che può essere usato in modo efficiente dai database relazionali:

```
glueContext.write_dynamic_frame.from_options(
       frame = medicare_nest_dyf,
```

```
        connection_type = "s3",
        connection_options = {"path": "s3://glue-sample-target/output-dir/
medicare_parquet"},
        format = "parquet")
```

## AWS Riferimento PySpark alle estensioni Glue

AWS Glue ha creato le seguenti estensioni al PySpark dialetto Python.

- Accesso ai parametri utilizzando `getResolvedOptions`
- PySpark tipi di estensione
- DynamicFrame classe
- DynamicFrameCollection classe
- DynamicFrameWriter classe
- DynamicFrameReader classe
- GlueContext classe

### Accesso ai parametri utilizzando **`getResolvedOptions`**

La funzione dell'utilità AWS Glue `getResolvedOptions(args, options)` ti permette di accedere agli argomenti passati allo script quando esegui un processo. Per usare questa funzione, esegui l'importazione dal modulo AWS Glue `utils` insieme al modulo `sys`:

```
import sys
from awsglue.utils import getResolvedOptions
```

#### **`getResolvedOptions(args, options)`**

- `args`: elenco degli argomenti contenuti in `sys.argv`.
- `options`: una matrice Python dei nomi degli argomenti da recuperare.

Example Recupero degli argomenti passati a un JobRun

Supponiamo di aver creato un file JobRun in uno script, magari all'interno di una funzione Lambda:

```
response = client.start_job_run(
             JobName = 'my_test_Job',
```

```
            Arguments = {
              '--day_partition_key':   'partition_0',
              '--hour_partition_key':  'partition_1',
              '--day_partition_value':  day_partition_value,
              '--hour_partition_value': hour_partition_value } )
```

Per recuperare gli argomenti passati, puoi usare la funzione `getResolvedOptions` come segue:

```
import sys
from awsglue.utils import getResolvedOptions

args = getResolvedOptions(sys.argv,
                          ['JOB_NAME',
                           'day_partition_key',
                           'hour_partition_key',
                           'day_partition_value',
                           'hour_partition_value'])
print "The day-partition key is: ", args['day_partition_key']
print "and the day-partition value is: ", args['day_partition_value']
```

Si noti che gli argomenti vengono definiti con due trattini iniziali ma viene fatto riferimento a essi nello script senza i trattini Gli argomenti utilizzano solo trattini bassi, non trattini. I tuoi argomenti devono seguire questa convenzione per poter essere risolti.

PySpark tipi di estensione

I tipi utilizzati dalle AWS Glue PySpark estensioni.

DataType

La classe base per gli altri tipi di AWS Glue.

### `__init__(properties={})`

- `properties`: proprietà del tipo di dati (opzionale).

### `typeName(cls)`

Restituisce il tipo per la classe di tipo AWS Glue (ovvero il nome della classe senza "Type" nella parte finale).

- `cls`: un'istanza di classe AWS Glue derivata da `DataType`.

`jsonValue( )`

Restituisce un oggetto JSON contenente il tipo di dati e le proprietà della classe:

```
{
  "dataType": typeName,
  "properties": properties
}
```

AtomicType e derivati semplici

Eredita dalla classe DataType e la estende e funge da classe di base per tutti i tipi di dati atomici AWS Glue.

**`fromJsonValue(cls, json_value)`**

Inizializza un'istanza di classe con valori da un oggetto JSON.

- `cls`: un'istanza di classe di tipo AWS Glue da inizializzare.
- `json_value`: l'oggetto JSON dal quale caricare coppie chiave-valore.

I seguenti tipi sono derivate semplici della classe AtomicType:

- `BinaryType`: i dati binari.
- `BooleanType`: i valori booleani.
- `ByteType`: un valore di byte.
- `DateType`: un valore datetime.
- `DoubleType`: un valore doppio in virgola mobile.
- `IntegerType`: un valore intero.
- `LongType`: un valore intero lungo.
- `NullType`: un valore nullo.
- `ShortType`: un valore intero breve.
- `StringType`: una stringa di testo.

- `TimestampType`: un valore di timestamp (in genere in secondi dal 1/1/1970).
- `UnknownType`: un valore di tipo non identificato.

DecimalType(AtomicType)

Eredita la classe AtomicType e la estende per rappresentare un numero decimale (un numero espresso in cifre decimali, opposto ai numeri binari in base 2).

**`__init__(precision=10, scale=2, properties={})`**

- `precision`: il numero di cifre nel numero decimale (opzionale; il valore predefinito è 10).
- `scale`: il numero di cifre alla destra del punto decimale (opzionale; il valore predefinito è 2).
- `properties`: le proprietà del numero decimale (opzionale).

EnumType(AtomicType)

Eredita la classe AtomicType e la estende per rappresentare un'enumerazione delle opzioni valide.

**`__init__(options)`**

- `options`: un elenco delle opzioni enumerate.

Tipi di raccolta

- ArrayType(DataType)
- ChoiceType(DataType)
- MapType(DataType)
- Field(Object)
- StructType(DataType)
- EntityType(DataType)

ArrayType(DataType)

**`__init__(elementType=UnknownType(), properties={})`**

- `elementType`— Il tipo di elementi nell'array (opzionale; l'impostazione predefinita è UnknownType).

- `properties`: proprietà del tipo di matrice (opzionale).

ChoiceType(DataType)

**`__init__(choices=[], properties={})`**

- `choices`: un elenco di possibili scelte (opzionale).
- `properties`: proprietà di queste opzioni (opzionale).

**`add(new_choice)`**

Aggiunge una nuova opzione all'elenco di scelte possibili.

- `new_choice`: l'opzione da aggiungere all'elenco di scelte possibili.

**`merge(new_choices)`**

Unisce un elenco di nuove opzioni con quello esistente.

- `new_choices`: un elenco di nuove opzioni da unire con quelle esistenti.

MapType(DataType)

**`__init__(valueType=UnknownType, properties={})`**

- `valueType`— Il tipo di valori nella mappa (opzionale; l'impostazione predefinita è UnknownType).
- `properties`: proprietà della mappa (opzionale).

Field(Object)

Consente di creare un oggetto campo al di fuori di un oggetto che deriva da DataType.

**`__init__(name, dataType, properties={})`**

- `name`: il nome da assegnare al campo.
- `dataType`: l'oggetto dal quale creare un campo.

- `properties`: proprietà del campo (opzionale).

StructType(DataType)

Definisce una struttura di dati (`struct`).

**`__init__(fields=[], properties={})`**

- `fields`: un elenco dei campi (di tipo `Field`) da includere nella struttura (opzionale).
- `properties`: proprietà della struttura (opzionale).

**`add(field)`**

- `field`: un oggetto di tipo `Field` da aggiungere alla struttura.

**`hasField(field)`**

Restituisce `True` se questa struttura ha un campo con lo stesso nome, altrimenti `False`.

- `field`: un nome campo o un oggetto di tipo `Field` di cui viene utilizzato il nome.

**`getField(field)`**

- `field`: un nome campo o un oggetto di tipo `Field` di cui viene utilizzato il nome. Se la struttura ha un campo con lo stesso nome, viene restituito.

EntityType(DataType)

`__init__(entity, base_type, properties)`

Questa classe non è ancora implementata.

Altri tipi

- DataSource(oggetto)
- DataSink(oggetto)

DataSource(oggetto)

### `__init__(j_source, sql_ctx, name)`

- `j_source`: l'origine dei dati.
- `sql_ctx`: il contesto SQL.
- `name`: il nome data-source.

### `setFormat(format, **options)`

- `++format`: il formato da impostare per l'origine dei dati.
- `options`: un insieme di opzioni da impostare per l'origine dati. Per ulteriori informazioni sulle opzioni di formato, consulta la pagina the section called "Opzioni del formato dei dati".

`getFrame()`

Restituisce un `DynamicFrame` per l'origine dati.

DataSink(oggetto)

### `__init__(j_sink, sql_ctx)`

- `j_sink`: il sink da creare.
- `sql_ctx`: il contesto SQL per il sink dei dati.

### `setFormat(format, **options)`

- `format`: il formato da impostare per il sink dei dati.
- `options`: insieme di opzioni da impostare per il sink dei dati. Per ulteriori informazioni sulle opzioni di formato, consulta la pagina the section called "Opzioni del formato dei dati".

### `setAccumulableSize(size)`

- `size`: la dimensione accumulabile da impostare, in byte.

### `writeFrame(dynamic_frame, info="")`

- `dynamic_frame`: il `DynamicFrame` da scrivere.
- `info`: informazioni sul `DynamicFrame` (opzionale).

### `write(dynamic_frame_or_dfc, info="")`

Scrive un `DynamicFrame` o una `DynamicFrameCollection`.

- `dynamic_frame_or_dfc`: un oggetto `DynamicFrame` o un oggetto `DynamicFrameCollection` da scrivere.
- `info`: informazioni sulla `DynamicFrame` o `DynamicFrames` da scrivere (opzionale).

DynamicFrame classe

Una delle principali astrazioni in Apache Spark è SparkSQL `DataFrame`, che è simile al costrutto `DataFrame` di R e Pandas. A `DataFrame` è simile a una tabella e supporta operazioni in stile funzionale (map/reduce/filter/etc.) e operazioni SQL (select, project, aggregate).

I `DataFrames` sono potenti e ampiamente utilizzati, ma presentano delle limitazioni riguardo operazioni di estrazione, trasformazione e caricamento (ETL). Principalmente, richiedono che venga specificato uno schema prima di caricare qualsiasi dato. SparkSQL risolve il problema eseguendo due passaggi sui dati: il primo per dedurre lo schema e il secondo per caricare i dati. Tuttavia, l'inferenza è limitata e non gestisce i casi di dati non organizzati. Lo stesso campo, ad esempio, potrebbe essere di tipo diverso in record diversi. Apache Spark spesso si ferma e definisce il tipo come `string` utilizzando il testo del campo originale. Questo potrebbe non essere corretto e potrebbe essere richiesto un controllo più preciso sulle modalità di risoluzione delle discrepanze dello schema. Inoltre, per i set di dati di grandi dimensioni, un ulteriore passaggio sui dati di origine potrebbe essere proibitivo in termini di costi.

Per ovviare a queste limitazioni, AWS Glue introduce il`DynamicFrame`. Un `DynamicFrame` è simile a un `DataFrame`, con la differenza che ogni record è autodescrittivo, quindi inizialmente non è richiesto alcuno schema. Al contrario, AWS Glue calcola uno schema on-the-fly quando richiesto e codifica esplicitamente le incongruenze dello schema utilizzando un tipo di scelta (o unione). Puoi risolvere queste incongruenze per rendere i set di dati compatibili con i datastore che richiedono uno schema fisso.

Analogamente, un `DynamicRecord` rappresenta un record logico all'interno di un `DynamicFrame`. È come una riga in un `DataFrame` Spark, con la differenza che è autodescrittivo e può essere utilizzato per dati non conformi a uno schema fisso. Quando si utilizza AWS Glue with PySpark, in genere non si manipola in modo indipendente`DynamicRecords`. Viceversa, di solito si trasforma il set di dati nel suo complesso attraverso il rispettivo `DynamicFrame`.

Dopo aver risolto eventuali incongruenze dello schema, puoi convertire `DynamicFrames` in e da `DataFrames`.

- construction -

- __init__
- fromDF
- toDF

__init__

**`__init__(jdf, glue_ctx, name)`**

- `jdf`: un riferimento al frame di dati nella JVM (Java Virtual Machine).
- `glue_ctx`: un oggetto GlueContext classe.
- `name`: una stringa nome opzionale, vuota per impostazione predefinita.

fromDF

**`fromDF(dataframe, glue_ctx, name)`**

Converte un `DataFrame` in un `DynamicFrame` convertendo campi del `DataFrame` in campi del `DynamicRecord`. Restituisce il nuovo `DynamicFrame`.

Un `DynamicRecord` rappresenta un record logico all'interno di un `DynamicFrame`. È simile a una riga in un `DataFrame` Spark, con la differenza che è autodescrittivo e può essere utilizzato per dati non conformi a uno schema fisso.

Questa funzione prevede che le colonne con nomi duplicati nel `DataFrame` siano già state risolte.

- `dataframe`: il `DataFrame` Apache Spark SQL da convertire (obbligatorio).
- `glue_ctx`: l'oggetto GlueContext classe che specifica il contesto di questa trasformazione (richiesto).

- `name`— Il nome del risultato `DynamicFrame` (opzionale a partire da AWS Glue 3.0).

toDF

**`toDF(options)`**

Converte un `DynamicFrame` in un `DataFrame` Apache Spark, convertendo `DynamicRecords` in campi di `DataFrame`. Restituisce il nuovo `DataFrame`.

Un `DynamicRecord` rappresenta un record logico all'interno di un `DynamicFrame`. È simile a una riga in un `DataFrame` Spark, con la differenza che è autodescrittivo e può essere utilizzato per dati non conformi a uno schema fisso.

- `options`— Un elenco di `ResolveOption` oggetti che specificano come risolvere i tipi di scelta durante la conversione. Questo parametro viene utilizzato per gestire le incongruenze dello schema, non per le opzioni di formato come l'analisi del file CSV.

  Per l'analisi CSV e altre opzioni di formato, specificale nel metodo al momento della creazione, non nel `from_options` DynamicFrame metodo. `toDF`

  Ecco un esempio del modo corretto di gestire le opzioni di formato CSV:

```
from awsglue.context import GlueContext
from awsglue.dynamicframe import DynamicFrame
from pyspark.context import SparkContext

sc = SparkContext()
glueContext = GlueContext(sc)

# Correct: Specify format options in from_options
csv_dyf = glueContext.create_dynamic_frame.from_options(
    connection_type="s3",
    connection_options={"paths": ["s3://my-bucket/path/to/csv/"]},
    format="csv",
    format_options={
        "withHeader": True,
        "separator": ",",
        "inferSchema": True
    }
)

# Convert to DataFrame (no format options needed here)
```

```
csv_df = csv_dyf.toDF()
```

Il `options` parametro in `toDF` è specifico per la risoluzione dei tipi di scelta. Se scegli il tipo di operazione `Project` e `Cast`, devi specificare il tipo di destinazione. Gli esempi includono quanto segue.

```
>>>toDF([ResolveOption("a.b.c", "KeepAsStruct")])
>>>toDF([ResolveOption("a.b.c", "Project", DoubleType())])
```

- information -

- count
- schema
- printSchema
- show
- repartition
- coalesce

count

`count( )`: restituisce il numero di righe nell'oggetto sottostante `DataFrame`.

schema

`schema( )`: restituisce lo schema di questo `DynamicFrame` oppure, se non è disponibile, lo schema del `DataFrame` sottostante.

Per ulteriori informazioni sui tipi di `DynamicFrame` che compongono questo schema, consulta la pagina the section called "Tipi".

printSchema

`printSchema( )`: stampa lo schema dell'oggetto sottostante `DataFrame`.

show

`show(num_rows)`: stampa un numero di righe specificato dall'oggetto sottostante `DataFrame`.

repartition

`repartition(numPartitions)`: restituisce un nuovo oggetto `DynamicFrame` con partizioni `numPartitions`.

coalesce

`coalesce(numPartitions)`: restituisce un nuovo oggetto `DynamicFrame` con partizioni `numPartitions`.

- transforms -

- apply_mapping
- drop_fields
- filter
- join
- map
- mergeDynamicFrame
- relationalize
- rename_field
- resolveChoice
- select_fields
- spigot
- split_fields
- split_rows
- unbox
- the section called "unione"
- unnest
- unnest_ddb_json
- write

apply_mapping

### `apply_mapping(mappings, transformation_ctx="", info="", stageThreshold=0, totalThreshold=0)`

Applica una mappatura dichiarativa a `DynamicFrame` e restituisce un nuovo `DynamicFrame` con tali mappature applicate ai campi specificati. I campi non specificati vengono omessi dal nuovo `DynamicFrame`.

- `mappings`: un elenco di tuple di mappatura (obbligatorio). Ognuna è costituito da: colonna di origine, tipo di origine, colonna di destinazione, tipo di destinazione.

  Se il nome della colonna di origine include un punto (`.`), esso deve essere racchiuso tra apici inversi (``). Ad esempio, per mappare `this.old.name` (stringa) a `thisNewName`, devi utilizzare la tupla seguente:

  ```
  ("`this.old.name`", "string", "thisNewName", "string")
  ```

- `transformation_ctx`: una stringa univoca utilizzata per identificare informazioni sullo stato (opzionale).
- `info`: una stringa da associare alla segnalazione errori per questa trasformazione (opzionale).
- `stageThreshold`: il numero di errori rilevati durante questa trasformazione raggiunto il quale il processo deve interrompersi (facoltativo). L'impostazione predefinita è zero, indicando che il processo non deve terminare.
- `totalThreshold`: il numero massimo di errori rilevati durante questa trasformazione raggiunto il quale il processo deve interrompersi (facoltativo). L'impostazione predefinita è zero, indicando che il processo non deve terminare.

Esempio: usa apply_mapping per rinominare campi e modificare tipi di campo

L'esempio di codice seguente mostra come utilizzare il metodo `apply_mapping` per rinominare i campi selezionati e modificare tipi di campo.

> **Note**
>
> Per accedere al set di dati utilizzato in questo esempio, consulta Esempio di codice: unione e relazioni dei dati e segui le istruzioni in Fase 1: esecuzione del crawling sui dati nel bucket Amazon S3.

```
# Example: Use apply_mapping to reshape source data into
# the desired column names and types as a new DynamicFrame

from pyspark.context import SparkContext
from awsglue.context import GlueContext

# Create GlueContext
sc = SparkContext.getOrCreate()
glueContext = GlueContext(sc)

# Create a DynamicFrame and view its schema
persons = glueContext.create_dynamic_frame.from_catalog(
    database="legislators", table_name="persons_json"
)
print("Schema for the persons DynamicFrame:")
persons.printSchema()

# Select and rename fields, change field type
print("Schema for the persons_mapped DynamicFrame, created with apply_mapping:")
persons_mapped = persons.apply_mapping(
    [
        ("family_name", "String", "last_name", "String"),
        ("name", "String", "first_name", "String"),
        ("birth_date", "String", "date_of_birth", "Date"),
    ]
)
persons_mapped.printSchema()
```

## Output

```
Schema for the persons DynamicFrame:
root
|-- family_name: string
|-- name: string
|-- links: array
|    |-- element: struct
|    |    |-- note: string
|    |    |-- url: string
|-- gender: string
|-- image: string
|-- identifiers: array
|    |-- element: struct
```

```
|    |    |-- scheme: string
|    |    |-- identifier: string
|-- other_names: array
|    |-- element: struct
|    |    |-- lang: string
|    |    |-- note: string
|    |    |-- name: string
|-- sort_name: string
|-- images: array
|    |-- element: struct
|    |    |-- url: string
|-- given_name: string
|-- birth_date: string
|-- id: string
|-- contact_details: array
|    |-- element: struct
|    |    |-- type: string
|    |    |-- value: string
|-- death_date: string

Schema for the persons_mapped DynamicFrame, created with apply_mapping:
root
|-- last_name: string
|-- first_name: string
|-- date_of_birth: date
```

drop_fields

### `drop_fields(paths, transformation_ctx="", info="", stageThreshold=0, totalThreshold=0)`

Richiama la trasformazione FlatMap classe per rimuovere campi da un `DynamicFrame`. Restituisce un nuovo `DynamicFrame` con i campi specificati rimossi.

- `paths`: un elenco di stringhe. Ognuna contiene il percorso completo di un nodo del campo da rimuovere. Puoi utilizzare la notazione a punti per specificare campi nidificati. Ad esempio, se il campo `first` è figlio del campo `name` nell'albero, specifica `"name.first"` per il percorso.

  Se il nome di un nodo di campo contiene un punto (`.`) letterale, è necessario racchiudere il nome tra apici inversi (`` ` ``).

- `transformation_ctx`: una stringa univoca utilizzata per identificare informazioni sullo stato (opzionale).
- `info`: una stringa da associare alla segnalazione errori per questa trasformazione (opzionale).
- `stageThreshold`: il numero di errori rilevati durante questa trasformazione raggiunto il quale il processo deve interrompersi (facoltativo). L'impostazione predefinita è zero, indicando che il processo non deve terminare.
- `totalThreshold`: il numero massimo di errori rilevati durante questa trasformazione raggiunto il quale il processo deve interrompersi (facoltativo). L'impostazione predefinita è zero, indicando che il processo non deve terminare.

Esempio: utilizza drop_fields per rimuovere campi da un **`DynamicFrame`**

Questo esempio di codice utilizza il metodo `drop_fields` per rimuovere i campi di primo livello e i campi nidificati selezionati da un `DynamicFrame`.

Set di dati di esempio

L'esempio utilizza il set di dati seguente rappresentato dalla tabella `EXAMPLE-FRIENDS-DATA` nel codice:

```
{"name": "Sally", "age": 23, "location": {"state": "WY", "county": "Fremont"},
 "friends": []}
{"name": "Varun", "age": 34, "location": {"state": "NE", "county": "Douglas"},
 "friends": [{"name": "Arjun", "age": 3}]}
{"name": "George", "age": 52, "location": {"state": "NY"}, "friends": [{"name":
 "Fred"}, {"name": "Amy", "age": 15}]}
{"name": "Haruki", "age": 21, "location": {"state": "AK", "county": "Denali"}}
{"name": "Sheila", "age": 63, "friends": [{"name": "Nancy", "age": 22}]}
```

Esempio di codice

```
# Example: Use drop_fields to remove top-level and nested fields from a DynamicFrame.
# Replace MY-EXAMPLE-DATABASE with your Glue Data Catalog database name.
# Replace EXAMPLE-FRIENDS-DATA with your table name.

from pyspark.context import SparkContext
from awsglue.context import GlueContext

# Create GlueContext
```

```
sc = SparkContext.getOrCreate()
glueContext = GlueContext(sc)

# Create a DynamicFrame from Glue Data Catalog
glue_source_database = "MY-EXAMPLE-DATABASE"
glue_source_table = "EXAMPLE-FRIENDS-DATA"

friends = glueContext.create_dynamic_frame.from_catalog(
    database=glue_source_database, table_name=glue_source_table
)
print("Schema for friends DynamicFrame before calling drop_fields:")
friends.printSchema()

# Remove location.county, remove friends.age, remove age
friends = friends.drop_fields(paths=["age", "location.county", "friends.age"])
print("Schema for friends DynamicFrame after removing age, county, and friend age:")
friends.printSchema()
```

## Output

```
Schema for friends DynamicFrame before calling drop_fields:
root
|-- name: string
|-- age: int
|-- location: struct
|    |-- state: string
|    |-- county: string
|-- friends: array
|    |-- element: struct
|    |    |-- name: string
|    |    |-- age: int

Schema for friends DynamicFrame after removing age, county, and friend age:
root
|-- name: string
|-- location: struct
|    |-- state: string
|-- friends: array
|    |-- element: struct
|    |    |-- name: string
```

filter

### `filter(f, transformation_ctx="", info="", stageThreshold=0, totalThreshold=0)`

Restituisce un nuovo `DynamicFrame` contenente tutti i `DynamicRecords` nel `DynamicFrame` di input che soddisfano la funzione predicato specificata `f`.

- `f`: funzione predicato da applicare all'oggetto `DynamicFrame`. La funzione deve richiedere un `DynamicRecord` come argomento e restituire True se il `DynamicRecord` soddisfa i requisiti del filtro o False in caso contrario (obbligatorio).

  Un `DynamicRecord` rappresenta un record logico all'interno di un `DynamicFrame`. È simile a una riga in un `DataFrame` Spark, con la differenza che è autodescrittivo e può essere utilizzato per dati non conformi a uno schema fisso.
- `transformation_ctx`: una stringa univoca utilizzata per identificare informazioni sullo stato (opzionale).
- `info`: una stringa da associare alla segnalazione errori per questa trasformazione (opzionale).
- `stageThreshold`: il numero di errori rilevati durante questa trasformazione raggiunto il quale il processo deve interrompersi (facoltativo). L'impostazione predefinita è zero, indicando che il processo non deve terminare.
- `totalThreshold`: il numero massimo di errori rilevati durante questa trasformazione raggiunto il quale il processo deve interrompersi (facoltativo). L'impostazione predefinita è zero, indicando che il processo non deve terminare.

Esempio: usa il filtro per ottenere una selezione filtrata di campi

In questo esempio viene utilizzato il metodo `filter` per creare un nuovo `DynamicFrame` che include una selezione filtrata di campi di un altro `DynamicFrame`.

Come il metodo `map`, `filter` assume una funzione come argomento che viene applicato a ogni record nel `DynamicFrame` originario. La funzione accetta un record come input e restituisce un valore booleano. Se il valore restituito è vero, il record viene incluso nel `DynamicFrame` risultante. Se è falso, il record viene escluso.

> **Note**
>
> Per accedere al set di dati utilizzato in questo esempio, consulta Esempio di codice: preparazione dei dati utilizzando ResolveChoice, Lambda e ApplyMapping e segui le istruzioni in Fase 1: esecuzione del crawling sui dati nel bucket Amazon S3.

```
# Example: Use filter to create a new DynamicFrame
# with a filtered selection of records

from pyspark.context import SparkContext
from awsglue.context import GlueContext

# Create GlueContext
sc = SparkContext.getOrCreate()
glueContext = GlueContext(sc)

# Create DynamicFrame from Glue Data Catalog
medicare = glueContext.create_dynamic_frame.from_options(
    "s3",
    {
        "paths": [
            "s3://awsglue-datasets/examples/medicare/Medicare_Hospital_Provider.csv"
        ]
    },
    "csv",
    {"withHeader": True},
)

# Create filtered DynamicFrame with custom lambda
# to filter records by Provider State and Provider City
sac_or_mon = medicare.filter(
    f=lambda x: x["Provider State"] in ["CA", "AL"]
    and x["Provider City"] in ["SACRAMENTO", "MONTGOMERY"]
)

# Compare record counts
print("Unfiltered record count: ", medicare.count())
print("Filtered record count:  ", sac_or_mon.count())
```

Output

```
Unfiltered record count:  163065
Filtered record count:   564
```

join

### `join(paths1, paths2, frame2, transformation_ctx="", info="", stageThreshold=0, totalThreshold=0)`

Esegue un equi join con un altro `DynamicFrame` e restituisce il `DynamicFrame` risultante.

- `paths1`: un elenco delle chiavi di questo frame da unire.
- `paths2`: un elenco delle chiavi dell'altro frame da unire.
- `frame2`: l'altro `DynamicFrame` da unire.
- `transformation_ctx`: una stringa univoca utilizzata per identificare informazioni sullo stato (opzionale).
- `info`: una stringa da associare alla segnalazione errori per questa trasformazione (opzionale).
- `stageThreshold`: il numero di errori rilevati durante questa trasformazione raggiunto il quale il processo deve interrompersi (facoltativo). L'impostazione predefinita è zero, indicando che il processo non deve terminare.
- `totalThreshold`: il numero massimo di errori rilevati durante questa trasformazione raggiunto il quale il processo deve interrompersi (facoltativo). L'impostazione predefinita è zero, indicando che il processo non deve terminare.

Esempio: usa join per combinare **`DynamicFrames`**

Questo esempio utilizza il `join` metodo per eseguire un join su tre`DynamicFrames`. AWS Glue esegue l'unione in base alle chiavi di campo fornite. Il `DynamicFrame` risultante contiene le righe dei due frame originali in cui le chiavi specificate corrispondono.

Tieni presente che la trasformazione `join` mantiene intatti tutti i campi. Ciò significa che i campi specificati per la corrispondenza vengono visualizzati nel risultato DynamicFrame, anche se sono ridondanti e contengono le stesse chiavi. In questo esempio viene utilizzato `drop_fields` per rimuovere tali chiavi ridondanti dopo l'unione.

> **Note**
>
> Per accedere al set di dati utilizzato in questo esempio, consulta Esempio di codice: unione e relazioni dei dati e segui le istruzioni in Fase 1: esecuzione del crawling sui dati nel bucket Amazon S3.

```
# Example: Use join to combine data from three DynamicFrames

from pyspark.context import SparkContext
from awsglue.context import GlueContext

# Create GlueContext
sc = SparkContext.getOrCreate()
glueContext = GlueContext(sc)

# Load DynamicFrames from Glue Data Catalog
persons = glueContext.create_dynamic_frame.from_catalog(
    database="legislators", table_name="persons_json"
)
memberships = glueContext.create_dynamic_frame.from_catalog(
    database="legislators", table_name="memberships_json"
)
orgs = glueContext.create_dynamic_frame.from_catalog(
    database="legislators", table_name="organizations_json"
)
print("Schema for the persons DynamicFrame:")
persons.printSchema()
print("Schema for the memberships DynamicFrame:")
memberships.printSchema()
print("Schema for the orgs DynamicFrame:")
orgs.printSchema()

# Join persons and memberships by ID
persons_memberships = persons.join(
    paths1=["id"], paths2=["person_id"], frame2=memberships
)

# Rename and drop fields from orgs
# to prevent field name collisions with persons_memberships
orgs = (
    orgs.drop_fields(["other_names", "identifiers"])
```

```
    .rename_field("id", "org_id")
    .rename_field("name", "org_name")
)

# Create final join of all three DynamicFrames
legislators_combined = orgs.join(
    paths1=["org_id"], paths2=["organization_id"], frame2=persons_memberships
).drop_fields(["person_id", "org_id"])

# Inspect the schema for the joined data
print("Schema for the new legislators_combined DynamicFrame:")
legislators_combined.printSchema()
```

## Output

```
Schema for the persons DynamicFrame:
root
|-- family_name: string
|-- name: string
|-- links: array
|    |-- element: struct
|    |    |-- note: string
|    |    |-- url: string
|-- gender: string
|-- image: string
|-- identifiers: array
|    |-- element: struct
|    |    |-- scheme: string
|    |    |-- identifier: string
|-- other_names: array
|    |-- element: struct
|    |    |-- lang: string
|    |    |-- note: string
|    |    |-- name: string
|-- sort_name: string
|-- images: array
|    |-- element: struct
|    |    |-- url: string
|-- given_name: string
|-- birth_date: string
|-- id: string
|-- contact_details: array
|    |-- element: struct
```

```
|    |    |-- type: string
|    |    |-- value: string
|-- death_date: string

Schema for the memberships DynamicFrame:
root
|-- area_id: string
|-- on_behalf_of_id: string
|-- organization_id: string
|-- role: string
|-- person_id: string
|-- legislative_period_id: string
|-- start_date: string
|-- end_date: string

Schema for the orgs DynamicFrame:
root
|-- identifiers: array
|    |-- element: struct
|    |    |-- scheme: string
|    |    |-- identifier: string
|-- other_names: array
|    |-- element: struct
|    |    |-- lang: string
|    |    |-- note: string
|    |    |-- name: string
|-- id: string
|-- classification: string
|-- name: string
|-- links: array
|    |-- element: struct
|    |    |-- note: string
|    |    |-- url: string
|-- image: string
|-- seats: int
|-- type: string

Schema for the new legislators_combined DynamicFrame:
root
|-- role: string
|-- seats: int
|-- org_name: string
|-- links: array
|    |-- element: struct
```

```
|    |    |-- note: string
|    |    |-- url: string
|-- type: string
|-- sort_name: string
|-- area_id: string
|-- images: array
|    |-- element: struct
|    |    |-- url: string
|-- on_behalf_of_id: string
|-- other_names: array
|    |-- element: struct
|    |    |-- note: string
|    |    |-- name: string
|    |    |-- lang: string
|-- contact_details: array
|    |-- element: struct
|    |    |-- type: string
|    |    |-- value: string
|-- name: string
|-- birth_date: string
|-- organization_id: string
|-- gender: string
|-- classification: string
|-- legislative_period_id: string
|-- identifiers: array
|    |-- element: struct
|    |    |-- scheme: string
|    |    |-- identifier: string
|-- image: string
|-- given_name: string
|-- start_date: string
|-- family_name: string
|-- id: string
|-- death_date: string
|-- end_date: string
```

map

**`map(f, transformation_ctx="", info="", stageThreshold=0, totalThreshold=0)`**

Restituisce un nuovo `DynamicFrame` ottenuto applicando la funzione di mappatura specificata a tutti i record nel `DynamicFrame` originale.

- `f`: funzione di mappatura da applicare a tutti i record nell'oggetto `DynamicFrame`. La funzione deve richiedere un `DynamicRecord` come argomento e restituire un nuovo `DynamicRecord` (obbligatorio).

  Un `DynamicRecord` rappresenta un record logico all'interno di un `DynamicFrame`. È simile a una riga in un `DataFrame` Apache Spark, con la differenza che è autodescrittivo e può essere utilizzato per dati non conformi a uno schema fisso.
- `transformation_ctx`: una stringa univoca utilizzata per identificare informazioni sullo stato (opzionale).
- `info`: una stringa associata a errori nella trasformazione (facoltativo).
- `stageThreshold`: il numero massimo di errori che si possono verificare nella trasformazione prima che venga arrestata (facoltativo). Il valore di default è zero.
- `totalThreshold`: il numero massimo di errori che si possono verificare in totale prima che l'elaborazione venga arrestata (facoltativo). Il valore di default è zero.

Esempio: utilizza la mappa per applicare una funzione a ogni record in un **`DynamicFrame`**

In questo esempio viene utilizzato il metodo `map` per applicare una funzione a ogni record di un `DynamicFrame`. Nello specifico, questo esempio applica una funzione denominata `MergeAddress` a ogni record per unire diversi campi indirizzo in un singolo tipo `struct`.

> **Note**
>
> Per accedere al set di dati utilizzato in questo esempio, consulta Esempio di codice: preparazione dei dati utilizzando ResolveChoice, Lambda e ApplyMapping e segui le istruzioni in Fase 1: esecuzione del crawling sui dati nel bucket Amazon S3.

```
# Example: Use map to combine fields in all records
# of a DynamicFrame

from pyspark.context import SparkContext
from awsglue.context import GlueContext

# Create GlueContext
sc = SparkContext.getOrCreate()
glueContext = GlueContext(sc)
```

```
# Create a DynamicFrame and view its schema
medicare = glueContext.create_dynamic_frame.from_options(
        "s3",
        {"paths": ["s3://awsglue-datasets/examples/medicare/
Medicare_Hospital_Provider.csv"]},
        "csv",
        {"withHeader": True})
print("Schema for medicare DynamicFrame:")
medicare.printSchema()

# Define a function to supply to the map transform
# that merges address fields into a single field
def MergeAddress(rec):
  rec["Address"] = {}
  rec["Address"]["Street"] = rec["Provider Street Address"]
  rec["Address"]["City"] = rec["Provider City"]
  rec["Address"]["State"] = rec["Provider State"]
  rec["Address"]["Zip.Code"] = rec["Provider Zip Code"]
  rec["Address"]["Array"] = [rec["Provider Street Address"], rec["Provider City"],
 rec["Provider State"], rec["Provider Zip Code"]]
  del rec["Provider Street Address"]
  del rec["Provider City"]
  del rec["Provider State"]
  del rec["Provider Zip Code"]
  return rec


# Use map to apply MergeAddress to every record
mapped_medicare = medicare.map(f = MergeAddress)
print("Schema for mapped_medicare DynamicFrame:")
mapped_medicare.printSchema()
```

## Output

```
Schema for medicare DynamicFrame:
root
|-- DRG Definition: string
|-- Provider Id: string
|-- Provider Name: string
|-- Provider Street Address: string
|-- Provider City: string
|-- Provider State: string
|-- Provider Zip Code: string
```

```
|-- Hospital Referral Region Description: string
|-- Total Discharges: string
|-- Average Covered Charges: string
|-- Average Total Payments: string
|-- Average Medicare Payments: string

Schema for mapped_medicare DynamicFrame:
root
|-- Average Total Payments: string
|-- Average Covered Charges: string
|-- DRG Definition: string
|-- Average Medicare Payments: string
|-- Hospital Referral Region Description: string
|-- Address: struct
|    |-- Zip.Code: string
|    |-- City: string
|    |-- Array: array
|    |    |-- element: string
|    |-- State: string
|    |-- Street: string
|-- Provider Id: string
|-- Total Discharges: string
|-- Provider Name: string
```

mergeDynamicFrame

**`mergeDynamicFrame(stage_dynamic_frame, primary_keys, transformation_ctx = "", options = {}, info = "", stageThreshold = 0, totalThreshold = 0)`**

Unisce questo `DynamicFrame` con un `DynamicFrame` temporaneo basato sulle chiavi primarie specificate per identificare i record. I record duplicati (record con le stesse chiavi primarie) non vengono deduplicati. Se non è presente alcun record corrispondente nel frame temporaneo, tutti i record (inclusi i duplicati) vengono mantenuti dall'origine. Se il frame di staging dispone di record corrispondenti, i suoi registri sovrascrivono i registri nell'origine in AWS Glue.

- `stage_dynamic_frame`: il `DynamicFrame` di gestione temporanea da unire.
- `primary_keys`: l'elenco dei campi chiave primaria per abbinare i record dall'origine e dai frame dinamici di gestione temporanei.
- `transformation_ctx`: una stringa univoca utilizzata per recuperare i metadati relativi alla trasformazione corrente (opzionale).

- `options`: una stringa di coppie nome-valore JSON che forniscono informazioni aggiuntive per questa trasformazione. Questo argomento non è attualmente utilizzato.
- `info`: un `String`. Qualsiasi stringa da associare agli errori in questa trasformazione.
- `stageThreshold`: un `Long`. Il numero di errori nella trasformazione specificata per cui l'elaborazione deve restituire un errore.
- `totalThreshold`: un `Long`. Il numero totale di errori fino a questa trasformazione inclusa per i quali l'elaborazione deve restituire un errore.

Questo metodo restituisce un nuovo `DynamicFrame` ottenuto unendo questo `DynamicFrame` con il `DynamicFrame` temporaneo.

Il `DynamicFrame` restituito contiene il record A in questi casi:

- Se `A` esiste sia nel frame di origine che nel frame temporaneo, viene restituito `A` nel frame temporaneo.
- Se `A` si trova nella tabella di origine e `A.primaryKeys` non si trova nel `stagingDynamicFrame`, `A` non viene aggiornato nella tabella temporanea.

Il frame di origine e il frame temporaneo non devono avere lo stesso schema.

Esempio: mergeDynamicFrame da utilizzare per unire due in **`DynamicFrames`** base a una chiave primaria

Il seguente esempio di codice mostra come utilizzare il metodo `mergeDynamicFrame` per unire un `DynamicFrame` con un `DynamicFrame` di staging, in base alla chiave primaria `id`.

Set di dati di esempio

L'esempio utilizza due `DynamicFrames` da una `DynamicFrameCollection` chiamata `split_rows_collection`. Di seguito è riportato un elenco di chiavi in `split_rows_collection`.

```
dict_keys(['high', 'low'])
```

Esempio di codice

```
# Example: Use mergeDynamicFrame to merge DynamicFrames
# based on a set of specified primary keys
```

```
from pyspark.context import SparkContext
from awsglue.context import GlueContext
from awsglue.transforms import SelectFromCollection

# Inspect the original DynamicFrames
frame_low = SelectFromCollection.apply(dfc=split_rows_collection, key="low")
print("Inspect the DynamicFrame that contains rows where ID < 10")
frame_low.toDF().show()

frame_high = SelectFromCollection.apply(dfc=split_rows_collection, key="high")
print("Inspect the DynamicFrame that contains rows where ID > 10")
frame_high.toDF().show()

# Merge the DynamicFrames based on the "id" primary key
merged_high_low = frame_high.mergeDynamicFrame(
    stage_dynamic_frame=frame_low, primary_keys=["id"]
)

# View the results where the ID is 1 or 20
print("Inspect the merged DynamicFrame that contains the combined rows")
merged_high_low.toDF().where("id = 1 or id= 20").orderBy("id").show()
```

## Output

```
Inspect the DynamicFrame that contains rows where ID < 10
+---+-----+----------------------+----------------------+
| id|index|contact_details.val.type|contact_details.val.value|
+---+-----+----------------------+----------------------+
|  1|    0|                   fax|          202-225-3307|
|  1|    1|                 phone|          202-225-5731|
|  2|    0|                   fax|          202-225-3307|
|  2|    1|                 phone|          202-225-5731|
|  3|    0|                   fax|          202-225-3307|
|  3|    1|                 phone|          202-225-5731|
|  4|    0|                   fax|          202-225-3307|
|  4|    1|                 phone|          202-225-5731|
|  5|    0|                   fax|          202-225-3307|
|  5|    1|                 phone|          202-225-5731|
|  6|    0|                   fax|          202-225-3307|
|  6|    1|                 phone|          202-225-5731|
|  7|    0|                   fax|          202-225-3307|
|  7|    1|                 phone|          202-225-5731|
```

```
|  8|    0|                   fax|            202-225-3307|
|  8|    1|                 phone|            202-225-5731|
|  9|    0|                   fax|            202-225-3307|
|  9|    1|                 phone|            202-225-5731|
| 10|    0|                   fax|            202-225-6328|
| 10|    1|                 phone|            202-225-4576|
+---+-----+----------------------+------------------------+
only showing top 20 rows

Inspect the DynamicFrame that contains rows where ID > 10
+---+-----+----------------------+------------------------+
| id|index|contact_details.val.type|contact_details.val.value|
+---+-----+----------------------+------------------------+
| 11|    0|                   fax|            202-225-6328|
| 11|    1|                 phone|            202-225-4576|
| 11|    2|               twitter|          RepTrentFranks|
| 12|    0|                   fax|            202-225-6328|
| 12|    1|                 phone|            202-225-4576|
| 12|    2|               twitter|          RepTrentFranks|
| 13|    0|                   fax|            202-225-6328|
| 13|    1|                 phone|            202-225-4576|
| 13|    2|               twitter|          RepTrentFranks|
| 14|    0|                   fax|            202-225-6328|
| 14|    1|                 phone|            202-225-4576|
| 14|    2|               twitter|          RepTrentFranks|
| 15|    0|                   fax|            202-225-6328|
| 15|    1|                 phone|            202-225-4576|
| 15|    2|               twitter|          RepTrentFranks|
| 16|    0|                   fax|            202-225-6328|
| 16|    1|                 phone|            202-225-4576|
| 16|    2|               twitter|          RepTrentFranks|
| 17|    0|                   fax|            202-225-6328|
| 17|    1|                 phone|            202-225-4576|
+---+-----+----------------------+------------------------+
only showing top 20 rows

Inspect the merged DynamicFrame that contains the combined rows
+---+-----+----------------------+------------------------+
| id|index|contact_details.val.type|contact_details.val.value|
+---+-----+----------------------+------------------------+
|  1|    0|                   fax|            202-225-3307|
|  1|    1|                 phone|            202-225-5731|
| 20|    0|                   fax|            202-225-5604|
| 20|    1|                 phone|            202-225-6536|
```

```
| 20|    2|                 twitter|                USRepLong|
+---+-----+------------------------+-------------------------+
```

relationalize

### `relationalize(root_table_name, staging_path, options, transformation_ctx="", info="", stageThreshold=0, totalThreshold=0)`

Converte un `DynamicFrame` in un modulo che si inserisce in un database relazionale. La relazionalizzazione di un `DynamicFrame` è particolarmente utile quando si desidera spostare dati da un ambiente NoSQL come DynamoDB a un database relazionale come MySQL.

La trasformazione genera un elenco di frame rimuovendo le colonne annidate e ruotando le colonne dell'array. La colonna matrice trasformata mediante pivot può essere unita alla tabella root utilizzando la join-key generata durante la fase di annullamento dell'annidamento.

- `root_table_name`: il nome della tabella root.
- `staging_path`: il percorso in cui il metodo può archiviare le partizioni di tabelle trasformate mediante pivot in formato CSV (facoltativo). Le tabelle trasformate mediante pivot vengono rilette da questo percorso.
- `options`: un dizionario dei parametri opzionali.
- `transformation_ctx`: una stringa univoca utilizzata per identificare informazioni sullo stato (opzionale).
- `info`: una stringa da associare alla segnalazione errori per questa trasformazione (opzionale).
- `stageThreshold`: il numero di errori rilevati durante questa trasformazione raggiunto il quale il processo deve interrompersi (facoltativo). L'impostazione predefinita è zero, indicando che il processo non deve terminare.
- `totalThreshold`: il numero massimo di errori rilevati durante questa trasformazione raggiunto il quale il processo deve interrompersi (facoltativo). L'impostazione predefinita è zero, indicando che il processo non deve terminare.

Esempio: usa la relazionalizzazione per livellare uno schema annidato in un **`DynamicFrame`**

Questo esempio di codice utilizza il metodo `relationalize` per livellare uno schema annidato in una forma adatta a un database relazionale.

Set di dati di esempio

L'esempio utilizza un `DynamicFrame` chiamato `legislators_combined` con lo schema seguente. `legislators_combined` ha più campi annidati come `links`, `images`, e`contact_details`, che verranno livellati dalla trasformazione `relationalize`.

```
root
|-- role: string
|-- seats: int
|-- org_name: string
|-- links: array
|    |-- element: struct
|    |    |-- note: string
|    |    |-- url: string
|-- type: string
|-- sort_name: string
|-- area_id: string
|-- images: array
|    |-- element: struct
|    |    |-- url: string
|-- on_behalf_of_id: string
|-- other_names: array
|    |-- element: struct
|    |    |-- note: string
|    |    |-- name: string
|    |    |-- lang: string
|-- contact_details: array
|    |-- element: struct
|    |    |-- type: string
|    |    |-- value: string
|-- name: string
|-- birth_date: string
|-- organization_id: string
|-- gender: string
|-- classification: string
|-- legislative_period_id: string
|-- identifiers: array
|    |-- element: struct
|    |    |-- scheme: string
|    |    |-- identifier: string
|-- image: string
|-- given_name: string
|-- start_date: string
```

```
|-- family_name: string
|-- id: string
|-- death_date: string
|-- end_date: string
```

### Esempio di codice

```
# Example: Use relationalize to flatten
# a nested schema into a format that fits
# into a relational database.
# Replace DOC-EXAMPLE-S3-BUCKET/tmpDir with your own location.

from pyspark.context import SparkContext
from awsglue.context import GlueContext

# Create GlueContext
sc = SparkContext.getOrCreate()
glueContext = GlueContext(sc)

# Apply relationalize and inspect new tables
legislators_relationalized = legislators_combined.relationalize(
    "l_root", "s3://DOC-EXAMPLE-BUCKET/tmpDir"
)
legislators_relationalized.keys()

# Compare the schema of the contact_details
# nested field to the new relationalized table that
# represents it
legislators_combined.select_fields("contact_details").printSchema()
legislators_relationalized.select("l_root_contact_details").toDF().where(
    "id = 10 or id = 75"
).orderBy(["id", "index"]).show()
```

### Output

L'output seguente consente di confrontare lo schema del campo annidato chiamato `contact_details` con la tabella creata dalla trasformazione `relationalize`. Si noti che i record della tabella rimandano alla tabella principale utilizzando una chiave esterna chiamata `id` e una colonna `index` che rappresenta le posizioni dell'array.

```
dict_keys(['l_root', 'l_root_images', 'l_root_links', 'l_root_other_names',
 'l_root_contact_details', 'l_root_identifiers'])
```

```
root
|-- contact_details: array
|    |-- element: struct
|    |    |-- type: string
|    |    |-- value: string

+---+-----+-----------------------+-----------------------+
| id|index|contact_details.val.type|contact_details.val.value|
+---+-----+-----------------------+-----------------------+
| 10|    0|                    fax|           202-225-4160|
| 10|    1|                  phone|           202-225-3436|
| 75|    0|                    fax|           202-225-6791|
| 75|    1|                  phone|           202-225-2861|
| 75|    2|                twitter|             RepSamFarr|
+---+-----+-----------------------+-----------------------+
```

rename_field

### `rename_field(oldName, newName, transformation_ctx="", info="", stageThreshold=0, totalThreshold=0)`

Rinomina un campo in questo `DynamicFrame` e restituisce un nuovo `DynamicFrame` con il campo rinominato.

- `oldName`: il percorso completo al nodo che desideri rinominare.

  Se il vecchio nome contiene dei punti, `RenameField` non funziona a meno che non venga racchiuso tra virgolette (`). Ad esempio, per sostituire `this.old.name` con `thisNewName`, chiamare rename_field come segue:

  ```
  newDyF = oldDyF.rename_field("`this.old.name`", "thisNewName")
  ```

- `newName`: il nuovo nome, come un percorso completo.
- `transformation_ctx`: una stringa univoca utilizzata per identificare informazioni sullo stato (opzionale).
- `info`: una stringa da associare alla segnalazione errori per questa trasformazione (opzionale).
- `stageThreshold`: il numero di errori rilevati durante questa trasformazione raggiunto il quale il processo deve interrompersi (facoltativo). L'impostazione predefinita è zero, indicando che il processo non deve terminare.

- `totalThreshold`: il numero massimo di errori rilevati durante questa trasformazione raggiunto il quale il processo deve interrompersi (facoltativo). L'impostazione predefinita è zero, indicando che il processo non deve terminare.

### Esempio: usa rename_field per rinominare i campi in un **`DynamicFrame`**

Questo esempio di codice utilizza il metodo `rename_field` per rinominare i campi in un `DynamicFrame`. Si noti che l'esempio utilizza il concatenamento di metodi per rinominare più campi contemporaneamente.

> **Note**
>
> Per accedere al set di dati utilizzato in questo esempio, consulta Esempio di codice: unione e relazioni dei dati e segui le istruzioni in Fase 1: esecuzione del crawling sui dati nel bucket Amazon S3.

#### Esempio di codice

```
# Example: Use rename_field to rename fields
# in a DynamicFrame

from pyspark.context import SparkContext
from awsglue.context import GlueContext

# Create GlueContext
sc = SparkContext.getOrCreate()
glueContext = GlueContext(sc)

# Inspect the original orgs schema
orgs = glueContext.create_dynamic_frame.from_catalog(
    database="legislators", table_name="organizations_json"
)
print("Original orgs schema: ")
orgs.printSchema()

# Rename fields and view the new schema
orgs = orgs.rename_field("id", "org_id").rename_field("name", "org_name")
print("New orgs schema with renamed fields: ")
orgs.printSchema()
```

## Output

```
Original orgs schema:
root
|-- identifiers: array
|    |-- element: struct
|    |    |-- scheme: string
|    |    |-- identifier: string
|-- other_names: array
|    |-- element: struct
|    |    |-- lang: string
|    |    |-- note: string
|    |    |-- name: string
|-- id: string
|-- classification: string
|-- name: string
|-- links: array
|    |-- element: struct
|    |    |-- note: string
|    |    |-- url: string
|-- image: string
|-- seats: int
|-- type: string

New orgs schema with renamed fields:
root
|-- identifiers: array
|    |-- element: struct
|    |    |-- scheme: string
|    |    |-- identifier: string
|-- other_names: array
|    |-- element: struct
|    |    |-- lang: string
|    |    |-- note: string
|    |    |-- name: string
|-- classification: string
|-- org_id: string
|-- org_name: string
|-- links: array
|    |-- element: struct
|    |    |-- note: string
|    |    |-- url: string
|-- image: string
|-- seats: int
```

```
|-- type: string
```

resolveChoice

**`resolveChoice(specs = None, choice = "" , database = None , table_name = None , transformation_ctx="", info="", stageThreshold=0, totalThreshold=0, catalog_id = None)`**

Risolve un tipo di scelta all'interno di questo `DynamicFrame` e restituisce il nuovo `DynamicFrame`.

- `specs`: elenco di ambiguità specifiche da risolvere, ognuna sotto forma di tupla: `(field_path, action)`.

  Ci sono due modi per utilizzare `resolveChoice`. Il primo consiste nell'indicare un argomento `specs` per specificare una sequenza di campi specifici e come risolverli. L'altra modalità per `resolveChoice` è usare un singolo argomento `choice` per specificare una singola risoluzione per tutti i `ChoiceTypes`.

  I valori per `specs` vengono specificati come tuple costituiti da coppie `(field_path, action)`. Il valore `field_path` identifica un elemento ambiguo specifico e il valore `action` identifica la soluzione corrispondente. Sono disponibili le operazioni seguenti:

  - `cast:`*`type`*: tenta di trasmettere tutti i valori al tipo specificato. Ad esempio: `cast:int`.
  - `make_cols`: converte ogni tipo distinto in colonna con il nome *`columnName_type`*. Risolve una potenziale ambiguità livellando i dati. Ad esempio, se `columnA` fosse un `int` o una `string`, la soluzione consisterebbe nel produrre due colonne denominate `columnA_int` e `columnA_string` nel `DynamicFrame` risultante.
  - `make_struct`: risolve una potenziale ambiguità utilizzando una `struct` per rappresentare i dati. Ad esempio, se i dati in una colonna sono un `int` o una `string`, con l'utilizzo dell'operazione `make_struct` viene prodotta una colonna di strutture nel risultante `DynamicFrame`. Ogni struttura contiene sia un `int` che un `string`.
  - `project:`*`type`*: risolve una potenziale ambiguità proiettando tutti i dati su uno dei tipi di dati possibili. Ad esempio, se i dati in una colonna sono un `int` o una `string`, utilizzando un'operazione `project:string` viene prodotta una colonna nel `DynamicFrame` risultante, dove tutti i valori `int` sono stati convertiti in stringhe.

Se il `field_path` identifica un array, inserisci parentesi quadre vuote dopo il nome dell'array per evitare ambiguità. Ad esempio, supponiamo che tu stia lavorando con dati strutturati nel seguente modo:

```
"myList": [
  { "price": 100.00 },
  { "price": "$100.00" }
]
```

Puoi selezionare la versione numerica invece di quella di stringa del prezzo impostando il `field_path` su `"myList[].price"` e la `action` su `"cast:double"`.

> **Note**
>
> Può essere utilizzato solo uno dei parametri `specs` e `choice`. Se il parametro `specs` non è `None`, allora il parametro `choice` deve essere una stringa vuota. Viceversa, se `choice` non è una stringa vuota, allora il parametro `specs` deve essere `None`.

- `choice`: specifica una singola risoluzione per tutti i `ChoiceTypes`. Puoi usare questa modalità nei casi in cui l'elenco completo di `ChoiceTypes` non è noto prima del runtime. Oltre alle operazioni elencate in precedenza per `specs`, questa modalità supporta anche l'operazione seguente:
  - `match_catalog`: tenta di trasmettere ogni `ChoiceType` al tipo corrispondente nella tabella del catalogo specificata.
- `database`: il database del catalogo dati da usare con l'operazione `match_catalog`.
- `table_name`: la tabella del catalogo dati da usare con l'operazione `match_catalog`.
- `transformation_ctx`: una stringa univoca utilizzata per identificare informazioni sullo stato (opzionale).
- `info`: una stringa da associare alla segnalazione errori per questa trasformazione (opzionale).
- `stageThreshold`: il numero di errori rilevati durante questa trasformazione raggiunto il quale il processo deve interrompersi (facoltativo). L'impostazione predefinita è zero, indicando che il processo non deve terminare.
- `totalThreshold`: il numero di errori riscontrati fino a questa trasformazione compresa, raggiunto il quale il processo dovrebbe interrompersi (opzionale: impostazione predefinita: zero, a indicare che il processo non dovrebbe interrompersi).

- `catalog_id`: l'ID catalogo del catalogo dati a cui si accede (l'ID account del catalogo dati). Se impostato su `None` (valore predefinito), utilizza l'ID catalogo dell'account chiamante.

Esempio: utilizzare resolveChoice per gestire una colonna che contiene più tipi

Questo esempio di codice utilizza il metodo `resolveChoice` per specificare come gestire una colonna `DynamicFrame` che contiene valori di più tipi. L'esempio mostra due modi comuni per gestire una colonna con tipi diversi:

- Trasforma la colonna in un singolo tipo di dati.
- Conserva tutti i tipi in colonne separate.

Set di dati di esempio

> **Note**
>
> Per accedere al set di dati utilizzato in questo esempio, consulta Esempio di codice: preparazione dei dati utilizzando ResolveChoice, Lambda e ApplyMapping e segui le istruzioni in Fase 1: esecuzione del crawling sui dati nel bucket Amazon S3.

L'esempio utilizza un `DynamicFrame` chiamato `medicare` con il seguente schema:

```
root
|-- drg definition: string
|-- provider id: choice
|    |-- long
|    |-- string
|-- provider name: string
|-- provider street address: string
|-- provider city: string
|-- provider state: string
|-- provider zip code: long
|-- hospital referral region description: string
|-- total discharges: long
|-- average covered charges: string
|-- average total payments: string
|-- average medicare payments: string
```

Esempio di codice

```
# Example: Use resolveChoice to handle
# a column that contains multiple types

from pyspark.context import SparkContext
from awsglue.context import GlueContext

# Create GlueContext
sc = SparkContext.getOrCreate()
glueContext = GlueContext(sc)

# Load the input data and inspect the "provider id" column
medicare = glueContext.create_dynamic_frame.from_catalog(
    database="payments", table_name="medicare_hospital_provider_csv"
)
print("Inspect the provider id column:")
medicare.toDF().select("provider id").show()

# Cast provider id to type long
medicare_resolved_long = medicare.resolveChoice(specs=[("provider id", "cast:long")])
print("Schema after casting provider id to type long:")
medicare_resolved_long.printSchema()
medicare_resolved_long.toDF().select("provider id").show()

# Create separate columns
# for each provider id type
medicare_resolved_cols = medicare.resolveChoice(choice="make_cols")
print("Schema after creating separate columns for each type:")
medicare_resolved_cols.printSchema()
medicare_resolved_cols.toDF().select("provider id_long", "provider id_string").show()
```

## Output

```
Inspect the 'provider id' column:
+-----------+
|provider id|
+-----------+
|   [10001,]|
|   [10005,]|
|   [10006,]|
|   [10011,]|
|   [10016,]|
|   [10023,]|
|   [10029,]|
```

```
|   [10033,]|
|   [10039,]|
|   [10040,]|
|   [10046,]|
|   [10055,]|
|   [10056,]|
|   [10078,]|
|   [10083,]|
|   [10085,]|
|   [10090,]|
|   [10092,]|
|   [10100,]|
|   [10103,]|
+-----------+
only showing top 20 rows

Schema after casting 'provider id' to type long:
root
|-- drg definition: string
|-- provider id: long
|-- provider name: string
|-- provider street address: string
|-- provider city: string
|-- provider state: string
|-- provider zip code: long
|-- hospital referral region description: string
|-- total discharges: long
|-- average covered charges: string
|-- average total payments: string
|-- average medicare payments: string

+-----------+
|provider id|
+-----------+
|      10001|
|      10005|
|      10006|
|      10011|
|      10016|
|      10023|
|      10029|
|      10033|
|      10039|
|      10040|
```

```
|      10046|
|      10055|
|      10056|
|      10078|
|      10083|
|      10085|
|      10090|
|      10092|
|      10100|
|      10103|
+-----------+
only showing top 20 rows

Schema after creating separate columns for each type:
root
|-- drg definition: string
|-- provider id_string: string
|-- provider id_long: long
|-- provider name: string
|-- provider street address: string
|-- provider city: string
|-- provider state: string
|-- provider zip code: long
|-- hospital referral region description: string
|-- total discharges: long
|-- average covered charges: string
|-- average total payments: string
|-- average medicare payments: string

+----------------+------------------+
|provider id_long|provider id_string|
+----------------+------------------+
|           10001|              null|
|           10005|              null|
|           10006|              null|
|           10011|              null|
|           10016|              null|
|           10023|              null|
|           10029|              null|
|           10033|              null|
|           10039|              null|
|           10040|              null|
|           10046|              null|
|           10055|              null|
```

```
|           10056|              null|
|           10078|              null|
|           10083|              null|
|           10085|              null|
|           10090|              null|
|           10092|              null|
|           10100|              null|
|           10103|              null|
+----------------+------------------+
only showing top 20 rows
```

select_fields

### `select_fields(paths, transformation_ctx="", info="", stageThreshold=0, totalThreshold=0)`

Restituisce un nuovo `DynamicFrame` contenente i campi selezionati.

- `paths`: un elenco di stringhe. Ogni stringa è un percorso completo di un nodo di livello superiore da selezionare.
- `transformation_ctx`: una stringa univoca utilizzata per identificare informazioni sullo stato (opzionale).
- `info`: una stringa da associare alla segnalazione errori per questa trasformazione (opzionale).
- `stageThreshold`: il numero di errori rilevati durante questa trasformazione raggiunto il quale il processo deve interrompersi (facoltativo). L'impostazione predefinita è zero, indicando che il processo non deve terminare.
- `totalThreshold`: il numero massimo di errori rilevati durante questa trasformazione raggiunto il quale il processo deve interrompersi (facoltativo). L'impostazione predefinita è zero, indicando che il processo non deve terminare.

Esempio: usa select_fields per creare un nuovo **`DynamicFrame`** con i campi scelti

L'esempio di codice seguente mostra come utilizzare il metodo `select_fields` per creare un nuovo `DynamicFrame` con un elenco di campi scelto da un `DynamicFrame` esistente.

> **Note**
>
> Per accedere al set di dati utilizzato in questo esempio, consulta Esempio di codice: unione e relazioni dei dati e segui le istruzioni in Fase 1: esecuzione del crawling sui dati nel bucket Amazon S3.

```
# Example: Use select_fields to select specific fields from a DynamicFrame

from pyspark.context import SparkContext
from awsglue.context import GlueContext

# Create GlueContext
sc = SparkContext.getOrCreate()
glueContext = GlueContext(sc)

# Create a DynamicFrame and view its schema
persons = glueContext.create_dynamic_frame.from_catalog(
    database="legislators", table_name="persons_json"
)
print("Schema for the persons DynamicFrame:")
persons.printSchema()

# Create a new DynamicFrame with chosen fields
names = persons.select_fields(paths=["family_name", "given_name"])
print("Schema for the names DynamicFrame, created with select_fields:")
names.printSchema()
names.toDF().show()
```

Output

```
Schema for the persons DynamicFrame:
root
|-- family_name: string
|-- name: string
|-- links: array
|    |-- element: struct
|    |    |-- note: string
|    |    |-- url: string
|-- gender: string
|-- image: string
|-- identifiers: array
```

```
|    |-- element: struct
|    |    |-- scheme: string
|    |    |-- identifier: string
|-- other_names: array
|    |-- element: struct
|    |    |-- lang: string
|    |    |-- note: string
|    |    |-- name: string
|-- sort_name: string
|-- images: array
|    |-- element: struct
|    |    |-- url: string
|-- given_name: string
|-- birth_date: string
|-- id: string
|-- contact_details: array
|    |-- element: struct
|    |    |-- type: string
|    |    |-- value: string
|-- death_date: string

Schema for the names DynamicFrame:
root
|-- family_name: string
|-- given_name: string

+-----------+----------+
|family_name|given_name|
+-----------+----------+
|    Collins|   Michael|
|   Huizenga|      Bill|
|    Clawson|    Curtis|
|    Solomon|    Gerald|
|     Rigell|    Edward|
|      Crapo|   Michael|
|      Hutto|      Earl|
|      Ertel|     Allen|
|     Minish|    Joseph|
|    Andrews|    Robert|
|     Walden|      Greg|
|      Kazen|   Abraham|
|     Turner|   Michael|
|      Kolbe|     James|
|  Lowenthal|      Alan|
```

```
|    Capuano|   Michael|
|   Schrader|      Kurt|
|     Nadler|   Jerrold|
|     Graves|       Tom|
|   McMillan|      John|
+-----------+----------+
only showing top 20 rows
```

simplify_ddb_json

### `simplify_ddb_json(): DynamicFrame`

Semplifica le colonne nidificate in un `DynamicFrame` che si trovano specificamente nella struttura JSON DynamoDB e restituisce un nuovo `DynamicFrame` semplificato. Se sono presenti più tipi o più tipi di Mappa in un tipo di Elenco, gli elementi contenuti nell'Elenco non saranno semplificati. Si noti che si tratta di un tipo specifico di trasformazione che si comporta in modo diverso dalla normale trasformazione `unnest` e richiede che i dati siano già presenti nella struttura JSON DynamoDB. Per ulteriori informazioni, consulta DynamoDB JSON.

Ad esempio, lo schema di lettura di un'esportazione con la struttura JSON DynamoDB potrebbe apparire come segue:

```
root
|-- Item: struct
|    |-- parentMap: struct
|    |    |-- M: struct
|    |    |    |-- childMap: struct
|    |    |    |    |-- M: struct
|    |    |    |    |    |-- appName: struct
|    |    |    |    |    |    |-- S: string
|    |    |    |    |    |-- packageName: struct
|    |    |    |    |    |    |-- S: string
|    |    |    |    |    |-- updatedAt: struct
|    |    |    |    |    |    |-- N: string
|    |-- strings: struct
|    |    |-- SS: array
|    |    |    |-- element: string
|    |-- numbers: struct
|    |    |-- NS: array
|    |    |    |-- element: string
|    |-- binaries: struct
```

```
|    |    |-- BS: array
|    |    |    |-- element: string
|    |-- isDDBJson: struct
|    |    |-- BOOL: boolean
|    |-- nullValue: struct
|    |    |-- NULL: boolean
```

La trasformazione di `simplify_ddb_json()` lo convertirebbe in:

```
root
|-- parentMap: struct
|    |-- childMap: struct
|    |    |-- appName: string
|    |    |-- packageName: string
|    |    |-- updatedAt: string
|-- strings: array
|    |-- element: string
|-- numbers: array
|    |-- element: string
|-- binaries: array
|    |-- element: string
|-- isDDBJson: boolean
|-- nullValue: null
```

Esempio: utilizzo di simply_ddb_json per invocare una semplificazione JSON DynamoDB

Questo esempio di codice utilizza il `simplify_ddb_json` metodo per utilizzare il connettore di esportazione AWS Glue DynamoDB, richiamare un DynamoDB JSON simplify e stampare il numero di partizioni.

Esempio di codice

```
from pyspark.context import SparkContext
from awsglue.context import GlueContext

sc = SparkContext()
glueContext = GlueContext(sc)

dynamicFrame = glueContext.create_dynamic_frame.from_options(
    connection_type = "dynamodb",
    connection_options = {
        'dynamodb.export': 'ddb',
```

```
        'dynamodb.tableArn': '<table arn>',
        'dynamodb.s3.bucket': '<bucket name>',
        'dynamodb.s3.prefix': '<bucket prefix>',
        'dynamodb.s3.bucketOwner': '<account_id of bucket>'
    }
)
simplified = dynamicFrame.simplify_ddb_json()
print(simplified.getNumPartitions())
```

spigot

### `spigot(path, options={})`

Scrive record di esempio in una destinazione specificata per aiutarti a verificare le trasformazioni eseguite dal tuo lavoro.

- `path`: il percorso della destinazione in cui scrivere (obbligatorio).
- `options`: coppie chiave-valore che specificano opzioni (opzionale). L'opzione `"topk"` specifica che devono essere scritti i primi record k. L'opzione `"prob"` specifica la probabilità (sotto forma di valore decimale) di scelta di un dato record. Puoi usarlo per selezionare i record da scrivere.
- `transformation_ctx`: una stringa univoca utilizzata per identificare informazioni sullo stato (opzionale).

Esempio: usa lo spigot per scrivere campi di esempio da **`DynamicFrame`** ad Amazon S3

Questo esempio di codice utilizza il metodo `spigot` per scrivere record di esempio in un bucket Amazon S3 dopo aver applicato la trasformazione `select_fields`.

Set di dati di esempio

> **Note**
>
> Per accedere al set di dati utilizzato in questo esempio, consulta Esempio di codice: unione e relazioni dei dati e segui le istruzioni in Fase 1: esecuzione del crawling sui dati nel bucket Amazon S3.

L'esempio utilizza un `DynamicFrame` chiamato `persons` con il seguente schema:

```
root
```

```
|-- family_name: string
|-- name: string
|-- links: array
|    |-- element: struct
|    |    |-- note: string
|    |    |-- url: string
|-- gender: string
|-- image: string
|-- identifiers: array
|    |-- element: struct
|    |    |-- scheme: string
|    |    |-- identifier: string
|-- other_names: array
|    |-- element: struct
|    |    |-- lang: string
|    |    |-- note: string
|    |    |-- name: string
|-- sort_name: string
|-- images: array
|    |-- element: struct
|    |    |-- url: string
|-- given_name: string
|-- birth_date: string
|-- id: string
|-- contact_details: array
|    |-- element: struct
|    |    |-- type: string
|    |    |-- value: string
|-- death_date: string
```

## Esempio di codice

```
# Example: Use spigot to write sample records
# to a destination during a transformation
# from pyspark.context import SparkContext.
# Replace DOC-EXAMPLE-BUCKET with your own location.

from pyspark.context import SparkContext
from awsglue.context import GlueContext

# Create GlueContext
sc = SparkContext.getOrCreate()
glueContext = GlueContext(sc)
```

```
# Load table data into a DynamicFrame
persons = glueContext.create_dynamic_frame.from_catalog(
    database="legislators", table_name="persons_json"
)

# Perform the select_fields on the DynamicFrame
persons = persons.select_fields(paths=["family_name", "given_name", "birth_date"])

# Use spigot to write a sample of the transformed data
# (the first 10 records)
spigot_output = persons.spigot(
    path="s3://DOC-EXAMPLE-BUCKET", options={"topk": 10}
)
# Example: Use spigot to write sample records
# to a destination during a transformation
# from pyspark.context import SparkContext.
# Replace DOC-EXAMPLE-BUCKET with your own location.

from pyspark.context import SparkContext
from awsglue.context import GlueContext

# Create GlueContext
sc = SparkContext.getOrCreate()
glueContext = GlueContext(sc)

# Load table data into a DynamicFrame
persons = glueContext.create_dynamic_frame.from_catalog(
    database="legislators", table_name="persons_json"
)

# Perform the select_fields on the DynamicFrame
persons = persons.select_fields(paths=["family_name", "given_name", "birth_date"])

# Use spigot to write a sample of the transformed data
# (the first 10 records)
spigot_output = persons.spigot(
    path="s3://DOC-EXAMPLE-BUCKET", options={"topk": 10}
)
```

Output

Di seguito è riportato un esempio di dati che `spigot` scrive su Amazon S3. Poiché è stato specificato il codice di esempio `options={"topk": 10}`, i dati di esempio contengono i primi 10 record.

```
{"family_name":"Collins","given_name":"Michael","birth_date":"1944-10-15"}
{"family_name":"Huizenga","given_name":"Bill","birth_date":"1969-01-31"}
{"family_name":"Clawson","given_name":"Curtis","birth_date":"1959-09-28"}
{"family_name":"Solomon","given_name":"Gerald","birth_date":"1930-08-14"}
{"family_name":"Rigell","given_name":"Edward","birth_date":"1960-05-28"}
{"family_name":"Crapo","given_name":"Michael","birth_date":"1951-05-20"}
{"family_name":"Hutto","given_name":"Earl","birth_date":"1926-05-12"}
{"family_name":"Ertel","given_name":"Allen","birth_date":"1937-11-07"}
{"family_name":"Minish","given_name":"Joseph","birth_date":"1916-09-01"}
{"family_name":"Andrews","given_name":"Robert","birth_date":"1957-08-04"}
```

split_fields

### `split_fields(paths, name1, name2, transformation_ctx="", info="", stageThreshold=0, totalThreshold=0)`

Restituisce un nuovo `DynamicFrameCollection` che ne contiene due `DynamicFrames`. Il primo `DynamicFrame` contiene tutti i nodi che sono stati separati e il secondo contiene i nodi rimanenti.

- `paths`: elenco di stringhe, ciascuna delle quali è un percorso completo di un nodo da separare in un nuovo oggetto `DynamicFrame`.
- `name1`: una stringa nome per il `DynamicFrame` separato.
- `name2`: una stringa nome per il `DynamicFrame` che rimane dopo aver separato i nodi specificati.
- `transformation_ctx`: una stringa univoca utilizzata per identificare informazioni sullo stato (opzionale).
- `info`: una stringa da associare alla segnalazione errori per questa trasformazione (opzionale).
- `stageThreshold`: il numero di errori rilevati durante questa trasformazione raggiunto il quale il processo deve interrompersi (facoltativo). L'impostazione predefinita è zero, indicando che il processo non deve terminare.
- `totalThreshold`: il numero massimo di errori rilevati durante questa trasformazione raggiunto il quale il processo deve interrompersi (facoltativo). L'impostazione predefinita è zero, indicando che il processo non deve terminare.

Esempio: usare split_fields per dividere i campi selezionati in un campo separato **`DynamicFrame`**

Questo esempio di codice utilizza il metodo `split_fields` per dividere un elenco di campi specificati in un campo separato `DynamicFrame`.

Set di dati di esempio

L'esempio utilizza un `DynamicFrame` chiamato `l_root_contact_details` che proviene da una raccolta denominata `legislators_relationalized`.

`l_root_contact_details` ha il seguente schema e le seguenti voci.

```
root
|-- id: long
|-- index: int
|-- contact_details.val.type: string
|-- contact_details.val.value: string

+---+-----+-----------------------+-----------------------+
| id|index|contact_details.val.type|contact_details.val.value|
+---+-----+-----------------------+-----------------------+
|  1|    0|                  phone|             202-225-5265|
|  1|    1|                twitter|              kathyhochul|
|  2|    0|                  phone|             202-225-3252|
|  2|    1|                twitter|            repjackyrosen|
|  3|    0|                    fax|             202-225-1314|
|  3|    1|                  phone|             202-225-3772|
...
```

Esempio di codice

```
# Example: Use split_fields to split selected
# fields into a separate DynamicFrame

from pyspark.context import SparkContext
from awsglue.context import GlueContext

# Create GlueContext
sc = SparkContext.getOrCreate()
glueContext = GlueContext(sc)

# Load the input DynamicFrame and inspect its schema
frame_to_split = legislators_relationalized.select("l_root_contact_details")
```

```
print("Inspect the input DynamicFrame schema:")
frame_to_split.printSchema()

# Split id and index fields into a separate DynamicFrame
split_fields_collection = frame_to_split.split_fields(["id", "index"], "left", "right")

# Inspect the resulting DynamicFrames
print("Inspect the schemas of the DynamicFrames created with split_fields:")
split_fields_collection.select("left").printSchema()
split_fields_collection.select("right").printSchema()
```

Output

```
Inspect the input DynamicFrame's schema:
root
|-- id: long
|-- index: int
|-- contact_details.val.type: string
|-- contact_details.val.value: string

Inspect the schemas of the DynamicFrames created with split_fields:
root
|-- id: long
|-- index: int

root
|-- contact_details.val.type: string
|-- contact_details.val.value: string
```

split_rows

### `split_rows(comparison_dict, name1, name2, transformation_ctx="", info="", stageThreshold=0, totalThreshold=0)`

Suddivide una o più righe di un `DynamicFrame` in un nuovo `DynamicFrame`.

Il metodo restituisce un nuovo `DynamicFrameCollection` che contiene due `DynamicFrames`. Il primo `DynamicFrame` contiene tutti i nodi che sono stati separati e il secondo contiene i nodi rimanenti.

- `comparison_dict`: un dizionario in cui la chiave è un percorso verso una colonna e il valore è un altro dizionario per la mappatura di comparatori rispetto a valori con i quali vengono confrontati i

valori di colonna. Ad esempio, `{"age": {">": 10, "<": 20}}` divide tutte le righe il cui valore nella colonna età è superiore a 10 e inferiore a 20.

- `name1`: una stringa nome per il `DynamicFrame` separato.
- `name2`: una stringa nome per il `DynamicFrame` che rimane dopo aver separato i nodi specificati.
- `transformation_ctx`: una stringa univoca utilizzata per identificare informazioni sullo stato (opzionale).
- `info`: una stringa da associare alla segnalazione errori per questa trasformazione (opzionale).
- `stageThreshold`: il numero di errori rilevati durante questa trasformazione raggiunto il quale il processo deve interrompersi (facoltativo). L'impostazione predefinita è zero, indicando che il processo non deve terminare.
- `totalThreshold`: il numero massimo di errori rilevati durante questa trasformazione raggiunto il quale il processo deve interrompersi (facoltativo). L'impostazione predefinita è zero, indicando che il processo non deve terminare.

Esempio: usare split_rows per dividere le righe in un **`DynamicFrame`**

Questo esempio di codice utilizza il metodo `split_rows` per dividere le righe in un `DynamicFrame` in base al valore del campo `id`.

Set di dati di esempio

L'esempio utilizza un `DynamicFrame` chiamato `l_root_contact_details` che proviene da una raccolta denominata `legislators_relationalized`.

`l_root_contact_details` ha il seguente schema e le seguenti voci.

```
root
|-- id: long
|-- index: int
|-- contact_details.val.type: string
|-- contact_details.val.value: string

+---+-----+-----------------------+-----------------------+
| id|index|contact_details.val.type|contact_details.val.value|
+---+-----+-----------------------+-----------------------+
|  1|    0|                  phone|             202-225-5265|
|  1|    1|                twitter|              kathyhochul|
|  2|    0|                  phone|             202-225-3252|
```

```
|  2|    1|                 twitter|            repjackyrosen|
|  3|    0|                     fax|             202-225-1314|
|  3|    1|                   phone|             202-225-3772|
|  3|    2|                 twitter|          MikeRossUpdates|
|  4|    0|                     fax|             202-225-1314|
|  4|    1|                   phone|             202-225-3772|
|  4|    2|                 twitter|          MikeRossUpdates|
|  5|    0|                     fax|             202-225-1314|
|  5|    1|                   phone|             202-225-3772|
|  5|    2|                 twitter|          MikeRossUpdates|
|  6|    0|                     fax|             202-225-1314|
|  6|    1|                   phone|             202-225-3772|
|  6|    2|                 twitter|          MikeRossUpdates|
|  7|    0|                     fax|             202-225-1314|
|  7|    1|                   phone|             202-225-3772|
|  7|    2|                 twitter|          MikeRossUpdates|
|  8|    0|                     fax|             202-225-1314|
+---+-----+------------------------+-------------------------+
```

## Esempio di codice

```
# Example: Use split_rows to split up
# rows in a DynamicFrame based on value

from pyspark.context import SparkContext
from awsglue.context import GlueContext

# Create GlueContext
sc = SparkContext.getOrCreate()
glueContext = GlueContext(sc)

# Retrieve the DynamicFrame to split
frame_to_split = legislators_relationalized.select("l_root_contact_details")

# Split up rows by ID
split_rows_collection = frame_to_split.split_rows({"id": {">": 10}}, "high", "low")

# Inspect the resulting DynamicFrames
print("Inspect the DynamicFrame that contains IDs < 10")
split_rows_collection.select("low").toDF().show()
print("Inspect the DynamicFrame that contains IDs > 10")
split_rows_collection.select("high").toDF().show()
```

## Output

```
Inspect the DynamicFrame that contains IDs < 10
+---+-----+-----------------------+-----------------------+
| id|index|contact_details.val.type|contact_details.val.value|
+---+-----+-----------------------+-----------------------+
|  1|    0|                  phone|             202-225-5265|
|  1|    1|                twitter|              kathyhochul|
|  2|    0|                  phone|             202-225-3252|
|  2|    1|                twitter|            repjackyrosen|
|  3|    0|                    fax|             202-225-1314|
|  3|    1|                  phone|             202-225-3772|
|  3|    2|                twitter|          MikeRossUpdates|
|  4|    0|                    fax|             202-225-1314|
|  4|    1|                  phone|             202-225-3772|
|  4|    2|                twitter|          MikeRossUpdates|
|  5|    0|                    fax|             202-225-1314|
|  5|    1|                  phone|             202-225-3772|
|  5|    2|                twitter|          MikeRossUpdates|
|  6|    0|                    fax|             202-225-1314|
|  6|    1|                  phone|             202-225-3772|
|  6|    2|                twitter|          MikeRossUpdates|
|  7|    0|                    fax|             202-225-1314|
|  7|    1|                  phone|             202-225-3772|
|  7|    2|                twitter|          MikeRossUpdates|
|  8|    0|                    fax|             202-225-1314|
+---+-----+-----------------------+-----------------------+
only showing top 20 rows

Inspect the DynamicFrame that contains IDs > 10
+---+-----+-----------------------+-----------------------+
| id|index|contact_details.val.type|contact_details.val.value|
+---+-----+-----------------------+-----------------------+
| 11|    0|                  phone|             202-225-5476|
| 11|    1|                twitter|             RepDavidYoung|
| 12|    0|                  phone|             202-225-4035|
| 12|    1|                twitter|           RepStephMurphy|
| 13|    0|                    fax|             202-226-0774|
| 13|    1|                  phone|             202-225-6335|
| 14|    0|                    fax|             202-226-0774|
| 14|    1|                  phone|             202-225-6335|
| 15|    0|                    fax|             202-226-0774|
| 15|    1|                  phone|             202-225-6335|
| 16|    0|                    fax|             202-226-0774|
```

```
| 16|    1|                  phone|             202-225-6335|
| 17|    0|                    fax|             202-226-0774|
| 17|    1|                  phone|             202-225-6335|
| 18|    0|                    fax|             202-226-0774|
| 18|    1|                  phone|             202-225-6335|
| 19|    0|                    fax|             202-226-0774|
| 19|    1|                  phone|             202-225-6335|
| 20|    0|                    fax|             202-226-0774|
| 20|    1|                  phone|             202-225-6335|
+---+-----+-----------------------+-------------------------+
only showing top 20 rows
```

unbox

**`unbox(path, format, transformation_ctx="", info="", stageThreshold=0, totalThreshold=0, **options)`**

Esegue la conversione unboxing di un campo stringa in un oggetto `DynamicFrame` e restituisce un nuovo oggetto `DynamicFrame` che contiene gli oggetti `DynamicRecords` sottoposti a conversione unboxing.

Un `DynamicRecord` rappresenta un record logico all'interno di un `DynamicFrame`. È simile a una riga in un `DataFrame` Apache Spark, con la differenza che è autodescrittivo e può essere utilizzato per dati non conformi a uno schema fisso.

- `path`: un percorso completo al nodo stringa che desideri cancellare.
- `format`: una specifica del formato (facoltativa). Viene usata per una connessione Amazon S3 o AWS Glue che supporta più formati. Consulta Opzioni del formato dati per input e output in AWS Glue per Spark per informazioni sui formati supportati.
- `transformation_ctx`: una stringa univoca utilizzata per identificare informazioni sullo stato (opzionale).
- `info`: una stringa da associare alla segnalazione errori per questa trasformazione (opzionale).
- `stageThreshold`: il numero di errori rilevati durante questa trasformazione raggiunto il quale il processo deve interrompersi (facoltativo). L'impostazione predefinita è zero, indicando che il processo non deve terminare.
- `totalThreshold`: il numero massimo di errori rilevati durante questa trasformazione raggiunto il quale il processo deve interrompersi (facoltativo). L'impostazione predefinita è zero, indicando che il processo non deve terminare.

- `options`: una o più delle seguenti:
  - `separator`: una stringa che contiene il carattere separatore.
  - `escaper`: una stringa che contiene il carattere escape.
  - `skipFirst`: un valore Boolean che indica se saltare la prima istanza.
  - `withSchema`: una stringa contenente una rappresentazione JSON dello schema del nodo. Il formato della rappresentazione JSON di uno schema è definito dall'output di `StructType.json()`.
  - `withHeader`: un valore Boolean che indica se è inclusa un'intestazione.

Esempio: usare unbox per decomprimere un campo di stringa in un campo struct

Questo esempio di codice utilizza il metodo `unbox` per decomprimere o riformattare un campo di tipo stringa `DynamicFrame` in un campo di tipo struct.

Set di dati di esempio

L'esempio utilizza un `DynamicFrame` chiamato `mapped_with_string` con i seguenti schema e voci:

Nota il campo denominato `AddressString`. Questo è il campo che di cui l'esempio esegue l'unboxing in un campo struct.

```
root
|-- Average Total Payments: string
|-- AddressString: string
|-- Average Covered Charges: string
|-- DRG Definition: string
|-- Average Medicare Payments: string
|-- Hospital Referral Region Description: string
|-- Address: struct
|    |-- Zip.Code: string
|    |-- City: string
|    |-- Array: array
|    |    |-- element: string
|    |-- State: string
|    |-- Street: string
|-- Provider Id: string
|-- Total Discharges: string
|-- Provider Name: string
```

```
+----------------------+--------------------+-----------------------
+--------------------+------------------------+-----------------------------------
+--------------------+----------+---------------+--------------------+
|Average Total Payments|       AddressString|Average Covered Charges|      DRG
 Definition|Average Medicare Payments|Hospital Referral Region Description|
 Address|Provider Id|Total Discharges|       Provider Name|
+----------------------+--------------------+-----------------------
+--------------------+------------------------+-----------------------------------
+--------------------+----------+---------------+--------------------+
|              $5777.24|{"Street": "1108 ...|             $32963.07|039 -
 EXTRACRANIA...|                $4763.73|                        AL - Dothan|[36301,
 DOTHAN, [...|      10001|             91|SOUTHEAST ALABAMA...|
|              $5787.57|{"Street": "2505 ...|             $15131.85|039 -
 EXTRACRANIA...|                $4976.71|                    AL - Birmingham|[35957,
 BOAZ, [25...|      10005|             14|MARSHALL MEDICAL ...|
|              $5434.95|{"Street": "205 M...|             $37560.37|039 -
 EXTRACRANIA...|                $4453.79|                    AL - Birmingham|[35631,
 FLORENCE,...|      10006|             24|ELIZA COFFEE MEMO...|
|              $5417.56|{"Street": "50 ME...|             $13998.28|039 -
 EXTRACRANIA...|                $4129.16|                    AL - Birmingham|[35235,
 BIRMINGHA...|      10011|             25|   ST VINCENT'S EAST|
...
```

## Esempio di codice

```
# Example: Use unbox to unbox a string field
# into a struct in a DynamicFrame

from pyspark.context import SparkContext
from awsglue.context import GlueContext

# Create GlueContext
sc = SparkContext.getOrCreate()
glueContext = GlueContext(sc)

unboxed = mapped_with_string.unbox("AddressString", "json")
unboxed.printSchema()
unboxed.toDF().show()
```

## Output

```
root
```

```
|-- Average Total Payments: string
|-- AddressString: struct
|    |-- Street: string
|    |-- City: string
|    |-- State: string
|    |-- Zip.Code: string
|    |-- Array: array
|    |    |-- element: string
|-- Average Covered Charges: string
|-- DRG Definition: string
|-- Average Medicare Payments: string
|-- Hospital Referral Region Description: string
|-- Address: struct
|    |-- Zip.Code: string
|    |-- City: string
|    |-- Array: array
|    |    |-- element: string
|    |-- State: string
|    |-- Street: string
|-- Provider Id: string
|-- Total Discharges: string
|-- Provider Name: string

+----------------------+--------------------+-----------------------
+--------------------+------------------------+-----------------------------------
+--------------------+-----------+---------------+--------------------+
|Average Total Payments|       AddressString|Average Covered Charges|      DRG
 Definition|Average Medicare Payments|Hospital Referral Region Description|
 Address|Provider Id|Total Discharges|       Provider Name|
+----------------------+--------------------+-----------------------
+--------------------+------------------------+-----------------------------------
+--------------------+-----------+---------------+--------------------+
|              $5777.24|[1108 ROSS CLARK ...|             $32963.07|039 -
 EXTRACRANIA...|                $4763.73|                        AL - Dothan|[36301,
 DOTHAN, [...|      10001|             91|SOUTHEAST ALABAMA...|
|              $5787.57|[2505 U S HIGHWAY...|             $15131.85|039 -
 EXTRACRANIA...|                $4976.71|                    AL - Birmingham|[35957,
 BOAZ, [25...|      10005|             14|MARSHALL MEDICAL ...|
|              $5434.95|[205 MARENGO STRE...|             $37560.37|039 -
 EXTRACRANIA...|                $4453.79|                    AL - Birmingham|[35631,
 FLORENCE,...|      10006|             24|ELIZA COFFEE MEMO...|
|              $5417.56|[50 MEDICAL PARK ...|             $13998.28|039 -
 EXTRACRANIA...|                $4129.16|                    AL - Birmingham|[35235,
 BIRMINGHA...|      10011|             25|   ST VINCENT'S EAST|
```

```
|              $5658.33|[1000 FIRST STREE...|              $31633.27|039 -
 EXTRACRANIA...|                 $4851.44|                     AL - Birmingham|[35007,
 ALABASTER...|      10016|              18|SHELBY BAPTIST ME...|
|              $6653.80|[2105 EAST SOUTH ...|              $16920.79|039 -
 EXTRACRANIA...|                 $5374.14|                     AL - Montgomery|[36116,
 MONTGOMER...|      10023|              67|BAPTIST MEDICAL C...|
|              $5834.74|[2000 PEPPERELL P...|              $11977.13|039 -
 EXTRACRANIA...|                 $4761.41|                     AL - Birmingham|[36801,
 OPELIKA, ...|      10029|              51|EAST ALABAMA MEDI...|
|              $8031.12|[619 SOUTH 19TH S...|              $35841.09|039 -
 EXTRACRANIA...|                 $5858.50|                     AL - Birmingham|[35233,
 BIRMINGHA...|      10033|              32|UNIVERSITY OF ALA...|
|              $6113.38|[101 SIVLEY RD, H...|              $28523.39|039 -
 EXTRACRANIA...|                 $5228.40|                     AL - Huntsville|[35801,
 HUNTSVILL...|      10039|             135| HUNTSVILLE HOSPITAL|
|              $5541.05|[1007 GOODYEAR AV...|              $75233.38|039 -
 EXTRACRANIA...|                 $4386.94|                     AL - Birmingham|[35903,
 GADSDEN, ...|      10040|              34|GADSDEN REGIONAL ...|
|              $5461.57|[600 SOUTH THIRD ...|              $67327.92|039 -
 EXTRACRANIA...|                 $4493.57|                     AL - Birmingham|[35901,
 GADSDEN, ...|      10046|              14|RIVERVIEW REGIONA...|
|              $5356.28|[4370 WEST MAIN S...|              $39607.28|039 -
 EXTRACRANIA...|                 $4408.20|                         AL - Dothan|[36305,
 DOTHAN, [...|      10055|              45|    FLOWERS HOSPITAL|
|              $5374.65|[810 ST VINCENT'S...|              $22862.23|039 -
 EXTRACRANIA...|                 $4186.02|                     AL - Birmingham|[35205,
 BIRMINGHA...|      10056|              43|ST VINCENT'S BIRM...|
|              $5366.23|[400 EAST 10TH ST...|              $31110.85|039 -
 EXTRACRANIA...|                 $4376.23|                     AL - Birmingham|[36207,
 ANNISTON,...|      10078|              21|NORTHEAST ALABAMA...|
|              $5282.93|[1613 NORTH MCKEN...|              $25411.33|039 -
 EXTRACRANIA...|                 $4383.73|                         AL - Mobile|[36535,
 FOLEY, [1...|      10083|              15|SOUTH BALDWIN REG...|
|              $5676.55|[1201 7TH STREET ...|               $9234.51|039 -
 EXTRACRANIA...|                 $4509.11|                     AL - Huntsville|[35609,
 DECATUR, ...|      10085|              27|DECATUR GENERAL H...|
|              $5930.11|[6801 AIRPORT BOU...|              $15895.85|039 -
 EXTRACRANIA...|                 $3972.85|                         AL - Mobile|[36608,
 MOBILE, [...|      10090|              27| PROVIDENCE HOSPITAL|
|              $6192.54|[809 UNIVERSITY B...|              $19721.16|039 -
 EXTRACRANIA...|                 $5179.38|                     AL - Tuscaloosa|[35401,
 TUSCALOOS...|      10092|              31|D C H REGIONAL ME...|
```

```
|              $4968.00|[750 MORPHY AVENU...|              $10710.88|039 -
 EXTRACRANIA...|                 $3898.88|                       AL - Mobile|[36532,
 FAIRHOPE,...|      10100|              18|     THOMAS HOSPITAL|
|              $5996.00|[701 PRINCETON AV...|              $51343.75|039 -
 EXTRACRANIA...|                 $4962.45|                   AL - Birmingham|[35211,
 BIRMINGHA...|      10103|              33|BAPTIST MEDICAL C...|
+----------------------+--------------------+-----------------------
+--------------------+------------------------+-----------------------------------
+--------------------+----------+---------------+--------------------+
only showing top 20 rows
```

unione

**union(frame1, frame2, transformation_ctx = "", info = "", stageThreshold = 0, totalThreshold = 0)**

Unione due. DynamicFrames Restituzioni DynamicFrame contenenti tutti i record di entrambi gli input DynamicFrames. Questa trasformazione può restituire risultati diversi dall'unione di due DataFrames con dati equivalenti. Se hai bisogno del comportamento dell' DataFrame unione Spark, prendi in considerazione l'utilizzo `toDF` di.

- `frame1`— I primi DynamicFrame a unirsi.
- `frame2`— Seconda dopo DynamicFrame l'unione.
- `transformation_ctx`: (facoltativo) una stringa univoca utilizzata per identificare informazioni su statistiche/stato
- `info`: (facoltativo) qualsiasi stringa da associare agli errori nella trasformazione
- `stageThreshold`: (facoltativo) numero massimo di errori nella trasformazione fino a che l'elaborazione si interrompe a causa di un errore
- `totalThreshold`: (facoltativo) numero massimo di errori totali fino a che l'elaborazione si interrompe a causa di un errore.

unnest

**unnest(transformation_ctx="", info="", stageThreshold=0, totalThreshold=0)**

Annulla l'annidamento di oggetti nidificati in un `DynamicFrame` rendendoli oggetti di primo livello e restituendo un nuovo `DynamicFrame` non nidificato.

- `transformation_ctx`: una stringa univoca utilizzata per identificare informazioni sullo stato (opzionale).
- `info`: una stringa da associare alla segnalazione errori per questa trasformazione (opzionale).
- `stageThreshold`: il numero di errori rilevati durante questa trasformazione raggiunto il quale il processo deve interrompersi (facoltativo). L'impostazione predefinita è zero, indicando che il processo non deve terminare.
- `totalThreshold`: il numero massimo di errori rilevati durante questa trasformazione raggiunto il quale il processo deve interrompersi (facoltativo). L'impostazione predefinita è zero, indicando che il processo non deve terminare.

Esempio: usare unnest per trasformare i campi annidati in campi di primo livello

Questo esempio di codice utilizza il metodo `unnest` per raggruppare tutti i campi annidati di a `DynamicFrame` in campi di primo livello.

Set di dati di esempio

L'esempio utilizza un `DynamicFrame` chiamato `mapped_medicare` con il seguente schema. Nota che il campo `Address` è l'unico campo che contiene dati annidati.

```
root
|-- Average Total Payments: string
|-- Average Covered Charges: string
|-- DRG Definition: string
|-- Average Medicare Payments: string
|-- Hospital Referral Region Description: string
|-- Address: struct
|    |-- Zip.Code: string
|    |-- City: string
|    |-- Array: array
|    |    |-- element: string
|    |-- State: string
|    |-- Street: string
|-- Provider Id: string
|-- Total Discharges: string
|-- Provider Name: string
```

Esempio di codice

```
# Example: Use unnest to unnest nested
```

```
# objects in a DynamicFrame

from pyspark.context import SparkContext
from awsglue.context import GlueContext

# Create GlueContext
sc = SparkContext.getOrCreate()
glueContext = GlueContext(sc)

# Unnest all nested fields
unnested = mapped_medicare.unnest()
unnested.printSchema()
```

Output

```
root
|-- Average Total Payments: string
|-- Average Covered Charges: string
|-- DRG Definition: string
|-- Average Medicare Payments: string
|-- Hospital Referral Region Description: string
|-- Address.Zip.Code: string
|-- Address.City: string
|-- Address.Array: array
|    |-- element: string
|-- Address.State: string
|-- Address.Street: string
|-- Provider Id: string
|-- Total Discharges: string
|-- Provider Name: string
```

unnest_ddb_json

Snidifica le colonne nidificate in un `DynamicFrame` che si trovano specificamente nella struttura JSON di DynamoDB e restituisce un nuovo `DynamicFrame` non annidato. Le colonne che sono di un array di struct non verranno annidate. Si noti che si tratta di un tipo specifico di trasformazione di snidamento che si comporta in modo diverso dalla normale trasformazione di `unnest` e richiede che i dati siano già nella struttura JSON di DynamoDB. Per ulteriori informazioni, consulta DynamoDB JSON.

**`unnest_ddb_json(transformation_ctx="", info="", stageThreshold=0, totalThreshold=0)`**

- `transformation_ctx`: una stringa univoca utilizzata per identificare informazioni sullo stato (opzionale).
- `info`: una stringa da associare alla segnalazione errori per questa trasformazione (opzionale).
- `stageThreshold`: il numero di errori riscontrati durante questa trasformazione, raggiunto il quale il processo dovrebbe interrompersi (opzionale impostazione predefinita: zero, a indicare che il processo non dovrebbe interrompersi).
- `totalThreshold`: il numero di errori riscontrati fino a questa trasformazione compresa, raggiunto il quale il processo dovrebbe interrompersi (opzionale: impostazione predefinita: zero, a indicare che il processo non dovrebbe interrompersi).

Ad esempio, lo schema di lettura di un'esportazione con la struttura JSON DynamoDB potrebbe apparire come segue:

```
root
|-- Item: struct
|    |-- ColA: struct
|    |    |-- S: string
|    |-- ColB: struct
|    |    |-- S: string
|    |-- ColC: struct
|    |    |-- N: string
|    |-- ColD: struct
|    |    |-- L: array
|    |    |    |-- element: null
```

La trasformazione di `unnest_ddb_json()` lo convertirebbe in:

```
root
|-- ColA: string
|-- ColB: string
|-- ColC: string
|-- ColD: array
|    |-- element: null
```

L'esempio di codice seguente mostra come utilizzare il connettore di esportazione AWS Glue DynamoDB, richiamare un unnest JSON di DynamoDB e stampare il numero di partizioni:

```
import sys
from pyspark.context import SparkContext
from awsglue.context import GlueContext
from awsglue.job import Job
from awsglue.utils import getResolvedOptions

args = getResolvedOptions(sys.argv, ["JOB_NAME"])
glue_context= GlueContext(SparkContext.getOrCreate())
job = Job(glue_context)
job.init(args["JOB_NAME"], args)

dynamicFrame = glue_context.create_dynamic_frame.from_options(
    connection_type="dynamodb",
    connection_options={
        "dynamodb.export": "ddb",
        "dynamodb.tableArn": "<test_source>",
        "dynamodb.s3.bucket": "<bucket name>",
        "dynamodb.s3.prefix": "<bucket prefix>",
        "dynamodb.s3.bucketOwner": "<account_id>",
    }
)
unnested = dynamicFrame.unnest_ddb_json()
print(unnested.getNumPartitions())

job.commit()
```

write

### `write(connection_type, connection_options, format, format_options, accumulator_size)`

Ottiene un DataSink(oggetto) del tipo di connessione specificata da GlueContext classe di questo `DynamicFrame` e lo utilizza per formattare e scrivere i contenuti di questo `DynamicFrame`. Restituisce il nuovo `DynamicFrame` formattato e scritto come specificato.

- `connection_type`: il tipo di connessione da utilizzare. I valori validi sono `s3`, `mysql`, `postgresql`, `redshift`, `sqlserver` e `oracle`.
- `connection_options`: l'opzione di connessione da utilizzare (opzionale). Per un `connection_type` di `s3` è definito un percorso Amazon S3.

```
connection_options = {"path": "s3://aws-glue-target/temp"}
```

Per le connessioni JDBC, diverse proprietà devono essere definite. Il nome del database deve fare parte dell'URL. Può opzionalmente essere incluso nelle opzioni di connessione.

> **Warning**
>
> Si consiglia di non archiviare le password nello script. Valuta la possibilità `boto3` di utilizzarli per recuperarli da Gestione dei segreti AWS o dal AWS Glue Data Catalog.

```
connection_options = {"url": "jdbc-url/database", "user": "username",
 "password": passwordVariable,"dbtable": "table-name", "redshiftTmpDir": "s3-tempdir-path"}
```

- `format`: una specifica del formato (facoltativa). Viene utilizzato per un servizio Amazon Simple Storage Service (Amazon S3) o una connessione AWS Glue che supporta più formati. Consultare Opzioni del formato dati per input e output in AWS Glue per Spark per informazioni sui formati supportati.
- `format_options`: opzioni di formato per il formato specificato. Consultare Opzioni del formato dati per input e output in AWS Glue per Spark per informazioni sui formati supportati.
- `accumulator_size`: la dimensione accumulabile da utilizzare, in byte (facoltativa).

- errori -

- assertErrorThreshold
- errorsAsDynamicCornice
- errorsCount
- stageErrorsCount

assertErrorThreshold

`assertErrorThreshold( )`: asserzione per gli errori nelle trasformazioni che hanno creato questo oggetto `DynamicFrame`. Restituisce una `Exception` dal `DataFrame` sottostante.

errorsAsDynamicCornice

`errorsAsDynamicFrame( )`: restituisce un `DynamicFrame` che ha record di errore nidificati al suo interno.

### Esempio: utilizzare errorsAsDynamic Frame per visualizzare i record di errori

L'esempio di codice seguente mostra come utilizzare il metodo `errorsAsDynamicFrame` per visualizzare un record degli errori per un `DynamicFrame`.

#### Set di dati di esempio

L'esempio utilizza il set di dati seguente che puoi caricare in Amazon S3 come JSON. Tieni presente che il formato del secondo record non è corretto. I dati con formato non corretto generalmente interrompono l'analisi dei file quando utilizzi SparkSQL. `DynamicFrame`, tuttavia, riconosce i problemi di formato non corretto e trasforma le righe con formato non corretto in record degli errori che puoi gestire singolarmente.

```
{"id": 1, "name": "george", "surname": "washington", "height": 178}
{"id": 2, "name": "benjamin", "surname": "franklin",
{"id": 3, "name": "alexander", "surname": "hamilton", "height": 171}
{"id": 4, "name": "john", "surname": "jay", "height": 190}
```

#### Esempio di codice

```
# Example: Use errorsAsDynamicFrame to view error records.
# Replace s3://DOC-EXAMPLE-S3-BUCKET/error_data.json with your location.

from pyspark.context import SparkContext
from awsglue.context import GlueContext

# Create GlueContext
sc = SparkContext.getOrCreate()
glueContext = GlueContext(sc)

# Create errors DynamicFrame, view schema
errors = glueContext.create_dynamic_frame.from_options(
    "s3", {"paths": ["s3://DOC-EXAMPLE-S3-BUCKET/error_data.json"]}, "json"
)
print("Schema of errors DynamicFrame:")
errors.printSchema()

# Show that errors only contains valid entries from the dataset
print("errors contains only valid records from the input dataset (2 of 4 records)")
errors.toDF().show()

# View errors
```

```
print("Errors count:", str(errors.errorsCount()))
print("Errors:")
errors.errorsAsDynamicFrame().toDF().show()

# View error fields and error data
error_record = errors.errorsAsDynamicFrame().toDF().head()

error_fields = error_record["error"]
print("Error fields: ")
print(error_fields.asDict().keys())

print("\nError record data:")
for key in error_fields.asDict().keys():
    print("\n", key, ": ", str(error_fields[key]))
```

## Output

```
Schema of errors DynamicFrame:
root
|-- id: int
|-- name: string
|-- surname: string
|-- height: int

errors contains only valid records from the input dataset (2 of 4 records)
+---+------+----------+------+
| id|  name|   surname|height|
+---+------+----------+------+
|  1|george|washington|   178|
|  4|  john|       jay|   190|
+---+------+----------+------+

Errors count: 1
Errors:
+--------------------+
|               error|
+--------------------+
|[[  File "/tmp/20...|
+--------------------+

Error fields:
dict_keys(['callsite', 'msg', 'stackTrace', 'input', 'bytesread', 'source',
 'dynamicRecord'])
```

```
Error record data:

 callsite :  Row(site='  File "/tmp/2060612586885849088", line 549, in <module>\n
 sys.exit(main())\n  File "/tmp/2060612586885849088", line 523, in main\n    response
 = handler(content)\n  File "/tmp/2060612586885849088", line 197, in execute_request
\n    result = node.execute()\n  File "/tmp/2060612586885849088", line 103, in
 execute\n    exec(code, global_dict)\n  File "<stdin>", line 10, in <module>\n
 File "/opt/amazon/lib/python3.6/site-packages/awsglue/dynamicframe.py", line 625, in
 from_options\n    format_options, transformation_ctx, push_down_predicate, **kwargs)\n
  File "/opt/amazon/lib/python3.6/site-packages/awsglue/context.py", line 233, in
 create_dynamic_frame_from_options\n    source.setFormat(format, **format_options)\n',
 info='')

 msg :  error in jackson reader

 stackTrace :  com.fasterxml.jackson.core.JsonParseException: Unexpected character
 ('{' (code 123)): was expecting either valid name character (for unquoted name) or
 double-quote (for quoted) to start field name
 at [Source: com.amazonaws.services.glue.readers.BufferedStream@73492578; line: 3,
 column: 2]
 at com.fasterxml.jackson.core.JsonParser._constructError(JsonParser.java:1581)
 at
 com.fasterxml.jackson.core.base.ParserMinimalBase._reportError(ParserMinimalBase.java:533)
 at
 com.fasterxml.jackson.core.base.ParserMinimalBase._reportUnexpectedChar(ParserMinimalBase.java
 at
 com.fasterxml.jackson.core.json.UTF8StreamJsonParser._handleOddName(UTF8StreamJsonParser.java:
 at
 com.fasterxml.jackson.core.json.UTF8StreamJsonParser._parseName(UTF8StreamJsonParser.java:1650
 at
 com.fasterxml.jackson.core.json.UTF8StreamJsonParser.nextToken(UTF8StreamJsonParser.java:740)
 at com.amazonaws.services.glue.readers.JacksonReader$$anonfun$hasNextGoodToken
$1.apply(JacksonReader.scala:57)
 at com.amazonaws.services.glue.readers.JacksonReader$$anonfun$hasNextGoodToken
$1.apply(JacksonReader.scala:57)
 at scala.collection.Iterator$$anon$9.next(Iterator.scala:162)
 at scala.collection.Iterator$$anon$16.hasNext(Iterator.scala:599)
 at scala.collection.Iterator$$anon$16.hasNext(Iterator.scala:598)
 at scala.collection.Iterator$class.foreach(Iterator.scala:891)
 at scala.collection.AbstractIterator.foreach(Iterator.scala:1334)
 at com.amazonaws.services.glue.readers.JacksonReader$$anonfun
$1.apply(JacksonReader.scala:120)
```

```
 at com.amazonaws.services.glue.readers.JacksonReader$$anonfun
$1.apply(JacksonReader.scala:116)
 at
 com.amazonaws.services.glue.DynamicRecordBuilder.handleErr(DynamicRecordBuilder.scala:209)
 at
 com.amazonaws.services.glue.DynamicRecordBuilder.handleErrorWithException(DynamicRecordBuilder
 at
 com.amazonaws.services.glue.readers.JacksonReader.nextFailSafe(JacksonReader.scala:116)
 at com.amazonaws.services.glue.readers.JacksonReader.next(JacksonReader.scala:109)
 at com.amazonaws.services.glue.readers.JSONReader.next(JSONReader.scala:247)
 at
 com.amazonaws.services.glue.hadoop.TapeHadoopRecordReaderSplittable.nextKeyValue(TapeHadoopRec
 at org.apache.spark.rdd.NewHadoopRDD$$anon$1.hasNext(NewHadoopRDD.scala:230)
 at org.apache.spark.InterruptibleIterator.hasNext(InterruptibleIterator.scala:37)
 at scala.collection.Iterator$$anon$11.hasNext(Iterator.scala:409)
 at scala.collection.Iterator$$anon$11.hasNext(Iterator.scala:409)
 at scala.collection.Iterator$$anon$13.hasNext(Iterator.scala:462)
 at scala.collection.Iterator$$anon$11.hasNext(Iterator.scala:409)
 at scala.collection.Iterator$$anon$11.hasNext(Iterator.scala:409)
 at scala.collection.Iterator$$anon$13.hasNext(Iterator.scala:462)
 at scala.collection.Iterator$$anon$11.hasNext(Iterator.scala:409)
 at scala.collection.Iterator$$anon$11.hasNext(Iterator.scala:409)
 at org.apache.spark.sql.execution.SparkPlan$$anonfun$2.apply(SparkPlan.scala:255)
 at org.apache.spark.sql.execution.SparkPlan$$anonfun$2.apply(SparkPlan.scala:247)
 at org.apache.spark.rdd.RDD$$anonfun$mapPartitionsInternal$1$$anonfun$apply
$24.apply(RDD.scala:836)
 at org.apache.spark.rdd.RDD$$anonfun$mapPartitionsInternal$1$$anonfun$apply
$24.apply(RDD.scala:836)
 at org.apache.spark.rdd.MapPartitionsRDD.compute(MapPartitionsRDD.scala:52)
 at org.apache.spark.rdd.RDD.computeOrReadCheckpoint(RDD.scala:324)
 at org.apache.spark.rdd.RDD.iterator(RDD.scala:288)
 at org.apache.spark.rdd.MapPartitionsRDD.compute(MapPartitionsRDD.scala:52)
 at org.apache.spark.rdd.RDD.computeOrReadCheckpoint(RDD.scala:324)
 at org.apache.spark.rdd.RDD.iterator(RDD.scala:288)
 at org.apache.spark.scheduler.ResultTask.runTask(ResultTask.scala:90)
 at org.apache.spark.scheduler.Task.run(Task.scala:121)
 at org.apache.spark.executor.Executor$TaskRunner$$anonfun$10.apply(Executor.scala:408)
 at org.apache.spark.util.Utils$.tryWithSafeFinally(Utils.scala:1360)
 at org.apache.spark.executor.Executor$TaskRunner.run(Executor.scala:414)
 at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
 at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
 at java.lang.Thread.run(Thread.java:750)
```

```
 input :

 bytesread :  252

 source :

 dynamicRecord :  Row(id=2, name='benjamin', surname='franklin')
```

DynamicFrame Esempi completi

Gli esempi seguenti mostrano vari modi per creare e utilizzare scenari diversi da quelli di DynamicFrames base del catalogo Glue.

Caricamento da PostgreSQL con query SQL SELECT

Questo esempio mostra come caricare dati da un database PostgreSQL utilizzando una query SQL SELECT personalizzata:

```
from awsglue.context import GlueContext
from pyspark.context import SparkContext

sc = SparkContext()
glueContext = GlueContext(sc)

# Load specific data from PostgreSQL with custom query
postgres_dyf = glueContext.create_dynamic_frame.from_options(
    connection_type="postgresql",
    connection_options={
        "url": "jdbc:postgresql://your-postgres-host:5432/your-database",
        "user": "your-username",
        "password": "your-password",
        "dbtable": "(SELECT customer_id, customer_name, email FROM customers WHERE
 active = true) AS filtered_customers"
    }
)
```

Caricamento di colonne specifiche per evitare scansioni complete della tabella

Questo esempio dimostra come caricare solo colonne specifiche da una tabella di database di grandi dimensioni:

```
from awsglue.context import GlueContext
```

```
from pyspark.context import SparkContext

sc = SparkContext()
glueContext = GlueContext(sc)

# Load only specific columns from a large table
selected_columns_dyf = glueContext.create_dynamic_frame.from_options(
    connection_type="mysql",
    connection_options={
        "url": "jdbc:mysql://your-mysql-host:3306/your-database",
        "user": "your-username",
        "password": "your-password",
        "dbtable": "(SELECT order_id, customer_id FROM large_orders_table) AS
 selected_data"
    }
)

# Alternative approach using column selection in query
efficient_load_dyf = glueContext.create_dynamic_frame.from_options(
    connection_type="postgresql",
    connection_options={
        "url": "jdbc:postgresql://your-postgres-host:5432/your-database",
        "user": "your-username",
        "password": "your-password",
        "query": "SELECT product_id, product_name FROM products WHERE category =
 'electronics'"
    }
)
```

Filtraggio a livello di riga tramite connessioni JDBC

Questo esempio mostra come utilizzare il filtro a livello di riga per caricare solo righe specifiche da una tabella di database:

```
from awsglue.context import GlueContext
from pyspark.context import SparkContext

sc = SparkContext()
glueContext = GlueContext(sc)

# Load filtered rows using WHERE clause
filtered_rows_dyf = glueContext.create_dynamic_frame.from_options(
    connection_type="postgresql",
```

```
    connection_options={
        "url": "jdbc:postgresql://your-postgres-host:5432/your-database",
        "user": "your-username",
        "password": "your-password",
        "dbtable": "(SELECT * FROM transactions WHERE transaction_date >= '2024-01-01'
 AND amount > 100) AS recent_large_transactions"
    }
)

# Using partitionColumn for parallel loading with filtering
partitioned_load_dyf = glueContext.create_dynamic_frame.from_options(
    connection_type="mysql",
    connection_options={
        "url": "jdbc:mysql://your-mysql-host:3306/your-database",
        "user": "your-username",
        "password": "your-password",
        "dbtable": "sales_data",
        "partitionColumn": "sale_date",
        "lowerBound": "2024-01-01",
        "upperBound": "2024-12-31",
        "numPartitions": "10"
    }
)
```

Creazione DynamicFrame da dati Python in memoria

Questo esempio dimostra come creare elenchi, tuple o dizionari in Python: DynamicFrame

```
from awsglue.context import GlueContext
from awsglue.dynamicframe import DynamicFrame
from pyspark.context import SparkContext
from pyspark.sql import Row

sc = SparkContext()
glueContext = GlueContext(sc)

# Method 1: From list of tuples
data_tuples = [
    ("John", "Doe", 30, "Engineer"),
    ("Jane", "Smith", 25, "Designer"),
    ("Bob", "Johnson", 35, "Manager")
]

# Convert to RDD of Rows
```

```
rdd = sc.parallelize([Row(first_name=row[0], last_name=row[1], age=row[2], job=row[3])
 for row in data_tuples])
df = glueContext.spark_session.createDataFrame(rdd)
dyf_from_tuples = DynamicFrame.fromDF(df, glueContext, "employees_from_tuples")

# Method 2: From list of dictionaries
data_dicts = [
    {"product_id": 1, "product_name": "Laptop", "price": 999.99, "category":
 "Electronics"},
    {"product_id": 2, "product_name": "Book", "price": 19.99, "category": "Education"},
    {"product_id": 3, "product_name": "Chair", "price": 149.99, "category":
 "Furniture"}
]

df_from_dicts = glueContext.spark_session.createDataFrame(data_dicts)
dyf_from_dicts = DynamicFrame.fromDF(df_from_dicts, glueContext, "products_from_dicts")

# Method 3: From nested data structures
nested_data = [
    {
        "customer_id": 1,
        "customer_info": {
            "name": "Alice Brown",
            "email": "alice@example.com"
        },
        "orders": [
            {"order_id": 101, "amount": 250.00},
            {"order_id": 102, "amount": 175.50}
        ]
    }
]

df_nested = glueContext.spark_session.createDataFrame(nested_data)
dyf_nested = DynamicFrame.fromDF(df_nested, glueContext, "customers_with_orders")
```

Ottimizzazione delle prestazioni per set di dati di grandi dimensioni

Quando lavori con set di dati di grandi dimensioni, prendi in considerazione queste tecniche di ottimizzazione delle prestazioni:

```
# Use partitioning for parallel reads
large_table_dyf = glueContext.create_dynamic_frame.from_options(
    connection_type="postgresql",
```

```
    connection_options={
        "url": "jdbc:postgresql://your-postgres-host:5432/your-database",
        "user": "your-username",
        "password": "your-password",
        "dbtable": "large_table",
        "partitionColumn": "id",
        "lowerBound": "1",
        "upperBound": "1000000",
        "numPartitions": "20"
    }
)

# Use pushdown predicates to filter at source
filtered_dyf = glueContext.create_dynamic_frame.from_options(
    connection_type="mysql",
    connection_options={
        "url": "jdbc:mysql://your-mysql-host:3306/your-database",
        "user": "your-username",
        "password": "your-password",
        "dbtable": "transactions"
    },
    push_down_predicate="transaction_date >= '2024-01-01'"
)
```

errorsCount

`errorsCount( )`: restituisce il numero totale di errori in un oggetto `DynamicFrame`.

stageErrorsCount

`stageErrorsCount`: restituisce il numero di errori che si sono verificati nel processo di generazione di questo oggetto `DynamicFrame`.

DynamicFrameCollection classe

Un `DynamicFrameCollection` è un dizionario di oggetti DynamicFrame classe, in cui le chiavi sono i nomi di `DynamicFrames` e i valori sono gli oggetti `DynamicFrame`.

__init__

**`__init__(dynamic_frames, glue_ctx)`**

- `dynamic_frames`: un dizionario di oggetti DynamicFrame classe.

- `glue_ctx`: un oggetto GlueContext classe.

Chiavi

`keys( )`: restituisce un elenco di chiavi in questa raccolta, che in genere contiene i nomi dei valori `DynamicFrame` corrispondenti.

Valori

`values(key)`: restituisce un elenco di valori `DynamicFrame` in questa raccolta.

Select

**`select(key)`**

Restituisce il `DynamicFrame` corrispondente alla chiave specificata (che in genere è il nome di `DynamicFrame`).

- `key`: una chiave in `DynamicFrameCollection`, che in genere rappresenta il nome di un `DynamicFrame`.

Eseguire la mappatura

**`map(callable, transformation_ctx="")`**

Utilizza una funzione passata per creare e restituire un nuovo `DynamicFrameCollection` basato su `DynamicFrames` in questa raccolta.

- `callable`: funzione che impiega `DynamicFrame` e il contesto di trasformazione specificato come parametri e restituisce un `DynamicFrame`.
- `transformation_ctx`: un contesto di trasformazione che deve essere utilizzato dal callable (facoltativo).

Flatmap

**`flatmap(f, transformation_ctx="")`**

Utilizza una funzione passata per creare e restituire un nuovo `DynamicFrameCollection` basato su `DynamicFrames` in questa raccolta.

- `f`: funzione che impiega `DynamicFrame` come parametro e restituisce uno `DynamicFrame` o `DynamicFrameCollection`.
- `transformation_ctx`: un contesto di trasformazione che deve essere utilizzato dalla funzione (facoltativo).

DynamicFrameWriter classe

Metodi

- __init__
- from_options
- from_catalog
- from_jdbc_conf

__init__

**`__init__(glue_context)`**

- `glue_context`: il GlueContext classe da usare.

from_options

**`from_options(frame, connection_type, connection_options={}, format=None, format_options={}, transformation_ctx="")`**

Scrive un `DynamicFrame` usando la connessione e il formato specificati.

- `frame`: il `DynamicFrame` da scrivere.
- `connection_type`: il tipo di connessione. I valori validi sono `s3`, `mysql`, `postgresql`, `redshift`, `sqlserver` e `oracle`.
- `connection_options`: opzioni di connessione, come tabella di database e percorso (opzionale). Per un `connection_type` di `s3` è definito un percorso Amazon S3.

  ```
  connection_options = {"path": "s3://aws-glue-target/temp"}
  ```

  Per le connessioni JDBC, diverse proprietà devono essere definite. Il nome del database deve fare parte dell'URL. Può opzionalmente essere incluso nelle opzioni di connessione.

> **Warning**
>
> Si consiglia di non archiviare le password nello script. Valuta la possibilità `boto3` di utilizzarli per recuperarli da Gestione dei segreti AWS o dal AWS Glue Data Catalog.

```
connection_options = {"url": "jdbc-url/database", "user": "username",
 "password": passwordVariable,"dbtable": "table-name", "redshiftTmpDir": "s3-tempdir-
path"}
```

La proprietà `dbtable` è il nome della tabella JDBC. Per i archivi dati JDBC che supportano schemi all'interno di un database, specificare `schema.table-name`. Se non viene fornito alcuno schema, viene usato lo schema "pubblico" predefinito.

Per ulteriori informazioni, consulta Tipi e opzioni di connessione per ETL in AWS Glue per Spark.

- `format`: una specifica del formato (facoltativa). Viene utilizzato per un servizio Amazon Simple Storage Service (Amazon S3) o una connessione AWS Glue che supporta più formati. Consultare Opzioni del formato dati per input e output in AWS Glue per Spark per informazioni sui formati supportati.
- `format_options`: opzioni di formato per il formato specificato. Consultare Opzioni del formato dati per input e output in AWS Glue per Spark per informazioni sui formati supportati.
- `transformation_ctx`: un contesto di trasformazione da usare (opzionale).

from_catalog

**`from_catalog(frame, name_space, table_name, redshift_tmp_dir="", transformation_ctx="")`**

Scrive un `DynamicFrame` utilizzando il nome della tabella e il database del catalogo specificati.

- `frame`: il `DynamicFrame` da scrivere.
- `name_space`: il database da usare.
- `table_name`: il `table_name` da usare.
- `redshift_tmp_dir`: una directory temporanea Amazon Redshift da usare (opzionale).
- `transformation_ctx`: un contesto di trasformazione da usare (opzionale).

- `additional_options`: opzioni aggiuntive fornite a AWS Glue.

  Per scrivere su tabelle governate da Lake Formation, è possibile utilizzare queste opzioni aggiuntive:

  - `transactionId`: (stringa) l'ID transazione in cui eseguire la scrittura nella tabella Governed. Di questa transazione non può essere già stato eseguito il commit, né può essere interrotta, diversamente la scrittura non andrà a buon fine.
  - `callDeleteObjectsOnCancel` : (booleano, facoltativo) Se impostato su `true` (default), AWS Glue chiama automaticamente l'API `DeleteObjectsOnCancel` dopo che l'oggetto è stato scritto su Amazon S3. Per ulteriori informazioni, consulta la sezione DeleteObjectsOnCancel nella Guida per gli sviluppatori di AWS Lake Formation .

  Example Esempio: scrittura su una tabella governata in Lake Formation

```
txId = glueContext.start_transaction(read_only=False)
glueContext.write_dynamic_frame.from_catalog(
    frame=dyf,
    database = db,
    table_name = tbl,
    transformation_ctx = "datasource0",
    additional_options={"transactionId":txId})
...
glueContext.commit_transaction(txId)
```

from_jdbc_conf

**`from_jdbc_conf(frame, catalog_connection, connection_options={}, redshift_tmp_dir = "", transformation_ctx="")`**

Scrive un `DynamicFrame` usando le informazioni sulla connessione JDBC specificate.

- `frame`: il `DynamicFrame` da scrivere.
- `catalog_connection`: una connessione del catalogo da utilizzare.
- `connection_options`: opzioni di connessione, come tabella di database e percorso (opzionale).
- `redshift_tmp_dir`: una directory temporanea Amazon Redshift da usare (opzionale).
- `transformation_ctx`: un contesto di trasformazione da usare (opzionale).

Esempio di write_dynamic_frame

Questo esempio scrive l'output localmente utilizzando un `connection_type` di S3 con un argomento percorso POSIX in `connection_options`, che consente di scrivere su storage locale.

```
glueContext.write_dynamic_frame.from_options(\
frame = dyf_splitFields,\
connection_options = {'path': '/home/glue/GlueLocalOutput/'},\
connection_type = 's3',\
format = 'json')
```

DynamicFrameReader classe

- metodi -

- __init__
- from_rdd
- from_options
- from_catalog

__init__

**`__init__(glue_context)`**

- `glue_context`: il GlueContext classe da usare.

from_rdd

**`from_rdd(data, name, schema=None, sampleRatio=None)`**

Legge un `DynamicFrame` da un Resilient Distributed Dataset (RDD).

- `data`: il set di dati da cui leggere.
- `name`: il nome da cui leggere.
- `schema`: lo schema da leggere (opzionale).
- `sampleRatio`: il rapporto di esempio (facoltativo).

from_options

**`from_options(connection_type, connection_options={}, format=None, format_options={}, transformation_ctx="")`**

Legge un `DynamicFrame` usando la connessione e il formato specificati.

- `connection_type`: il tipo di connessione. I valori validi includono `s3`, `mysql`, `postgresql`, `redshift`, `sqlserver`, `oracle`, `dynamodb` e `snowflake`.
- `connection_options`: opzioni di connessione, come tabella di database e percorso (opzionale). Per ulteriori informazioni, consulta Tipi di connessione e opzioni per ETL in AWS Glue for Spark. Per un `connection_type` di `s3`, i percorsi Amazon S3 sono definiti in una matrice.

```
connection_options = {"paths": [ "s3://amzn-s3-demo-bucket/object_a", "s3://amzn-s3-demo-bucket/object_b"]}
```

Per le connessioni JDBC, diverse proprietà devono essere definite. Il nome del database deve fare parte dell'URL. Può opzionalmente essere incluso nelle opzioni di connessione.

> **Warning**
>
> Si consiglia di non archiviare le password nello script. Valuta la possibilità `boto3` di utilizzarli per recuperarli da Gestione dei segreti AWS o dal AWS Glue Data Catalog.

```
connection_options = {"url": "jdbc-url/database", "user": "username",
 "password": passwordVariable,"dbtable": "table-name", "redshiftTmpDir": "s3-tempdir-path"}
```

Per una connessione JDBC che esegue letture parallele, è possibile impostare l'opzione hashfield. Ad esempio:

```
connection_options = {"url": "jdbc-url/database", "user": "username",
 "password": passwordVariable,"dbtable": "table-name", "redshiftTmpDir": "s3-tempdir-path" , "hashfield": "month"}
```

Per ulteriori informazioni, consulta Lettura in parallelo dalle tabelle JDBC.

- `format`: una specifica del formato (facoltativa). Viene utilizzato per un servizio Amazon Simple Storage Service (Amazon S3) o una connessione AWS Glue che supporta più formati. Consultare Opzioni del formato dati per input e output in AWS Glue per Spark per informazioni sui formati supportati.
- `format_options`: opzioni di formato per il formato specificato. Consultare Opzioni del formato dati per input e output in AWS Glue per Spark per informazioni sui formati supportati.
- `transformation_ctx`: il contesto di trasformazione da usare (opzionale).
- `push_down_predicate`: filtra le partizioni senza dover elencare e leggere tutti i file nel set di dati. Per ulteriori informazioni, consulta Prefiltraggio con i predicati pushdown.

from_catalog

**`from_catalog(database, table_name, redshift_tmp_dir="", transformation_ctx="", push_down_predicate="", additional_options={})`**

Legge un `DynamicFrame` utilizzando il nome della tabella e il namespace del catalogo specificati.

- `database`: il database da cui leggere.
- `table_name`: il nome della tabella da cui leggere.
- `redshift_tmp_dir`: una directory Amazon Redshift temporanea da utilizzare (facoltativa se non si leggono i dati da Redshift).
- `transformation_ctx`: il contesto di trasformazione da usare (opzionale).
- `push_down_predicate`: filtra le partizioni senza dover elencare e leggere tutti i file nel set di dati. Per ulteriori informazioni, consulta Prefiltraggio con i predicati pushdown.
- `additional_options`: opzioni aggiuntive fornite a AWS Glue.
  - Per usare una connessione JDBC che esegue letture parallele, è possibile impostare l'opzione `hashfield`, `hashexpression` o `hashpartitions`. Ad esempio:

    ```
    additional_options = {"hashfield": "month"}
    ```

    Per ulteriori informazioni, consulta Lettura in parallelo dalle tabelle JDBC.
  - Per passare un'espressione di catalogo per filtrare in base alle colonne di indice, vedi l'opzione `catalogPartitionPredicate`.

    `catalogPartitionPredicate`: è possibile passare un'espressione di catalogo per filtrare in base alle colonne di indice. Questo esegue il push down del filtro sul lato server. Per ulteriori

informazioni, consultare la pagina relativa agli indici di partizionamento di AWS Glue. Tenere presente che `push_down_predicate` e `catalogPartitionPredicate` usano sintassi diverse. Il primo utilizza la sintassi standard Spark SQL e il secondo utilizza il parser JSQL.

Per ulteriori informazioni, consulta Gestione delle partizioni per l'output ETL in AWS Glue.

GlueContext classe

Racchiude l'SparkContextoggetto Apache Spark e fornisce quindi meccanismi per interagire con la piattaforma Apache Spark.

__init__

**`__init__(sparkContext)`**

- `sparkContext`: il contesto Apache Spark da usare.

Creazione in corso

- __init__
- getSource
- create_dynamic_frame_from_rdd
- create_dynamic_frame_from_catalog
- create_dynamic_frame_from_options
- create_sample_dynamic_frame_from_catalog
- create_sample_dynamic_frame_from_options
- add_ingestion_time_columns
- create_data_frame_from_catalog
- create_data_frame_from_options
- forEachBatch

getSource

**`getSource(connection_type, transformation_ctx = "", **options)`**

Crea un oggetto `DataSource` che può essere utilizzato per leggere `DynamicFrames` da fonti esterne.

- `connection_type`: il tipo di connessione da utilizzare, ad esempio Amazon Simple Storage Service (Amazon S3), Amazon Redshift e JDBC. I valori validi includono `s3`, `mysql`, `postgresql`, `redshift`, `sqlserver`, `oracle` e `dynamodb`.
- `transformation_ctx`: il contesto di trasformazione da usare (opzionale).
- `options`: una raccolta di coppie nome/valore opzionali. Per ulteriori informazioni, consulta Tipi e opzioni di connessione per ETL in AWS Glue per Spark.

Di seguito è riportato un esempio dell'utilizzo di `getSource`:

```
>>> data_source = context.getSource("file", paths=["/in/path"])
>>> data_source.setFormat("json")
>>> myFrame = data_source.getFrame()
```

create_dynamic_frame_from_rdd

**`create_dynamic_frame_from_rdd(data, name, schema=None, sample_ratio=None, transformation_ctx="")`**

Restituisce un `DynamicFrame` che viene creato da un Apache Spark Resilient Distributed Dataset (RDD).

- `data`: l'origine dati da usare.
- `name`: il nome dei dati da usare.
- `schema`: lo schema da usare (opzionale).
- `sample_ratio`: il rapporto di esempio da usare (opzionale).
- `transformation_ctx`: il contesto di trasformazione da usare (opzionale).

create_dynamic_frame_from_catalog

**`create_dynamic_frame_from_catalog(database, table_name, redshift_tmp_dir, transformation_ctx = "", push_down_predicate= "", additional_options = {}, catalog_id = None)`**

Restituisce un `DynamicFrame` creato utilizzando un database del catalogo dati e un nome della tabella. Quando si utilizza questo metodo, si forniscono `format_options` tramite tabella le proprietà della tabella AWS Glue Data Catalog specificata e altre opzioni tramite l'`additional_options`argomento.

- `Database`: il database da cui leggere.
- `table_name`: il nome della tabella da cui leggere.
- `redshift_tmp_dir`: una directory temporanea Amazon Redshift da usare (opzionale).
- `transformation_ctx`: il contesto di trasformazione da usare (opzionale).
- `push_down_predicate`: filtra le partizioni senza dover elencare e leggere tutti i file nel set di dati. Per le fonti e le limitazioni supportate, consulta Ottimizzazione delle letture con pushdown in AWS Glue ETL. Per ulteriori informazioni, consulta Prefiltraggio con i predicati pushdown.
- `additional_options`: una raccolta di coppie nome/valore opzionali. Le opzioni possibili includono quelle elencate in Tipi e opzioni di connessione per ETL in AWS Glue per Spark ad eccezione di `endpointUrl`, `streamName`, `bootstrap.servers`, `security.protocol`, `topicName`, `classification` e `delimiter`. Un'altra opzione supportata è `catalogPartitionPredicate`:

  `catalogPartitionPredicate`: è possibile passare un'espressione di catalogo per filtrare in base alle colonne di indice. Questo esegue il push down del filtro sul lato server. Per ulteriori informazioni, consultare la pagina relativa agli indici di partizionamento di AWS Glue. Tenere presente che `push_down_predicate` e `catalogPartitionPredicate` usano sintassi diverse. Il primo utilizza la sintassi standard Spark SQL e il secondo utilizza il parser JSQL.
- `catalog_id`: l'ID catalogo (ID account) del catalogo dati a cui si accede. Se Nessuno, viene utilizzato l'ID account predefinito del chiamante.

create_dynamic_frame_from_options

**`create_dynamic_frame_from_options(connection_type, connection_options={}, format=None, format_options={}, transformation_ctx = "")`**

Restituisce un `DynamicFrame` creato con la connessione e il formato specificati.

- `connection_type`: il tipo di connessione, come Amazon S3, Amazon Redshift e JDBC. I valori validi includono `s3`, `mysql`, `postgresql`, `redshift`, `sqlserver`, `oracle` e `dynamodb`.
- `connection_options`: opzioni di connessione, come tabella di database e percorsi (opzionale). Per un oggetto `connection_type` di `s3`, viene definito un elenco di percorsi Amazon S3.

```
connection_options = {"paths": ["s3://aws-glue-target/temp"]}
```

Per le connessioni JDBC, diverse proprietà devono essere definite. Il nome del database deve fare parte dell'URL. Può opzionalmente essere incluso nelle opzioni di connessione.

> **Warning**
>
> Si consiglia di non archiviare le password nello script. Valuta la possibilità `boto3` di utilizzarli per recuperarli da Gestione dei segreti AWS o dal AWS Glue Data Catalog.

```
connection_options = {"url": "jdbc-url/database", "user": "username",
 "password": passwordVariable,"dbtable": "table-name", "redshiftTmpDir": "s3-tempdir-path"}
```

La proprietà `dbtable` è il nome della tabella JDBC. Per i archivi dati JDBC che supportano schemi all'interno di un database, specificare `schema.table-name`. Se non viene fornito alcuno schema, viene usato lo schema "pubblico" predefinito.

Per ulteriori informazioni, consultare Tipi e opzioni di connessione per ETL in AWS Glue per Spark.

- `format`: una specifica di formato. Viene usata per una connessione Amazon S3 o AWS Glue che supporta più formati. Consultare Opzioni del formato dati per input e output in AWS Glue per Spark per informazioni sui formati supportati.
- `format_options`: opzioni di formato per il formato specificato. Consultare Opzioni del formato dati per input e output in AWS Glue per Spark per informazioni sui formati supportati.
- `transformation_ctx`: il contesto di trasformazione da usare (opzionale).
- `push_down_predicate`: filtra le partizioni senza dover elencare e leggere tutti i file nel set di dati. Per le fonti e le limitazioni supportate, consulta Ottimizzazione delle letture con pushdown in AWS Glue ETL. Per ulteriori informazioni, consulta Prefiltraggio con i predicati pushdown.

create_sample_dynamic_frame_from_catalog

**`create_sample_dynamic_frame_from_catalog(database, table_name, num, redshift_tmp_dir, transformation_ctx = "", push_down_predicate= "", additional_options = {}, sample_options = {}, catalog_id = None)`**

Restituisce un `DynamicFrame` di esempio creato utilizzando un database del catalogo dati e un nome della tabella. La `DynamicFrame` contiene solo i primi `num` registri da un'origine dati.

- `database`: il database da cui leggere.
- `table_name`: il nome della tabella da cui leggere.
- `num`: il numero massimo di registri nel frame dinamico di esempio restituito.
- `redshift_tmp_dir`: una directory temporanea Amazon Redshift da usare (opzionale).
- `transformation_ctx`: il contesto di trasformazione da usare (opzionale).
- `push_down_predicate`: filtra le partizioni senza dover elencare e leggere tutti i file nel set di dati. Per ulteriori informazioni, consultare Prefiltraggio con i predicati pushdown.
- `additional_options`: una raccolta di coppie nome/valore opzionali. Le opzioni possibili includono quelle elencate in Tipi e opzioni di connessione per ETL in AWS Glue per Spark ad eccezione di `endpointUrl`, `streamName`, `bootstrap.servers`, `security.protocol`, `topicName`, `classification` e `delimiter`.
- `sample_options`: parametri per controllare il comportamento di campionamento (opzionale). Parametri attuali disponibili per le origini Amazon S3:
  - `maxSamplePartitions`: il numero massimo di partizioni che il campionamento leggerà. Il valore predefinito è 10
  - `maxSampleFilesPerPartition`: il numero massimo di file che il campionamento leggerà in una partizione. Il valore predefinito è 10.

    Questi parametri aiutano a ridurre il tempo impiegato dall'elenco dei file. Ad esempio, supponiamo che il set di dati contenga 1000 partizioni e ogni partizione contenga 10 file. Se è impostato `maxSamplePartitions` = 10 e `maxSampleFilesPerPartition` = 10, invece di elencare tutti i 10.000 file, il campionamento elencherà e leggerà solo le prime 10 partizioni con i primi 10 file in ognuna di esse: 10*10 = 100 file in totale.
- `catalog_id`: l'ID catalogo del catalogo dati a cui si accede (l'ID account del catalogo dati). Impostato su `None` per default. L'impostazione predefinita di `None` è l'ID catalogo dell'account chiamante nel servizio.

create_sample_dynamic_frame_from_options

**`create_sample_dynamic_frame_from_options(connection_type, connection_options={}, num, sample_options={}, format=None, format_options={}, transformation_ctx = "")`**

Restituisce un `DynamicFrame` di esempio creato con la connessione e il formato specificati. La `DynamicFrame` contiene solo i primi `num` registri da un'origine dati.

- `connection_type`: il tipo di connessione, come Amazon S3, Amazon Redshift e JDBC. I valori validi includono `s3`, `mysql`, `postgresql`, `redshift`, `sqlserver`, `oracle` e `dynamodb`.
- `connection_options`: opzioni di connessione, come tabella di database e percorsi (opzionale). Per ulteriori informazioni, consultare Tipi e opzioni di connessione per ETL in AWS Glue per Spark.
- `num`: il numero massimo di registri nel frame dinamico di esempio restituito.
- `sample_options`: parametri per controllare il comportamento di campionamento (opzionale). Parametri attuali disponibili per le origini Amazon S3:
  - `maxSamplePartitions`: il numero massimo di partizioni che il campionamento leggerà. Il valore predefinito è 10
  - `maxSampleFilesPerPartition`: il numero massimo di file che il campionamento leggerà in una partizione. Il valore predefinito è 10.

    Questi parametri aiutano a ridurre il tempo impiegato dall'elenco dei file. Ad esempio, supponiamo che il set di dati contenga 1000 partizioni e ogni partizione contenga 10 file. Se è impostato `maxSamplePartitions` = 10 e `maxSampleFilesPerPartition` = 10, invece di elencare tutti i 10.000 file, il campionamento elencherà e leggerà solo le prime 10 partizioni con i primi 10 file in ognuna di esse: 10*10 = 100 file in totale.
- `format`: una specifica di formato. Viene usata per una connessione Amazon S3 o AWS Glue che supporta più formati. Consultare Opzioni del formato dati per input e output in AWS Glue per Spark per informazioni sui formati supportati.
- `format_options`: opzioni di formato per il formato specificato. Consultare Opzioni del formato dati per input e output in AWS Glue per Spark per informazioni sui formati supportati.
- `transformation_ctx`: il contesto di trasformazione da usare (opzionale).
- `push_down_predicate`: filtra le partizioni senza dover elencare e leggere tutti i file nel set di dati. Per ulteriori informazioni, consulta Prefiltraggio con i predicati pushdown.

add_ingestion_time_columns

**`add_ingestion_time_columns(dataFrame, timeGranularity = "")`**

Aggiunge colonne del tempo di importazione dati come `ingest_year`, `ingest_month`, `ingest_day`, `ingest_hour`, `ingest_minute` al `DataFrame` di input. Questa funzione viene generata automaticamente nello script generato da AWS Glue quando si specifica una tabella del catalogo dati con Amazon S3 come destinazione. Questa funzione aggiorna automaticamente la partizione con le colonne del tempo di importazione dati nella tabella di output. Ciò consente ai dati

di output di venire partizionati automaticamente nel tempo di importazione dati senza necessitare di colonne di tempo di inserimento esplicite nei dati di input.

- `dataFrame`: il `dataFrame` al quale aggiungere le colonne del tempo di importazione dati.
- `timeGranularity`: la granularità delle colonne temporali. I valori validi sono "`day`", "`hour`" e "`minute`". Ad esempio, se "`hour`" viene passato alla funzione, il `dataFrame` originale avrà "`ingest_year`", "`ingest_month`", "`ingest_day`" e "`ingest_hour`" colonne temporali aggiunte.

Restituisce il frame di dati dopo l'aggiunta di colonne di granularità di tempo.

Esempio:

```
dynamic_frame = DynamicFrame.fromDF(glueContext.add_ingestion_time_columns(dataFrame,
 "hour"))
```

create_data_frame_from_catalog

**`create_data_frame_from_catalog(database, table_name, transformation_ctx = "", additional_options = {})`**

Restituisce un `DataFrame` creato utilizzando le informazioni da una tabella del catalogo dati.

- `database`: il database del catalogo dati da cui leggere.
- `table_name`: il nome della tabella de catalogo dati da cui leggere.
- `transformation_ctx`: il contesto di trasformazione da usare (opzionale).
- `additional_options`: una raccolta di coppie nome/valore opzionali. Le opzioni possibili includono quelle elencate in Tipi e opzioni di connessione per ETL in AWS Glue per Spark per le origini di streaming, ad esempio `startingPosition`, `maxFetchTimeInMs`, e `startingOffsets`.
  - `useSparkDataSource`— Se impostato su true, forza AWS Glue a utilizzare l'API Spark Data Source nativa per leggere la tabella. L'API Spark Data Source supporta i seguenti formati: AVRO, binario, CSV, JSON, ORC, Parquet e testo. In una tabella del catalogo dati, il formato può essere specificato utilizzando la proprietà `classification`. Per ulteriori informazioni sull'API Spark Data Source, consulta la documentazione ufficiale di Apache Spark.

    L'uso di `create_data_frame_from_catalog` con `useSparkDataSource` offre i seguenti vantaggi:

- Restituisce direttamente un `DataFrame` e fornisce un'alternativa a `create_dynamic_frame.from_catalog().toDF()`.
- Supporta il controllo delle autorizzazioni AWS Lake Formation a livello di tabella per i formati nativi.
- Supporta la lettura dei formati Data Lake senza il controllo delle autorizzazioni a AWS Lake Formation livello di tabella. Per ulteriori informazioni, consulta Utilizzo di framework di data lake con AWS Glue processi ETL.

Quando abilitiuseSparkDataSource, puoi anche aggiungere una qualsiasi delle opzioni di Spark Data Source secondo necessità. `additional_options` AWS Glue trasmette queste opzioni direttamente al lettore Spark.

- `useCatalogSchema`— Se impostato su true, AWS Glue applica lo schema Data Catalog al risultato`DataFrame`. Altrimenti, il lettore deduce lo schema dai dati. Se si abilita l'opzione `useCatalogSchema`, è necessario impostare anche `useSparkDataSource` su true.

Limitazioni

Quando si utilizza l'opzione `useSparkDataSource` considerare le seguenti limitazioni:

- Quando lo usiuseSparkDataSource, AWS Glue ne crea una nuova `DataFrame` in una sessione Spark separata, diversa dalla sessione Spark originale.
- Il filtro DataFrame delle partizioni Spark non funziona con le seguenti funzionalità di AWS Glue.
  - Segnalibri di processo
  - Esclusione delle classi di archiviazione di Amazon S3
  - Predicati di partizione di catalogo

  Per utilizzare il filtraggio delle partizioni con queste funzionalità, è possibile utilizzare il predicato pushdown AWS Glue. Per ulteriori informazioni, consulta Prefiltraggio con i predicati pushdown. Il filtraggio sulle colonne non partizionate non viene modificato.

  Lo script di esempio seguente dimostra il modo errato di eseguire il filtraggio delle partizioni con l'opzione `excludeStorageClasses`.

```
// Incorrect partition filtering using Spark filter with excludeStorageClasses
read_df = glueContext.create_data_frame.from_catalog(
    database=database_name,
    table_name=table_name,
```

```
    additional_options = {
      "useSparkDataSource": True,
      "excludeStorageClasses" : ["GLACIER", "DEEP_ARCHIVE"]
    }
)

//  Suppose year and month are partition keys.
//  Filtering on year and month won't work, the filtered_df will still
//  contain data with other year/month values.
filtered_df = read_df.filter("year == '2017 and month == '04' and 'state == 'CA'")
```

Lo script di esempio seguente dimostra il modo corretto di utilizzare un predicato pushdown in modo da eseguire il filtraggio delle partizioni con l'opzione `excludeStorageClasses`.

```
// Correct partition filtering using the AWS Glue pushdown predicate
// with excludeStorageClasses
read_df = glueContext.create_data_frame.from_catalog(
    database=database_name,
    table_name=table_name,
    //  Use AWS Glue pushdown predicate to perform partition filtering
    push_down_predicate = "(year=='2017' and month=='04')"
    additional_options = {
      "useSparkDataSource": True,
      "excludeStorageClasses" : ["GLACIER", "DEEP_ARCHIVE"]
    }
)

//  Use Spark filter only on non-partitioned columns
filtered_df = read_df.filter("state == 'CA'")
```

Esempio: creazione di una tabella CSV utilizzando il lettore di origini dati Spark

```
//  Read a CSV table with '\t' as separator
read_df = glueContext.create_data_frame.from_catalog(
    database=<database_name>,
    table_name=<table_name>,
    additional_options = {"useSparkDataSource": True,  "sep": '\t'}
)
```

create_data_frame_from_options

**`create_data_frame_from_options(connection_type, connection_options={}, format=None, format_options={}, transformation_ctx = "")`**

Questa API è obsoleta. Utilizza invece le API `getSource()`. Restituisce un `DataFrame` creato con la connessione e il formato specificati. Utilizza questa funzione solo con origini di streaming AWS Glue.

- `connection_type`: il tipo di connessione streaming. I valori validi includono `kinesis` e `kafka`.
- `connection_options`: opzioni di connessione, che sono diverse per Kinesis e Kafka. È possibile trovare l'elenco di tutte le opzioni di connessione per ogni origine dati di streaming all'indirizzo Tipi e opzioni di connessione per ETL in AWS Glue per Spark. Di seguito vengono illustrate le differenze delle opzioni di connessione di streaming:
  - Le origini di streaming di Kinesis richiedono `streamARN`, `startingPosition`, `inferSchema` e `classification`.
  - Le origini di streaming di Kafka richiedono `connectionName`, `topicName`, `startingOffsets`, `inferSchema` e `classification`.
- `format`: una specifica di formato. Viene usata per una connessione Amazon S3 o AWS Glue che supporta più formati. Per ulteriori informazioni sui formati supportati, consulta Opzioni del formato dati per input e output in AWS Glue per Spark.
- `format_options`: opzioni di formato per il formato specificato. Per ulteriori informazioni sulle opzioni di formato supportate, consulta Opzioni del formato dati per input e output in AWS Glue per Spark.
- `transformation_ctx`: il contesto di trasformazione da usare (opzionale).

Esempio per l'origine di streaming Amazon Kinesis:

```
kinesis_options =
   { "streamARN": "arn:aws:kinesis:us-east-2:777788889999:stream/fromOptionsStream",
     "startingPosition": "TRIM_HORIZON",
     "inferSchema": "true",
     "classification": "json"
   }
data_frame_datasource0 =
 glueContext.create_data_frame.from_options(connection_type="kinesis",
 connection_options=kinesis_options)
```

Esempio per l'origine di streaming Kafka:

```
kafka_options =
    { "connectionName": "ConfluentKafka",
      "topicName": "kafka-auth-topic",
      "startingOffsets": "earliest",
      "inferSchema": "true",
      "classification": "json"
    }
data_frame_datasource0 =
 glueContext.create_data_frame.from_options(connection_type="kafka",
 connection_options=kafka_options)
```

forEachBatch

### `forEachBatch(frame, batch_function, options)`

Applica la `batch_function` passata a ogni micro batch che viene letto dall'origine di streaming.

- `frame`— Il file DataFrame contenente il microbatch corrente.
- `batch_function`: una funzione che verrà applicata per ogni micro batch.
- `options`: una raccolta di coppie chiave-valore che contiene informazioni su come elaborare micro batch. Sono richieste le seguenti opzioni:
  - `windowSize`: la quantità di tempo da dedicare all'elaborazione di ciascun batch.
  - `checkpointLocation`: la posizione in cui sono archiviati i checkpoint per il processo ETL di streaming.
  - `batchMaxRetries`: numero massimo di tentativi per riprovare il processo se il batch ha esito negativo. Il valore predefinito è 3. Questa opzione è configurabile solo per Glue versione 2.0 e successive.

Esempio:

```
glueContext.forEachBatch(
    frame = data_frame_datasource0,
    batch_function = processBatch,
    options = {
        "windowSize": "100 seconds",
        "checkpointLocation": "s3://kafka-auth-dataplane/confluent-test/output/
checkpoint/"
```

```
    }
)

def processBatch(data_frame, batchId):
    if (data_frame.count() > 0):
        datasource0 = DynamicFrame.fromDF(
          glueContext.add_ingestion_time_columns(data_frame, "hour"),
          glueContext, "from_data_frame"
        )
        additionalOptions_datasink1 = {"enableUpdateCatalog": True}
        additionalOptions_datasink1["partitionKeys"] = ["ingest_yr", "ingest_mo",
 "ingest_day"]
        datasink1 = glueContext.write_dynamic_frame.from_catalog(
          frame = datasource0,
          database = "tempdb",
          table_name = "kafka-auth-table-output",
          transformation_ctx = "datasink1",
          additional_options = additionalOptions_datasink1
        )
```

Utilizzo di set di dati in Amazon S3

- purge_table
- purge_s3_path
- transition_table
- transition_s3_path

purge_table

**purge_table(catalog_id=None, database="", table_name="", options={}, transformation_ctx="")**

Elimina i file da Amazon S3 per il database e la tabella del catalogo specificati. Se tutti i file in una partizione vengono eliminati, anche la partizione viene eliminata dal catalogo. L'operazione purge_table sulle tabelle registrate con Lake Formation non è supportata.

Per poter recuperare gli oggetti eliminati, è possibile abilitare la funzione di controllo delle versioni degli oggetti nel bucket Amazon S3. Quando un oggetto viene eliminato da un bucket per il quale non è abilitata la funzione Versioni multiple degli oggetti, l'oggetto non può essere recuperato. Per ulteriori informazioni su come recuperare gli oggetti eliminati in un bucket abilitato per le versioni, consultare

In che modo può essere recuperato un oggetto Amazon S3 che è stato eliminato? nel Portale del sapere di Supporto AWS .

- `catalog_id`: l'ID catalogo del catalogo dati a cui si accede (l'ID account del catalogo dati). Impostato su `None` per default. L'impostazione predefinita di `None` è l'ID catalogo dell'account chiamante nel servizio.
- `database`: il database da usare.
- `table_name`: il nome della tabella da usare.
- `options`: opzioni per filtrare i file da eliminare e per la generazione di file manifesto.
  - `retentionPeriod`: specifica un periodo in numero di ore per la conservazione dei file. I file più recenti del periodo di conservazione vengono mantenuti. Impostato su 168 ore (7 giorni) per impostazione predefinita.
  - `partitionPredicate`: le partizioni che soddisfano questo predicato vengono eliminate. I file all'interno del periodo di conservazione in queste partizioni non vengono eliminati. Impostato su `""`: vuoto per impostazione predefinita.
  - `excludeStorageClasses`: i file con classe di storage nel `excludeStorageClasses` non vengono eliminati. L'impostazione di default è `Set()`: un set vuoto.
  - `manifestFilePath`: un percorso opzionale per la generazione di file manifesto. Tutti i file che sono stati eliminati correttamente vengono registrati in `Success.csv` e quelli che non sono riusciti in `Failed.csv`
- `transformation_ctx`: il contesto di trasformazione da usare (opzionale). Utilizzato nel percorso del file manifest.

Example

```
glueContext.purge_table("database", "table", {"partitionPredicate": "(month=='march')",
 "retentionPeriod": 1, "excludeStorageClasses": ["STANDARD_IA"], "manifestFilePath":
 "s3://bucketmanifest/"})
```

purge_s3_path

**`purge_s3_path(s3_path, options={}, transformation_ctx="")`**

Elimina i file dal percorso Amazon S3 specificato in modo ricorsivo.

Per poter recuperare gli oggetti eliminati, puoi abilitare la funzione di controllo delle versioni degli oggetti nel bucket Amazon S3. Quando un oggetto viene eliminato da un bucket per il quale non è

abilitata la funzione di controllo delle versioni degli oggetti, l'oggetto non può essere recuperato. Per ulteriori informazioni su come recuperare oggetti eliminati in un bucket con il controllo delle versioni, vedi Come posso recuperare un oggetto Amazon S3 che è stato eliminato? nel Knowledge Center. Supporto

- `s3_path`: il percorso in Amazon S3 dei file da eliminare nel formato `s3://<bucket>/<prefix>/`
- `options`: opzioni per filtrare i file da eliminare e per la generazione di file manifesto.
  - `retentionPeriod`: specifica un periodo in numero di ore per la conservazione dei file. I file più recenti del periodo di conservazione vengono mantenuti. Impostato su 168 ore (7 giorni) per impostazione predefinita.
  - `excludeStorageClasses`: i file con classe di storage nel `excludeStorageClasses` non vengono eliminati. L'impostazione di default è `Set()`: un set vuoto.
  - `manifestFilePath`: un percorso opzionale per la generazione di file manifesto. Tutti i file che sono stati eliminati correttamente vengono registrati in `Success.csv` e quelli che non sono riusciti in `Failed.csv`
- `transformation_ctx`: il contesto di trasformazione da usare (opzionale). Utilizzato nel percorso del file manifest.

Example

```
glueContext.purge_s3_path("s3://bucket/path/", {"retentionPeriod": 1,
 "excludeStorageClasses": ["STANDARD_IA"], "manifestFilePath": "s3://bucketmanifest/"})
```

transition_table

**`transition_table(database, table_name, transition_to, options={}, transformation_ctx="", catalog_id=None)`**

Esegue la transizione della classe di storage dei file archiviati su Amazon S3 per il database e la tabella del catalogo specificati.

È possibile eseguire la transizione tra due classi di archiviazione qualsiasi. Per le classi di archiviazione `GLACIER` e `DEEP_ARCHIVE`, è possibile passare a queste classi. Tuttavia, si dovrebbe utilizzare un `S3 RESTORE` per eseguire la transizione dalle classi di archiviazione `GLACIER` a `DEEP_ARCHIVE`.

Se si eseguono processi ETL AWS Glue che leggono file o partizioni da Amazon S3, è possibile escludere alcuni tipi di classe di archiviazione Amazon S3. Per ulteriori informazioni, consultare Esclusione delle classi di archiviazione Amazon S3.

- `database`: il database da usare.
- `table_name`: il nome della tabella da usare.
- `transition_to`: la classe di storage Amazon S3 in cui eseguire la transizione.
- `options`: opzioni per filtrare i file da eliminare e per la generazione di file manifesto.
  - `retentionPeriod`: specifica un periodo in numero di ore per la conservazione dei file. I file più recenti del periodo di conservazione vengono mantenuti. Impostato su 168 ore (7 giorni) per impostazione predefinita.
  - `partitionPredicate`: le partizioni che soddisfano questo predicato vengono trasferite. I file all'interno del periodo di conservazione in queste partizioni non vengono passati. Impostato su `""`: vuoto per impostazione predefinita.
  - `excludeStorageClasses`: i file con classe di storage nel set `excludeStorageClasses` non vengono passati. L'impostazione di default è `Set()`: un set vuoto.
  - `manifestFilePath`: un percorso facoltativo per la generazione di file manifesto. Tutti i file che sono stati passati correttamente vengono registrati in `Success.csv` e quelli che non sono riusciti in `Failed.csv`
  - `accountId`: l'ID account Amazon Web Services per eseguire la trasformazione di transizione. Obbligatorio per questa trasformazione.
  - `roleArn`— Il AWS ruolo per eseguire la trasformazione di transizione. Obbligatorio per questa trasformazione.
- `transformation_ctx`: il contesto di trasformazione da usare (opzionale). Utilizzato nel percorso del file manifest.
- `catalog_id`: l'ID catalogo del catalogo dati a cui si accede (l'ID account del catalogo dati). Impostato su `None` per default. L'impostazione predefinita di `None` è l'ID catalogo dell'account chiamante nel servizio.

Example

```
glueContext.transition_table("database", "table", "STANDARD_IA", {"retentionPeriod": 1,
 "excludeStorageClasses": ["STANDARD_IA"], "manifestFilePath": "s3://bucketmanifest/",
```

```
 "accountId": "12345678901", "roleArn": "arn:aws:iam::123456789012:user/example-
username"})
```

transition_s3_path

### `transition_s3_path(s3_path, transition_to, options={}, transformation_ctx="")`

Esegue la transizione della classe di storage nel percorso Amazon S3 specificato in modo ricorsivo.

È possibile eseguire la transizione tra due classi di archiviazione qualsiasi. Per le classi di archiviazione `GLACIER` e `DEEP_ARCHIVE`, è possibile passare a queste classi. Tuttavia, si dovrebbe utilizzare un `S3 RESTORE` per eseguire la transizione dalle classi di archiviazione `GLACIER` a `DEEP_ARCHIVE`.

Se si eseguono processi ETL AWS Glue che leggono file o partizioni da Amazon S3, è possibile escludere alcuni tipi di classe di archiviazione Amazon S3. Per ulteriori informazioni, consultare Esclusione delle classi di archiviazione Amazon S3.

- `s3_path`: il percorso in Amazon S3 dei file da convertire nel formato `s3://<bucket>/<prefix>/`
- `transition_to`: la classe di storage Amazon S3 in cui eseguire la transizione.
- `options`: opzioni per filtrare i file da eliminare e per la generazione di file manifesto.
  - `retentionPeriod`: specifica un periodo in numero di ore per la conservazione dei file. I file più recenti del periodo di conservazione vengono mantenuti. Impostato su 168 ore (7 giorni) per impostazione predefinita.
  - `partitionPredicate`: le partizioni che soddisfano questo predicato vengono trasferite. I file all'interno del periodo di conservazione in queste partizioni non vengono passati. Impostato su `""`: vuoto per impostazione predefinita.
  - `excludeStorageClasses`: i file con classe di storage nel set `excludeStorageClasses` non vengono passati. L'impostazione di default è `Set()`: un set vuoto.
  - `manifestFilePath`: un percorso facoltativo per la generazione di file manifesto. Tutti i file che sono stati passati correttamente vengono registrati in `Success.csv` e quelli che non sono riusciti in `Failed.csv`
  - `accountId`: l'ID account Amazon Web Services per eseguire la trasformazione di transizione. Obbligatorio per questa trasformazione.

  - `roleArn`— Il AWS ruolo per eseguire la trasformazione di transizione. Obbligatorio per questa trasformazione.
- `transformation_ctx`: il contesto di trasformazione da usare (opzionale). Utilizzato nel percorso del file manifest.

Example

```
glueContext.transition_s3_path("s3://bucket/prefix/", "STANDARD_IA",
 {"retentionPeriod": 1, "excludeStorageClasses": ["STANDARD_IA"],
 "manifestFilePath": "s3://bucketmanifest/", "accountId": "12345678901", "roleArn":
 "arn:aws:iam::123456789012:user/example-username"})
```

Estrazione in corso

- extract_jdbc_conf

extract_jdbc_conf

**`extract_jdbc_conf(connection_name, catalog_id = None)`**

Restituisce un `dict` con chiavi con le proprietà di configurazione dall'oggetto di connessione AWS Glue nel catalogo dati.

- `user`: il nome utente del database.
- `password`: la password del database.
- `vendor`: specifica un fornitore (`mysql`, `postgresql`, `oracle`, `sqlserver` e così via).
- `enforceSSL`: una stringa booleana che indica se è necessaria una connessione sicura.
- `customJDBCCert`: utilizza un certificato client specifico dal percorso Amazon S3 indicato.
- `skipCustomJDBCCertValidation`: una stringa booleana che indica se `customJDBCCert` deve essere convalidato da una CA.
- `customJDBCCertString`: informazioni aggiuntive sul certificato personalizzato, specifico per il tipo di driver.
- `url` (obsoleto): l'URL JDBC con solo protocollo, server e porta.
- `fullUrl`: l'URL JDBC immesso al momento della creazione della connessione (disponibile in AWS Glueversione 3.0 o successive).

Esempio di recupero delle configurazioni JDBC:

```
jdbc_conf = glueContext.extract_jdbc_conf(connection_name="your_glue_connection_name")
print(jdbc_conf)
>>> {'enforceSSL': 'false', 'skipCustomJDBCCertValidation': 'false', 'url':
 'jdbc:mysql://myserver:3306', 'fullUrl': 'jdbc:mysql://myserver:3306/mydb',
 'customJDBCCertString': '', 'user': 'admin', 'customJDBCCert': '', 'password': '1234',
 'vendor': 'mysql'}
```

Transazioni

- start_transaction
- commit_transaction
- cancel_transaction

start_transaction

**`start_transaction(read_only)`**

Avvia una nuova transazione. Chiama internamente l'API Lake Formation startTransaction.

- `read_only`: (booleano) indica se questa transazione debba essere di sola lettura o lettura e scrittura. Le scritture effettuate utilizzando un ID transazione di sola lettura verranno rifiutate. Il commit delle transazioni di sola lettura non deve essere eseguito.

Restituisce l'ID transazione.

commit_transaction

**`commit_transaction(transaction_id, wait_for_commit = True)`**

Tenta di eseguire il commit della transazione specificata. `commit_transaction` può rispondere prima che la transazione abbia terminato il commit. Chiama internamente l'API Lake Formation commitTransaction.

- `transaction_id` : (stringa) la transazione di cui eseguire il commit.
- `wait_for_commit`: (booleano) determina se il `commit_transaction` restituisce immediatamente. Il valore di default è true. Se false, `commit_transaction` effettua il polling e aspetta che sia stato eseguito il commit della transazione. Il tempo di attesa è limitato a 1 minuto utilizzando il backoff esponenziale con un massimo di 6 tentativi.

Restituisce un valore booleano per indicare se il commit sia stato eseguito o meno.

cancel_transaction

**`cancel_transaction(transaction_id)`**

Tenta di annullare la transazione specificata. Restituisce un'eccezione `TransactionCommittedException` se è stato precedentemente eseguito il commit della transazione. Richiama internamente l'CancelTransactionAPI Lake Formation.

- `transaction_id`: (stringa) la transazione da annullare.

Scrittura

- getSink
- write_dynamic_frame_from_options
- write_from_options
- write_dynamic_frame_from_catalog
- write_data_frame_from_catalog
- write_dynamic_frame_from_jdbc_conf
- write_from_jdbc_conf

getSink

**`getSink(connection_type, format = None, transformation_ctx = "", **options)`**

Ottiene un oggetto `DataSink` che può essere utilizzato per scrivere `DynamicFrames` su fonti esterne. Verifica prima il `format` SparkSQL per essere certo di ricevere il sink previsto.

- `connection_type`: il tipo di connessione da utilizzare, come Amazon S3, Amazon Redshift e JDBC. I valori validi includono `s3`, `mysql`, `postgresql`, `redshift`, `sqlserver`, `oracle`, `kinesis` e `kafka`.
- `format`: il formato SparkSQL da utilizzare (opzionale).
- `transformation_ctx`: il contesto di trasformazione da usare (opzionale).
- `options`: raccolta di coppie nome-valore utilizzate per specificare le opzioni di connessione. Alcuni dei valori possibili sono:
  - `user` e `password`: per l'autorizzazione

- `url`: l'endpoint per il archivio dati
- `dbtable`: il nome della tabella di destinazione
- `bulkSize`: il grado di parallelismo per le operazioni di inserimento

Le opzioni che è possibile specificare dipendono dal tipo di connessione. Per ulteriori valori ed esempi, consultare Tipi e opzioni di connessione per ETL in AWS Glue per Spark.

Esempio:

```
>>> data_sink = context.getSink("s3")
>>> data_sink.setFormat("json"),
>>> data_sink.writeFrame(myFrame)
```

write_dynamic_frame_from_options

**`write_dynamic_frame_from_options(frame, connection_type, connection_options={}, format=None, format_options={}, transformation_ctx = "")`**

Legge e restituisce un `DynamicFrame` usando la connessione e il formato specificati.

- `frame`: il `DynamicFrame` da scrivere.
- `connection_type`: il tipo di connessione, come Amazon S3, Amazon Redshift e JDBC. I valori validi includono `s3`, `mysql`, `postgresql`, `redshift`, `sqlserver`, `oracle`, `kinesis` e `kafka`.
- `connection_options`: opzioni di connessione, come tabella di database e percorso (opzionale). Per un `connection_type` di `s3` è definito un percorso Amazon S3.

  ```
  connection_options = {"path": "s3://aws-glue-target/temp"}
  ```

  Per le connessioni JDBC, diverse proprietà devono essere definite. Il nome del database deve fare parte dell'URL. Può opzionalmente essere incluso nelle opzioni di connessione.

  > **Warning**
  >
  > Si consiglia di non archiviare le password nello script. Valuta la possibilità `boto3` di utilizzarli per recuperarli da Gestione dei segreti AWS o dal AWS Glue Data Catalog.

```
connection_options = {"url": "jdbc-url/database", "user": "username",
 "password": passwordVariable,"dbtable": "table-name", "redshiftTmpDir": "s3-tempdir-path"}
```

La proprietà `dbtable` è il nome della tabella JDBC. Per i archivi dati JDBC che supportano schemi all'interno di un database, specificare `schema.table-name`. Se non viene fornito alcuno schema, viene usato lo schema "pubblico" predefinito.

Per ulteriori informazioni, consultare Tipi e opzioni di connessione per ETL in AWS Glue per Spark.

- `format`: una specifica di formato. Viene usata per una connessione Amazon S3 o AWS Glue che supporta più formati. Consultare Opzioni del formato dati per input e output in AWS Glue per Spark per informazioni sui formati supportati.
- `format_options`: opzioni di formato per il formato specificato. Consultare Opzioni del formato dati per input e output in AWS Glue per Spark per informazioni sui formati supportati.
- `transformation_ctx`: un contesto di trasformazione da usare (opzionale).

write_from_options

**`write_from_options(frame_or_dfc, connection_type, connection_options={}, format={}, format_options={}, transformation_ctx = "")`**

Scrive e restituisce un `DynamicFrame` o una `DynamicFrameCollection` creati con la connessione e le informazioni di formattazione specificati.

- `frame_or_dfc`: il `DynamicFrame` o la `DynamicFrameCollection` per scrivere.
- `connection_type`: il tipo di connessione, come Amazon S3, Amazon Redshift e JDBC. I valori validi sono `s3`, `mysql`, `postgresql`, `redshift`, `sqlserver` e `oracle`.
- `connection_options`: opzioni di connessione, come tabella di database e percorso (opzionale). Per un `connection_type` di `s3` è definito un percorso Amazon S3.

```
connection_options = {"path": "s3://aws-glue-target/temp"}
```

Per le connessioni JDBC, diverse proprietà devono essere definite. Il nome del database deve fare parte dell'URL. Può opzionalmente essere incluso nelle opzioni di connessione.

> **Warning**
>
> Si consiglia di non archiviare le password nello script. Valuta la possibilità `boto3` di utilizzarli per recuperarli da Gestione dei segreti AWS o dal AWS Glue Data Catalog.

```
connection_options = {"url": "jdbc-url/database", "user": "username",
 "password": passwordVariable,"dbtable": "table-name", "redshiftTmpDir": "s3-tempdir-
path"}
```

La proprietà `dbtable` è il nome della tabella JDBC. Per i archivi dati JDBC che supportano schemi all'interno di un database, specificare `schema.table-name`. Se non viene fornito alcuno schema, viene usato lo schema "pubblico" predefinito.

Per ulteriori informazioni, consultare Tipi e opzioni di connessione per ETL in AWS Glue per Spark.

- `format`: una specifica di formato. Viene usata per una connessione Amazon S3 o AWS Glue che supporta più formati. Consultare Opzioni del formato dati per input e output in AWS Glue per Spark per informazioni sui formati supportati.
- `format_options`: opzioni di formato per il formato specificato. Consultare Opzioni del formato dati per input e output in AWS Glue per Spark per informazioni sui formati supportati.
- `transformation_ctx`: un contesto di trasformazione da usare (opzionale).

write_dynamic_frame_from_catalog

**`write_dynamic_frame_from_catalog(frame, database, table_name, redshift_tmp_dir, transformation_ctx = "", additional_options = {}, catalog_id = None)`**

Scrive e restituisce un `DynamicFrame` utilizzando un database del catalogo dati e una tabella.

- `frame`: il `DynamicFrame` da scrivere.
- `Database`: il database del catalogo dati che contiene la tabella.
- `table_name`: il nome della tabella del catalogo dati associata alla destinazione.
- `redshift_tmp_dir`: una directory temporanea Amazon Redshift da usare (opzionale).
- `transformation_ctx`: il contesto di trasformazione da usare (opzionale).

- `additional_options`: una raccolta di coppie nome/valore opzionali.
- `catalog_id`: l'ID catalogo (ID account) del catalogo dati a cui si accede. Se Nessuno, viene utilizzato l'ID account predefinito del chiamante.

write_data_frame_from_catalog

**`write_data_frame_from_catalog(frame, database, table_name, redshift_tmp_dir, transformation_ctx = "", additional_options = {}, catalog_id = None)`**

Scrive e restituisce un `DataFrame` utilizzando un database del catalogo dati e una tabella. Questo metodo supporta la scrittura nei formati di data lake (Hudi, Iceberg e Delta Lake). Per ulteriori informazioni, consultare Utilizzo di framework di data lake con AWS Glue processi ETL.

- `frame`: il `DataFrame` da scrivere.
- `Database`: il database del catalogo dati che contiene la tabella.
- `table_name`: il nome della tabella del catalogo dati associata alla destinazione.
- `redshift_tmp_dir`: una directory temporanea Amazon Redshift da usare (opzionale).
- `transformation_ctx`: il contesto di trasformazione da usare (opzionale).
- `additional_options`: una raccolta di coppie nome/valore opzionali.
  - `useSparkDataSink`— Se impostato su true, forza AWS Glue a utilizzare l'API nativa Spark Data Sink per scrivere sulla tabella. Quando abiliti questa opzione, puoi aggiungere qualsiasi opzione di Spark Data Source a seconda delle `additional_options` necessità. AWS Glue passa queste opzioni direttamente allo scrittore Spark.
- `catalog_id`: l'ID catalogo (ID account) del catalogo dati a cui si accede. Se non si specifica un valore, verrà utilizzato l'ID account predefinito del chiamante.

Limitazioni

Quando si utilizza l'opzione `useSparkDataSink` considerare le seguenti limitazioni:

- L'opzione `enableUpdateCatalog` non è supportata quando si utilizza l'opzione `useSparkDataSink`.

Esempio: scrittura su una tabella Hudi utilizzando lo scrittore Spark Data Source

```
hudi_options = {
    'useSparkDataSink': True,
    'hoodie.table.name': <table_name>,
    'hoodie.datasource.write.storage.type': 'COPY_ON_WRITE',
    'hoodie.datasource.write.recordkey.field': 'product_id',
    'hoodie.datasource.write.table.name': <table_name>,
    'hoodie.datasource.write.operation': 'upsert',
    'hoodie.datasource.write.precombine.field': 'updated_at',
    'hoodie.datasource.write.hive_style_partitioning': 'true',
    'hoodie.upsert.shuffle.parallelism': 2,
    'hoodie.insert.shuffle.parallelism': 2,
    'hoodie.datasource.hive_sync.enable': 'true',
    'hoodie.datasource.hive_sync.database': <database_name>,
    'hoodie.datasource.hive_sync.table': <table_name>,
    'hoodie.datasource.hive_sync.use_jdbc': 'false',
    'hoodie.datasource.hive_sync.mode': 'hms'}

glueContext.write_data_frame.from_catalog(
    frame = <df_product_inserts>,
    database = <database_name>,
    table_name = <table_name>,
    additional_options = hudi_options
)
```

write_dynamic_frame_from_jdbc_conf

### `write_dynamic_frame_from_jdbc_conf(frame, catalog_connection, connection_options={}, redshift_tmp_dir = "", transformation_ctx = "", catalog_id = None)`

Legge e restituisce un `DynamicFrame` usando le informazioni sulla connessione JDBC specificate.

- `frame`: il `DynamicFrame` da scrivere.
- `catalog_connection`: una connessione del catalogo da utilizzare.
- `connection_options`: opzioni di connessione, come tabella di database e percorso (opzionale). Per ulteriori informazioni, consultare Tipi e opzioni di connessione per ETL in AWS Glue per Spark.
- `redshift_tmp_dir`: una directory temporanea Amazon Redshift da usare (opzionale).
- `transformation_ctx`: un contesto di trasformazione da usare (opzionale).
- `catalog_id`: l'ID catalogo (ID account) del catalogo dati a cui si accede. Se Nessuno, viene utilizzato l'ID account predefinito del chiamante.

write_from_jdbc_conf

**`write_from_jdbc_conf(frame_or_dfc, catalog_connection, connection_options={}, redshift_tmp_dir = "", transformation_ctx = "", catalog_id = None)`**

Legge e restituisce un `DynamicFrame` o una `DynamicFrameCollection` usando le informazioni sulla connessione JDBC specificate.

- `frame_or_dfc`: il `DynamicFrame` o la `DynamicFrameCollection` per scrivere.
- `catalog_connection`: una connessione del catalogo da utilizzare.
- `connection_options`: opzioni di connessione, come tabella di database e percorso (opzionale). Per ulteriori informazioni, consultare Tipi e opzioni di connessione per ETL in AWS Glue per Spark.
- `redshift_tmp_dir`: una directory temporanea Amazon Redshift da usare (opzionale).
- `transformation_ctx`: un contesto di trasformazione da usare (opzionale).
- `catalog_id`: l'ID catalogo (ID account) del catalogo dati a cui si accede. Se Nessuno, viene utilizzato l'ID account predefinito del chiamante.

## AWS Glue PySpark trasforma il riferimento

AWS Glue fornisce le seguenti trasformazioni integrate che è possibile utilizzare nelle operazioni PySpark ETL. I dati passano da una trasformazione all'altra in una struttura di dati chiamata a DynamicFrame, che è un'estensione di Apache Spark SQL. `DataFrame` `DynamicFrame` contiene i tuoi dati e il suo schema di riferimento per elaborare i dati.

La maggior parte di queste trasformazioni esiste anche come metodi della classe `DynamicFrame`. Per ulteriori informazioni, consulta DynamicFrame trasformazioni.

- GlueTransform classe base
- ApplyMapping classe
- DropFields classe
- DropNullFields classe
- ErrorsAsDynamicFrame classe
- EvaluateDataQuality classe
- FillMissingValues classe
- Classe filtro

- FindIncrementalMatches classe
- FindMatches classe
- FlatMap classe
- Classe join
- Classe mappatura
- MapToCollection classe
- mergeDynamicFrame
- Classe relazionalizzazione
- RenameField classe
- ResolveChoice classe
- SelectFields classe
- SelectFromCollection classe
- Classe Simplify_ddb_json
- Classe Spigot
- SplitFields classe
- SplitRows classe
- Classe unbox
- UnnestFrame classe

GlueTransform classe base

La classe di base che tutte le classi `awsglue.transforms` ereditano.

Tutte le classi definiscono un `__call__` metodo. Sostituiscono i metodi della classe `GlueTransform` elencati nelle seguenti sezioni, oppure sono denominate utilizzando il nome della classe per impostazione predefinita.

Metodi

- apply(cls, *args, **kwargs)
- name(cls)
- describeArgs(cls)
- describeReturn(cls)
- describeTransform(cls)

- describeErrors(cls)
- describe(cls)

apply(cls, *args, **kwargs)

Applica la trasformazione chiamando la classe di trasformazione e restituisce il risultato.

- `cls`: l'oggetto `self` della classe.

name(cls)

Restituisce il nome della classe di trasformazione derivata.

- `cls`: l'oggetto `self` della classe.

describeArgs(cls)

- `cls`: l'oggetto `self` della classe.

Restituisce un elenco di dizionari, ciascuno corrispondente a un argomento denominato, nel formato seguente:

```
[
  {
    "name": "(name of argument)",
    "type": "(type of argument)",
    "description": "(description of argument)",
    "optional": "(Boolean, True if the argument is optional)",
    "defaultValue": "(Default value string, or None)(String; the default value, or
 None)"
  },
...
]
```

Solleva un'eccezione `NotImplementedError` quando viene chiamato in una trasformazione derivata dove non è stato implementato.

describeReturn(cls)

- `cls`: l'oggetto `self` della classe.

Restituisce un dizionario con informazioni sul tipo di restituzione, nel formato seguente:

```
{
  "type": "(return type)",
  "description": "(description of output)"
}
```

Solleva un'eccezione `NotImplementedError` quando viene chiamato in una trasformazione derivata dove non è stato implementato.

describeTransform(cls)

Restituisce una stringa che descrive la trasformazione.

- `cls`: l'oggetto `self` della classe.

Solleva un'eccezione `NotImplementedError` quando viene chiamato in una trasformazione derivata dove non è stato implementato.

describeErrors(cls)

- `cls`: l'oggetto `self` della classe.

Restituisce un elenco di dizionari, ognuno dei quali descrive una possibile eccezione generata da questa trasformazione, nel formato seguente:

```
[
  {
    "type": "(type of error)",
    "description": "(description of error)"
  },
...
]
```

describe(cls)

- `cls`: l'oggetto `self` della classe.

Restituisce un oggetto con il seguente formato:

```
{
  "transform" : {
    "name" : cls.name( ),
    "args" : cls.describeArgs( ),
    "returns" : cls.describeReturn( ),
    "raises" : cls.describeErrors( ),
    "location" : "internal"
  }
}
```

ApplyMapping classe

Applica una mappatura in un `DynamicFrame`.

Esempio

Ti consigliamo di utilizzare il metodo `DynamicFrame.apply_mapping()` per applicare una mappatura in un `DynamicFrame`. Per visualizzare un esempio di codice, consulta Esempio: usa apply_mapping per rinominare campi e modificare tipi di campo.

Metodi

- __call__
- apply
- name
- describeArgs
- describeReturn
- describeTransform
- describeErrors
- Describe

__call__(frame, mappings, transformation_ctx = "", info = "", stageThreshold = 0, totalThreshold = 0)

Applica una mappatura dichiarativa a un `DynamicFrame` specificato.

- `frame`: il `DynamicFrame` in cui applicare la mappatura (obbligatorio).
- `mappings`: un elenco di tuple di mappatura (obbligatorio). Ognuna è costituito da: colonna di origine, tipo di origine, colonna di destinazione, tipo di destinazione.

Se la colonna di origine include un punto "`.`" nel nome, deve essere racchiuso tra apici inversi "```". Ad esempio, per mappare `this.old.name` (stringa) a `thisNewName`, devi utilizzare la tupla seguente:

```
("`this.old.name`", "string", "thisNewName", "string")
```

- `transformation_ctx`: una stringa univoca utilizzata per identificare informazioni sullo stato (opzionale).
- `info`: una stringa associata a errori nella trasformazione (facoltativo).
- `stageThreshold`: il numero massimo di errori che si possono verificare nella trasformazione prima che venga arrestata (facoltativo). Il valore di default è zero.
- `totalThreshold`: il numero massimo di errori che si possono verificare in totale prima che l'elaborazione venga arrestata (facoltativo). Il valore di default è zero.

Restituisce solo i campi del `DynamicFrame` specificato nelle tuple di "mappatura".

apply(cls, *args, **kwargs)

Ereditato da `GlueTransform` apply.

name(cls)

Ereditato da `GlueTransform` name.

describeArgs(cls)

Ereditato da `GlueTransform` describeArgs.

describeReturn(cls)

Ereditato da `GlueTransform` describeReturn.

describeTransform(cls)

Ereditato da `GlueTransform` describeTransform.

describeErrors(cls)

Ereditato da `GlueTransform` describeErrors.

describe(cls)

Ereditato da `GlueTransform` describe.

DropFields classe

Elimina i campi all'interno di un `DynamicFrame`.

Esempio

Ti consigliamo di utilizzare il metodo `DynamicFrame.drop_fields()` per eliminare i campi da un `DynamicFrame`. Per visualizzare un esempio di codice, consulta Esempio: utilizza drop_fields per rimuovere campi da un `DynamicFrame`.

Metodi

- __call__
- apply
- name
- describeArgs
- describeReturn
- describeTransform
- describeErrors
- Describe

__call__ (frame, percorsi, transformation_ctx = "", info = "", stageThreshold = 0, totalThreshold = 0)

Elimina i nodi all'interno di un `DynamicFrame`.

- `frame`: il `DynamicFrame` in cui rimuovere i nodi (obbligatorio).
- `paths`: un elenco di percorsi completi dei nodi da rilasciare (obbligatorio).
- `transformation_ctx`: una stringa univoca utilizzata per identificare informazioni sullo stato (opzionale).
- `info`: una stringa associata a errori nella trasformazione (opzionale).
- `stageThreshold`: il numero massimo di errori che si possono verificare nella trasformazione prima che venga arrestata (facoltativo). Il valore di default è zero.
- `totalThreshold`: il numero massimo di errori che si possono verificare in totale prima che l'elaborazione venga arrestata (facoltativo). Il valore di default è zero.

Restituisce un nuovo `DynamicFrame` senza i campi specificati.

apply(cls, *args, **kwargs)

Ereditato da `GlueTransform` apply.

name(cls)

Ereditato da `GlueTransform` name.

describeArgs(cls)

Ereditato da `GlueTransform` describeArgs.

describeReturn(cls)

Ereditato da `GlueTransform` describeReturn.

describeTransform(cls)

Ereditato da `GlueTransform` describeTransform.

describeErrors(cls)

Ereditato da `GlueTransform` describeErrors.

describe(cls)

Ereditato da `GlueTransform` describe.

DropNullFields classe

Elimina tutti i campi nulli in un `DynamicFrame` il cui tipo è `NullType`. Questi sono campi con valori mancanti o nulli in ogni record nel set di dati `DynamicFrame`.

Esempio

Questo esempio usa `DropNullFields` per crearne una nuovo `DynamicFrame` dove sono stati rimossi i campi di tipo `NullType`. Per dimostrare `DropNullFields`, aggiungiamo una nuova colonna `empty_column` di tipo null al set di dati `persons` già caricato.

> **Note**
>
> Per accedere al set di dati utilizzato in questo esempio, consulta Esempio di codice: unione e relazioni dei dati e segui le istruzioni in Fase 1: esecuzione del crawling sui dati nel bucket Amazon S3.

```
# Example: Use DropNullFields to create a new DynamicFrame without NullType fields

from pyspark.context import SparkContext
from awsglue.context import GlueContext
from pyspark.sql.functions import lit
from pyspark.sql.types import NullType
from awsglue.dynamicframe import DynamicFrame
from awsglue.transforms import DropNullFields

# Create GlueContext
sc = SparkContext.getOrCreate()
glueContext = GlueContext(sc)

# Create DynamicFrame
persons = glueContext.create_dynamic_frame.from_catalog(
    database="legislators", table_name="persons_json"
)
print("Schema for the persons DynamicFrame:")
persons.printSchema()

# Add new column "empty_column" with NullType
persons_with_nulls = persons.toDF().withColumn("empty_column",
 lit(None).cast(NullType()))
persons_with_nulls_dyf = DynamicFrame.fromDF(persons_with_nulls, glueContext,
 "persons_with_nulls")
print("Schema for the persons_with_nulls_dyf DynamicFrame:")
persons_with_nulls_dyf.printSchema()

# Remove the NullType field
persons_no_nulls = DropNullFields.apply(persons_with_nulls_dyf)
print("Schema for the persons_no_nulls DynamicFrame:")
persons_no_nulls.printSchema()
```

## Output

```
Schema for the persons DynamicFrame:
root
|-- family_name: string
|-- name: string
|-- links: array
|    |-- element: struct
|    |    |-- note: string
|    |    |-- url: string
|-- gender: string
|-- image: string
|-- identifiers: array
|    |-- element: struct
|    |    |-- scheme: string
|    |    |-- identifier: string
|-- other_names: array
|    |-- element: struct
|    |    |-- lang: string
|    |    |-- note: string
|    |    |-- name: string
|-- sort_name: string
|-- images: array
|    |-- element: struct
|    |    |-- url: string
|-- given_name: string
|-- birth_date: string
|-- id: string
|-- contact_details: array
|    |-- element: struct
|    |    |-- type: string
|    |    |-- value: string
|-- death_date: string

Schema for the persons_with_nulls_dyf DynamicFrame:
root
|-- family_name: string
|-- name: string
|-- links: array
|    |-- element: struct
|    |    |-- note: string
|    |    |-- url: string
|-- gender: string
|-- image: string
```

```
|-- identifiers: array
|    |-- element: struct
|    |    |-- scheme: string
|    |    |-- identifier: string
|-- other_names: array
|    |-- element: struct
|    |    |-- lang: string
|    |    |-- note: string
|    |    |-- name: string
|-- sort_name: string
|-- images: array
|    |-- element: struct
|    |    |-- url: string
|-- given_name: string
|-- birth_date: string
|-- id: string
|-- contact_details: array
|    |-- element: struct
|    |    |-- type: string
|    |    |-- value: string
|-- death_date: string
|-- empty_column: null

null_fields ['empty_column']
Schema for the persons_no_nulls DynamicFrame:
root
|-- family_name: string
|-- name: string
|-- links: array
|    |-- element: struct
|    |    |-- note: string
|    |    |-- url: string
|-- gender: string
|-- image: string
|-- identifiers: array
|    |-- element: struct
|    |    |-- scheme: string
|    |    |-- identifier: string
|-- other_names: array
|    |-- element: struct
|    |    |-- lang: string
|    |    |-- note: string
|    |    |-- name: string
|-- sort_name: string
```

```
|-- images: array
|    |-- element: struct
|    |    |-- url: string
|-- given_name: string
|-- birth_date: string
|-- id: string
|-- contact_details: array
|    |-- element: struct
|    |    |-- type: string
|    |    |-- value: string
|-- death_date: string
```

Metodi

- __call__
- apply
- name
- describeArgs
- describeReturn
- describeTransform
- describeErrors
- Describe

__call__(frame, transformation_ctx = "", info = "", stageThreshold = 0, totalThreshold = 0)

Elimina tutti i campi nulli in un `DynamicFrame` il cui tipo è `NullType`. Questi sono campi con valori mancanti o nulli in ogni record nel set di dati `DynamicFrame`.

- `frame`: il `DynamicFrame` in cui rimuovere i campi nulli (obbligatorio).
- `transformation_ctx`: una stringa univoca utilizzata per identificare informazioni sullo stato (opzionale).
- `info`: una stringa associata a errori nella trasformazione (opzionale).
- `stageThreshold`: il numero massimo di errori che si possono verificare nella trasformazione prima che venga arrestata (facoltativo). Il valore di default è zero.
- `totalThreshold`: il numero massimo di errori che si possono verificare in totale prima che l'elaborazione venga arrestata (facoltativo). Il valore di default è zero.

Restituisce un nuovo `DynamicFrame` senza campi nulli.

apply(cls, *args, **kwargs)

- `cls`: cls

name(cls)

- `cls`: cls

describeArgs(cls)

- `cls`: cls

describeReturn(cls)

- `cls`: cls

describeTransform(cls)

- `cls`: cls

describeErrors(cls)

- `cls`: cls

describe(cls)

- `cls`: cls

ErrorsAsDynamicFrame classe

Restituisce un `DynamicFrame` contenente record nidificati per gli errori che si sono verificati durante la creazione del `DynamicFrame` di origine.

Esempio

Ti consigliamo di utilizzare il metodo `DynamicFrame.errorsAsDynamicFrame()` per recuperare e visualizzare i record degli errori. Per visualizzare un esempio di codice, consulta Esempio: utilizzare errorsAsDynamic Frame per visualizzare i record di errori.

Metodi

- __call__
- apply
- name
- describeArgs
- describeReturn
- describeTransform
- describeErrors
- Describe

__call__(frame)

Restituisce un `DynamicFrame` contenente record nidificati degli errori correlati al `DynamicFrame` di origine.

- `frame`: il `DynamicFrame` di origine (obbligatorio).

apply(cls, *args, **kwargs)

- `cls`: cls

name(cls)

- `cls`: cls

describeArgs(cls)

- `cls`: cls

describeReturn(cls)

- `cls`: cls

describeTransform(cls)

- `cls`: cls

describeErrors(cls)

- `cls`: cls

describe(cls)

- `cls`: cls

EvaluateDataQuality classe

Valuta un set di regole di qualità dei dati rispetto ai dati in un `DynamicFrame` e restituisce un nuovo `DynamicFrame` con i risultati della valutazione.

Esempio

Il seguente codice di esempio dimostra come valutare la qualità dei dati per un `DynamicFrame` e quindi visualizzare i risultati.

```
from awsglue.transforms import *
from pyspark.context import SparkContext
from awsglue.context import GlueContext
from awsgluedq.transforms import EvaluateDataQuality

#Create Glue context
sc = SparkContext.getOrCreate()
glueContext = GlueContext(sc)

# Define DynamicFrame
legislatorsAreas = glueContext.create_dynamic_frame.from_catalog(
    database="legislators", table_name="areas_json")

# Create data quality ruleset
```

```
ruleset = """Rules = [ColumnExists "id", IsComplete "id"]"""

# Evaluate data quality
dqResults = EvaluateDataQuality.apply(
    frame=legislatorsAreas,
    ruleset=ruleset,
    publishing_options={
        "dataQualityEvaluationContext": "legislatorsAreas",
        "enableDataQualityCloudWatchMetrics": True,
        "enableDataQualityResultsPublishing": True,
        "resultsS3Prefix": "amzn-s3-demo-bucket1",
    },
)


# Inspect data quality results
dqResults.printSchema()
dqResults.toDF().show()
```

## Output

```
root
|-- Rule: string
|-- Outcome: string
|-- FailureReason: string
|-- EvaluatedMetrics: map
|    |-- keyType: string
|    |-- valueType: double


+----------------------+-------+-------------+--------------------------------------+
|Rule                  |Outcome|FailureReason|EvaluatedMetrics                      |
+----------------------+-------+-------------+--------------------------------------+
|ColumnExists "id"     |Passed |null         |{}                                    |
|IsComplete "id"       |Passed |null         |{Column.first_name.Completeness -> 1.0}|
+----------------------+-------+-------------+--------------------------------------+
```

## Metodi

- __call__
- apply
- name

- describeArgs
- describeReturn
- describeTransform
- describeErrors
- describe

__call__(frame, ruleset, publishing_options = {})

- `frame`: il `DynamicFrame` di cui desideri valutare la qualità dei dati.
- `ruleset`: un set di regole del Data Quality Definition Language (DQDL) in formato stringa. Per ulteriori informazioni su DQDL, consulta la guida di Riferimento a Data Quality Definition Language (DQDL).
- `publishing_options`: un dizionario che specifica le seguenti opzioni per la pubblicazione dei risultati e dei parametri di valutazione:
  - `dataQualityEvaluationContext`— Una stringa che specifica lo spazio dei nomi in cui AWS Glue deve pubblicare le Amazon CloudWatch metriche e i risultati sulla qualità dei dati. Le metriche aggregate vengono visualizzate in CloudWatch, mentre i risultati completi vengono visualizzati nell'interfaccia AWS Glue Studio.
    - Obbligatorio: no
    - Valore predefinito: `default_context`
  - `enableDataQualityCloudWatchMetrics`— Speciifica se i risultati della valutazione della qualità dei dati devono essere pubblicati su. CloudWatch Uno spazio dei nomi per i parametri viene specificato utilizzando l'opzione `dataQualityEvaluationContext`.
    - Obbligatorio: no
    - Valore predefinito: False
  - `enableDataQualityResultsPublishing`: specifica se i risultati della qualità dei dati devono essere visibili nella scheda Data Quality (Qualità dei dati) nell'interfaccia di AWS Glue Studio.
    - Obbligatorio: no
    - Valore predefinito: true
  - `resultsS3Prefix`— Speciifica la posizione di Amazon S3 in cui AWS Glue può scrivere i risultati della valutazione della qualità dei dati.
    - Obbligatorio: no
    - Valore predefinito: "" (stringa vuota)

apply(cls, *args, **kwargs)

Ereditato da `GlueTransform` apply.

name(cls)

Ereditato da `GlueTransform` name.

describeArgs(cls)

Ereditato da `GlueTransform` describeArgs.

describeReturn(cls)

Ereditato da `GlueTransform` describeReturn.

describeTransform(cls)

Ereditato da `GlueTransform` describeTransform.

describeErrors(cls)

Ereditato da `GlueTransform` describeErrors.

describe(cls)

Ereditato da `GlueTransform` describe.

FillMissingValues classe

La classe `FillMissingValues` individua i valori null e stringhe vuote in un `DynamicFrame` specificato e utilizza metodi di machine learning, come la regressione lineare e la foresta casuale, per prevedere i valori mancanti. Il processo ETL utilizza i valori nel set di dati di input per addestrare il modello di machine learning, che prevede quindi quali devono essere i valori mancanti.

> **Tip**
>
> Se si utilizzano set di dati incrementali, ogni set incrementale viene utilizzato come dati di addestramento per il modello di machine learning, pertanto i risultati potrebbero non essere molto accurati.

Per l'importazione:

```
from awsglueml.transforms import FillMissingValues
```

Metodi

- Applica

apply(frame, missing_values_column, output_column ="", transformation_ctx ="", info ="", stageThreshold = 0, totalThreshold = 0)

Riempie i valori mancanti di un frame dinamico in una colonna specificata e restituisce un frame dinamico con stime in una nuova colonna. Per le righe senza valori mancanti, il valore della colonna specificato viene duplicato nella nuova colonna.

- `frame` il `DynamicFrame` in cui inserire i valori mancanti. Obbligatorio.
- `missing_values_column`: la colonna contenente valori mancanti (valori `null` e stringhe vuote). Obbligatorio.
- `output_column`: il nome della nuova colonna che conterrà i valori stimati per tutte le righe il cui valore era mancante. Facoltativo; il valore di default è il nome di `missing_values_column` con suffisso formato da `"_filled"`.
- `transformation_ctx`: una stringa univoca utilizzata per identificare informazioni sullo stato (opzionale).
- `info`: una stringa associata a errori nella trasformazione (opzionale).
- `stageThreshold`: il numero massimo di errori che si possono verificare nella trasformazione prima che venga arrestata (opzionale; il numero predefinito è zero).
- `totalThreshold`: il numero massimo di errori che si possono verificare in totale prima che l'elaborazione venga arrestata (opzionale; il numero predefinito è zero).

Restituisce un nuovo `DynamicFrame` con una colonna aggiuntiva che contiene stime per le righe con valori mancanti e il valore attuale per le altre righe.

Classe filtro

Crea un nuovo `DynamicFrame` contenente record dall'input `DynamicFrame` che soddisfano una funzione predicato specificata.

Esempio

Ti consigliamo di utilizzare il metodo `DynamicFrame.filter()` per filtrare i record in un `DynamicFrame`. Per visualizzare un esempio di codice, consulta Esempio: usa il filtro per ottenere una selezione filtrata di campi.

Metodi

- __call__
- apply
- name
- describeArgs
- describeReturn
- describeTransform
- describeErrors
- describe

__call__(frame, f, transformation_ctx="", info="", stageThreshold=0, totalThreshold=0))

Restituisce un nuovo `DynamicFrame` creato selezionando i record dall'input `DynamicFrame` che soddisfano una funzione predicato specificata.

- `frame`: l'origine `DynamicFrame` da applicare alla funzione filtro specificata (campo obbligatorio).
- `f`: la funzione di predicato da applicare a ciascun `DynamicRecord` nella `DynamicFrame`. La funzione deve assumere un `DynamicRecord` come argomento e restituire True se il `DynamicRecord` soddisfa i requisiti del filtro, altrimenti False (obbligatorio).

  Un `DynamicRecord` rappresenta un record logico all'interno di un `DynamicFrame`. È simile a una riga in un `DataFrame` Spark, con la differenza che è autodescrittivo e può essere utilizzato per dati non conformi a uno schema fisso.
- `transformation_ctx`: una stringa univoca utilizzata per identificare informazioni sullo stato (opzionale).
- `info`: una stringa associata a errori nella trasformazione (facoltativo).
- `stageThreshold`: il numero massimo di errori che si possono verificare nella trasformazione prima che venga arrestata (facoltativo). Il valore di default è zero.

- `totalThreshold`: il numero massimo di errori che si possono verificare in totale prima che l'elaborazione venga arrestata (facoltativo). Il valore di default è zero.

apply(cls, *args, **kwargs)

Ereditato da `GlueTransform` apply.

name(cls)

Ereditato da `GlueTransform` name.

describeArgs(cls)

Ereditato da `GlueTransform` describeArgs.

describeReturn(cls)

Ereditato da `GlueTransform` describeReturn.

describeTransform(cls)

Ereditato da `GlueTransform` describeTransform.

describeErrors(cls)

Ereditato da `GlueTransform` describeErrors.

describe(cls)

Ereditato da `GlueTransform` describe.

FindIncrementalMatches classe

Identifica i record corrispondenti nel `DynamicFrame` esistente e incrementale e crea un nuovo `DynamicFrame` con un identificatore univoco assegnato a ciascun gruppo di record corrispondenti.

Per l'importazione:

```
from awsglueml.transforms import FindIncrementalMatches
```

Metodi

- Applica

applica (ExistingFrame, IncrementalFrame, transformId, transformation_ctx = «», info = «», stageThreshold = 0, totalThreshold = 0, EnforcedMatches = nessuno, punteggi = 0) computeMatchConfidence

Identifica i record corrispondenti nel `DynamicFrame` di input e crea un nuovo `DynamicFrame` con un identificatore univoco assegnato a ciascun gruppo di record corrispondenti.

- `existingFrame`— L'`DynamicFrame`esistente e FindIncrementalMatches quello precorrispondente per applicare la trasformazione. Obbligatorio.
- `incrementalFrame`— L'incrementale `DynamicFrame` da applicare alla FindIncrementalMatches trasformazione da confrontare con. `existingFrame` Obbligatorio.
- `transformId`— L'ID univoco associato alla FindIncrementalMatches trasformazione da applicare ai record in. `DynamicFrames` Obbligatorio.
- `transformation_ctx`— Una stringa univoca utilizzata per identificare stats/state le informazioni. Opzionale.
- `info`: una stringa associata a errori nella trasformazione. Opzionale.
- `stageThreshold`: il numero massimo di errori che si possono verificare nella trasformazione prima che venga arrestata. Opzionale. Il valore di default è zero.
- `totalThreshold`: il numero massimo di errori che si possono verificare in totale prima che l'elaborazione venga arrestata. Opzionale. Il valore di default è zero.
- `enforcedMatches`: il `DynamicFrame` utilizzato per applicare le corrispondenze. Opzionale. Il valore predefinito è None (Nessuno).
- `computeMatchConfidenceScores`: un valore booleano che indica se calcolare un punteggio di confidenza per ciascun gruppo di record corrispondenti. Opzionale. Il valore predefinito è false.

Restituisce un nuovo `DynamicFrame` con un identificatore univoco assegnato a ciascun gruppo di record corrispondenti.

FindMatches classe

Identifica i record corrispondenti nel `DynamicFrame` di input e crea un nuovo `DynamicFrame` con un identificatore univoco assegnato a ciascun gruppo di record corrispondenti.

Per l'importazione:

```
from awsglueml.transforms import FindMatches
```

Metodi

- Applica

applica (frame, transformId, transformation_ctx = «», info = «», stageThreshold = 0, totalThreshold = 0, EnforcedMatches = nessuno, punteggi = 0) computeMatchConfidence

Identifica i record corrispondenti nel `DynamicFrame` di input e crea un nuovo `DynamicFrame` con un identificatore univoco assegnato a ciascun gruppo di record corrispondenti.

- `frame`— `DynamicFrame` Per applicare la trasformazione. FindMatches Obbligatorio.
- `transformId`— L'ID univoco associato alla FindMatches trasformazione da applicare ai record in`DynamicFrame`. Obbligatorio.
- `transformation_ctx`— Una stringa univoca utilizzata per identificare stats/state le informazioni. Opzionale.
- `info`: una stringa associata a errori nella trasformazione. Opzionale.
- `stageThreshold`: il numero massimo di errori che si possono verificare nella trasformazione prima che venga arrestata. Opzionale. Il valore di default è zero.
- `totalThreshold`: il numero massimo di errori che si possono verificare in totale prima che l'elaborazione venga arrestata. Opzionale. Il valore di default è zero.
- `enforcedMatches`: il `DynamicFrame` utilizzato per applicare le corrispondenze. Opzionale. Il valore predefinito è None (Nessuno).
- `computeMatchConfidenceScores`: un valore booleano che indica se calcolare un punteggio di confidenza per ciascun gruppo di record corrispondenti. Opzionale. Il valore predefinito è false.

Restituisce un nuovo `DynamicFrame` con un identificatore univoco assegnato a ciascun gruppo di record corrispondenti.

FlatMap classe

Applica una trasformazione a ogni `DynamicFrame` in una raccolta. I risultati non vengono appiattiti in un unico `DynamicFrame` ma conservati come una raccolta.

Esempi per FlatMap

Il seguente frammento di esempio mostra come utilizzare la trasformazione `ResolveChoice` su una raccolta di frame dinamici quando applicata a una `FlatMap`. I dati utilizzati per l'input sono nel JSON

situato all'indirizzo segnaposto `s3://bucket/path-for-data/sample.json` di Amazon S3 e contengono i seguenti dati.

Dati JSON di esempio

```
[{
    "firstname": "Arnav",
    "lastname": "Desai",
    "address": {
        "street": "6 Anyroad Avenue",
        "city": "London",
        "state": "England",
        "country": "UK"
    },
    "phone": 17235550101,
    "affiliations": [
        "General Anonymous Example Products",
        "Example Independent Research",
        "Government Department of Examples"
    ]
},
{
    "firstname": "Mary",
    "lastname": "Major",
    "address": {
        "street": "7821 Spot Place",
        "city": "Centerville",
        "state": "OK",
        "country": "US"
    },
    "phone": 19185550023,
    "affiliations": [
        "Example Dot Com",
        "Example Independent Research",
        "Example.io"
    ]
},
{
    "firstname": "Paulo",
    "lastname": "Santos",
    "address": {
        "street": "123 Maple Street",
        "city": "London",
        "state": "Ontario",
```

```
        "country": "CA"
    },
    "phone": 12175550181,
    "affiliations": [
        "General Anonymous Example Products",
        "Example Dot Com"
    ]
}]
```

Example Applica ResolveChoice a un output DynamicFrameCollection e mostra.

```
#Read DynamicFrame
datasource = glueContext.create_dynamic_frame_from_options("s3", connection_options =
 {"paths":["s3://bucket/path/to/file/mysamplejson.json"]}, format="json")
datasource.printSchema()
datasource.show()

## Split to create a DynamicFrameCollection
split_frame=datasource.split_fields(["firstname","lastname","address"],"personal_info","busines
split_frame.keys()
print("---")

## Use FlatMap to run ResolveChoice
kwargs = {"choice": "cast:string"}
flat = FlatMap.apply(split_frame, ResolveChoice, frame_name="frame",
 transformation_ctx='tcx', **kwargs)
flat.keys()

##Select one of the DynamicFrames
personal_info = flat.select("personal_info")
personal_info.printSchema()
personal_info.show()
print("---")

business_info = flat.select("business_info")
business_info.printSchema()
business_info.show()
```

> **Important**
>
> Quando si chiama `FlatMap.apply`, il parametro `frame_name` deve essere `"frame"`. Nessun altro valore è attualmente accettato.

## Output di esempio

```
root
|-- firstname: string
|-- lastname: string
|-- address: struct
|    |-- street: string
|    |-- city: string
|    |-- state: string
|    |-- country: string
|-- phone: long
|-- affiliations: array
|    |-- element: string
---
{
    "firstname": "Mary",
    "lastname": "Major",
    "address": {
        "street": "7821 Spot Place",
        "city": "Centerville",
        "state": "OK",
        "country": "US"
    },
    "phone": 19185550023,
    "affiliations": [
        "Example Dot Com",
        "Example Independent Research",
        "Example.io"
    ]
}

{
    "firstname": "Paulo",
    "lastname": "Santos",
    "address": {
        "street": "123 Maple Street",
        "city": "London",
        "state": "Ontario",
        "country": "CA"
    },
    "phone": 12175550181,
    "affiliations": [
        "General Anonymous Example Products",
        "Example Dot Com"
```

```
    ]
}
---
root
|-- firstname: string
|-- lastname: string
|-- address: struct
|    |-- street: string
|    |-- city: string
|    |-- state: string
|    |-- country: string

{
    "firstname": "Mary",
    "lastname": "Major",
    "address": {
        "street": "7821 Spot Place",
        "city": "Centerville",
        "state": "OK",
        "country": "US"
    }
}

{
    "firstname": "Paulo",
    "lastname": "Santos",
    "address": {
        "street": "123 Maple Street",
        "city": "London",
        "state": "Ontario",
        "country": "CA"
    }
}
---
root
|-- phone: long
|-- affiliations: array
|    |-- element: string

{
    "phone": 19185550023,
    "affiliations": [
        "Example Dot Com",
        "Example Independent Research",
```

```
        "Example.io"
    ]
}

{
    "phone": 12175550181,
    "affiliations": [
        "General Anonymous Example Products",
        "Example Dot Com"
    ]
}
```

Metodi

- __call__
- Applica
- Name
- describeArgs
- describeReturn
- describeTransform
- describeErrors
- Describe

__call__ (dfc, frame_name BaseTransform, transformation_ctx = «», **base_kwargs)

Si applica alla trasformazione per ogni `DynamicFrame` in una raccolta e appiattisce i risultati.

- `dfc`: la `DynamicFrameCollection` su cui applicare la classe flatmap (obbligatorio).
- `BaseTransform`: una trasformazione derivata da `GlueTransform` da applicare a ciascun membro della raccolta (obbligatorio).
- `frame_name`: Il nome dell'argomento a cui passare gli elementi della raccolta (obbligatorio).
- `transformation_ctx`: una stringa univoca utilizzata per identificare informazioni sullo stato (opzionale).
- `base_kwargs`: argomenti per passare alla trasformazione di base (obbligatorio).

Restituisce una nuova `DynamicFrameCollection` creata applicando la trasformazione a ciascun `DynamicFrame` nella `DynamicFrameCollection` di origine.

apply(cls, *args, **kwargs)

Ereditato da `GlueTransform` apply.

name(cls)

Ereditato da `GlueTransform` name.

describeArgs(cls)

Ereditato da `GlueTransform` describeArgs.

describeReturn(cls)

Ereditato da `GlueTransform` describeReturn.

describeTransform(cls)

Ereditato da `GlueTransform` describeTransform.

describeErrors(cls)

Ereditato da `GlueTransform` describeErrors.

describe(cls)

Ereditato da `GlueTransform` describe.

Classe join

Esegue un equi join su due `DynamicFrames`.

Esempio

Ti consigliamo di utilizzare il metodo `DynamicFrame.join()` per unire `DynamicFrames`. Per visualizzare un esempio di codice, consulta Esempio: usa join per combinare `DynamicFrames`.

Metodi

- __call__
- apply
- name

- describeArgs
- describeReturn
- describeTransform
- describeErrors
- Describe

__call__ (frame1, frame2, keys1, keys2, transformation_ctx = "")

Esegue un equi join su due `DynamicFrames`.

- `frame1`: il primo `DynamicFrame` da unire (obbligatorio).
- `frame2`: il secondo `DynamicFrame` da unire (obbligatorio).
- `keys1`: le chiavi da unire per il primo frame (obbligatorio).
- `keys2`: le chiavi da unire per il secondo frame (obbligatorio).
- `transformation_ctx`: una stringa univoca utilizzata per identificare informazioni sullo stato (opzionale).

Restituisce un nuovo `DynamicFrame` che viene creato unendo i due `DynamicFrames`.

apply(cls, *args, **kwargs)

Ereditato da `GlueTransform` apply.

name(cls)

Ereditato da `GlueTransform` name.

describeArgs(cls)

Ereditato da `GlueTransform` describeArgs.

describeReturn(cls)

Ereditato da `GlueTransform` describeReturn.

describeTransform(cls)

Ereditato da `GlueTransform` describeTransform.

describeErrors(cls)

Ereditato da `GlueTransform` describeErrors.

describe(cls)

Ereditato da `GlueTransform` describe.

Classe mappatura

Crea un nuovo `DynamicFrame` applicando una funzione a tutti i record nell'input `DynamicFrame`.

Esempio

Ti consigliamo di utilizzare il metodo `DynamicFrame.map()` per applicare una funzione a tutti i record in un `DynamicFrame`. Per visualizzare un esempio di codice, consulta Esempio: utilizza la mappa per applicare una funzione a ogni record in un `DynamicFrame`.

Metodi

- __call__
- apply
- name
- describeArgs
- describeReturn
- describeTransform
- describeErrors
- Describe

__call__(frame, f, transformation_ctx="", info="", stageThreshold=0, totalThreshold=0)

Restituisce un nuovo `DynamicFrame` ottenuto applicando la funzione specificata a tutti i `DynamicRecords` nel `DynamicFrame` originale.

- `frame`: il `DynamicFrame` originale a cui applicare la funzione di mappatura (obbligatorio).
- `f`: la funzione da applicare a tutti i `DynamicRecords` nel `DynamicFrame`. La funzione deve assumere un `DynamicRecord` come argomento e restituire un nuovo `DynamicRecord` prodotto dalla mappatura (obbligatorio).

Un `DynamicRecord` rappresenta un record logico all'interno di un `DynamicFrame`. È simile a una riga in un `DataFrame` Apache Spark, con la differenza che è autodescrittivo e può essere utilizzato per dati non conformi a uno schema fisso.

- `transformation_ctx`: una stringa univoca utilizzata per identificare informazioni sullo stato (opzionale).
- `info`: una stringa associata a errori nella trasformazione (opzionale).
- `stageThreshold`: il numero massimo di errori che si possono verificare nella trasformazione prima che venga arrestata (facoltativo). Il valore di default è zero.
- `totalThreshold`: il numero massimo di errori che si possono verificare in totale prima che l'elaborazione venga arrestata (facoltativo). Il valore di default è zero.

Restituisce un nuovo `DynamicFrame` ottenuto applicando la funzione specificata a tutti i `DynamicRecords` nel `DynamicFrame` originale.

apply(cls, *args, **kwargs)

Ereditato da `GlueTransform` apply.

name(cls)

Ereditato da `GlueTransform` name.

describeArgs(cls)

Ereditato da `GlueTransform` describeArgs.

describeReturn(cls)

Ereditato da `GlueTransform` describeReturn.

describeTransform(cls)

Ereditato da `GlueTransform` describeTransform.

describeErrors(cls)

Ereditato da `GlueTransform` describeErrors.

describe(cls)

Ereditato da `GlueTransform` describe.

MapToCollection classe

Applica una trasformazione a ogni `DynamicFrame` nella `DynamicFrameCollection` specificata.

Metodi

- __call__
- Applica
- Name
- describeArgs
- describeReturn
- describeTransform
- describeErrors
- Describe

__call__ (dfc, frame_name BaseTransform, transformation_ctx = «», **base_kwargs)

Applica una funzione di trasformazione a ogni `DynamicFrame` nella `DynamicFrameCollection` specificata.

- `dfc`: la `DynamicFrameCollection` a cui applicare la funzione di trasformazione (obbligatorio).
- `callable`: la funzione di trasformazione chiamabile da applicare a ciascun membro della raccolta (obbligatorio).
- `transformation_ctx`: una stringa univoca utilizzata per identificare informazioni sullo stato (opzionale).

Restituisce una nuova `DynamicFrameCollection` creata applicando la trasformazione a ciascun `DynamicFrame` nella `DynamicFrameCollection` di origine.

apply(cls, *args, **kwargs)

Ereditato da `GlueTransform` apply

name(cls)

Ereditato da `GlueTransform` name.

describeArgs(cls)

Ereditato da `GlueTransform` describeArgs.

describeReturn(cls)

Ereditato da `GlueTransform` describeReturn.

describeTransform(cls)

Ereditato da `GlueTransform` describeTransform.

describeErrors(cls)

Ereditato da `GlueTransform` describeErrors.

describe(cls)

Ereditato da `GlueTransform` describe.

Classe relazionalizzazione

Livella uno schema nidificato in un `DynamicFrame` e trasforma le colonne della matrice tramite pivoting da un frame appiattito.

Esempio

Ti consigliamo di utilizzare il metodo `DynamicFrame.relationalize()` per relazionare `DynamicFrame`. Per visualizzare un esempio di codice, consulta Esempio: usa la relazionalizzazione per livellare uno schema annidato in un `DynamicFrame`.

Metodi

- __call__
- apply
- name

- describeArgs
- describeReturn
- describeTransform
- describeErrors
- describe

__call__(frame, staging_path=None, name='roottable', options=None, transformation_ctx = "", info = "", stageThreshold = 0, totalThreshold = 0)

Mette in relazione un `DynamicFrame` e produce un elenco di frame che vengono generati annullando l'annidamento di colonne nidificate e trasformando colonne della matrice mediante pivot. La colonna matrice trasformata mediante pivot può essere unita alla tabella root utilizzando la chiave di join generata durante la fase di annullamento dell'annidamento.

- `frame`: il `DynamicFrame` da mettere in relazione (obbligatorio).
- `staging_path`: il percorso in cui il metodo può archiviare le partizioni di tabelle trasformate mediante pivot in formato CSV (facoltativo). Le tabelle trasformate mediante pivot vengono rilette da questo percorso.
- `name`: il nome della tabella root (opzionale).
- `options`: un dizionario dei parametri opzionali. Attualmente inutilizzato.
- `transformation_ctx`: una stringa univoca utilizzata per identificare informazioni sullo stato (opzionale).
- `info`: una stringa associata a errori nella trasformazione (opzionale).
- `stageThreshold`: il numero massimo di errori che si possono verificare nella trasformazione prima che venga arrestata (facoltativo). Il valore di default è zero.
- `totalThreshold`: il numero massimo di errori che si possono verificare in totale prima che l'elaborazione venga arrestata (facoltativo). Il valore di default è zero.

apply(cls, *args, **kwargs)

Ereditato da `GlueTransform` apply.

name(cls)

Ereditato da `GlueTransform` name.

describeArgs(cls)

Ereditato da `GlueTransform` describeArgs.

describeReturn(cls)

Ereditato da `GlueTransform` describeReturn.

describeTransform(cls)

Ereditato da `GlueTransform` describeTransform.

describeErrors(cls)

Ereditato da `GlueTransform` describeErrors.

describe(cls)

Ereditato da `GlueTransform` describe.

RenameField classe

Rinomina un nodo all'interno di un `DynamicFrame`.

Esempio

Ti consigliamo di utilizzare il metodo `DynamicFrame.rename_field()` per rinominare un campo in un `DynamicFrame`. Per visualizzare un esempio di codice, consulta Esempio: usa rename_field per rinominare i campi in un `DynamicFrame`.

Metodi

- __call__
- apply
- name
- describeArgs
- describeReturn
- describeTransform
- describeErrors
- describe

__call__(frame, old_name, new_name, transformation_ctx = "", info = "", stageThreshold = 0, totalThreshold = 0)

Rinomina un nodo all'interno di un `DynamicFrame`.

- `frame`: il `DynamicFrame` in cui rinominare un nodo (obbligatorio).
- `old_name`: il percorso completo del nodo da rinominare (obbligatorio).

  Se il vecchio nome contiene dei punti, non RenameField funzionerà a meno che non gli si posizionino dei backtick attorno (). ``` Ad esempio, per sostituire `this.old.name` con`thisNewName`, dovresti chiamare RenameField come segue:

  ```
  newDyF = RenameField(oldDyF, "`this.old.name`", "thisNewName")
  ```

- `new_name`: il nuovo nome, incluso il percorso completo (obbligatorio).
- `transformation_ctx`: una stringa univoca utilizzata per identificare informazioni sullo stato (opzionale).
- `info`: una stringa associata a errori nella trasformazione (opzionale).
- `stageThreshold`: il numero massimo di errori che si possono verificare nella trasformazione prima che venga arrestata (facoltativo). Il valore di default è zero.
- `totalThreshold`: il numero massimo di errori che si possono verificare in totale prima che l'elaborazione venga arrestata (facoltativo). Il valore di default è zero.

apply(cls, *args, **kwargs)

Ereditato da `GlueTransform` apply.

name(cls)

Ereditato da `GlueTransform` name.

describeArgs(cls)

Ereditato da `GlueTransform` describeArgs.

describeReturn(cls)

Ereditato da `GlueTransform` describeReturn.

describeTransform(cls)

Ereditato da `GlueTransform` describeTransform.

describeErrors(cls)

Ereditato da `GlueTransform` describeErrors.

describe(cls)

Ereditato da `GlueTransform` describe.

ResolveChoice classe

Risolve un tipo di scelta all'interno di un `DynamicFrame`.

Esempio

Ti consigliamo di utilizzare il metodo `DynamicFrame.resolveChoice()` per gestire campi che contengono più tipi in un `DynamicFrame`. Per visualizzare un esempio di codice, consulta Esempio: utilizzare resolveChoice per gestire una colonna che contiene più tipi.

Metodi

- __call__
- apply
- name
- describeArgs
- describeReturn
- describeTransform
- describeErrors
- describe

__call__(frame, specs = none, choice = "", transformation_ctx = "", info = "", stageThreshold = 0, totalThreshold = 0)

Fornisce informazioni per risolvere tipi ambigui all'interno di un `DynamicFrame`. Restituisce il `DynamicFrame` risultante.

- `frame`: il `DynamicFrame` in cui risolvere il tipo di scelta (obbligatorio).
- `specs`: elenco di ambiguità specifiche da risolvere, ognuna sotto forma di tupla:`(path, action)`. Il valore `path` identifica un elemento ambiguo specifico e il valore `action` identifica la soluzione corrispondente.

  Può essere utilizzato solo uno dei parametri `spec` e `choice`. Se il parametro `spec` non è `None`, allora il parametro `choice` deve essere una stringa vuota. Viceversa, se `choice` non è una stringa vuota, allora il parametro `spec` deve essere `None`. Se non viene fornito alcun parametro, AWS Glue tenta di analizzare lo schema e di utilizzarlo per risolvere le ambiguità.

  La parte `action` di una tupla `specs` può specificare una delle quattro strategie di risoluzione possibili:

  - `cast`: consente di specificare un tipo verso cui trasmettere (ad esempio, `cast:int`).
  - `make_cols`: risolve una potenziale ambiguità appiattendo i dati. Ad esempio, se `columnA` è un `int` o una `string`, la soluzione consiste nel produrre due colonne denominate `columnA_int` e `columnA_string` nel `DynamicFrame` risultante.
  - `make_struct`: risolve una potenziale ambiguità utilizzando una struttura per rappresentare i dati. Ad esempio, se i dati in una colonna sono un `int` o una `string`, utilizzando l'operazione `make_struct` viene prodotta una colonna di strutture nel `DynamicFrame` risultante, ognuna contenente sia un `int` che una `string`.
  - `project`: risolve un potenziale ambiguità conservando solo i valori di un tipo specificato nel `DynamicFrame` risultante. Ad esempio, se i dati in una colonna `ChoiceType` possono essere di tipo `int` o `string`, specificando un'operazione `project:string` si rimuovono i valori dal `DynamicFrame` risultante che non sono di tipo `string`.

  Se il `path` identifica un array, inserisci parentesi quadre vuote dopo il nome dell'array per evitare ambiguità. Ad esempio, supponiamo che tu stia lavorando con dati strutturati nel seguente modo:

  ```
  "myList": [
    { "price": 100.00 },
    { "price": "$100.00" }
  ]
  ```

  Puoi selezionare la versione numerica invece di quella di stringa del prezzo impostando il `path` su `"myList[].price"` e la `action` su `"cast:double"`.
- `choice`: l'operazione di risoluzione di default se il parametro `specs` è `None`. Se il parametro `specs` non è `None`, allora deve essere impostato solo su una stringa vuota.

Oltre alle operazioni elencate in precedenza per `specs`, questo argomento supporta anche l'operazione seguente:

  - `MATCH_CATALOG`: tenta di trasmettere ogni `ChoiceType` al tipo corrispondente nella tabella del catalogo specificata.
- `database`— Il database AWS Glue Data Catalog da utilizzare con la `MATCH_CATALOG` scelta (obbligatorio per`MATCH_CATALOG`).
- `table_name`— Il nome della tabella AWS Glue Data Catalog da utilizzare con l'`MATCH_CATALOG`azione (obbligatorio per`MATCH_CATALOG`).
- `transformation_ctx`: una stringa univoca utilizzata per identificare informazioni sullo stato (opzionale).
- `info`: una stringa associata a errori nella trasformazione (opzionale).
- `stageThreshold`: il numero massimo di errori che si possono verificare nella trasformazione prima che venga arrestata (facoltativo). Il valore di default è zero.
- `totalThreshold`: il numero massimo di errori che si possono verificare in totale prima che l'elaborazione venga arrestata (facoltativo). Il valore di default è zero.

apply(cls, *args, **kwargs)

Ereditato da `GlueTransform` apply.

name(cls)

Ereditato da `GlueTransform` name.

describeArgs(cls)

Ereditato da `GlueTransform` describeArgs.

describeReturn(cls)

Ereditato da `GlueTransform` describeReturn.

describeTransform(cls)

Ereditato da `GlueTransform` describeTransform.

describeErrors(cls)

Ereditato da `GlueTransform` describeErrors.

describe(cls)

Ereditato da `GlueTransform` describe.

SelectFields classe

La classe `SelectFields` crea un nuovo `DynamicFrame` da un `DynamicFrame` esistente e mantiene solo i campi specificati. `SelectFields` fornisce funzionalità simili a quelle di un'istruzione `SELECT` SQL.

Esempio

Ti consigliamo di utilizzare il metodo `DynamicFrame.select_fields()` per selezionare i campi da `DynamicFrame`. Per visualizzare un esempio di codice, consulta Esempio: usa select_fields per creare un nuovo `DynamicFrame` con i campi scelti.

Metodi

- __call__
- apply
- name
- describeArgs
- describeReturn
- describeTransform
- describeErrors
- Describe

__call__ (frame, percorsi, transformation_ctx = "", info = "", stageThreshold = 0, totalThreshold = 0)

Ottiene i campi (nodi) in un `DynamicFrame`.

- `frame`: il `DynamicFrame` in cui selezionare i campi (obbligatorio).
- `paths`: un elenco di percorsi completi ai campi da selezionare (obbligatorio).
- `transformation_ctx`: una stringa univoca utilizzata per identificare informazioni sullo stato (opzionale).
- `info`: una stringa associata a errori nella trasformazione (facoltativo).
- `stageThreshold`: il numero massimo di errori che si possono verificare nella trasformazione prima che venga arrestata (facoltativo). Il valore di default è zero.

- `totalThreshold`: il numero massimo di errori che si possono verificare in totale prima che l'elaborazione venga arrestata (facoltativo). Il valore di default è zero.

Restituisce un nuovo `DynamicFrame` contenente solo i campi specificati.

apply(cls, *args, **kwargs)

Ereditato da `GlueTransform` apply.

name(cls)

Ereditato da `GlueTransform` name.

describeArgs(cls)

Ereditato da `GlueTransform` describeArgs.

describeReturn(cls)

Ereditato da `GlueTransform` describeReturn.

describeTransform(cls)

Ereditato da `GlueTransform` describeTransform.

describeErrors(cls)

Ereditato da `GlueTransform` describeErrors.

describe(cls)

Ereditato da `GlueTransform` describe.

SelectFromCollection classe

Seleziona un `DynamicFrame` in una `DynamicFrameCollection`.

Esempio

Questo esempio utilizza `SelectFromCollection` per selezionare un `DynamicFrame` da una `DynamicFrameCollection`.

Set di dati di esempio

L'esempio seleziona due `DynamicFrames` da una `DynamicFrameCollection` chiamata `split_rows_collection`. Di seguito è riportato un elenco di chiavi in `split_rows_collection`.

```
dict_keys(['high', 'low'])
```

### Esempio di codice

```
# Example: Use SelectFromCollection to select
# DynamicFrames from a DynamicFrameCollection

from pyspark.context import SparkContext
from awsglue.context import GlueContext
from awsglue.transforms import SelectFromCollection

# Create GlueContext
sc = SparkContext.getOrCreate()
glueContext = GlueContext(sc)

# Select frames and inspect entries
frame_low = SelectFromCollection.apply(dfc=split_rows_collection, key="low")
frame_low.toDF().show()

frame_high = SelectFromCollection.apply(dfc=split_rows_collection, key="high")
frame_high.toDF().show()
```

### Output

```
+---+-----+-----------------------+-----------------------+
| id|index|contact_details.val.type|contact_details.val.value|
+---+-----+-----------------------+-----------------------+
|  1|    0|                    fax|           202-225-3307|
|  1|    1|                  phone|           202-225-5731|
|  2|    0|                    fax|           202-225-3307|
|  2|    1|                  phone|           202-225-5731|
|  3|    0|                    fax|           202-225-3307|
|  3|    1|                  phone|           202-225-5731|
|  4|    0|                    fax|           202-225-3307|
|  4|    1|                  phone|           202-225-5731|
|  5|    0|                    fax|           202-225-3307|
|  5|    1|                  phone|           202-225-5731|
|  6|    0|                    fax|           202-225-3307|
```

```
|  6|    1|                  phone|             202-225-5731|
|  7|    0|                    fax|             202-225-3307|
|  7|    1|                  phone|             202-225-5731|
|  8|    0|                    fax|             202-225-3307|
|  8|    1|                  phone|             202-225-5731|
|  9|    0|                    fax|             202-225-3307|
|  9|    1|                  phone|             202-225-5731|
| 10|    0|                    fax|             202-225-6328|
| 10|    1|                  phone|             202-225-4576|
+---+-----+----------------------+-------------------------+
only showing top 20 rows

+---+-----+----------------------+-------------------------+
| id|index|contact_details.val.type|contact_details.val.value|
+---+-----+----------------------+-------------------------+
| 11|    0|                    fax|             202-225-6328|
| 11|    1|                  phone|             202-225-4576|
| 11|    2|                twitter|           RepTrentFranks|
| 12|    0|                    fax|             202-225-6328|
| 12|    1|                  phone|             202-225-4576|
| 12|    2|                twitter|           RepTrentFranks|
| 13|    0|                    fax|             202-225-6328|
| 13|    1|                  phone|             202-225-4576|
| 13|    2|                twitter|           RepTrentFranks|
| 14|    0|                    fax|             202-225-6328|
| 14|    1|                  phone|             202-225-4576|
| 14|    2|                twitter|           RepTrentFranks|
| 15|    0|                    fax|             202-225-6328|
| 15|    1|                  phone|             202-225-4576|
| 15|    2|                twitter|           RepTrentFranks|
| 16|    0|                    fax|             202-225-6328|
| 16|    1|                  phone|             202-225-4576|
| 16|    2|                twitter|           RepTrentFranks|
| 17|    0|                    fax|             202-225-6328|
| 17|    1|                  phone|             202-225-4576|
+---+-----+----------------------+-------------------------+
only showing top 20 rows
```

## Metodi

- __call__
- apply

- name
- describeArgs
- describeReturn
- describeTransform
- describeErrors
- describe

__call__ (dfc, chiave, transformation_ctx = "")

Ottiene un `DynamicFrame` da una `DynamicFrameCollection`.

- `dfc`: la `DynamicFrameCollection` da cui il `DynamicFrame` deve essere selezionato (obbligatorio).
- `key`: la chiave del `DynamicFrame` da selezionare (obbligatorio).
- `transformation_ctx`: una stringa univoca utilizzata per identificare informazioni sullo stato (opzionale).

apply(cls, *args, **kwargs)

Ereditato da `GlueTransform` apply.

name(cls)

Ereditato da `GlueTransform` name.

describeArgs(cls)

Ereditato da `GlueTransform` describeArgs.

describeReturn(cls)

Ereditato da `GlueTransform` describeReturn.

describeTransform(cls)

Ereditato da `GlueTransform` describeTransform.

describeErrors(cls)

Ereditato da `GlueTransform` describeErrors.

describe(cls)

Ereditato da `GlueTransform` describe.

Classe Simplify_ddb_json

Semplifica le colonne nidificate in un `DynamicFrame` che si trovano specificamente nella struttura JSON DynamoDB e restituisce un nuovo `DynamicFrame` semplificato.

Esempio

Si consiglia di utilizzare il metodo `DynamicFrame.simplify_ddb_json()` per semplificare le colonne nidificate in un ambiente `DynamicFrame` che si trova specificamente nella struttura JSON DynamoDB. Per visualizzare un esempio di codice, consultare Esempio: utilizzo di simply_ddb_json per invocare una semplificazione JSON DynamoDB.

Classe Spigot

Scrive record di esempio in una destinazione specificata per aiutarti a verificare le trasformazioni eseguite dal processo AWS Glue.

Esempio

Si consiglia di utilizzare il metodo `DynamicFrame.spigot()` per scrivere un sottoinsieme di record da un `DynamicFrame` a una destinazione specificata. Per visualizzare un esempio di codice, consulta Esempio: usa lo spigot per scrivere campi di esempio da `DynamicFrame` ad Amazon S3.

Metodi

- __call__
- apply
- name
- describeArgs
- describeReturn
- describeTransform
- describeErrors
- describe

__call__(frame, path, options, transformation_ctx = "")

Scrive record di esempio in una destinazione specificata durante una trasformazione.

- `frame`: il `DynamicFrame` da sottoporre allo spigot (obbligatorio).
- `path`: il percorso della destinazione in cui scrivere (obbligatorio).
- `options`: coppie chiave-valore JSON che specificano opzioni (opzionale). L'opzione `"topk"` specifica che devono essere scritti i primi record k. L'opzione `"prob"` specifica la probabilità (sotto forma di valore decimale) di selezione di un dato record. Puoi usarlo per selezionare i record da scrivere.
- `transformation_ctx`: una stringa univoca utilizzata per identificare informazioni sullo stato (opzionale).

apply(cls, *args, **kwargs)

Ereditato da `GlueTransform` apply

name(cls)

Ereditato da `GlueTransform` name

describeArgs(cls)

Ereditato da `GlueTransform` describeArgs

describeReturn(cls)

Ereditato da `GlueTransform` describeReturn

describeTransform(cls)

Ereditato da `GlueTransform` describeTransform

describeErrors(cls)

Ereditato da `GlueTransform` describeErrors

describe(cls)

Ereditato da `GlueTransform` describe

SplitFields classe

Divide un `DynamicFrame` in due, in base ai campi specificati.

Esempio

Ti consigliamo di utilizzare il metodo `DynamicFrame.split_fields()` per dividere i campi in un `DynamicFrame`. Per visualizzare un esempio di codice, consulta Esempio: usare split_fields per dividere i campi selezionati in un campo separato `DynamicFrame`.

Metodi

- __call__
- apply
- name
- describeArgs
- describeReturn
- describeTransform
- describeErrors
- describe

__call__(frame, paths, name1 = none, name2 = none, transformation_ctx = "", info = "", stageThreshold = 0, totalThreshold = 0)

Divide uno o più campi in un `DynamicFrame` disattivato in un nuovo `DynamicFrame` e crea un altro nuovo `DynamicFrame` che contiene i campi che rimangono.

- `frame`: il `DynamicFrame` di origine da dividere in due nuovi elementi (obbligatorio).
- `paths`: un elenco di percorsi completi per campi da suddividere (obbligatorio).
- `name1`: il nome da assegnare al `DynamicFrame` che contiene i campi che devono essere separati (facoltativo). Se non viene fornito nessun nome, il nome del frame di origine viene utilizzato con "1" aggiunto.
- `name2`: il nome da assegnare al `DynamicFrame` che contiene i campi che rimangono una volta che i campi specificati vengono suddivisi (facoltativo). Se non viene fornito alcun nome, il nome del frame di origine viene utilizzato con "2" aggiunto.

- `transformation_ctx`: una stringa univoca utilizzata per identificare informazioni sullo stato (opzionale).
- `info`: una stringa associata a errori nella trasformazione (opzionale).
- `stageThreshold`: il numero massimo di errori che si possono verificare nella trasformazione prima che venga arrestata (facoltativo). Il valore di default è zero.
- `totalThreshold`: il numero massimo di errori che si possono verificare in totale prima che l'elaborazione venga arrestata (facoltativo). Il valore di default è zero.

apply(cls, *args, **kwargs)

Ereditato da `GlueTransform` apply.

name(cls)

Ereditato da `GlueTransform` name.

describeArgs(cls)

Ereditato da `GlueTransform` describeArgs.

describeReturn(cls)

Ereditato da `GlueTransform` describeReturn.

describeTransform(cls)

Ereditato da `GlueTransform` describeTransform.

describeErrors(cls)

Ereditato da `GlueTransform` describeErrors.

describe(cls)

Ereditato da `GlueTransform` describe.

SplitRows classe

Crea una `DynamicFrameCollection` che contiene due `DynamicFrames`. Un `DynamicFrame` contiene solo le righe specificate da suddividere e l'altro con tutte le righe rimanenti.

Esempio

Ti consigliamo di utilizzare il metodo `DynamicFrame.split_rows()` per dividere le righe in un `DynamicFrame`. Per visualizzare un esempio di codice, consulta Esempio: usare split_rows per dividere le righe in un `DynamicFrame`.

Metodi

- __call__
- apply
- name
- describeArgs
- describeReturn
- describeTransform
- describeErrors
- describe

__call__(frame, comparison_dict, name1="frame1", name2="frame2", transformation_ctx = "", info = none, stageThreshold = 0, totalThreshold = 0)

Suddivide una o più righe di un `DynamicFrame` in un nuovo `DynamicFrame`.

- `frame`: il `DynamicFrame` di origine da dividere in due nuovi elementi (obbligatorio).
- `comparison_dict`: un dizionario in cui la chiave è il percorso completo verso una colonna e il valore è un altro dizionario per la mappatura di comparatori rispetto a valori con i quali vengono confrontati i valori di colonna. Ad esempio, `{"age": {">": 10, "<": 20}}` divide le righe in cui il valore di "age" (età) è compreso tra 10 e 20 (non inclusi) dalle righe dove "age" non è compreso in tale intervallo (obbligatorio).
- `name1`: il nome da assegnare al `DynamicFrame` che contiene le righe da dividere (opzionale).
- `name2`: il nome da assegnare al `DynamicFrame` che contiene le righe che rimangono dopo la divisione delle righe specificate (opzionale).
- `transformation_ctx`: una stringa univoca utilizzata per identificare informazioni sullo stato (opzionale).
- `info`: una stringa associata a errori nella trasformazione (opzionale).
- `stageThreshold`: il numero massimo di errori che si possono verificare nella trasformazione prima che venga arrestata (facoltativo). Il valore di default è zero.

- `totalThreshold`: il numero massimo di errori che si possono verificare in totale prima che l'elaborazione venga arrestata (facoltativo). Il valore di default è zero.

apply(cls, *args, **kwargs)

Ereditato da `GlueTransform` apply.

name(cls)

Ereditato da `GlueTransform` name.

describeArgs(cls)

Ereditato da `GlueTransform` describeArgs.

describeReturn(cls)

Ereditato da `GlueTransform` describeReturn.

describeTransform(cls)

Ereditato da `GlueTransform` describeTransform.

describeErrors(cls)

Ereditato da `GlueTransform` describeErrors.

describe(cls)

Ereditato da `GlueTransform` describe.

Classe unbox

Sottopone a conversione unboxing (riformatta) un campo stringa in un oggetto `DynamicFrame`.

Esempio

Ti consigliamo di utilizzare il metodo `DynamicFrame.unbox()` per eseguire la conversione unboxing di un campo in un `DynamicFrame`. Per visualizzare un esempio di codice, consulta Esempio: usare unbox per decomprimere un campo di stringa in un campo struct.

Metodi

- __call__

- apply
- name
- describeArgs
- describeReturn
- describeTransform
- describeErrors
- describe

__call__(frame, path, format, transformation_ctx = "", info="", stageThreshold=0, totalThreshold=0, **options)

Sottopone a conversione unboxing un campo stringa in un oggetto `DynamicFrame`.

- `frame`: il `DynamicFrame` nel quale sottoporre a conversione unboxing un campo (obbligatorio).
- `path`: il percorso completo del `StringNode` da cancellare (obbligatorio).
- `format`: una specifica del formato (facoltativa). Viene usata per una connessione Amazon S3 o AWS Glue che supporta più formati. Consulta Opzioni del formato dati per input e output in AWS Glue per Spark per informazioni sui formati supportati.
- `transformation_ctx`: una stringa univoca utilizzata per identificare informazioni sullo stato (opzionale).
- `info`: una stringa associata a errori nella trasformazione (opzionale).
- `stageThreshold`: il numero massimo di errori che si possono verificare nella trasformazione prima che venga arrestata (facoltativo). Il valore di default è zero.
- `totalThreshold`: il numero massimo di errori che si possono verificare in totale prima che l'elaborazione venga arrestata (facoltativo). Il valore di default è zero.
- `separator`: un token di separazione (opzionale).
- `escaper`: un token di escape (opzionale).
- `skipFirst`: `True` se la prima riga di dati deve essere ignorata oppure `False` se non deve essere ignorata (opzionale).
- withSchema: una stringa che contiene lo schema per i dati da rimuovere (opzionale). Deve essere sempre creato utilizzando `StructType.json`.
- `withHeader`: `True` se i dati decompressi includono un'intestazione oppure `False` se non la includono (opzionale).

apply(cls, *args, **kwargs)

Ereditato da `GlueTransform` apply.

name(cls)

Ereditato da `GlueTransform` name.

describeArgs(cls)

Ereditato da `GlueTransform` describeArgs.

describeReturn(cls)

Ereditato da `GlueTransform` describeReturn.

describeTransform(cls)

Ereditato da `GlueTransform` describeTransform.

describeErrors(cls)

Ereditato da `GlueTransform` describeErrors.

describe(cls)

Ereditato da `GlueTransform` describe.

UnnestFrame classe

Annulla l'annidamento di un `DynamicFrame`, livella gli oggetti nidificati a elementi di primo livello e genera chiavi di join per oggetti di array.

Esempio

Ti consigliamo di utilizzare il metodo `DynamicFrame.unnest()` per livellare le strutture annidate in un`DynamicFrame`. Per visualizzare un esempio di codice, consulta Esempio: usare unnest per trasformare i campi annidati in campi di primo livello.

Metodi

- __call__

- apply
- name
- describeArgs
- describeReturn
- describeTransform
- describeErrors
- describe

__call__(frame, transformation_ctx = "", info="", stageThreshold=0, totalThreshold=0)

Annulla l'annidamento di un `DynamicFrame`, livella gli oggetti nidificati a elementi di primo livello e genera chiavi di join per oggetti di array.

- `frame`: il `DynamicFrame` per annullare l'annidamento (obbligatorio).
- `transformation_ctx`: una stringa univoca utilizzata per identificare informazioni sullo stato (opzionale).
- `info`: una stringa associata a errori nella trasformazione (opzionale).
- `stageThreshold`: il numero massimo di errori che si possono verificare nella trasformazione prima che venga arrestata (facoltativo). Il valore di default è zero.
- `totalThreshold`: il numero massimo di errori che si possono verificare in totale prima che l'elaborazione venga arrestata (facoltativo). Il valore di default è zero.

apply(cls, *args, **kwargs)

Ereditato da `GlueTransform` apply.

name(cls)

Ereditato da `GlueTransform` name.

describeArgs(cls)

Ereditato da `GlueTransform` describeArgs.

describeReturn(cls)

Ereditato da `GlueTransform` describeReturn.

describeTransform(cls)

Ereditato da `GlueTransform` describeTransform.

describeErrors(cls)

Ereditato da `GlueTransform` describeErrors.

describe(cls)

Ereditato da `GlueTransform` describe.

FlagDuplicatesInColumn classe

La trasformazione `FlagDuplicatesInColumn` restituisce una nuova colonna che contiene un valore specificato in ogni riga che indica se la il valore nella colonna di origine di tale riga corrisponde a un valore in una riga precedente della colonna di origine. Quando vengono trovate corrispondenze, i valori vengono contrassegnati come duplicati. L'occorrenza iniziale non viene contrassegnata poiché non corrisponde a una riga precedente.

Esempio

```
from pyspark.context import SparkContext
from pyspark.sql import SparkSession
from awsgluedi.transforms import *

sc = SparkContext()
spark = SparkSession(sc)

datasource1 = spark.read.json("s3://${BUCKET}/json/zips/raw/data")

try:
    df_output = column.FlagDuplicatesInColumn.apply(
        data_frame=datasource1,
        spark_context=sc,
        source_column="city",
        target_column="flag_col",
        true_string="True",
        false_string="False"
    )
except:
    print("Unexpected Error happened ")
    raise
```

Output

La `FlagDuplicatesInColumn` trasformazione aggiungerà una nuova colonna `flag_col` al `df_output`. DataFrame Questa colonna conterrà una valore di stringa che indica se per la riga corrispondente è presente o meno un valore duplicato nella colonna `city`. Se per una riga è presente un valore `city` duplicato, `flag_col` conterrà il valore `true_string` "True". Se per una riga è presente un valore `city` univoco, `flag_col` conterrà il valore `false_string` "False".

La `df_output` risultante conterrà tutte le colonne del `datasource1` originale, più la colonna DataFrame `flag_col` aggiuntiva che indica i valori `city` duplicati. DataFrame

Metodi

- __call__
- apply
- name
- describeArgs
- describeReturn
- describeTransform
- describeErrors
- describe

__call__(spark_context, data_frame, source_column, target_column, true_string=DEFAULT_TRUE_STRING, false_string=DEFAULT_FALSE_STRING)

La trasformazione `FlagDuplicatesInColumn` restituisce una nuova colonna che contiene un valore specificato in ogni riga che indica se la il valore nella colonna di origine di tale riga corrisponde a un valore in una riga precedente della colonna di origine. Quando vengono trovate corrispondenze, i valori vengono contrassegnati come duplicati. L'occorrenza iniziale non viene contrassegnata poiché non corrisponde a una riga precedente.

- `source_column`: nome della colonna di origine.
- `target_column`: nome della colonna di destinazione.
- `true_string`: stringa da inserire nella colonna di destinazione quando per un valore della colonna di origine è presente un duplicato di un valore precedente in tale colonna.

- `false_string`: stringa da inserire nella colonna di destinazione quando un valore della colonna di origine è diverso dai valori precedenti in tale colonna.

apply(cls, *args, **kwargs)

Ereditato da `GlueTransform` apply.

name(cls)

Ereditato da `GlueTransform` name.

describeArgs(cls)

Ereditato da `GlueTransform` describeArgs.

describeReturn(cls)

Ereditato da `GlueTransform` describeReturn.

describeTransform(cls)

Ereditato da `GlueTransform` describeTransform.

describeErrors(cls)

Ereditato da `GlueTransform` describeErrors.

describe(cls)

Ereditato da `GlueTransform` describe.

FormatPhoneNumber classe

La trasformazione `FormatPhoneNumber` restituisce una colonna in cui una stringa con numero di telefono viene convertita in un valore formattato.

Esempio

```
from pyspark.context import SparkContext
from pyspark.sql import SparkSession
from awsgluedi.transforms import *

sc = SparkContext()
spark = SparkSession(sc)
```

```
input_df = spark.createDataFrame(
    [
        ("408-341-5669",),
        ("4083415669",)
    ],
    ["phone"],
)

try:
    df_output = column_formatting.FormatPhoneNumber.apply(
        data_frame=input_df,
        spark_context=sc,
        source_column="phone",
        default_region="US"
    )
    df_output.show()
except:
    print("Unexpected Error happened ")
    raise
```

Output

L'output sarà:

```
```
+---------------+
| phone|
+---------------+
|(408) 341-5669|
|(408) 341-5669|
+---------------+
```
```

La trasformazione `FormatPhoneNumber` assegna `source_column` come `“phone”` e `default_region` come `“US”`.

La trasformazione formatta correttamente entrambi i numeri di telefono, indipendentemente dal formato iniziale, applicando il formato standard statunitense `(408) 341-5669`.

Metodi

- __call__

- apply
- name
- describeArgs
- describeReturn
- describeTransform
- describeErrors
- describe

__call__(spark_context, data_frame, source_column, phone_number_format=None, default_region=None, default_region_column=None)

La trasformazione `FormatPhoneNumber` restituisce una colonna in cui una stringa con numero di telefono viene convertita in un valore formattato.

- `source_column`: il nome di una colonna esistente.
- `phone_number_format`: il formato in cui convertire il numero di telefono. Se non viene specificato alcun formato, il valore predefinito è `E.164`, un formato standard per i numeri di telefono riconosciuto a livello internazionale. I valori validi includono quanto segue:
  - E164 (omettere il punto dopo la E)
- `default_region`: un codice regionale valido costituito da due o tre lettere maiuscole che specifica la regione del numero di telefono quando non è presente alcun prefisso nel numero stesso. Può essere indicato al massimo uno di `defaultRegion` o `defaultRegionColumn`.
- `default_region_column`: il nome di una colonna del tipo di dati avanzato `Country`. Il codice regionale della colonna specificata viene utilizzato per determinare il prefisso del numero di telefono quando non è presente nel numero stesso. Può essere indicato al massimo uno di `defaultRegion` o `defaultRegionColumn`.

apply(cls, *args, **kwargs)

Ereditato da `GlueTransform` apply.

name(cls)

Ereditato da `GlueTransform` name.

describeArgs(cls)

Ereditato da `GlueTransform` describeArgs.

describeReturn(cls)

Ereditato da `GlueTransform` describeReturn.

describeTransform(cls)

Ereditato da `GlueTransform` describeTransform.

describeErrors(cls)

Ereditato da `GlueTransform` describeErrors.

describe(cls)

Ereditato da `GlueTransform` describe.

FormatCase classe

La trasformazione `FormatCase` modifica ogni stringa di una colonna applicando la formattazione specificata.

Esempio

```
from pyspark.context import SparkContext
from pyspark.sql import SparkSession
from awsgluedi.transforms import *

sc = SparkContext()
spark = SparkSession(sc)

datasource1 = spark.read.json("s3://${BUCKET}/json/zips/raw/data")

try:
    df_output = data_cleaning.FormatCase.apply(
        data_frame=datasource1,
        spark_context=sc,
        source_column="city",
        case_type="LOWER"
    )
```

```
except:
    print("Unexpected Error happened ")
    raise
```

Output

La trasformazione `FormatCase` convertirà i valori nella colonna `city` in lettere minuscole in base al parametro `case_type="LOWER"`. Il `df_output` risultante DataFrame conterrà tutte le colonne dell'originale `datasource1`, ma con i valori della colonna DataFrame `city` in minuscolo.

Metodi

- __call__
- apply
- name
- describeArgs
- describeReturn
- describeTransform
- describeErrors
- describe

__call__(spark_context, data_frame, source_column, case_type)

La trasformazione `FormatCase` modifica ogni stringa di una colonna applicando la formattazione specificata.

- `source_column`: il nome di una colonna esistente.
- `case_type`: i tipi di formattazione supportati sono `CAPITAL`, `LOWER`, `UPPER`, `SENTENCE`.

apply(cls, *args, **kwargs)

Ereditato da `GlueTransform` apply.

name(cls)

Ereditato da `GlueTransform` name.

describeArgs(cls)

Ereditato da `GlueTransform` describeArgs.

describeReturn(cls)

Ereditato da `GlueTransform` describeReturn.

describeTransform(cls)

Ereditato da `GlueTransform` describeTransform.

describeErrors(cls)

Ereditato da `GlueTransform` describeErrors.

describe(cls)

Ereditato da `GlueTransform` describe.

FillWithMode classe

La trasformazione `FillWithMode` formatta una colonna secondo il formato del numero di telefono specificato. È inoltre possibile specificare la logica di spareggio, in cui alcuni dei valori sono identici. Considerare, ad esempio, i seguenti valori: `1 2 2 3 3 4`

Un modeType `MINIMUM` porta `FillWithMode` a restituire 2 come valore della modalità. Se modeType è `MAXIMUM`, la modalità è 3. Per `AVERAGE`, la modalità è 2,5.

Esempio

```
from awsglue.context import *
from pyspark.sql import SparkSession
from awsgluedi.transforms import *

sc = SparkContext()
spark = SparkSession(sc)

input_df = spark.createDataFrame(
    [
        (105.111, 13.12),
        (1055.123, 13.12),
```

```
        (None, 13.12),
        (13.12, 13.12),
        (None, 13.12),
    ],
    ["source_column_1", "source_column_2"],
)

try:
    df_output = data_quality.FillWithMode.apply(
        data_frame=input_df,
        spark_context=sc,
        source_column="source_column_1",
        mode_type="MAXIMUM"
    )
    df_output.show()
except:
    print("Unexpected Error happened ")
    raise
```

Output

L'output del codice dato sarà:

```
```
+---------------+---------------+
|source_column_1|source_column_2|
+---------------+---------------+
| 105.111| 13.12|
| 1055.123| 13.12|
| 1055.123| 13.12|
| 13.12| 13.12|
| 1055.123| 13.12|
+---------------+---------------+
```
```

La `FillWithMode` trasformazione dal modulo `awsglue.data_quality` viene applicata a `input_df`. DataFrame Sostituisce i valori `null` nella colonna `source_column_1` con il valore massimo (`mode_type="maximum"`) dei valori non null in quella colonna.

In questo caso, il valore massimo nella colonna `source_column_1` è `1055.123`. Pertanto, i valori `null` in vengono sostituiti da `1055.123` nell'output `df_output`. `source_column_1` DataFrame

Metodi

- __call__
- apply
- name
- describeArgs
- describeReturn
- describeTransform
- describeErrors
- describe

__call__(spark_context, data_frame, source_column, mode_type)

La trasformazione `FillWithMode` modifica la formattazione delle stringhe in una colonna.

- `source_column`: il nome di una colonna esistente.
- `mode_type`: come risolvere i valori pari nei dati. Questo valore deve essere uno dei seguenti: `MINIMUM`, `NONE`, `AVERAGE` o `MAXIMUM`.

apply(cls, *args, **kwargs)

Ereditato da `GlueTransform` apply.

name(cls)

Ereditato da `GlueTransform` name.

describeArgs(cls)

Ereditato da `GlueTransform` describeArgs.

describeReturn(cls)

Ereditato da `GlueTransform` describeReturn.

describeTransform(cls)

Ereditato da `GlueTransform` describeTransform.

describeErrors(cls)

Ereditato da `GlueTransform` describeErrors.

describe(cls)

Ereditato da `GlueTransform` describe.

FlagDuplicateRows classe

La trasformazione `FlagDuplicateRows` restituisce una nuova colonna che contiene un valore specificato in ogni riga che indica se la stessa corrisponde esattamente a una riga precedente nel set di dati. Quando vengono trovate corrispondenze, i valori vengono contrassegnati come duplicati. L'occorrenza iniziale non viene contrassegnata poiché non corrisponde a una riga precedente.

Esempio

```
from pyspark.context import SparkContext
from pyspark.sql import SparkSession
from awsgluedi.transforms import *

sc = SparkContext()
spark = SparkSession(sc)

input_df = spark.createDataFrame(
    [
        (105.111, 13.12),
        (13.12, 13.12),
        (None, 13.12),
        (13.12, 13.12),
        (None, 13.12),
    ],
    ["source_column_1", "source_column_2"],
)

try:
    df_output = data_quality.FlagDuplicateRows.apply(
        data_frame=input_df,
        spark_context=sc,
        target_column="flag_row",
        true_string="True",
        false_string="False",
        target_index=1
```

```
    )
except:
    print("Unexpected Error happened ")
    raise
```

Output

L'output sarà PySpark DataFrame con una colonna aggiuntiva `flag_row` che indica se una riga è duplicata o meno, in base alla `source_column_1` colonna. Il `df_output` DataFrame risultante conterrà le seguenti righe:

```
```
+---------------+---------------+--------+
|source_column_1|source_column_2|flag_row|
+---------------+---------------+--------+
| 105.111| 13.12| False|
| 13.12| 13.12| True|
| null| 13.12| True|
| 13.12| 13.12| True|
| null| 13.12| True|
+---------------+---------------+--------+
```
```

La colonna `flag_row` indica se una riga è duplicata o meno. Il valore di `true_string` è impostato su "True" e quello di `false_string` su "False". Il `target_index` è impostato su 1, il che significa che la `flag_row` colonna verrà inserita nella seconda posizione (indice 1) nell'output. DataFrame

Metodi

- __call__
- apply
- name
- describeArgs
- describeReturn
- describeTransform
- describeErrors
- describe

__call__(spark_context, data_frame, target_column, true_string=DEFAULT_TRUE_STRING, false_string=DEFAULT_FALSE_STRING, target_index=None)

La trasformazione `FlagDuplicateRows` restituisce una nuova colonna che contiene un valore specificato in ogni riga che indica se la stessa corrisponde esattamente a una riga precedente nel set di dati. Quando vengono trovate corrispondenze, i valori vengono contrassegnati come duplicati. L'occorrenza iniziale non viene contrassegnata poiché non corrisponde a una riga precedente.

- `true_string`: valore da inserire se la riga corrisponde a una riga precedente.
- `false_string`: valore da inserire se la riga è univoca.
- `target_column`: nome della nuova colonna inserita nel set di dati.

apply(cls, *args, **kwargs)

Ereditato da `GlueTransform` apply.

name(cls)

Ereditato da `GlueTransform` name.

describeArgs(cls)

Ereditato da `GlueTransform` describeArgs.

describeReturn(cls)

Ereditato da `GlueTransform` describeReturn.

describeTransform(cls)

Ereditato da `GlueTransform` describeTransform.

describeErrors(cls)

Ereditato da `GlueTransform` describeErrors.

describe(cls)

Ereditato da `GlueTransform` describe.

### RemoveDuplicates classe

La trasformazione `RemoveDuplicates` elimina un'intera riga se viene rilevato un valore duplicato in una colonna di origine selezionata.

### Esempio

```
from pyspark.context import SparkContext
from pyspark.sql import SparkSession
from awsgluedi.transforms import *

sc = SparkContext()
spark = SparkSession(sc)

input_df = spark.createDataFrame(
    [
        (105.111, 13.12),
        (13.12, 13.12),
        (None, 13.12),
        (13.12, 13.12),
        (None, 13.12),
    ],
    ["source_column_1", "source_column_2"],
)

try:
    df_output = data_quality.RemoveDuplicates.apply(
        data_frame=input_df,
        spark_context=sc,
        source_column="source_column_1"
    )
except:
    print("Unexpected Error happened ")
    raise
```

### Output

L'output sarà un PySpark DataFrame con i duplicati rimossi in base alla `source_column_1` colonna. Il `df_output` DataFrame risultante conterrà le seguenti righe:

```
```

```
+---------------+---------------+
|source_column_1|source_column_2|
+---------------+---------------+
| 105.111| 13.12|
| 13.12| 13.12|
| null| 13.12|
+---------------+---------------+
```

Nota che le righe con `source_column_1` i valori `13.12` e `null` appaiono solo una volta nell'output DataFrame, poiché i duplicati sono stati rimossi in base alla colonna. `source_column_1`

Metodi

- __call__
- apply
- name
- describeArgs
- describeReturn
- describeTransform
- describeErrors
- describe

__call__(spark_context, data_frame, source_column)

La trasformazione `RemoveDuplicates` elimina un'intera riga se viene rilevato un valore duplicato in una colonna di origine selezionata.

- `source_column`: il nome di una colonna esistente.

apply(cls, *args, **kwargs)

Ereditato da `GlueTransform` apply.

name(cls)

Ereditato da `GlueTransform` name.

describeArgs(cls)

Ereditato da `GlueTransform` describeArgs.

describeReturn(cls)

Ereditato da `GlueTransform` describeReturn.

describeTransform(cls)

Ereditato da `GlueTransform` describeTransform.

describeErrors(cls)

Ereditato da `GlueTransform` describeErrors.

describe(cls)

Ereditato da `GlueTransform` describe.

MonthName classe

La trasformazione `MonthName` crea una nuova colonna che contiene il nome del mese, a partire da una stringa che rappresenta una data.

Esempio

```
from pyspark.context import SparkContext
from pyspark.sql import SparkSession
from awsgluedi.transforms import *

sc = SparkContext()
spark = SparkSession(sc)

spark.conf.set("spark.sql.legacy.timeParserPolicy", "LEGACY")

input_df = spark.createDataFrame(
    [
        ("20-2018-12",),
        ("2018-20-12",),
        ("20182012",),
        ("12202018",),
        ("20122018",),
        ("20-12-2018",),
```

```
        ("12/20/2018",),
        ("02/02/02",),
        ("02 02 2009",),
        ("02/02/2009",),
        ("August/02/2009",),
        ("02/june/2009",),
        ("02/2020/june",),
        ("2013-02-21 06:35:45.658505",),
        ("August 02 2009",),
        ("2013/02/21",),
        (None,),
    ],
    ["column_1"],
)

try:
    df_output = datetime_functions.MonthName.apply(
        data_frame=input_df,
        spark_context=sc,
        source_column="column_1",
        target_column="target_column"
    )
    df_output.show()
except:
    print("Unexpected Error happened ")
    raise
```

## Output

L'output sarà:

```
```
+------------+------------+
| column_1|target_column|
+------------+------------+
|20-2018-12 | December |
|2018-20-12 | null |
| 20182012| null |
| 12202018| null |
| 20122018| null |
|20-12-2018 | December |
|12/20/2018 | December |
| 02/02/02 | February |
```
```

```
|02 02 2009 | February |
|02/02/2009 | February |
|August/02/2009| August |
|02/june/2009| null |
|02/2020/june| null |
|2013-02-21 06:35:45.658505| February |
|August 02 2009| August |
| 2013/02/21| February |
| null | null |
+------------+------------+
```

La trasformazione `MonthName` assegna `source_column` come `“column_1”` e `target_column` come `“target_column”`. Tenta di estrarre il nome del mese dalle date/time stringhe nella colonna `"column_1"` e lo colloca nella colonna `"target_column"`. Se la date/time stringa è in un formato non riconosciuto o non può essere analizzata, il valore `"target_column"` è impostato su `null`.

La trasformazione estrae correttamente il nome del mese da vari date/time formati, come «20-12-2018", «20/12/2018", «02/02/2009", «2013-02-21 06:35:45.658 505" e «02 agosto 2009".

Metodi

- __call__
- apply
- name
- describeArgs
- describeReturn
- describeTransform
- describeErrors
- describe

__call__(spark_context, data_frame, target_column, source_column=None, value=None)

La trasformazione `MonthName` crea una nuova colonna che contiene il nome del mese, a partire da una stringa che rappresenta una data.

- `source_column`: il nome di una colonna esistente.
- `value`: una stringa di caratteri da valutare.

- `target_column`: un nome per la nuova colonna creata.

apply(cls, *args, **kwargs)

Ereditato da `GlueTransform` apply.

name(cls)

Ereditato da `GlueTransform` name.

describeArgs(cls)

Ereditato da `GlueTransform` describeArgs.

describeReturn(cls)

Ereditato da `GlueTransform` describeReturn.

describeTransform(cls)

Ereditato da `GlueTransform` describeTransform.

describeErrors(cls)

Ereditato da `GlueTransform` describeErrors.

describe(cls)

Ereditato da `GlueTransform` describe.

IsEven classe

La trasformazione `IsEven` restituisce un valore booleano in una nuova colonna che indica se la colonna o il valore di origine sono pari. Se la colonna o il valore di origine sono decimali, il risultato è false.

Esempio

```
from pyspark.context import SparkContext
from pyspark.sql import SparkSession
from awsgluedi.transforms import *
```

```
sc = SparkContext()
spark = SparkSession(sc)

input_df = spark.createDataFrame(
    [(5,), (0,), (-1,), (2,), (None,)],
    ["source_column"],
)

try:
    df_output = math_functions.IsEven.apply(
        data_frame=input_df,
        spark_context=sc,
        source_column="source_column",
        target_column="target_column",
        value=None,
        true_string="Even",
        false_string="Not even",
    )
    df_output.show()
except:
    print("Unexpected Error happened ")
    raise
```

Output

L'output sarà:

```
```
+------------+------------+
|source_column|target_column|
+------------+------------+
| 5| Not even|
| 0| Even|
| -1| Not even|
| 2| Even|
| null| null|
+------------+------------+
```
```

La trasformazione `IsEven` assegna `source_column` come “source_column” e `target_column` come “target_column”. Verifica se il valore nella `“source_column”` è pari o meno. Se il valore è pari, imposta il valore `“target_column”` su `true_string` “Even”. Se il valore è dispari, imposta il

valore `"target_column"` su `false_string` "Not even". Se il valore `source_column` è `null`, il valore `target_column` è impostato su `null`.

La trasformazione identifica correttamente i numeri pari (0 e 2) e imposta il valore `"target_column"` su "Even". Per i numeri dispari (5 e -1), imposta il valore `"target_column"` su "Not even". Per il valore `null` in `source_column`, il valore `target_column` è impostato su `null`.

Metodi

- __call__
- apply
- name
- describeArgs
- describeReturn
- describeTransform
- describeErrors
- describe

__call__(spark_context, data_frame, target_column, source_column=None, true_string=DEFAULT_TRUE_STRING, false_string=DEFAULT_FALSE_STRING, value=None)

La trasformazione `IsEven` restituisce un valore booleano in una nuova colonna che indica se la colonna o il valore di origine sono pari. Se la colonna o il valore di origine sono decimali, il risultato è false.

- `source_column`: il nome di una colonna esistente.
- `target_column`: il nome della nuova colonna da creare.
- `true_string`: una stringa che indica se il valore è pari.
- `false_string`: una stringa che indica se il valore non è pari.

apply(cls, *args, **kwargs)

Ereditato da `GlueTransform` apply.

name(cls)

Ereditato da `GlueTransform` name.

describeArgs(cls)

Ereditato da `GlueTransform` describeArgs.

describeReturn(cls)

Ereditato da `GlueTransform` describeReturn.

describeTransform(cls)

Ereditato da `GlueTransform` describeTransform.

describeErrors(cls)

Ereditato da `GlueTransform` describeErrors.

describe(cls)

Ereditato da `GlueTransform` describe.

CryptographicHash classe

La trasformazione `CryptographicHash` applica un algoritmo per eseguire l'hash dei valori nella colonna.

Esempio

```
from pyspark.context import SparkContext
from pyspark.sql import SparkSession
from awsgluedi.transforms import *

secret = "${SECRET}"
sc = SparkContext()
spark = SparkSession(sc)

input_df = spark.createDataFrame(
    [
        (1, "1234560000"),
        (2, "1234560001"),
        (3, "1234560002"),
        (4, "1234560003"),
        (5, "1234560004"),
        (6, "1234560005"),
```

```
        (7, "1234560006"),
        (8, "1234560007"),
        (9, "1234560008"),
        (10, "1234560009"),
    ],
    ["id", "phone"],
)

try:
    df_output = pii.CryptographicHash.apply(
        data_frame=input_df,
        spark_context=sc,
        source_columns=["id", "phone"],
        secret_id=secret,
        algorithm="HMAC_SHA256",
        output_format="BASE64",
    )
    df_output.show()
except:
    print("Unexpected Error happened ")
    raise
```

Output

L'output sarà:

```
+---+------------+------------------+------------------+
| id| phone | id_hashed | phone_hashed |
+---+------------+------------------+------------------+
| 1| 1234560000 | QUI1zXTJiXmfIb... | juDBAmiRnnO3g... |
| 2| 1234560001 | ZAUWiZ3dVTzCo... | vC8lgUqBVDMNQ... |
| 3| 1234560002 | ZP4VvZWkqYifu... | Kl3QAkgswYpzB... |
| 4| 1234560003 | 3u8vO3wQ8EQfj... | CPBzK1P8PZZkV... |
| 5| 1234560004 | eWkQJk4zAOIzx... | aLf7+mHcXqbLs... |
| 6| 1234560005 | xtI9fZCJZCvsa... | dy2DFgdYWmr0p... |
| 7| 1234560006 | iW9hew7jnHuOf... | wwfGMCOEv6oOv... |
| 8| 1234560007 | H9V1pqvgkFhfS... | g9WKhagIXy9ht... |
| 9| 1234560008 | xDhEuHaxAUbU5... | b3uQLKPY+Q5vU... |
| 10| 1234560009 | GRN6nFXkxk349... | VJdsKt8VbxBbt... |
+---+------------+------------------+------------------+
```

La trasformazione calcola gli hash crittografici dei valori nelle colonne `id` e `phone` utilizzando l'algoritmo e la chiave segreta specificati, poi codifica gli hash nel formato Base64. Il `df_output` risultante DataFrame contiene tutte le colonne dell'originale `input_df`, più le colonne aggiuntive `id_hashed` e `phone_hashed` con gli DataFrame hash calcolati.

Metodi

- __call__
- apply
- name
- describeArgs
- describeReturn
- describeTransform
- describeErrors
- describe

__call__(spark_context, data_frame, source_columns, secret_id, algorithm=None, secret_version=None, create_secret_if_missing=False, output_format=None, entity_type_filter=None)

La trasformazione `CryptographicHash` applica un algoritmo per eseguire l'hash dei valori nella colonna.

- `source_columns`: una matrice di colonne esistenti.
- `secret_id`: l'ARN della chiave segreta di Secrets Manager. La chiave utilizzata nell'algoritmo di prefisso del Codice di autenticazione dei messaggi basato su hash (HMAC) per eseguire l'hash delle colonne di origine.
- `secret_version` : Opzionale. Per impostazione predefinita, utilizza la versione più recente del segreto.
- `entity_type_filter`: matrice facoltativa dei tipi di entità. Può essere utilizzata per crittografare solo le informazioni di identificazione personale (PII) rilevate nella colonna a testo libero.
- `create_secret_if_missing`: booleano facoltativo. Se true, cercherà di creare il segreto per conto del chiamante.
- `algorithm`: l'algoritmo utilizzato per eseguire l'hash dei dati. Valori enum validi: MD5 SHA1, SHA256,, MD5 HMAC_, SHA1 SHA256 HMAC_, HMAC_SHA512, HMAC_. SHA512

apply(cls, *args, **kwargs)

Ereditato da `GlueTransform` apply.

name(cls)

Ereditato da `GlueTransform` name.

describeArgs(cls)

Ereditato da `GlueTransform` describeArgs.

describeReturn(cls)

Ereditato da `GlueTransform` describeReturn.

describeTransform(cls)

Ereditato da `GlueTransform` describeTransform.

describeErrors(cls)

Ereditato da `GlueTransform` describeErrors.

describe(cls)

Ereditato da `GlueTransform` describe.

Classe Decrypt

La `Decrypt` trasformazione viene decrittografata all'interno di AWS Glue. I tuoi dati possono essere decrittografati anche al di fuori di AWS Glue con AWS Encryption SDK. Se l'ARN della chiave KMS fornito non corrisponde a quello utilizzato per crittografare la colonna, l'operazione di decrittografia non riesce.

Esempio

```
from pyspark.context import SparkContext
from pyspark.sql import SparkSession
from awsgluedi.transforms import *

kms = "${KMS}"
sc = SparkContext()
spark = SparkSession(sc)
```

```
input_df = spark.createDataFrame(
    [
        (1, "1234560000"),
        (2, "1234560001"),
        (3, "1234560002"),
        (4, "1234560003"),
        (5, "1234560004"),
        (6, "1234560005"),
        (7, "1234560006"),
        (8, "1234560007"),
        (9, "1234560008"),
        (10, "1234560009"),
    ],
    ["id", "phone"],
)

try:
    df_encrypt = pii.Encrypt.apply(
        data_frame=input_df,
        spark_context=sc,
        source_columns=["phone"],
        kms_key_arn=kms
    )
    df_decrypt = pii.Decrypt.apply(
        data_frame=df_encrypt,
        spark_context=sc,
        source_columns=["phone"],
        kms_key_arn=kms
    )
    df_decrypt.show()
except:
    print("Unexpected Error happened ")
    raise
```

Output

L'output sarà composto dalla colonna `id` originale e dalla colonna `phone` decrittografata: PySpark DataFrame

```
```
+---+------------+
| id| phone|
```
```

```
+---+------------+
| 1| 1234560000|
| 2| 1234560001|
| 3| 1234560002|
| 4| 1234560003|
| 5| 1234560004|
| 6| 1234560005|
| 7| 1234560006|
| 8| 1234560007|
| 9| 1234560008|
| 10| 1234560009|
+---+------------+
```

La trasformazione `Encrypt` prende le `source_columns`, come `["phone"]` e `kms_key_arn`, come valore della variabile di ambiente `${KMS}`. La trasformazione esegue la crittografia dei valori nella colonna `phone` utilizzando la chiave KMS specificata. Il DataFrame `df_encrypt` crittografato viene quindi passato alla trasformazione dal modulo `awsglue.pii`. `Decrypt` Prende le `source_columns`, come `["phone"]` e `kms_key_arn`, come valore della variabile di ambiente `${KMS}`. La trasformazione esegue la decrittografia dei valori crittografati nella colonna `phone` utilizzando la stessa chiave KMS. La `df_decrypt` risultante contiene la colonna `id` originale e la colonna `phone` decrittografata. DataFrame

Metodi

- __call__
- apply
- name
- describeArgs
- describeReturn
- describeTransform
- describeErrors
- describe

__call__(spark_context, data_frame, source_columns, kms_key_arn)

La `Decrypt` trasformazione viene decrittografata all'interno di AWS Glue. I tuoi dati possono essere decrittografati anche al di fuori di AWS Glue con AWS Encryption SDK. Se l'ARN della chiave KMS

fornito non corrisponde a quello utilizzato per crittografare la colonna, l'operazione di decrittografia non riesce.

- `source_columns`: una matrice di colonne esistenti.
- `kms_key_arn`— L'ARN della chiave del servizio di gestione delle AWS chiavi da utilizzare per decrittografare le colonne di origine.

apply(cls, *args, **kwargs)

Ereditato da `GlueTransform` apply.

name(cls)

Ereditato da `GlueTransform` name.

describeArgs(cls)

Ereditato da `GlueTransform` describeArgs.

describeReturn(cls)

Ereditato da `GlueTransform` describeReturn.

describeTransform(cls)

Ereditato da `GlueTransform` describeTransform.

describeErrors(cls)

Ereditato da `GlueTransform` describeErrors.

describe(cls)

Ereditato da `GlueTransform` describe.

Classe Encrypt

La `Encrypt` trasformazione crittografa le colonne di origine utilizzando la chiave del servizio di gestione delle AWS chiavi. La trasformazione `Encrypt` può crittografare fino a 128 MiB per cella. Cercherà di mantenere il formato durante la decrittografia. Per mantenere il tipo di dati, i metadati correlati devono essere serializzati a meno di 1 KB. In caso contrario, è necessario impostare il parametro `preserve_data_type` su false. I metadati del tipo di dati saranno archiviati in formato di testo semplice nel contesto della crittografia.

### Esempio

```
from pyspark.context import SparkContext
from pyspark.sql import SparkSession
from awsgluedi.transforms import *

kms = "${KMS}"
sc = SparkContext()
spark = SparkSession(sc)

input_df = spark.createDataFrame(
    [
        (1, "1234560000"),
        (2, "1234560001"),
        (3, "1234560002"),
        (4, "1234560003"),
        (5, "1234560004"),
        (6, "1234560005"),
        (7, "1234560006"),
        (8, "1234560007"),
        (9, "1234560008"),
        (10, "1234560009"),
    ],
    ["id", "phone"],
)

try:
    df_encrypt = pii.Encrypt.apply(
        data_frame=input_df,
        spark_context=sc,
        source_columns=["phone"],
        kms_key_arn=kms
    )
except:
    print("Unexpected Error happened ")
    raise
```

### Output

L'output sarà composto dalla colonna `id` originale e da una colonna aggiuntiva contenente i valori crittografati della colonna `phone`. PySpark DataFrame

```
```
+---+------------+-------------------------+
| id| phone | phone_encrypted |
+---+------------+-------------------------+
| 1| 1234560000| EncryptedData1234...abc |
| 2| 1234560001| EncryptedData5678...def |
| 3| 1234560002| EncryptedData9012...ghi |
| 4| 1234560003| EncryptedData3456...jkl |
| 5| 1234560004| EncryptedData7890...mno |
| 6| 1234560005| EncryptedData1234...pqr |
| 7| 1234560006| EncryptedData5678...stu |
| 8| 1234560007| EncryptedData9012...vwx |
| 9| 1234560008| EncryptedData3456...yz0 |
| 10| 1234560009| EncryptedData7890...123 |
+---+------------+-------------------------+
```
```

La trasformazione `Encrypt` prende le `source_columns`, come `["phone"]` e `kms_key_arn`, come valore della variabile di ambiente `${KMS}`. La trasformazione esegue la crittografia dei valori nella colonna `phone` utilizzando la chiave KMS specificata. La `df_encrypt` risultante DataFrame contiene la colonna `id` originale, la colonna `phone` originale e una colonna aggiuntiva denominata `phone_encrypted` contenente i valori criptati della colonna `phone`.

Metodi

- __call__
- apply
- name
- describeArgs
- describeReturn
- describeTransform
- describeErrors
- describe

__call__(spark_context, data_frame, source_columns, kms_key_arn, entity_type_filter=None, preserve_data_type=None)

`Encrypt`La trasformazione crittografa le AWS colonne di origine utilizzando la chiave del servizio di gestione delle chiavi.

- `source_columns`: una matrice di colonne esistenti.
- `kms_key_arn`— L'ARN della chiave del servizio di gestione delle AWS chiavi da utilizzare per crittografare le colonne di origine.
- `entity_type_filter`: matrice facoltativa dei tipi di entità. Può essere utilizzata per crittografare solo le informazioni di identificazione personale (PII) rilevate nella colonna a testo libero.
- `preserve_data_type`: booleano facoltativo. Il valore predefinito è true. Se false, il tipo di dati non sarà archiviato.

apply(cls, *args, **kwargs)

Ereditato da `GlueTransform` apply.

name(cls)

Ereditato da `GlueTransform` name.

describeArgs(cls)

Ereditato da `GlueTransform` describeArgs.

describeReturn(cls)

Ereditato da `GlueTransform` describeReturn.

describeTransform(cls)

Ereditato da `GlueTransform` describeTransform.

describeErrors(cls)

Ereditato da `GlueTransform` describeErrors.

describe(cls)

Ereditato da `GlueTransform` describe.

### IntToIp classe

La `IntToIp` trasformazione converte il valore intero della colonna di origine o di un altro valore nel IPv4 valore corrispondente nella colonna di destinazione e restituisce il risultato in una nuova colonna.

### Esempio

```
from pyspark.context import SparkContext
from pyspark.sql import SparkSession
from awsgluedi.transforms import *

sc = SparkContext()
spark = SparkSession(sc)

input_df = spark.createDataFrame(
    [
        (3221225473,),
        (0,),
        (1,),
        (100,),
        (168430090,),
        (4294967295,),
        (4294967294,),
        (4294967296,),
        (-1,),
        (None,),
    ],
    ["source_column_int"],
)

try:
    df_output = web_functions.IntToIp.apply(
        data_frame=input_df,
        spark_context=sc,
        source_column="source_column_int",
        target_column="target_column",
        value=None
    )
    df_output.show()
except:
    print("Unexpected Error happened ")
    raise
```

Output

L'output sarà:

```
```
+---------------+---------------+
|source_column_int|target_column|
+---------------+---------------+
| 3221225473| 192.0.0.1 |
| 0| 0.0.0.0 |
| 1| 0.0.0.1 |
| 100| 0.0.0.100|
| 168430090 | 10.0.0.10 |
| 4294967295| 255.255.255.255|
| 4294967294| 255.255.255.254|
| 4294967296| null |
| -1| null |
| null| null |
+---------------+---------------+
```
```

La `IntToIp.apply` trasformazione prende `source_column` come `"source_column_int"` e la `target_column` come `"target_column"` e converte i valori interi nella colonna `source_column_int` nella rappresentazione dell' IPv4 indirizzo corrispondente e memorizza il risultato nella colonna `target_column`.

Per i valori interi validi all'interno dell'intervallo di IPv4 indirizzi (da 0 a 4294967295), la trasformazione li converte correttamente nella loro rappresentazione degli IPv4 indirizzi (ad esempio, 192.0.0.1, 0.0.0.0, 10.0.0.10, 255.255.255.255).

Per i valori interi esterni all'intervallo valido (ad esempio, 4294967296, -1), il valore `target_column` è impostato su `null`. Per i valori `null` presenti nella colonna `source_column_int`, anche il valore `target_column` è impostato su `null`.

Metodi

- __call__
- apply
- name

- describeArgs
- describeReturn
- describeTransform
- describeErrors
- describe

__call__(spark_context, data_frame, target_column, source_column=None, value=None)

La `IntToIp` trasformazione converte il valore intero della colonna di origine o di un altro valore nel valore corrispondente nella colonna di destinazione e restituisce il risultato in una nuova colonna. IPv4

- `sourceColumn`: il nome di una colonna esistente.
- `value`: una stringa di caratteri da valutare.
- `targetColumn`: il nome della nuova colonna da creare.

apply(cls, *args, **kwargs)

Ereditato da `GlueTransform` apply.

name(cls)

Ereditato da `GlueTransform` name.

describeArgs(cls)

Ereditato da `GlueTransform` describeArgs.

describeReturn(cls)

Ereditato da `GlueTransform` describeReturn.

describeTransform(cls)

Ereditato da `GlueTransform` describeTransform.

describeErrors(cls)

Ereditato da `GlueTransform` describeErrors.

describe(cls)

Ereditato da `GlueTransform` describe.

IpToInt classe

La `IpToInt` trasformazione converte il valore del protocollo Internet versione 4 (IPv4) della colonna di origine o altro valore nel valore intero corrispondente nella colonna di destinazione e restituisce il risultato in una nuova colonna.

Esempio

Per AWS Glue 4.0 e versioni successive, crea o aggiorna gli argomenti del lavoro con `key: --enable-glue-di-transforms, value: true`

```
from pyspark.context import SparkContext
from awsgluedi.transforms import *

sc = SparkContext()

input_df = spark.createDataFrame(
    [
        ("192.0.0.1",),
        ("10.10.10.10",),
        ("1.2.3.4",),
        ("1.2.3.6",),
        ("http://12.13.14.15",),
        ("https://16.17.18.19",),
        ("1.2.3.4",),
        (None,),
        ("abc",),
        ("abc.abc.abc.abc",),
        ("321.123.123.123",),
        ("244.4.4.4",),
        ("255.255.255.255",),
    ],
    ["source_column_ip"],
)

    df_output = web_functions.IpToInt.apply(
        data_frame=input_df,
        spark_context=sc,
        source_column="source_column_ip",
```

```
        target_column="target_column",
        value=None
    )
    df_output.show()
```

Output

L'output sarà:

```
```
+----------------+---------------+
|source_column_ip| target_column|
+----------------+---------------+
| 192.0.0.1| 3221225473|
| 10.10.10.10| 168427722|
| 1.2.3.4| 16909060|
| 1.2.3.6| 16909062|
|http://12.13.14.15| null|
|https://16.17.18.19| null|
| 1.2.3.4| 16909060|
| null| null|
| abc| null|
|abc.abc.abc.abc| null|
| 321.123.123.123| null|
| 244.4.4.4| 4102444804|
| 255.255.255.255| 4294967295|
+----------------+---------------+
```
```

La `IpToInt` trasformazione prende la `source_column` come `"source_column_ip"` e la `target_column` come `"target_column"` e converte le stringhe di IPv4 indirizzo valide nella colonna `source_column_ip` nella corrispondente rappresentazione intera a 32 bit e memorizza il risultato nella colonna `target_column`.

Per le stringhe di IPv4 indirizzo valide (ad esempio, «192.0.0.1", «10.10.10.10", «1.2.3.4"), la trasformazione le converte correttamente nella loro rappresentazione intera (ad esempio 3221225473, 168427722, 16909060). Per le stringhe che non sono IPv4 indirizzi validi (ad esempio URLs, stringhe non IP come «abc», formati IP non validi come «abc.abc.abc.abc»), il valore `target_column` è impostato su `null`. Per i valori `null` presenti nella colonna `source_column_ip`, anche il valore `target_column` è impostato su `null`.

Metodi

- __call__
- apply
- name
- describeArgs
- describeReturn
- describeTransform
- describeErrors
- describe

__call__(spark_context, data_frame, target_column, source_column=None, value=None)

La `IpToInt` trasformazione converte il valore del protocollo Internet versione 4 (IPv4) della colonna di origine o altro valore nel valore intero corrispondente nella colonna di destinazione e restituisce il risultato in una nuova colonna.

- `sourceColumn`: il nome di una colonna esistente.
- `value`: una stringa di caratteri da valutare.
- `targetColumn`: il nome della nuova colonna da creare.

apply(cls, *args, **kwargs)

Ereditato da `GlueTransform` apply.

name(cls)

Ereditato da `GlueTransform` name.

describeArgs(cls)

Ereditato da `GlueTransform` describeArgs.

describeReturn(cls)

Ereditato da `GlueTransform` describeReturn.

describeTransform(cls)

Ereditato da `GlueTransform` describeTransform.

describeErrors(cls)

Ereditato da `GlueTransform` describeErrors.

describe(cls)

Ereditato da `GlueTransform` describe.

Trasformazioni di integrazione dei dati

Per AWS Glue 4.0 e versioni successive, crea o aggiorna gli argomenti del lavoro con`key: --enable-glue-di-transforms, value: true`.

Esempio di script di processo:

```
from pyspark.context import SparkContext

from awsgluedi.transforms import *
sc = SparkContext()

input_df = spark.createDataFrame(
    [(5,), (0,), (-1,), (2,), (None,)],
    ["source_column"],
)

try:
    df_output = math_functions.IsEven.apply(
        data_frame=input_df,
        spark_context=sc,
        source_column="source_column",
        target_column="target_column",
        value=None,
        true_string="Even",
        false_string="Not even",
    )
    df_output.show()
except:
    print("Unexpected Error happened ")
    raise
```

### Sessioni di esempio con notebook

```
%idle_timeout 2880
%glue_version 4.0
%worker_type G.1X
%number_of_workers 5
%region eu-west-1
```

```
%%configure
{
    "--enable-glue-di-transforms": "true"
}
```

```
from pyspark.context import SparkContext
from awsgluedi.transforms import *

sc = SparkContext()

input_df = spark.createDataFrame(
    [(5,), (0,), (-1,), (2,), (None,)],
    ["source_column"],
)

try:
    df_output = math_functions.IsEven.apply(
        data_frame=input_df,
        spark_context=sc,
        source_column="source_column",
        target_column="target_column",
        value=None,
        true_string="Even",
        false_string="Not even",
    )
    df_output.show()
except:
    print("Unexpected Error happened ")
    raise
```

### Sessioni di esempio con AWS CLI

```
aws glue create-session --default-arguments "--enable-glue-di-transforms=true"
```

Trasformazioni di integrazione dei dati:

- FlagDuplicatesInColumn classe
- FormatPhoneNumber classe
- FormatCase classe
- FillWithMode classe
- FlagDuplicateRows classe
- RemoveDuplicates classe
- MonthName classe
- IsEven classe
- CryptographicHash classe
- Classe Decrypt
- Classe Encrypt
- IntToIp classe
- IpToInt classe

Maven: creazione di bundle per il plug-in con le applicazioni Spark

È possibile raggruppare le dipendenze delle trasformazioni con le applicazioni Spark e le distribuzioni Spark (versione 3.3) aggiungendo la dipendenza del plug-in nel file Maven `pom.xml` mentre si sviluppano le applicazioni Spark in locale.

```
<repositories>
   ...
    <repository>
        <id>aws-glue-etl-artifacts</id>
        <url>https://aws-glue-etl-artifacts.s3.amazonaws.com/release/ </url>
    </repository>
</repositories>
...
<dependency>
    <groupId>com.amazonaws</groupId>
    <artifactId>AWSGlueTransforms</artifactId>
    <version>4.0.0</version>
</dependency>
```

In alternativa, puoi scaricare i file binari direttamente dagli artefatti di AWS Glue Maven e includerli nella tua applicazione Spark come segue.

```
#!/bin/bash
sudo wget -v https://aws-glue-etl-artifacts.s3.amazonaws.com/release/com/amazonaws/
AWSGlueTransforms/4.0.0/AWSGlueTransforms-4.0.0.jar -P /usr/lib/spark/jars/
```

## Programmazione di script ETL AWS Glue in Scala

Puoi trovare esempi di codice e utilità per Scala AWS Glue nell'archivio degli AWS Glue esempi sul GitHub sito web.

AWS Glue supporta un'estensione del dialetto PySpark Scala per lo scripting di lavori di estrazione, trasformazione e caricamento (ETL). Le sezioni seguenti descrivono come usare la libreria Scala AWS Glue e l'API AWS Glue negli script ETL e forniscono la documentazione di riferimento per la libreria.

Indice


- Uso di Scala per programmare gli script ETL AWS Glue
  - Test di un programma Scala ETL in un notebook Jupyter su un endpoint di sviluppo
  - Test di un programma Scala ETL in una REPL Scala
- Esempio di script Scala - Streaming ETL
- APIs nella libreria AWS Glue Scala
  - com.amazonaws.services.glue
  - com.amazonaws.services.glue.ml
  - com.amazonaws.services.glue.dq
  - com.amazonaws.services.glue.types
  - com.amazonaws.services.glue.util
  - AWS GlueScala ChoiceOption APIs
    - ChoiceOption tratto
    - ChoiceOption oggetto
      - Applicazione di def
    - Classe Case ChoiceOptionWithResolver
    - Classe del caso MatchCatalogSchemaChoiceOption
  - DataSink Classe astratta

        - Def writeDynamicFrame
        - Cornice Def pyWriteDynamic
        - Def writeDataFrame
        - Cornice Def pyWriteData
        - Def setCatalogInfo
        - Def supportsFormat
        - Def setFormat
        - Def withFormat
        - Def setAccumulableSize
        - Def getOutputError RecordsAccumulable
        - Cornice Def errorsAsDynamic
        - DataSink oggetto
            - Def recordMetrics
- AWS Glue DataSource Tratto di Scala
- AWS GlueScala DynamicFrame APIs
    - AWS Glue DynamicFrame Classe Scala
        - Val errorsCount
        - Def applyMapping
        - Def assertErrorThreshold
        - Def conteggio
        - Def dropField
        - Def dropFields
        - Def dropNulls
        - Cornice Def errorsAsDynamic
        - Def filtro
        - Def getName
        - Def getNumPartitions
        - Def calcolato getSchemaIf
        - Def isSchemaComputed
        - Def javaToPython

    - Def join
    - Def mappa
    - Def mergeDynamicFrames
    - Def printSchema
    - Def recomputeSchema
    - Def relazionalizzazione
    - Def renameField
    - Def ripartizione
    - Def resolveChoice
    - Def schema
    - Def selectField
    - Def selectFields
    - Def mostra
    - Def semplifica DDBJson
    - Def spigot
    - Def splitFields
    - Def splitRows
    - Def stageErrorsCount
    - Def toDF
    - Def unbox
    - Def unnest
    - Def unnest DDBJson
    - Def withFrameSchema
    - Def withName
    - Def withTransformationContext
- L' DynamicFrame oggetto
    - Applicazione di def
    - Def emptyDynamicFrame
    - Def fromPythonRDD
    - Def ignoreErrors

        - Def inlineErrors
        - Errori di def newFrameWith
- AWS Glue DynamicRecord Classe Scala
    - Def addField
    - Def dropField
    - Def setError
    - Def isError
    - Def getError
    - Def clearError
    - Def scrittura
    - Def readFields
    - Def clone
    - Def schema
    - Def getRoot
    - Def toJson
    - Def getFieldNode
    - Def getField
    - Def hashCode
    - Def equals
    - DynamicRecord oggetto
        - Applicazione di def
    - RecordTraverser tratto
- AWS GlueScala GlueContext APIs
    - def Colonne addIngestionTime
    - def createDataFrame FromOptions
    - forEachBatch
    - - def getCatalogSink
    - def getCatalogSource
    - def ottenere JDBCSink
    - def getSink

    - Formato def getSinkWith
    - def getSource
    - formato def getSourceWith
    - def getSparkSession
    - def startTransaction
    - def commitTransaction
    - def cancelTransaction
    - def this
    - def this
    - def this
- MappingSpec
    - MappingSpec classe di casi
    - MappingSpec oggetto
    - Val orderingByTarget
    - Applicazione di def
    - Applicazione di def
    - Applicazione di def
- AWS GlueScala ResolveSpec APIs
    - ResolveSpec oggetto
        - Def
        - Def
    - ResolveSpec classe case
        - ResolveSpec metodi def
- AWS GlueScala ArrayNode APIs
    - ArrayNode classe di casi
        - ArrayNode metodi def
- AWS GlueScala BinaryNode APIs
    - BinaryNode classe di casi
        - BinaryNode campi val
        - BinaryNode metodi def

- AWS GlueScala BooleanNode APIs
  - BooleanNode classe di casi
    - BooleanNode campi val
    - BooleanNode metodi def
- AWS GlueScala ByteNode APIs
  - ByteNode classe di casi
    - ByteNode campi val
    - ByteNode metodi def
- AWS GlueScala DateNode APIs
  - DateNode classe di casi
    - DateNode campi val
    - DateNode metodi def
- AWS GlueScala DecimalNode APIs
  - DecimalNode classe di casi
    - DecimalNode campi val
    - DecimalNode metodi def
- AWS GlueScala DoubleNode APIs
  - DoubleNode classe di casi
    - DoubleNode campi val
    - DoubleNode metodi def
- AWS GlueScala DynamicNode APIs
  - DynamicNode classe
    - DynamicNode metodi def
  - DynamicNode oggetto
    - DynamicNode metodi def
- EvaluateDataQuality classe
  - Applicazione di def
  - Esempio
- AWS GlueScala FloatNode APIs
  - FloatNode classe di casi

        - FloatNode campi val
        - FloatNode metodi def
- FillMissingValues classe
    - Applicazione di def
- FindMatches classe
    - Applicazione di def
- FindIncrementalMatches classe
    - Applicazione di def
- AWS GlueScala IntegerNode APIs
    - IntegerNode classe di casi
        - IntegerNode campi val
        - IntegerNode metodi def
- AWS GlueScala LongNode APIs
    - LongNode classe di casi
        - LongNode campi val
        - LongNode metodi def
- AWS GlueScala MapLikeNode APIs
    - MapLikeNode classe
        - MapLikeNode metodi def
- AWS GlueScala MapNode APIs
    - MapNode classe di casi
        - MapNode metodi def
- AWS GlueScala NullNode APIs
    - NullNode classe
    - NullNode oggetto case
- AWS GlueScala ObjectNode APIs
    - ObjectNode oggetto
        - ObjectNode metodi def
    - ObjectNode classe case
        - ObjectNode metodi def

- AWS GlueScala ScalarNode APIs
  - ScalarNode classe
    - ScalarNode metodi def
  - ScalarNode oggetto
    - ScalarNode metodi def
- AWS GlueScala ShortNode APIs
  - ShortNode classe di casi
    - ShortNode campi val
    - ShortNode metodi def
- AWS GlueScala StringNode APIs
  - StringNode classe di casi
    - StringNode campi val
    - StringNode metodi def
- AWS GlueScala TimestampNode APIs
  - TimestampNode classe di casi
    - TimestampNode campi val
    - TimestampNode metodi def
- AWS GlueScala GlueArgParser APIs
  - GlueArgParser oggetto
    - GlueArgParser metodi def
- AWS Gluelavoro Scala APIs
  - Oggetto del processo
    - Metodi def del processo


## Uso di Scala per programmare gli script ETL AWS Glue

È possibile generare automaticamente un programma di estrazione, trasformazione e caricamento (ETL) Scala usando la console AWS Glue ed effettuare le modifiche necessarie prima dell'assegnazione a un processo. In alternativa, è possibile scrivere il proprio programma da zero. Per altre informazioni, consulta Configurazione delle proprietà dei job per i job Spark in AWS Glue. AWS Glue compila quindi il programma Scala nel server prima di eseguire il processo associato.

Per garantire che il programma venga compilato senza errori e venga eseguito come previsto, è importante caricarlo su un endpoint di sviluppo in un REPL (Read-Eval-Print Loop) o in un Jupyter Notebook e testarlo lì prima di eseguirlo in un lavoro. Poiché il processo di compilazione viene effettuato sul server, non sarà possibile individuare chiaramente eventuali problemi.

Test di un programma Scala ETL in un notebook Jupyter su un endpoint di sviluppo

Per testare un programma Scala in un endpoint di sviluppo AWS Glue, configura l'endpoint di sviluppo come descritto in Aggiunta di un endpoint di sviluppo.

Connettilo quindi a un notebook Jupyter in esecuzione sul computer in locale o in remoto su un server notebook Amazon EC2. Per installare una versione locale di un notebook Jupyter, segui le istruzioni riportate alla pagina Tutorial: Notebook Jupyter in JupyterLab.

L'unica differenza tra l'esecuzione del codice Scala e l'esecuzione del PySpark codice sul Notebook è che è necessario iniziare ogni paragrafo sul Notebook con quanto segue:

```
%spark
```

Ciò impedisce al server Notebook di utilizzare come impostazione predefinita l'interprete PySpark Spark.

Test di un programma Scala ETL in una REPL Scala

Come testare un programma Scala in un endpoint di sviluppo usando un ciclo REPL Scala AWS Glue. Segui le istruzioni riportate inTutorial: usa un notebook basato sull' SageMaker intelligenza artificiale, tranne che alla fine del SSH-to-REPL comando, sostituisci `-t gluepyspark` con`-t glue-spark-shell`. In questo modo viene richiamato il ciclo REPL Scala AWS Glue.

Per chiudere la REPL quando si è terminato, digitare `sys.exit`.

## Esempio di script Scala - Streaming ETL

Example

Lo script di esempio seguente si connette ad Amazon Kinesis Data Streams, utilizza uno schema del catalogo dati per analizzare un flusso dei dati, unisce il flusso a un set di dati statico su Amazon S3 e genera i risultati uniti in Amazon S3 in formato parquet.

```
// This script connects to an Amazon Kinesis stream, uses a schema from the data
 catalog to parse the stream,
```

```
// joins the stream to a static dataset on Amazon S3, and outputs the joined results to
 Amazon S3 in parquet format.
import com.amazonaws.services.glue.GlueContext
import com.amazonaws.services.glue.util.GlueArgParser
import com.amazonaws.services.glue.util.Job
import java.util.Calendar
import org.apache.spark.SparkContext
import org.apache.spark.sql.Dataset
import org.apache.spark.sql.Row
import org.apache.spark.sql.SaveMode
import org.apache.spark.sql.SparkSession
import org.apache.spark.sql.functions.from_json
import org.apache.spark.sql.streaming.Trigger
import scala.collection.JavaConverters._

object streamJoiner {
  def main(sysArgs: Array[String]) {
    val spark: SparkContext = new SparkContext()
    val glueContext: GlueContext = new GlueContext(spark)
    val sparkSession: SparkSession = glueContext.getSparkSession
    import sparkSession.implicits._
    // @params: [JOB_NAME]
    val args = GlueArgParser.getResolvedOptions(sysArgs, Seq("JOB_NAME").toArray)
    Job.init(args("JOB_NAME"), glueContext, args.asJava)

    val staticData = sparkSession.read          // read() returns type DataFrameReader
      .format("csv")
      .option("header", "true")
      .load("s3://amzn-s3-demo-bucket/inputs/productsStatic.csv")  // load() returns a
 DataFrame

    val datasource0 = sparkSession.readStream   // readstream() returns type
 DataStreamReader
      .format("kinesis")
      .option("streamName", "stream-join-demo")
      .option("endpointUrl", "https://kinesis.us-east-1.amazonaws.com")
      .option("startingPosition", "TRIM_HORIZON")
      .load                                     // load() returns a DataFrame

    val selectfields1 = datasource0.select(from_json($"data".cast("string"),
 glueContext.getCatalogSchemaAsSparkSchema("stream-demos", "stream-join-demo2")) as
 "data").select("data.*")
```

```
    val datasink2 = selectfields1.writeStream.foreachBatch { (dataFrame: Dataset[Row],
 batchId: Long) => {   //foreachBatch() returns type DataStreamWriter
      val joined = dataFrame.join(staticData, "product_id")
      val year: Int = Calendar.getInstance().get(Calendar.YEAR)
      val month :Int = Calendar.getInstance().get(Calendar.MONTH) + 1
      val day: Int = Calendar.getInstance().get(Calendar.DATE)
      val hour: Int = Calendar.getInstance().get(Calendar.HOUR_OF_DAY)

      if (dataFrame.count() > 0) {
        joined.write                            // joined.write returns type
 DataFrameWriter
          .mode(SaveMode.Append)
          .format("parquet")
          .option("quote", " ")
          .save("s3://amzn-s3-demo-bucket/output/" + "/year=" + "%04d".format(year)
 + "/month=" + "%02d".format(month) + "/day=" + "%02d".format(day) + "/hour=" +
 "%02d".format(hour) + "/")
      }
    }
    }  // end foreachBatch()
      .trigger(Trigger.ProcessingTime("100 seconds"))
      .option("checkpointLocation", "s3://amzn-s3-demo-bucket/checkpoint/")
      .start().awaitTermination()               // start() returns type StreamingQuery
    Job.commit()
  }
}
```

## APIs nella libreria AWS Glue Scala

AWS Gluesupporta un'estensione del dialetto PySpark Scala per la creazione di script di lavori di estrazione, trasformazione e caricamento (ETL). Le sezioni seguenti descrivono la libreria di APIs Scala. AWS Glue

com.amazonaws.services.glue

Il pacchetto com.amazonaws.services.glue nella libreria Scala contiene quanto segue: AWS Glue APIs

- ChoiceOption
- DataSink
- DataSource tratto
- DynamicFrame

- DynamicRecord
- GlueContext
- MappingSpec
- ResolveSpec

com.amazonaws.services.glue.ml

Il pacchetto com.amazonaws.services.glue.ml nella libreria Scala contiene quanto segue: AWS Glue APIs

- FillMissingValues
- FindIncrementalMatches
- FindMatches

com.amazonaws.services.glue.dq

Il pacchetto com.amazonaws.services.glue.dq nella libreria Scala contiene quanto segue: AWS Glue APIs

- EvaluateDataQuality

com.amazonaws.services.glue.types

Il pacchetto com.amazonaws.services.glue.types nella libreria Scala contiene quanto segue: AWS Glue APIs

- ArrayNode
- BinaryNode
- BooleanNode
- ByteNode
- DateNode
- DecimalNode
- DoubleNode
- DynamicNode

- FloatNode
- IntegerNode
- LongNode
- MapLikeNode
- MapNode
- NullNode
- ObjectNode
- ScalarNode
- ShortNode
- StringNode
- TimestampNode

com.amazonaws.services.glue.util

Il pacchetto com.amazonaws.services.glue.util nella libreria Scala contiene quanto segue: AWS Glue APIs

- GlueArgParser
- Processo

AWS GlueScala ChoiceOption APIs

Argomenti

- ChoiceOption tratto
- ChoiceOption oggetto
- Classe Case ChoiceOptionWithResolver
- Classe del caso MatchCatalogSchemaChoiceOption

Pacchetto: com.amazonaws.services.glue

ChoiceOption tratto

```
trait ChoiceOption extends Serializable
```

### ChoiceOption oggetto

#### ChoiceOption

```
object ChoiceOption
```

Una strategia generale per risolvere la scelta applicabile a tutti i nodi `ChoiceType` in un `DynamicFrame`.

- `val CAST`
- `val MAKE_COLS`
- `val MAKE_STRUCT`
- `val MATCH_CATALOG`
- `val PROJECT`

#### Applicazione di def

```
def apply(choice: String): ChoiceOption
```

### Classe Case ChoiceOptionWithResolver

```
case class ChoiceOptionWithResolver(name: String, choiceResolver: ChoiceResolver)
 extends ChoiceOption {}
```

### Classe del caso MatchCatalogSchemaChoiceOption

```
case class MatchCatalogSchemaChoiceOption() extends ChoiceOption {}
```

### DataSink Classe astratta

#### Argomenti

- Def writeDynamicFrame
- Cornice Def pyWriteDynamic
- Def writeDataFrame

- Cornice Def pyWriteData
- Def setCatalogInfo
- Def supportsFormat
- Def setFormat
- Def withFormat
- Def setAccumulableSize
- Def getOutputError RecordsAccumulable
- Cornice Def errorsAsDynamic
- DataSink oggetto


Pacchetto: com.amazonaws.services.glue

```
abstract class DataSink
```

Writer analogo a `DataSource`. `DataSink` incapsula una destinazione e un formato in cui può essere scritto un oggetto `DynamicFrame`.

Def writeDynamicFrame

```
def writeDynamicFrame( frame : DynamicFrame,
                       callSite : CallSite = CallSite("Not provided", "")
                     ) : DynamicFrame
```

Cornice Def pyWriteDynamic

```
def pyWriteDynamicFrame( frame : DynamicFrame,
                         site : String = "Not provided",
                         info : String = "" )
```

Def writeDataFrame

```
def writeDataFrame(frame: DataFrame,
                   glueContext: GlueContext,
                   callSite: CallSite = CallSite("Not provided", "")
                   ): DataFrame
```

### Cornice Def pyWriteData

```
def pyWriteDataFrame(frame: DataFrame,
                     glueContext: GlueContext,
                     site: String = "Not provided",
                     info: String = ""
                     ): DataFrame
```

### Def setCatalogInfo

```
def setCatalogInfo(catalogDatabase: String,
                   catalogTableName : String,
                   catalogId : String = "")
```

### Def supportsFormat

```
def supportsFormat( format : String ) : Boolean
```

### Def setFormat

```
def setFormat( format : String,
               options : JsonOptions
             ) : Unit
```

### Def withFormat

```
def withFormat( format : String,
                options : JsonOptions = JsonOptions.empty
              ) : DataSink
```

### Def setAccumulableSize

```
def setAccumulableSize( size : Int ) : Unit
```

### Def getOutputError RecordsAccumulable

```
def getOutputErrorRecordsAccumulable : Accumulable[List[OutputError], OutputError]
```

### Cornice Def errorsAsDynamic

```
def errorsAsDynamicFrame : DynamicFrame
```

### DataSink oggetto

```
object DataSink
```

### Def recordMetrics

```
def recordMetrics( frame : DynamicFrame,
                   ctxt : String
                 ) : DynamicFrame
```

## AWS Glue DataSource Tratto di Scala

Pacchetto: com.amazonaws.services.glue

Un'interfaccia di alto livello per la produzione di un `DynamicFrame`.

```
trait DataSource {

  def getDynamicFrame : DynamicFrame

  def getDynamicFrame( minPartitions : Int,
                       targetPartitions : Int
                     ) : DynamicFrame
  def getDataFrame : DataFrame

  /** @param num: the number of records for sampling.
    * @param options: optional parameters to control sampling behavior. Current
 available parameter for Amazon S3 sources in options:
    *  1. maxSamplePartitions: the maximum number of partitions the sampling will
 read.
    *  2. maxSampleFilesPerPartition: the maximum number of files the sampling will
 read in one partition.
```

```
    */
  def getSampleDynamicFrame(num:Int, options: JsonOptions = JsonOptions.empty):
 DynamicFrame

  def glueContext : GlueContext

  def setFormat( format : String,
                 options : String
               ) : Unit

  def setFormat( format : String,
                 options : JsonOptions
               ) : Unit

  def supportsFormat( format : String ) : Boolean

  def withFormat( format : String,
                  options : JsonOptions = JsonOptions.empty
                ) : DataSource
}
```

AWS GlueScala DynamicFrame APIs

Pacchetto: com.amazonaws.services.glue

Indice


- AWS Glue DynamicFrame Classe Scala
  - Val errorsCount
  - Def applyMapping
  - Def assertErrorThreshold
  - Def conteggio
  - Def dropField
  - Def dropFields
  - Def dropNulls
  - Cornice Def errorsAsDynamic
  - Def filtro
  - Def getName
  - Def getNumPartitions

- Def calcolato getSchemaIf
- Def isSchemaComputed
- Def javaToPython
- Def join
- Def mappa
- Def mergeDynamicFrames
- Def printSchema
- Def recomputeSchema
- Def relazionalizzazione
- Def renameField
- Def ripartizione
- Def resolveChoice
- Def schema
- Def selectField
- Def selectFields
- Def mostra
- Def semplifica DDBJson
- Def spigot
- Def splitFields
- Def splitRows
- Def stageErrorsCount
- Def toDF
- Def unbox
- Def unnest
- Def unnest DDBJson
- Def withFrameSchema
- Def withName
- Def withTransformationContext

- L' DynamicFrame oggetto
  - Applicazione di def

- Def emptyDynamicFrame
- Def fromPythonRDD
- Def ignoreErrors
- Def inlineErrors
- Errori di def newFrameWith

AWS Glue DynamicFrame Classe Scala

Pacchetto: com.amazonaws.services.glue

```
class DynamicFrame extends Serializable with Logging  (
           val glueContext : GlueContext,
           _records : RDD[DynamicRecord],
           val name : String = s"",
           val transformationContext : String = DynamicFrame.UNDEFINED,
           callSite : CallSite = CallSite("Not provided", ""),
           stageThreshold : Long = 0,
           totalThreshold : Long = 0,
           prevErrors : => Long = 0,
           errorExpr : => Unit = {} )
```

Un `DynamicFrame` è una raccolta distribuita di oggetti DynamicRecord autodescrittivi.

`DynamicFrame` sono stati progettati per fornire un modello di dati flessibile per le operazioni ETL (estrazione, trasformazione e caricamento). Questi oggetti non richiedono la creazione di uno schema e possono essere usati per leggere e trasformare i dati che contengono valori e tipi non organizzati e non coerenti. Un schema può essere calcolato on demand per le operazioni che ne richiedono uno.

`DynamicFrame` offrono un'ampia gamma di trasformazioni per la pulizia dei dati e operazioni ETL. Supportano anche la conversione da e verso SparkSQL DataFrames per l'integrazione con il codice esistente e le numerose operazioni di analisi che DataFrames forniscono.

I parametri seguenti vengono condivisi tra molte trasformazioni AWS Glue che costituiscono oggetti `DynamicFrame`:

- `transformationContext` — Identificatore per questo `DynamicFrame`. Il `transformationContext` viene usato come chiave per lo stato dei segnalibro di processo che viene mantenuto tra esecuzioni.

- `callSite` — Fornisce informazioni sul contesto per la segnalazione degli errori. Questi valori vengono impostati automaticamente durante la chiamata da Python.
- `stageThreshold` — Numero massimo di record di errore consentiti nel calcolo di questo `DynamicFrame` prima di generare un'eccezione, esclusi i record presenti nell'oggetto `DynamicFrame` precedente.
- `totalThreshold` — numero massimo di record di errore totali prima di generare un'eccezione, inclusi quelli dei frame precedenti.

Val errorsCount

```
val errorsCount
```

Numero di record di errore in questo oggetto `DynamicFrame`. Include gli errori restituiti dalle operazioni precedenti.

Def applyMapping

```
def applyMapping( mappings : Seq[Product4[String, String, String, String]],
                  caseSensitive : Boolean = true,
                  transformationContext : String = "",
                  callSite : CallSite = CallSite("Not provided", ""),
                  stageThreshold : Long = 0,
                  totalThreshold : Long = 0
                ) : DynamicFrame
```

- `mappings` — Sequenza di mappature per creare un nuovo oggetto `DynamicFrame`.
- `caseSensitive` — Specifica se considerare o meno le colonne di origine come colonne che fanno distinzione tra maiuscole e minuscole. L'impostazione su false potrebbe essere utile durante l'integrazione con archivi senza distinzione tra maiuscole e minuscole come AWS Glue Data Catalog.

Seleziona, proietta e trasmette le colonne in base a una sequenza di mappature.

Ogni mappatura è costituita da una colonna e un tipo di origine e da una colonna e un tipo target. Le mappature possono essere specificate come un 4-tuple (`source_path`, `source_type`, `target_path`, `target_type`) o un oggetto MappingSpec contenente le stesse informazioni.

Oltre che per semplici proiezioni e trasferimenti, le mappature possono essere usate per annidare campi o annullarne l'annidamento separando i componenti del percorso con "`.`" (punto).

Ad esempio, supponiamo di avere un `DynamicFrame` con lo schema seguente.

```
{{{
  root
  |-- name: string
  |-- age: int
  |-- address: struct
  |    |-- state: string
  |    |-- zip: int
}}}
```

Puoi effettuare la chiamata seguente per annullare l'annidamento dei campi `state` e `zip`.

```
{{{
  df.applyMapping(
   Seq(("name", "string", "name", "string"),
       ("age", "int", "age", "int"),
       ("address.state", "string", "state", "string"),
       ("address.zip", "int", "zip", "int")))
}}}
```

Lo schema risultante è il seguente.

```
{{{
  root
  |-- name: string
  |-- age: int
  |-- state: string
  |-- zip: int
}}}
```

Puoi anche usare `applyMapping` per riannidare le colonne. Ad esempio, il codice seguente inverte la trasformazione precedente e crea una struttura denominata `address` nel target.

```
{{{
  df.applyMapping(
   Seq(("name", "string", "name", "string"),
       ("age", "int", "age", "int"),
       ("state", "string", "address.state", "string"),
```

```
        ("zip", "int", "address.zip", "int")))
 }}}
```

I nomi dei campi che contengono i caratteri "`.`" (periodo) possono essere quotati utilizzando le virgolette (` `).

> **Note**
>
> Al momento, non puoi utilizzare il metodo `applyMapping` per mappare colonne annidate all'interno di matrici.

Def assertErrorThreshold

```
def assertErrorThreshold : Unit
```

Operazione che forza il calcolo e verifica che il numero di record di errore sia inferiore a `stageThreshold` e `totalThreshold`. Genera un'eccezione se una delle due condizioni non è vera.

Def conteggio

```
lazy
def count
```

Restituisce il numero di elementi inclusi in questo oggetto `DynamicFrame`.

Def dropField

```
def dropField( path : String,
               transformationContext : String = "",
               callSite : CallSite = CallSite("Not provided", ""),
               stageThreshold : Long = 0,
               totalThreshold : Long = 0
             ) : DynamicFrame
```

Restituisce un nuovo oggetto `DynamicFrame` con la colonna specificata rimossa.

Def dropFields

```
def dropFields( fieldNames : Seq[String],   // The column names to drop.
```

```
                transformationContext : String = "",
                callSite : CallSite = CallSite("Not provided", ""),
                stageThreshold : Long = 0,
                totalThreshold : Long = 0
              ) : DynamicFrame
```

Restituisce un nuovo oggetto `DynamicFrame` con le colonne specificate rimosse.

Questo metodo può essere usato per eliminare colonne annidate, incluse quelle all'interno di matrici, ma non per eliminare elementi di matrice specifici.

Def dropNulls

```
def dropNulls( transformationContext : String = "",
               callSite : CallSite = CallSite("Not provided", ""),
               stageThreshold : Long = 0,
               totalThreshold : Long = 0 )
```

Restituisce un nuovo oggetto `DynamicFrame` con tutte le colonne null rimosse.

> **Note**
>
> Rimuove solo le colonne di tipo `NullType`. I singoli valori null in altre colonne non vengono rimossi o modificati.

Cornice Def errorsAsDynamic

```
def errorsAsDynamicFrame
```

Restituisce un nuovo oggetto `DynamicFrame` contenente i record degli errori da questo `DynamicFrame`.

Def filtro

```
def filter( f : DynamicRecord => Boolean,
            errorMsg : String = "",
            transformationContext : String = "",
            callSite : CallSite = CallSite("Not provided"),
            stageThreshold : Long = 0,
            totalThreshold : Long = 0
```

```
          ) : DynamicFrame
```

Crea un nuovo oggetto `DynamicFrame` contenente solo i record per i quali la funzione "f" restituisce `true`. La funzione di filtro "f" non dovrebbe modificare il record di input.

Def getName

```
def getName : String
```

Restituisce il nome di questo oggetto `DynamicFrame`.

Def getNumPartitions

```
def getNumPartitions
```

Restituisce il numero di partizioni incluse in questo oggetto `DynamicFrame`.

Def calcolato getSchemaIf

```
def getSchemaIfComputed : Option[Schema]
```

Restituisce lo schema se è già stato calcolato. Non analizza i dati se lo schema non è stato ancora calcolato.

Def isSchemaComputed

```
def isSchemaComputed : Boolean
```

Restituisce `true` se lo schema è stato calcolato per questo oggetto `DynamicFrame` oppure restituisce `false` in caso contrario. Se questo metodo restituisce false, la chiamata del metodo `schema` richiede un altro passaggio sui record in questo oggetto `DynamicFrame`.

Def javaToPython

```
def javaToPython : JavaRDD[Array[Byte]]
```

Def join

```
def join( keys1 : Seq[String],
          keys2 : Seq[String],
```

```
          frame2 : DynamicFrame,
          transformationContext : String = "",
          callSite : CallSite = CallSite("Not provided", ""),
          stageThreshold : Long = 0,
          totalThreshold : Long = 0
        ) : DynamicFrame
```

- `keys1` — Le colonne in questo `DynamicFrame` da utilizzare per l'unione.
- `keys2` — Le colonne in `frame2` da utilizzare per l'unione. Deve avere la stessa lunghezza di `keys1`.
- `frame2` — `DynamicFrame` da unire.

Restituisce il risultato dell'esecuzione di una query equijoin con `frame2` usando le chiavi specificate.

Def mappa

```
def map( f : DynamicRecord => DynamicRecord,
         errorMsg : String = "",
         transformationContext : String = "",
         callSite : CallSite = CallSite("Not provided", ""),
         stageThreshold : Long = 0,
         totalThreshold : Long = 0
       ) : DynamicFrame
```

Restituisce un nuovo oggetto `DynamicFrame` creato applicando la funzione "`f`" specificata a ogni record in questo oggetto `DynamicFrame`.

Questo metodo copia ogni record prima di applicare la funzione specificata e di conseguenza è sicuro per la modifica dei record. Se la funzione di mappatura genera un'eccezione per un determinato record, il record verrà contrassegnato come errore e l'analisi dello stack verrà salvata come colonna nel record di errore.

Def mergeDynamicFrames

```
def mergeDynamicFrames( stageDynamicFrame: DynamicFrame,  primaryKeys: Seq[String],
 transformationContext: String = "",
                         options: JsonOptions = JsonOptions.empty, callSite: CallSite =
 CallSite("Not provided"),
                         stageThreshold: Long = 0, totalThreshold: Long = 0):
 DynamicFrame
```

- `stageDynamicFrame` — Il `DynamicFrame` di gestione temporanea da unire.
- `primaryKeys` — L'elenco dei campi chiave primaria per abbinare i record dall'origine e dal `DynamicFrame` di gestione temporanea.
- `transformationContext` — Una stringa univoca utilizzata per recuperare i metadati relativi alla trasformazione corrente (opzionale).
- `options`: una stringa di coppie nome-valore JSON che forniscono informazioni aggiuntive per questa trasformazione.
- `callSite` — Usato per fornire informazioni sul contesto per la segnalazione degli errori.
- `stageThreshold` — Un `Long`. Il numero di errori nella trasformazione specificata per cui l'elaborazione deve restituire un errore.
- `totalThreshold` — Un `Long`. Il numero totale di errori fino a questa trasformazione inclusa per i quali l'elaborazione deve restituire un errore.

Unisce questo `DynamicFrame` con un `DynamicFrame` temporaneo basato sulle chiavi primarie specificate per identificare i record. I registri duplicati (registri con le stesse chiavi primarie) non vengono deduplicati. Se non è presente alcun record corrispondente nel frame temporaneo, tutti i record (inclusi i duplicati) vengono mantenuti dall'origine. Se il frame di staging dispone di record corrispondenti, i suoi registri sovrascrivono i registri nell'origine in AWS Glue.

Il `DynamicFrame` restituito contiene il record A in questi casi:

1. Se `A` esiste sia nel frame di origine che nel frame temporaneo, viene restituito `A` nel frame temporaneo.
2. Se `A` si trova nella tabella di origine e `A.primaryKeys` non si trova nel `stagingDynamicFrame` (ciò significa che `A` non viene aggiornato nella tabella temporanea).

Il frame di origine e il frame temporaneo non devono avere lo stesso schema.

Example

```
val mergedFrame: DynamicFrame = srcFrame.mergeDynamicFrames(stageFrame, Seq("id1",
 "id2"))
```

Def printSchema

```
def printSchema : Unit
```

Stampa lo schema di questo oggetto `DynamicFrame` in `stdout` in un formato leggibile.

Def recomputeSchema

```
def recomputeSchema : Schema
```

Forza un nuovo calcolo dello schema. Questa operazione richiede una scansione dei dati, ma potrebbe "limitare" lo schema se lo schema corrente include dati che nono sono presenti nei dati.

Restituisce lo schema ricalcolato.

Def relazionalizzazione

```
def relationalize( rootTableName : String,
                   stagingPath : String,
                   options : JsonOptions = JsonOptions.empty,
                   transformationContext : String = "",
                   callSite : CallSite = CallSite("Not provided"),
                   stageThreshold : Long = 0,
                   totalThreshold : Long = 0
                 ) : Seq[DynamicFrame]
```

- `rootTableName`: il nome da usare per l'oggetto `DynamicFrame` di base nell'output. Gli oggetti `DynamicFrame` creati tramite il pivoting di matrici usano questo nome come prefisso.
- `stagingPath` — il percorso Amazon Simple Storage Service (Amazon S3) per la scrittura di dati intermedi.
- `options` — opzioni e configurazione per l'applicazione di relazioni. Attualmente inutilizzato.

Appiattisce tutte le strutture annidate e trasforma tramite pivoting le matrici in tabelle separate.

Questa operazione può essere usata per preparare dati annidati a più livelli per l'inserimento in un database relazionale. Le strutture annidate vengono appiattite allo stesso modo della trasformazione Unnest. Inoltre, le matrici vengono trasformate tramite pivoting in tabelle separate attraverso un'operazione in cui ogni elemento di matrice diventa una riga. Ad esempio, supponiamo di avere un `DynamicFrame` con i dati seguenti.

```
{"name": "Nancy", "age": 47, "friends": ["Fred", "Lakshmi"]}
{"name": "Stephanie", "age": 28, "friends": ["Yao", "Phil", "Alvin"]}
{"name": "Nathan", "age": 54, "friends": ["Nicolai", "Karen"]}
```

Eseguire il seguente codice.

```
{{{
  df.relationalize("people", "s3:/my_bucket/my_path", JsonOptions.empty)
}}}
```

Il codice produce due tabelle. La prima tabella è denominata "persone" e contiene quanto segue.

```
{{{
  {"name": "Nancy", "age": 47, "friends": 1}
  {"name": "Stephanie", "age": 28, "friends": 2}
  {"name": "Nathan", "age": 54, "friends": 3)
}}}
```

Qui le matrici friends sono state sostituite con una chiave di join generata automaticamente. Viene creata una tabella separata denominata `people.friends` con il contenuto seguente.

```
{{{
  {"id": 1, "index": 0, "val": "Fred"}
  {"id": 1, "index": 1, "val": "Lakshmi"}
  {"id": 2, "index": 0, "val": "Yao"}
  {"id": 2, "index": 1, "val": "Phil"}
  {"id": 2, "index": 2, "val": "Alvin"}
  {"id": 3, "index": 0, "val": "Nicolai"}
  {"id": 3, "index": 1, "val": "Karen"}
}}}
```

In questa tabella "`id`" è una chiave di join che identifica il record da cui proviene l'elemento della matrice, "`index`" fa riferimento alla posizione nella matrice originale e "`val`" è l'effettiva voce della matrice.

Il metodo `relationalize` restituisce la sequenza di oggetti `DynamicFrame` creati applicando questo processo in modo ricorsivo a tutte le matrici.

> **Note**
>
> La libreria AWS Glue genera automaticamente le chiavi di join per le nuove tabelle. Per garantire che le chiavi di join siano univoche tra esecuzioni di processi, devono essere abilitati i segnalibro di processo.

Def renameField

```
def renameField( oldName : String,
                 newName : String,
                 transformationContext : String = "",
                 callSite : CallSite = CallSite("Not provided", ""),
                 stageThreshold : Long = 0,
                 totalThreshold : Long = 0
               ) : DynamicFrame
```

- `oldName`: nome originale della colonna.
- `newName`: nuovo nome della colonna.

Restituisce un nuovo `DynamicFrame` con il campo specificato rinominato.

Questo metodo può essere usato per rinominare campi annidati. Ad esempio, il codice seguente rinominerebbe `state` in `state_code` all'interno della struttura address.

```
{{{
  df.renameField("address.state", "address.state_code")
}}}
```

Def ripartizione

```
def repartition( numPartitions : Int,
                 transformationContext : String = "",
                 callSite : CallSite = CallSite("Not provided", ""),
                 stageThreshold : Long = 0,
                 totalThreshold : Long = 0
               ) : DynamicFrame
```

Restituisce un nuovo oggetto `DynamicFrame` con partizioni `numPartitions`.

Def resolveChoice

```
def resolveChoice( specs : Seq[Product2[String, String]] = Seq.empty[ResolveSpec],
                   choiceOption : Option[ChoiceOption] = None,
                   database : Option[String] = None,
                   tableName : Option[String] = None,
```

```
    transformationContext : String = "",
    callSite : CallSite = CallSite("Not provided", ""),
    stageThreshold : Long = 0,
    totalThreshold : Long = 0
  ) : DynamicFrame
```

- `choiceOption` — Un'operazione da applicare a tutte le colonne `ChoiceType` non elencate nella sequenza delle specifiche.
- `database` — Il database del catalogo dati da usare con l'operazione `match_catalog`.
- `tableName` — La tabella del catalogo dati da usare con l'operazione `match_catalog`.

Restituisce un nuovo oggetto `DynamicFrame` sostituendo uno o più oggetti `ChoiceType` con un tipo più specifico.

Ci sono due modi per utilizzare `resolveChoice`. Il primo consiste nell'indicare una sequenza di colonne specifiche e come risolverle. Queste vengono specificate come tuple costituite da coppie (colonna, operazione).

Sono disponibili le operazioni seguenti:

- `cast:type`: tenta di trasmettere tutti i valori al tipo specificato.
- `make_cols`: converte ogni tipo distinto in colonna con il nome `columnName_type`.
- `make_struct`: converte una colonna in struttura con chiavi per ogni tipo distinto.
- `project:type` — mantiene solo i valori del tipo specificato.

L'altra modalità per `resolveChoice` è specificare una singola risoluzione per tutti gli oggetti `ChoiceType`. Puoi usare questa modalità nei casi in cui l'elenco completo di oggetti `ChoiceType` non è noto prima dell'esecuzione. Oltre alle operazioni elencate in precedenza, questa modalità supporta anche l'operazione seguente:

- `match_catalogChoiceType`: tenta di trasmettere ogni oggetto al tipo corrispondente nella tabella del catalogo specificata.

Esempi:

Risoluzione della colonna `user.id` mediante casting a un tipo int, facendo in modo che il campo `address` mantenga solo le strutture:

```
{{{
  df.resolveChoice(specs = Seq(("user.id", "cast:int"), ("address", "project:struct")))
}}}
```

Risoluzione di tutti gli oggetti `ChoiceType` mediante conversione di ogni scelta in una colonna separata:

```
{{{
  df.resolveChoice(choiceOption = Some(ChoiceOption("make_cols")))
}}}
```

Risoluzione di tutte gli oggetti `ChoiceType` mediante casting ai tipi nella tabella del catalogo specificata:

```
{{{
  df.resolveChoice(choiceOption = Some(ChoiceOption("match_catalog")),
                   database = Some("my_database"),
                   tableName = Some("my_table"))
}}}
```

Def schema

```
def schema : Schema
```

Restituisce lo schema di questo oggetto `DynamicFrame`.

Lo schema restituito è garantito per contenere ogni campo presente in un record in questo `DynamicFrame`. Tuttavia, in un esiguo numero di casi, può anche contenere campi aggiuntivi. Puoi utilizzare il metodo Unnest per "ridurre" lo schema in base ai record in questo `DynamicFrame`.

Def selectField

```
def selectField( fieldName : String,
                 transformationContext : String = "",
                 callSite : CallSite = CallSite("Not provided", ""),
                 stageThreshold : Long = 0,
                 totalThreshold : Long = 0
               ) : DynamicFrame
```

Restituisce un singolo campo come `DynamicFrame`.

Def selectFields

```
def selectFields( paths : Seq[String],
                  transformationContext : String = "",
                  callSite : CallSite = CallSite("Not provided", ""),
                  stageThreshold : Long = 0,
                  totalThreshold : Long = 0
                ) : DynamicFrame
```

- `paths` — La sequenza dei nomi delle colonne da selezionare.

Restituisce un nuovo oggetto `DynamicFrame` contenente le colonne specificate.

> **Note**
>
> Il metodo `selectFields` può essere usato solo per selezionare colonne di primo livello. Puoi utilizzare il metodo applyMapping per selezionare colonne annidate.

Def mostra

```
def show( numRows : Int = 20 ) : Unit
```

- `numRows` — Numero di righe da stampare.

Stampa le righe di questo oggetto `DynamicFrame` in formato JSON.

Def semplifica DDBJson

Le esportazioni di DynamoDB con il connettore di esportazione DynamoDB AWS Glue possono risultare in file JSON con specifiche strutture annidate. Per ulteriori informazioni, consulta gli Oggetti dati. `simplifyDDBJson` Semplifica le colonne annidate in un tipo DynamicFrame di dati di questo tipo e ne restituisce uno nuovo semplificato. DynamicFrame Se sono presenti più tipi o un tipo di Mappa contenuta in un tipo di Elenco, gli elementi contenuti nell'Elenco non saranno semplificati. Questo metodo supporta solo i dati nel formato JSON di esportazione DynamoDB. Prendi in considerazione `unnest` per apportare modifiche simili su altri tipi di dati.

```
def simplifyDDBJson() : DynamicFrame
```

Questo metodo non assume parametri.

Input di esempio

Prendi in considerazione lo schema seguente generato da un'esportazione DynamoDB:

```
root
|-- Item: struct
|    |-- parentMap: struct
|    |    |-- M: struct
|    |    |    |-- childMap: struct
|    |    |    |    |-- M: struct
|    |    |    |    |    |-- appName: struct
|    |    |    |    |    |    |-- S: string
|    |    |    |    |    |-- packageName: struct
|    |    |    |    |    |    |-- S: string
|    |    |    |    |    |-- updatedAt: struct
|    |    |    |    |    |    |-- N: string
|    |-- strings: struct
|    |    |-- SS: array
|    |    |    |-- element: string
|    |-- numbers: struct
|    |    |-- NS: array
|    |    |    |-- element: string
|    |-- binaries: struct
|    |    |-- BS: array
|    |    |    |-- element: string
|    |-- isDDBJson: struct
|    |    |-- BOOL: boolean
|    |-- nullValue: struct
|    |    |-- NULL: boolean
```

Esempio di codice

```
import com.amazonaws.services.glue.GlueContext
import com.amazonaws.services.glue.util.GlueArgParser
import com.amazonaws.services.glue.util.Job
import com.amazonaws.services.glue.util.JsonOptions
import com.amazonaws.services.glue.DynamoDbDataSink
import org.apache.spark.SparkContextimport scala.collection.JavaConverters._
```

```
object GlueApp {

  def main(sysArgs: Array[String]): Unit = {
    val glueContext = new GlueContext(SparkContext.getOrCreate())
    val args = GlueArgParser.getResolvedOptions(sysArgs, Seq("JOB_NAME").toArray)
    Job.init(args("JOB_NAME"), glueContext, args.asJava)

    val dynamicFrame = glueContext.getSourceWithFormat(
      connectionType = "dynamodb",
      options = JsonOptions(Map(
        "dynamodb.export" -> "ddb",
        "dynamodb.tableArn" -> "ddbTableARN",
        "dynamodb.s3.bucket" -> "exportBucketLocation",
        "dynamodb.s3.prefix" -> "exportBucketPrefix",
        "dynamodb.s3.bucketOwner" -> "exportBucketAccountID",
      ))
    ).getDynamicFrame()

    val simplified = dynamicFrame.simplifyDDBJson()
    simplified.printSchema()

    Job.commit()
  }

}
```

### Output di esempio

La trasformazione `simplifyDDBJson` semplificherà questo processo in:

```
root
|-- parentMap: struct
|    |-- childMap: struct
|    |    |-- appName: string
|    |    |-- packageName: string
|    |    |-- updatedAt: string
|-- strings: array
|    |-- element: string
|-- numbers: array
|    |-- element: string
|-- binaries: array
|    |-- element: string
|-- isDDBJson: boolean
```

```
|-- nullValue: null
```

Def spigot

```
def spigot( path : String,
            options : JsonOptions = new JsonOptions("{}"),
            transformationContext : String = "",
            callSite : CallSite = CallSite("Not provided"),
            stageThreshold : Long = 0,
            totalThreshold : Long = 0
          ) : DynamicFrame
```

Trasformazione passthrough che restituisce gli stessi record, ma scrive un sottoinsieme di record come effetto secondario.

- `path` — Il percorso in Amazon S3 in cui scrivere l'output, nel formato `s3://bucket//path`.
- `options` — Una mappa `JsonOptions` opzionale che descrive il comportamento di campionamento.

Restituisce un oggetto `DynamicFrame` contenente gli stessi record di questo.

Per impostazione predefinita, scrive 100 record arbitrari nel percorso specificato da `path`. Questo comportamento può essere personalizzato usando la mappa `options`. Le chiavi valide includono le seguenti:

- `topk` — Specifica il numero totale di record scritti. Il valore di default è 100.
- `prob`: specifica la probabilità (sotto forma di valore decimale) di inclusione di un singolo record. Il valore predefinito è 1.

Ad esempio, la chiamata seguente esegue il campionamento del set di dati selezionando ogni record con una probabilità del 20% e arrestandosi dopo la scrittura di 200 record.

```
{{{
  df.spigot("s3://my_bucket/my_path", JsonOptions(Map("topk" -&gt; 200, "prob" -&gt;
 0.2)))
}}}
```

Def splitFields

```
def splitFields( paths : Seq[String],
                 transformationContext : String = "",
                 callSite : CallSite = CallSite("Not provided", ""),
                 stageThreshold : Long = 0,
                 totalThreshold : Long = 0
               ) : Seq[DynamicFrame]
```

- `paths` — Percorsi da includere nel primo `DynamicFrame`.

Restituisce una sequenza di due oggetti `DynamicFrame`. Il primo oggetto `DynamicFrame` contiene i percorsi specificati, mentre il secondo contiene tutte le altre colonne.

Esempio

Questo esempio prende una tabella DynamicFrame creata dalla `persons` tabella nel `legislators` database del AWS Glue Data Catalog e la DynamicFrame divide in due, con i campi specificati che entrano nel primo DynamicFrame e i campi rimanenti in un secondo DynamicFrame. L'esempio sceglie quindi il primo DynamicFrame dal risultato.

```
val InputFrame = glueContext.getCatalogSource(database="legislators",
 tableName="persons",
transformationContext="InputFrame").getDynamicFrame()

val SplitField_collection = InputFrame.splitFields(paths=Seq("family_name", "name",
 "links.note",
"links.url", "gender", "image", "identifiers.scheme", "identifiers.identifier",
 "other_names.lang",
"other_names.note", "other_names.name"), transformationContext="SplitField_collection")

val ResultFrame = SplitField_collection(0)
```

Def splitRows

```
def splitRows( paths : Seq[String],
               values : Seq[Any],
               operators : Seq[String],
               transformationContext : String,
               callSite : CallSite,
               stageThreshold : Long,
```

```
              totalThreshold : Long
            ) : Seq[DynamicFrame]
```

Suddivide le righe in base a predicati che confrontano colonne e costanti.

- `paths` — Colonne da usare per il confronto.
- `values` — I valori di costante da usare per il confronto.
- `operators` — Gli operatori da usare per il confronto.

Restituisce una sequenza di due oggetti `DynamicFrame`. Il primo contiene le righe per cui il predicato è true, il secondo contiene quelle per cui è false.

I predicati vengono specificati usando tre sequenze: "`paths`" contiene i nomi delle colonne (possibilmente annidate), "`values`" contiene i valori di costante rispetto ai quali eseguire il confronto e "`operators`" contiene gli operatori da usare per il confronto. Le tre sequenze devono essere tutte della stessa lunghezza: l'nesimo operatore verrà usato per confrontare la nesima colonna con l'nesimo valore.

Gli operatori consentiti sono: "`!=`", "`=`", "`<=`", "`<`", "`>=`" o "`>`".

Ad esempio, la chiamata seguente divide un oggetto `DynamicFrame` in modo che il primo frame di output contenga i record di persone degli Stati Uniti di età maggiore di 65 anni e che il secondo contenga tutti gli altri record.

```
{{{
  df.splitRows(Seq("age", "address.country"), Seq(65, "USA"), Seq("&gt;=", "="))
}}}
```

Def stageErrorsCount

```
def stageErrorsCount
```

Restituisce il numero di record di errore creati durante il calcolo di questo oggetto `DynamicFrame`. Sono esclusi gli errori restituiti dalla operazioni precedenti passate a questo oggetto `DynamicFrame` come input.

Def toDF

```
def toDF( specs : Seq[ResolveSpec] = Seq.empty[ResolveSpec] ) : DataFrame
```

Converte questo `DynamicFrame` in un Apache Spark SQL `DataFrame` con lo stesso schema e gli stessi record.

> **Note**
>
> Poiché gli oggetti `DataFrame` non supportano oggetti `ChoiceType`, questo metodo converte automaticamente le colonne `ChoiceType` in oggetti `StructType`. Per ulteriori informazioni sulle opzioni per le scelte di risoluzione, consulta resolveChoice.

Def unbox

```
def unbox( path : String,
           format : String,
           optionString : String = "{}",
           transformationContext : String = "",
           callSite : CallSite = CallSite("Not provided"),
           stageThreshold : Long = 0,
           totalThreshold : Long = 0
         ) : DynamicFrame
```

- `path` — La colonna da analizzare. Deve essere di tipo String o Binary.
- `format`: formato da usare per l'analisi.
- `optionString`: opzioni da passare al formato, ad esempio il separatore per file CSV.

Analizza una stringa incorporata o una colonna binaria in base al formato specificato. Le colonne analizzate vengono annidate all'interno di una struttura con il nome di colonna originale.

Supponi, ad esempio, di avere un file CSV con una colonna JSON incorporata.

```
name, age, address
Sally, 36, {"state": "NE", "city": "Omaha"}
...
```

Dopo un'analisi iniziale, avresti un oggetto `DynamicFrame` con lo schema seguente.

```
{{{
  root
  |-- name: string
```

```
  |-- age: int
  |-- address: string
}}}
```

Puoi chiamare unbox nella colonna address per analizzare i componenti specifici.

```
{{{
  df.unbox("address", "json")
}}}
```

Otterrai un oggetto DynamicFrame con lo schema seguente.

```
{{{
  root
  |-- name: string
  |-- age: int
  |-- address: struct
  |    |-- state: string
  |    |-- city: string
}}}
```

Def unnest

```
def unnest( transformationContext : String = "",
            callSite : CallSite = CallSite("Not Provided"),
            stageThreshold : Long = 0,
            totalThreshold : Long = 0
          ) : DynamicFrame
```

Restituisce un nuovo oggetto DynamicFrame con tutte le strutture annidate appiattite. I nomi vengono creati usando il carattere "." (punto).

Ad esempio, supponiamo di avere un DynamicFrame con lo schema seguente.

```
{{{
  root
  |-- name: string
  |-- age: int
  |-- address: struct
  |    |-- state: string
  |    |-- city: string
```

```
}}}
```

La chiamata seguente annulla l'annidamento della struttura address.

```
{{{
  df.unnest()
}}}
```

Lo schema risultante è il seguente.

```
{{{
  root
  |-- name: string
  |-- age: int
  |-- address.state: string
  |-- address.city: string
}}}
```

Questo metodo, inoltre, annulla l'annidamento delle strutture all'interno di array. Tuttavia, per motivi storici, ai nomi di tali campi vengono anteposti il nome della matrice di chiusura e "`.val`".

Def unnest DDBJson

```
unnestDDBJson(transformationContext : String = "",
            callSite : CallSite = CallSite("Not Provided"),
            stageThreshold : Long = 0,
            totalThreshold : Long = 0): DynamicFrame
```

Snidifica le colonne nidificate in un `DynamicFrame` che si trovano specificamente nella struttura JSON di DynamoDB e restituisce un nuovo `DynamicFrame` non annidato. Le colonne che sono di un array di struct non verranno annidate. Si noti che si tratta di un tipo specifico di trasformazione di snidamento che si comporta in modo diverso dalla normale trasformazione di `unnest` e richiede che i dati siano già nella struttura JSON di DynamoDB. Per ulteriori informazioni, consulta DynamoDB JSON.

Ad esempio, lo schema di lettura di un'esportazione con la struttura JSON DynamoDB potrebbe apparire come segue:

```
root
```

```
|-- Item: struct
|    |-- ColA: struct
|    |    |-- S: string
|    |-- ColB: struct
|    |    |-- S: string
|    |-- ColC: struct
|    |    |-- N: string
|    |-- ColD: struct
|    |    |-- L: array
|    |    |    |-- element: null
```

La trasformazione di `unnestDDBJson()` lo convertirebbe in:

```
root
|-- ColA: string
|-- ColB: string
|-- ColC: string
|-- ColD: array
|    |-- element: null
```

L'esempio di codice seguente mostra come utilizzare il connettore di esportazione AWS Glue DynamoDB, richiamare un unnest JSON di DynamoDB e stampare il numero di partizioni:

```
import com.amazonaws.services.glue.GlueContext
import com.amazonaws.services.glue.util.GlueArgParser
import com.amazonaws.services.glue.util.Job
import com.amazonaws.services.glue.util.JsonOptions
import com.amazonaws.services.glue.DynamoDbDataSink
import org.apache.spark.SparkContext
import scala.collection.JavaConverters._


object GlueApp {

  def main(sysArgs: Array[String]): Unit = {
    val glueContext = new GlueContext(SparkContext.getOrCreate())
    val args = GlueArgParser.getResolvedOptions(sysArgs, Seq("JOB_NAME").toArray)
    Job.init(args("JOB_NAME"), glueContext, args.asJava)

    val dynamicFrame = glueContext.getSourceWithFormat(
      connectionType = "dynamodb",
      options = JsonOptions(Map(
        "dynamodb.export" -> "ddb",
```

```
        "dynamodb.tableArn" -> "<test_source>",
        "dynamodb.s3.bucket" -> "<bucket name>",
        "dynamodb.s3.prefix" -> "<bucket prefix>",
        "dynamodb.s3.bucketOwner" -> "<account_id of bucket>",
      ))
    ).getDynamicFrame()

    val unnested = dynamicFrame.unnestDDBJson()
    print(unnested.getNumPartitions())

    Job.commit()
  }

}
```

Def withFrameSchema

```
def withFrameSchema( getSchema : () => Schema ) : DynamicFrame
```

- `getSchema` — Una funzione che restituisce lo schema da usare. È specificata come funzione a zero parametri per posticipare i calcoli potenzialmente onerosi.

Imposta lo schema di questo oggetto `DynamicFrame` sul valore specificato. Viene usato per lo più internamente per evitare ricalcoli dello schema onerosi. Lo schema passato deve contenere tutte le colonne presenti nei dati.

Def withName

```
def withName( name : String ) : DynamicFrame
```

- `name` — Il nuovo nome da usare.

Restituisce una copia di questo oggetto `DynamicFrame` con un nuovo nome.

Def withTransformationContext

```
def withTransformationContext( ctx : String ) : DynamicFrame
```

Restituisce una copia di questo oggetto `DynamicFrame` con il contesto di trasformazione specificato.

## L' DynamicFrame oggetto

Pacchetto: com.amazonaws.services.glue

```
object DynamicFrame
```

## Applicazione di def

```
def apply( df : DataFrame,
           glueContext : GlueContext
         ) : DynamicFrame
```

## Def emptyDynamicFrame

```
def emptyDynamicFrame( glueContext : GlueContext ) : DynamicFrame
```

## Def fromPythonRDD

```
def fromPythonRDD( rdd : JavaRDD[Array[Byte]],
                   glueContext : GlueContext
                 ) : DynamicFrame
```

## Def ignoreErrors

```
def ignoreErrors( fn : DynamicRecord => DynamicRecord ) : DynamicRecord
```

## Def inlineErrors

```
def inlineErrors( msg : String,
                  callSite : CallSite
                ) : (DynamicRecord => DynamicRecord)
```

## Errori di def newFrameWith

```
def newFrameWithErrors( prevFrame : DynamicFrame,
                        rdd : RDD[DynamicRecord],
                        name : String = "",
                        transformationContext : String = "",
```

```
                        callSite : CallSite,
                        stageThreshold : Long,
                        totalThreshold : Long
                      ) : DynamicFrame
```

## AWS Glue DynamicRecord Classe Scala

### Argomenti


- Def addField
- Def dropField
- Def setError
- Def isError
- Def getError
- Def clearError
- Def scrittura
- Def readFields
- Def clone
- Def schema
- Def getRoot
- Def toJson
- Def getFieldNode
- Def getField
- Def hashCode
- Def equals
- DynamicRecord oggetto
- RecordTraverser tratto


### Pacchetto: com.amazonaws.services.glue

```
class DynamicRecord extends Serializable with Writable with Cloneable
```

Un `DynamicRecord` è una struttura di dati autodescrittiva che rappresenta una riga di dati nel set di dati in fase di elaborazione. È autodescrittiva nel senso che puoi ottenere lo schema della riga

rappresentata da `DynamicRecord` controllando il record stesso. Un `DynamicRecord` è simile a un `Row` in Apache Spark.

Def addField

```
def addField( path : String,
              dynamicNode : DynamicNode
            ) : Unit
```

Aggiunge un DynamicNode al percorso specificato.

- `path` — Percorso per il campo da aggiungere.
- `dynamicNode` — Il DynamicNode da aggiungere al percorso specificato.

Def dropField

```
def dropField(path: String, underRename: Boolean = false): Option[DynamicNode]
```

Rilascia un DynamicNode dal percorso specificato e restituisce il nodo rilasciato se non c'è un array nel percorso specificato.

- `path` — Percorso per il campo da rilasciare.
- `underRenamedropField` — True se fa parte di un'azione di rinomina o false in caso contrario (false per impostazione predefinita).

Restituisce una `scala.Option Option` (DynamicNode).

Def setError

```
def setError( error : Error )
```

Imposta il record come un record di errore, come specificato dal parametro `error`.

Restituisce una `DynamicRecord`.

Def isError

```
def isError
```

Verifica se il record è un record di errore.

Def getError

```
def getError
```

Ottiene `Error` se il record è un record di errore. Restituisce `scala.Some` Some (Error) se il record è un record di errore o altrimenti `scala.None`.

Def clearError

```
def clearError
```

Imposta `Error` su `scala.None.None`.

Def scrittura

```
override def write( out : DataOutput ) : Unit
```

Def readFields

```
override def readFields( in : DataInput ) : Unit
```

Def clone

```
override def clone : DynamicRecord
```

Clona questo record in un nuovo `DynamicRecord` e lo restituisce.

Def schema

```
def schema
```

Ottiene lo `Schema` controllando il record.

Def getRoot

```
def getRoot : ObjectNode
```

Ottiene la radice `ObjectNode` per il record.

Def toJson

```
def toJson : String
```

Ottiene la stringa JSON per il record.

Def getFieldNode

```
def getFieldNode( path : String ) : Option[DynamicNode]
```

Ottiene il valore del campo nel `path` specificato come opzione di `DynamicNode`.

Restituisce `scala.Some` Some (DynamicNode) se il campo esiste, altrimenti `scala.None.None`.

Def getField

```
def getField( path : String ) : Option[Any]
```

Ottiene il valore del campo nel `path` specificato come opzione di `DynamicNode`.

Restituisce `scala.Some` Some (valore).

Def hashCode

```
override def hashCode : Int
```

Def equals

```
override def equals( other : Any )
```

DynamicRecord oggetto

```
object DynamicRecord
```

Applicazione di def

```
def apply( row : Row,
           schema : SparkStructType )
```

Applica il metodo per convertire un Apache Spark SQL `Row` in un DynamicRecord.

- `row` — Una `Row` Spark SQL.
- `schema` — Lo `Schema` della riga in questione.

Restituisce una `DynamicRecord`.

RecordTraverser tratto

```
trait RecordTraverser {
  def nullValue(): Unit
  def byteValue(value: Byte): Unit
  def binaryValue(value: Array[Byte]): Unit
  def booleanValue(value: Boolean): Unit
  def shortValue(value: Short) : Unit
  def intValue(value: Int) : Unit
  def longValue(value: Long) : Unit
  def floatValue(value: Float): Unit
  def doubleValue(value: Double): Unit
  def decimalValue(value: BigDecimal): Unit
  def stringValue(value: String): Unit
  def dateValue(value: Date): Unit
  def timestampValue(value: Timestamp): Unit
  def objectStart(length: Int): Unit
  def objectKey(key: String): Unit
  def objectEnd(): Unit
  def mapStart(length: Int): Unit
  def mapKey(key: String): Unit
  def mapEnd(): Unit
  def arrayStart(length: Int): Unit
  def arrayEnd(): Unit
}
```

AWS GlueScala GlueContext APIs

Pacchetto: com.amazonaws.services.glue

```
class GlueContext extends SQLContext(sc) (
           @transient val sc : SparkContext,
           val defaultSourcePartitioner : PartitioningStrategy )
```

`GlueContext` è il punto di ingresso per la lettura e la scrittura di un DynamicFrame da e verso Amazon Simple Storage Service (Amazon S3), il catalogo dati AWS Glue, JDBC e così via. Questa

classe fornisce funzioni di utilità per creare oggetti DataSource tratto e DataSink che possono in cambio essere utilizzati per leggere e scrivere DynamicFrame.

GlueContext può anche essere usato per impostare un numero target di partizioni (per impostazione predefinita 20) nel DynamicFrame se il numero di partizioni create dalla sorgente è inferiore alla soglia minima delle partizioni (per impostazione predefinita 10).

def Colonne addIngestionTime

```
def addIngestionTimeColumns(
         df : DataFrame,
         timeGranularity : String = "") : dataFrame
```

Aggiunge colonne del tempo di importazione dati come ingest_year, ingest_month, ingest_day, ingest_hour, ingest_minute al DataFrame di input. Questa funzione viene generata automaticamente nello script generato da AWS Glue quando si specifica una tabella del catalogo dati con Amazon S3 come destinazione. Questa funzione aggiorna automaticamente la partizione con le colonne del tempo di importazione dati nella tabella di output. Ciò consente ai dati di output di venire partizionati automaticamente nel tempo di importazione dati senza necessitare di colonne di tempo di inserimento esplicite nei dati di input.

- dataFrame: il dataFrame al quale aggiungere le colonne del tempo di importazione dati.
- timeGranularity: la granularità delle colonne temporali. I valori validi sono “day”, “hour” e “minute”. Ad esempio, se “hour" viene passato alla funzione, il dataFrame originale avrà “ingest_year”, “ingest_month”, “ingest_day” e “ingest_hour” colonne temporali aggiunte.

Restituisce il frame di dati dopo l'aggiunta di colonne di granularità di tempo.

Esempio:

```
glueContext.addIngestionTimeColumns(dataFrame, "hour")
```

def createDataFrame FromOptions

```
def createDataFrameFromOptions( connectionType : String,
                        connectionOptions : JsonOptions,
                        transformationContext : String = "",
                        format : String = null,
                        formatOptions : JsonOptions = JsonOptions.empty
```

```
                        ) : DataSource
```

Restituisce un `DataFrame` creato con la connessione e il formato specificati. Usa questa funzione solo con le sorgenti di streaming AWS Glue.

- `connectionType`: il tipo di connessione streaming. I valori validi includono `kinesis` e `kafka`.
- `connectionOptions`: opzioni di connessione, che sono diverse per Kinesis e Kafka. È possibile trovare l'elenco di tutte le opzioni di connessione per ogni origine dati di streaming all'indirizzo Tipi e opzioni di connessione per ETL in AWS Glue per Spark. Di seguito vengono illustrate le differenze delle opzioni di connessione di streaming:
  - Le origini di streaming di Kinesis richiedono `streamARN`, `startingPosition`, `inferSchema` e `classification`.
  - Le origini di streaming di Kafka richiedono `connectionName`, `topicName`, `startingOffsets`, `inferSchema` e `classification`.
- `transformationContext`: il contesto di trasformazione da utilizzare (facoltativo).
- `format`: una specifica del formato (facoltativo). Viene usata per una connessione Amazon S3 o AWS Glue che supporta più formati. Per ulteriori informazioni sui formati supportati, consulta Opzioni del formato dati per input e output in AWS Glue per Spark.
- `formatOptions`: opzioni di formattazione per il formato specificato. Per ulteriori informazioni sulle opzioni di formato supportate, consulta Opzioni del formato dei dati.

Esempio per l'origine di streaming Amazon Kinesis:

```
val data_frame_datasource0 =
glueContext.createDataFrameFromOptions(transformationContext = "datasource0",
 connectionType = "kinesis",
connectionOptions = JsonOptions("""{"streamName": "example_stream", "startingPosition":
 "TRIM_HORIZON", "inferSchema": "true", "classification": "json"}}""" ))
```

Esempio per l'origine di streaming Kafka:

```
val data_frame_datasource0 =
glueContext.createDataFrameFromOptions(transformationContext = "datasource0",
 connectionType = "kafka",
connectionOptions = JsonOptions("""{"connectionName": "example_connection",
 "topicName": "example_topic", "startingPosition": "earliest", "inferSchema": "false",
 "classification": "json", "schema":"`column1` STRING, `column2` STRING"}""" ))
```

forEachBatch

### `forEachBatch(frame, batch_function, options)`

Applica la `batch_function` passata a ogni micro batch che viene letto dall'origine di streaming.

- `frame`— Il file DataFrame contenente il microbatch corrente.
- `batch_function`: una funzione che verrà applicata per ogni micro batch.
- `options`: una raccolta di coppie chiave-valore che contiene informazioni su come elaborare micro batch. Sono richieste le seguenti opzioni:
  - `windowSize`: la quantità di tempo da dedicare all'elaborazione di ciascun batch.
  - `checkpointLocation`: la posizione in cui sono archiviati i checkpoint per il processo ETL di streaming.
  - `batchMaxRetries`: numero massimo di tentativi per riprovare il processo se il batch ha esito negativo. Il valore predefinito è 3. Questa opzione è configurabile solo per Glue versione 2.0 e successive.

Esempio:

```
glueContext.forEachBatch(data_frame_datasource0, (dataFrame: Dataset[Row], batchId:
 Long) =>
   {
      if (dataFrame.count() > 0)
        {
          val datasource0 = DynamicFrame(glueContext.addIngestionTimeColumns(dataFrame,
 "hour"), glueContext)
          // @type: DataSink
          // @args: [database = "tempdb", table_name = "fromoptionsoutput",
 stream_batch_time = "100 seconds",
          //      stream_checkpoint_location = "s3://from-options-testing-eu-central-1/
fromOptionsOutput/checkpoint/",
          //      transformation_ctx = "datasink1"]
          // @return: datasink1
          // @inputs: [frame = datasource0]
          val options_datasink1 = JsonOptions(
             Map("partitionKeys" -> Seq("ingest_year", "ingest_month","ingest_day",
 "ingest_hour"),
             "enableUpdateCatalog" -> true))
          val datasink1 = glueContext.getCatalogSink(
             database = "tempdb",
```

```
            tableName = "fromoptionsoutput",
            redshiftTmpDir = "",
            transformationContext = "datasink1",
            additionalOptions = options_datasink1).writeDynamicFrame(datasource0)
        }
   }, JsonOptions("""{"windowSize" : "100 seconds",
         "checkpointLocation" : "s3://from-options-testing-eu-central-1/
fromOptionsOutput/checkpoint/"}"""))
```

- def getCatalogSink

```
def getCatalogSink( database : String,
        tableName : String,
        redshiftTmpDir : String = "",
        transformationContext : String = ""
        additionalOptions: JsonOptions = JsonOptions.empty,
        catalogId: String = null
) : DataSink
```

Crea un DataSink che scrive in una posizione specificata di una tabella definita nel catalogo dati.

- `database` — Il nome del database nel catalogo di dati.
- `tableName` — Il nome della tabella nel catalogo dati.
- `redshiftTmpDir` — La directory di gestione temporanea da utilizzare con alcuni sink di dati. Impostato su per impostazione predefinita.
- `transformationContext` — Il contesto di trasformazione associato al sink che i segnalibri di processo utilizzano. Impostato su per impostazione predefinita.
- `additionalOptions`: opzioni aggiuntive fornite a AWS Glue.
- `catalogId` — L'ID catalogo (ID account) relativo al catalogo dati a cui si accede. Se null, viene utilizzato l'ID account predefinito del chiamante.

Restituisce il `DataSink`.

def getCatalogSource

```
def getCatalogSource( database : String,
                      tableName : String,
                      redshiftTmpDir : String = "",
```

```
                    transformationContext : String = ""
                    pushDownPredicate : String = " "
                    additionalOptions: JsonOptions = JsonOptions.empty,
                    catalogId: String = null
                  ) : DataSource
```

Crea un DataSource tratto che legge dati da una definizione di tabella nel catalogo di dati.

- `database` — Il nome del database nel catalogo di dati.
- `tableName` — Il nome della tabella nel catalogo dati.
- `redshiftTmpDir` — La directory di gestione temporanea da utilizzare con alcuni sink di dati. Impostato su per impostazione predefinita.
- `transformationContext` — Il contesto di trasformazione associato al sink che i segnalibri di processo utilizzano. Impostato su per impostazione predefinita.
- `pushDownPredicate`: filtra le partizioni senza dover elencare e leggere tutti i file nel set di dati. Per ulteriori informazioni, consultare Prefiltraggio con i predicati pushdown.
- `additionalOptions`: una raccolta di coppie nome/valore opzionali. Le opzioni possibili includono quelle elencate in Tipi e opzioni di connessione per ETL in AWS Glue per Spark ad eccezione di `endpointUrl`, `streamName`, `bootstrap.servers`, `security.protocol`, `topicName`, `classification` e `delimiter`. Un'altra opzione supportata è `catalogPartitionPredicate`:

  `catalogPartitionPredicate`: è possibile passare un'espressione di catalogo per filtrare in base alle colonne di indice. Questo esegue il push down del filtro sul lato server. Per ulteriori informazioni, consultare la pagina relativa agli indici di partizionamento di AWS Glue. Tenere presente che `push_down_predicate` e `catalogPartitionPredicate` usano sintassi diverse. Il primo utilizza la sintassi standard Spark SQL e il secondo utilizza il parser JSQL.
- `catalogId` — L'ID catalogo (ID account) relativo al catalogo dati a cui si accede. Se null, viene utilizzato l'ID account predefinito del chiamante.

Restituisce il `DataSource`.

Esempio di origine di streaming

```
val data_frame_datasource0 = glueContext.getCatalogSource(
    database = "tempdb",
    tableName = "test-stream-input",
```

```
    redshiftTmpDir = "",
    transformationContext = "datasource0",
    additionalOptions = JsonOptions("""{
        "startingPosition": "TRIM_HORIZON", "inferSchema": "false"}""")
    ).getDataFrame()
```

def ottenere JDBCSink

```
def getJDBCSink( catalogConnection : String,
                 options : JsonOptions,
                 redshiftTmpDir : String = "",
                 transformationContext : String = "",
                 catalogId: String = null
               ) : DataSink
```

Crea un oggetto DataSink che scrive in un database JDBC specificato in un oggetto `Connection` nel catalogo dati. L'oggetto `Connection` dispone di informazioni per connettersi a un sink JDBC, compresi URL, nome utente, password, VPC, sottorete e gruppi di sicurezza.

- `catalogConnection` — Il nome della connessione nel catalogo dati contenente l'URL JDBC su cui scrivere.
- `options`: una stringa di coppie nome-valore JSON che forniscono informazioni aggiuntive necessarie per scrivere su un datastore JDBC. Questo include:
  - dbtable (obbligatorio): il nome della tabella JDBC. Per i archivi dati JDBC che supportano schemi all'interno di un database, specificare `schema.table-name`. Se non viene fornito alcuno schema, viene usato lo schema "pubblico" predefinito. L'esempio seguente mostra un parametro options che punta a uno schema denominato `test` e a una tabella denominata `test_table` nel database `test_db`.

    ```
    options = JsonOptions("""{"dbtable": "test.test_table", "database": "test_db"}""")
    ```

  - database (obbligatorio): il nome del database JDBC.
  - Le eventuali opzioni aggiuntive trasmesse direttamente allo scrittore SparkSQL JDBC. Per ulteriori informazioni, consulta la pagina relativa all'origine dati Redshift per Spark.
- `redshiftTmpDir` — Una directory di gestione temporanea da utilizzare con alcuni sink di dati. Impostato su per impostazione predefinita.
- `transformationContext` — Il contesto di trasformazione associato al sink che i segnalibri di processo utilizzano. Impostato su per impostazione predefinita.

- `catalogId` — L'ID catalogo (ID account) relativo al catalogo dati a cui si accede. Se null, viene utilizzato l'ID account predefinito del chiamante.

Codice di esempio:

```
getJDBCSink(catalogConnection = "my-connection-name", options =
 JsonOptions("""{"dbtable": "my-jdbc-table", "database": "my-jdbc-db"}"""),
 redshiftTmpDir = "", transformationContext = "datasink4")
```

Restituisce il `DataSink`.

def getSink

```
def getSink( connectionType : String,
             connectionOptions : JsonOptions,
             transformationContext : String = ""
           ) : DataSink
```

Crea un file DataSink che scrive dati su una destinazione come Amazon Simple Storage Service (Amazon S3), JDBC o Glue Data Catalog o AWS un flusso di dati Apache Kafka o Amazon Kinesis.

- `connectionType` — Il tipo di connessione. Per informazioni, consulta the section called "Parametri di connessione".
- `connectionOptions`: una stringa di coppie nome-valore JSON che forniscono informazioni aggiuntive per stabilire la connessione con il sink dei dati. Per informazioni, consulta the section called "Parametri di connessione".
- `transformationContext` — Il contesto di trasformazione associato al sink che i segnalibri di processo utilizzano. Impostato su per impostazione predefinita.

Restituisce il `DataSink`.

Formato def getSinkWith

```
def getSinkWithFormat( connectionType : String,
                       options : JsonOptions,
                       transformationContext : String = "",
                       format : String = null,
                       formatOptions : JsonOptions = JsonOptions.empty
```

```
                    ) : DataSink
```

Crea un DataSink che scrive dati su una destinazione come Amazon S3, JDBC o il catalogo dati o un Apache Kafka o il flusso di dati Amazon Kinesis. Configura anche il formato per i dati da scrivere nella destinazione.

- `connectionType` — Il tipo di connessione. Per informazioni, consulta the section called "Parametri di connessione".
- `options`: una stringa di coppie nome-valore JSON che forniscono informazioni aggiuntive per stabilire connessioni con il sink dei dati. Per informazioni, consulta the section called "Parametri di connessione".
- `transformationContext` — Il contesto di trasformazione associato al sink che i segnalibri di processo utilizzano. Impostato su per impostazione predefinita.
- `format` — Il formato dei dati da scrivere sulla destinazione.
- `formatOptions`: una stringa di coppie nome-valore JSON che forniscono opzioni aggiuntive per la formattazione dei dati nella destinazione. Per informazioni, consulta Opzioni del formato dei dati.

Restituisce il `DataSink`.

def getSource

```
def getSource( connectionType : String,
               connectionOptions : JsonOptions,
               transformationContext : String = ""
               pushDownPredicate
             ) : DataSource
```

Crea un file DataSource tratto che legge i dati da una fonte come Amazon S3, JDBC o Glue Data Catalog. AWS Supporta anche origini dati di streaming Kafka e Kinesis.

- `connectionType` — Il tipo di origine dati. Per informazioni, consulta the section called "Parametri di connessione".
- `connectionOptions`: una stringa di coppie nome-valore JSON che forniscono informazioni aggiuntive per stabilire una connessione con l'origine dati. Per ulteriori informazioni, consulta the section called "Parametri di connessione".

  Un'origine di streaming Kinesis richiede le seguenti opzioni di connessione: `streamARN`, `startingPosition`, `inferSchema` e `classification`.

Un'origine di streaming Kinesis richiede le seguenti opzioni di connessione: `connectionName`, `topicName`, `startingOffsets`, `inferSchema` e `classification`.

- `transformationContext` — Il contesto di trasformazione associato al sink che i segnalibri di processo utilizzano. Impostato su per impostazione predefinita.
- `pushDownPredicate` — Predicato sulle colonne delle partizioni.

Restituisce il `DataSource`.

Esempio per l'origine di streaming Amazon Kinesis:

```
val kinesisOptions = jsonOptions()
data_frame_datasource0 = glueContext.getSource("kinesis",
 kinesisOptions).getDataFrame()

private def jsonOptions(): JsonOptions = {
    new JsonOptions(
      s"""{"streamARN": "arn:aws:kinesis:eu-central-1:123456789012:stream/
fromOptionsStream",
         |"startingPosition": "TRIM_HORIZON",
         |"inferSchema": "true",
         |"classification": "json"}""".stripMargin)
}
```

Esempio per l'origine di streaming Kafka:

```
val kafkaOptions = jsonOptions()
val data_frame_datasource0 = glueContext.getSource("kafka",
 kafkaOptions).getDataFrame()

private def jsonOptions(): JsonOptions = {
    new JsonOptions(
      s"""{"connectionName": "ConfluentKafka",
         |"topicName": "kafka-auth-topic",
         |"startingOffsets": "earliest",
         |"inferSchema": "true",
         |"classification": "json"}""".stripMargin)
 }
```

formato def getSourceWith

```
def getSourceWithFormat( connectionType : String,
                         options : JsonOptions,
                         transformationContext : String = "",
                         format : String = null,
                         formatOptions : JsonOptions = JsonOptions.empty
                       ) : DataSource
```

Crea un file DataSource tratto che legge i dati da una fonte come Amazon S3, JDBC o AWS Glue Data Catalog e imposta anche il formato dei dati archiviati nell'origine.

- `connectionType` – Il tipo dell'origine dati. Per informazioni, consulta the section called “Parametri di connessione”.
- `options`: una stringa di coppie nome/valore JSON che forniscono informazioni aggiuntive per stabilire una connessione all'origine dati. Per informazioni, consulta the section called “Parametri di connessione”.
- `transformationContext` – Il contesto di trasformazione associato al sink che deve essere usato dai segnalibri dei processi. Impostato su per impostazione predefinita.
- `format` – Il formato dei dati archiviati nell'origine. Quando il `connectionType` è "s3", è anche possibile specificare `format`. Le possibilità sono “avro”, “csv”, “grokLog”, “ion”, “json”, “xml”, “parquet” oppure “orc”.
- `formatOptions`: una stringa di coppie nome/valore JSON che forniscono opzioni aggiuntive per l'analisi dei dati nell'origine. Per informazioni, consulta Opzioni del formato dei dati.

Restituisce il `DataSource`.

Esempi

Crea un file DynamicFrame da un'origine dati che sia un file con valori separati da virgole (CSV) su Amazon S3:

```
val datasource0 = glueContext.getSourceWithFormat(
    connectionType="s3",
    options =JsonOptions(s"""{"paths": [ "s3://csv/nycflights.csv"]}"""),
    transformationContext = "datasource0",
    format = "csv",
    formatOptions=JsonOptions(s"""{"withHeader":"true","separator": ","}""")
```

```
    ).getDynamicFrame()
```

Crea un file DynamicFrame da un'origine dati che sia PostgreSQL utilizzando una connessione JDBC:

```
val datasource0 = glueContext.getSourceWithFormat(
    connectionType="postgresql",
    options =JsonOptions(s"""{
      "url":"jdbc:postgresql://databasePostgres-1.rds.amazonaws.com:5432/testdb",
      "dbtable": "public.company",
      "redshiftTmpDir":"",
      "user":"username",
      "password":"password123"
    }"""),
    transformationContext = "datasource0").getDynamicFrame()
```

Crea un file DynamicFrame da un'origine dati che è MySQL utilizzando una connessione JDBC:

```
 val datasource0 = glueContext.getSourceWithFormat(
    connectionType="mysql",
    options =JsonOptions(s"""{
      "url":"jdbc:mysql://databaseMysql-1.rds.amazonaws.com:3306/testdb",
      "dbtable": "athenatest_nycflights13_csv",
      "redshiftTmpDir":"",
      "user":"username",
      "password":"password123"
    }"""),
    transformationContext = "datasource0").getDynamicFrame()
```

def getSparkSession

```
def getSparkSession : SparkSession
```

Ottiene l'oggetto `SparkSession` associato a questo GlueContext. Utilizzate questo SparkSession oggetto per registrare tabelle e UDFs per utilizzarlo con `DataFrame` created from DynamicFrames.

Restituisce il SparkSession.

def startTransaction

```
def startTransaction(readOnly: Boolean):String
```

Avvia una nuova transazione. Chiama internamente l'API Lake Formation startTransaction.

- `readOnly`: (booleano) indica se questa transazione debba essere di sola lettura o lettura e scrittura. Le scritture effettuate utilizzando un ID transazione di sola lettura verranno rifiutate. Il commit delle transazioni di sola lettura non deve essere eseguito.

Restituisce l'ID transazione.

def commitTransaction

```
def commitTransaction(transactionId: String, waitForCommit: Boolean): Boolean
```

Tenta di eseguire il commit della transazione specificata. `commitTransaction` può restituire prima che la transazione abbia terminato il commit. Chiama internamente l'API Lake Formation commitTransaction.

- `transactionId`: (stringa) la transazione di cui eseguire il commit.
- `waitForCommit`: (booleano) determina se il `commitTransaction` restituisce immediatamente. Il valore di default è true. Se false, `commitTransaction` effettua il polling e aspetta che sia stato eseguito il commit della transazione. Il tempo di attesa è limitato a 1 minuto utilizzando il backoff esponenziale con un massimo di 6 tentativi.

Restituisce un valore booleano per indicare se il commit sia stato eseguito o meno.

def cancelTransaction

```
def cancelTransaction(transactionId: String): Unit
```

Tenta di annullare la transazione specificata. Richiama internamente l'CancelTransactionAPI Lake Formation.

- `transactionId`: (stringa) la transazione da annullare.

Restituisce un'eccezione `TransactionCommittedException` se è stato precedentemente eseguito il commit della transazione.

def this

```
def this( sc : SparkContext,
          minPartitions : Int,
          targetPartitions : Int )
```

Crea un oggetto `GlueContext` utilizzando lo `SparkContext` specificato, le partizioni minime e quelle target.

- `sc` — Il carattere `SparkContext`.
- `minPartitions` — Il numero minimo di partizioni.
- `targetPartitions` — Il numero target di partizioni.

Restituisce il `GlueContext`.

def this

```
def this( sc : SparkContext )
```

Crea un oggetto `GlueContext` con il `SparkContext` fornito. Imposta il numero minimo di partizioni a 10 e a le partizioni target a 20.

- `sc` — Il carattere `SparkContext`.

Restituisce il `GlueContext`.

def this

```
def this( sparkContext : JavaSparkContext )
```

Crea un oggetto `GlueContext` con il `JavaSparkContext` fornito. Imposta il numero minimo di partizioni a 10 e a le partizioni target a 20.

- `sparkContext` — Il carattere `JavaSparkContext`.

Restituisce il `GlueContext`.

MappingSpec

Pacchetto: com.amazonaws.services.glue

MappingSpec classe di casi

```
case class MappingSpec( sourcePath: SchemaPath,
                        sourceType: DataType,
                        targetPath: SchemaPath,
                        targetType: DataTyp
                       ) extends Product4[String, String, String, String] {
  override def _1: String = sourcePath.toString
  override def _2: String = ExtendedTypeName.fromDataType(sourceType)
  override def _3: String = targetPath.toString
  override def _4: String = ExtendedTypeName.fromDataType(targetType)
}
```

- `sourcePath`: il `SchemaPath` del campo di origine.
- `sourceType`: il `DataType` del campo di origine.
- `targetPath`: il `SchemaPath` del campo di destinazione.
- `targetType`: il `DataType` del campo di destinazione.

Un `MappingSpec` specifica una mappatura da un percorso di origine e un tipo di dati di origine a un percorso di destinazione e un tipo di dati di destinazione. Il valore al percorso di origine nel frame di origine viene visualizzato nel frame di destinazione presso il percorso di destinazione. Il tipo di dati di origine è trasmesso al tipo di dati di destinazione.

Si estende da `Product4` in modo che sia possibile gestire qualsiasi `Product4` all'interno dell'interfaccia `applyMapping`.

MappingSpec oggetto

```
object MappingSpec
```

L'oggetto `MappingSpec` dispone dei seguenti membri:

Val orderingByTarget

```
val orderingByTarget: Ordering[MappingSpec]
```

Applicazione di def

```
def apply( sourcePath : String,
           sourceType : DataType,
           targetPath : String,
           targetType : DataType
         ) : MappingSpec
```

Crea un `MappingSpec`.

- `sourcePath`: rappresentazione di stringa del percorso di origine.
- `sourceType`: sorgente `DataType`.
- `targetPath`: rappresentazione di stringa del percorso di destinazione.
- `targetType`: destinazione `DataType`.

Restituisce un `MappingSpec`.

Applicazione di def

```
def apply( sourcePath : String,
           sourceTypeString : String,
           targetPath : String,
           targetTypeString : String
         ) : MappingSpec
```

Crea un `MappingSpec`.

- `sourcePath`: rappresentazione di stringa del percorso di origine.
- `sourceType`: rappresentazione di stringa del tipo di dati di origine.
- `targetPath`: rappresentazione di stringa del percorso di destinazione.
- `targetType`: rappresentazione di stringa del tipo di dati di destinazione.

Restituisce un MappingSpec.

Applicazione di def

```
def apply( product : Product4[String, String, String, String] ) : MappingSpec
```

Crea un `MappingSpec`.

- `product`: il `Product4` del percorso di origine, il tipo di dati di origine, il percorso di destinazione e il tipo di dati di destinazione.

Restituisce una `MappingSpec`.

AWS GlueScala ResolveSpec APIs

Argomenti

- ResolveSpec oggetto
- ResolveSpec classe case

Pacchetto: com.amazonaws.services.glue

ResolveSpec oggetto

ResolveSpec

```
object ResolveSpec
```

Def

```
def apply( path : String,
           action : String
         ) : ResolveSpec
```

Crea un `ResolveSpec`.

- `path`: rappresentazione di stringa del campo di scelta che deve essere risolto.
- `action`: un'operazione di risoluzione. L'operazione può essere una delle seguenti: `Project`, `KeepAsStruct` o `Cast`.

Restituisce il `ResolveSpec`.

Def

```
def apply( product : Product2[String, String] ) : ResolveSpec
```

Crea un `ResolveSpec`.

- `product` — `Product2` di: percorso di origine, operazione di risoluzione.

Restituisce il `ResolveSpec`.

ResolveSpec classe case

```
case class ResolveSpec extends Product2[String, String]  (
           path : SchemaPath,
           action : String )
```

Crea un `ResolveSpec`.

- `path`: `SchemaPath` del campo di scelta che deve essere risolto.
- `action`: un'operazione di risoluzione. L'operazione può essere una delle seguenti: `Project`, `KeepAsStruct` o `Cast`.

ResolveSpec metodi def

```
def _1 : String
```

```
def _2 : String
```

AWS GlueScala ArrayNode APIs

Pacchetto: com.amazonaws.services.glue.types

ArrayNode classe di casi

ArrayNode

```
case class ArrayNode extends DynamicNode  (
           value : ArrayBuffer[DynamicNode] )
```

ArrayNode metodi def

```
def add( node : DynamicNode )
```

```
def clone
```

```
def equals( other : Any )
```

```
def get( index : Int ) : Option[DynamicNode]
```

```
def getValue
```

```
def hashCode : Int
```

```
def isEmpty : Boolean
```

```
def nodeType
```

```
def remove( index : Int )
```

```
def this
```

```
def toIterator : Iterator[DynamicNode]
```

```
def toJson : String
```

```
def update( index : Int,
            node : DynamicNode )
```

## AWS GlueScala BinaryNode APIs

Pacchetto: com.amazonaws.services.glue.types

### BinaryNode classe di casi

### BinaryNode

```
case class BinaryNode extends ScalarNode(value, TypeCode.BINARY)  (
           value : Array[Byte] )
```

BinaryNode campi val

- `ordering`

BinaryNode metodi def

```
def clone
```

```
def equals( other : Any )
```

```
def hashCode : Int
```

AWS GlueScala BooleanNode APIs

Pacchetto: com.amazonaws.services.glue.types

BooleanNode classe di casi

BooleanNode

```
case class BooleanNode extends ScalarNode(value, TypeCode.BOOLEAN)  (
           value : Boolean )
```

BooleanNode campi val

- `ordering`

BooleanNode metodi def

```
def equals( other : Any )
```

AWS GlueScala ByteNode APIs

Pacchetto: com.amazonaws.services.glue.types

ByteNode classe di casi

ByteNode

```
case class ByteNode extends ScalarNode(value, TypeCode.BYTE)  (
```

```
            value : Byte )
```

ByteNode campi val

- `ordering`

ByteNode metodi def

```
def equals( other : Any )
```

AWS GlueScala DateNode APIs

Pacchetto: com.amazonaws.services.glue.types

DateNode classe di casi

DateNode

```
case class DateNode extends ScalarNode(value, TypeCode.DATE)  (
            value : Date )
```

DateNode campi val

- `ordering`

DateNode metodi def

```
def equals( other : Any )
```

```
def this( value : Int )
```

AWS GlueScala DecimalNode APIs

Pacchetto: com.amazonaws.services.glue.types

DecimalNode classe di casi

DecimalNode

```
case class DecimalNode extends ScalarNode(value, TypeCode.DECIMAL)  (
```

```
            value : BigDecimal )
```

DecimalNode campi val

- `ordering`

DecimalNode metodi def

```
def equals( other : Any )
```

```
def this( value : Decimal )
```

AWS GlueScala DoubleNode APIs

Pacchetto: com.amazonaws.services.glue.types

DoubleNode classe di casi

DoubleNode

```
case class DoubleNode extends ScalarNode(value, TypeCode.DOUBLE)  (
            value : Double )
```

DoubleNode campi val

- `ordering`

DoubleNode metodi def

```
def equals( other : Any )
```

AWS GlueScala DynamicNode APIs

Argomenti

- DynamicNode classe
- DynamicNode oggetto

Pacchetto: com.amazonaws.services.glue.types

### DynamicNode classe

#### DynamicNode

```
class DynamicNode extends Serializable with Cloneable
```

#### DynamicNode metodi def

```
def getValue : Any
```

Ottenere un valore normale associato al record corrente:

```
def nodeType : TypeCode
```

```
def toJson : String
```

Metodo per il debug:

```
def toRow( schema : Schema,
           options : Map[String, ResolveOption]
         ) : Row
```

```
def typeName : String
```

### DynamicNode oggetto

#### DynamicNode

```
object DynamicNode
```

#### DynamicNode metodi def

```
def quote( field : String,
           useQuotes : Boolean
         ) : String
```

```
def quote( node : DynamicNode,
           useQuotes : Boolean
         ) : String
```

EvaluateDataQuality classe

AWS Glue Data Quality è in versione di anteprima AWS Glue ed è soggetto a modifiche.

Pacchetto: com.amazonaws.services.glue.dq

```
object EvaluateDataQuality
```

Applicazione di def

```
def apply(frame: DynamicFrame,
            ruleset: String,
            publishingOptions: JsonOptions = JsonOptions.empty): DynamicFrame
```

Valuta un set di regole di qualità dei dati rispetto ai dati in un `DynamicFrame` e restituisce un nuovo `DynamicFrame` con i risultati della valutazione. Per ulteriori informazioni su AWS Glue Data Quality, consultaAWS Glue Qualità dei dati.

- `frame`: il `DynamicFrame` di cui desideri valutare la qualità dei dati.
- `ruleset`: un set di regole del Data Quality Definition Language (DQDL) in formato stringa. Per ulteriori informazioni su DQDL, consulta la guida di Riferimento a Data Quality Definition Language (DQDL).
- `publishingOptions`: un dizionario che specifica le seguenti opzioni per la pubblicazione dei risultati e dei parametri di valutazione:
  - `dataQualityEvaluationContext`— Una stringa che specifica lo spazio dei nomi in cui AWS Glue deve pubblicare le Amazon CloudWatch metriche e i risultati sulla qualità dei dati. Le metriche aggregate vengono visualizzate in CloudWatch, mentre i risultati completi vengono visualizzati nell'interfaccia AWS Glue Studio.
    - Obbligatorio: no
    - Valore predefinito: `default_context`
  - `enableDataQualityCloudWatchMetrics`— Speciifica se i risultati della valutazione della qualità dei dati devono essere pubblicati su. CloudWatch Uno spazio dei nomi per i parametri viene specificato utilizzando l'opzione `dataQualityEvaluationContext`.
    - Obbligatorio: no
    - Valore predefinito: False

- `enableDataQualityResultsPublishing`: specifica se i risultati della qualità dei dati devono essere visibili nella scheda Data Quality (Qualità dei dati) nell'interfaccia di AWS Glue Studio.
  - Obbligatorio: no
  - Valore predefinito: true
- `resultsS3Prefix`— Speciifica la posizione di Amazon S3 in cui AWS Glue può scrivere i risultati della valutazione della qualità dei dati.
  - Obbligatorio: no
  - Valore predefinito: "" (stringa vuota)

Esempio

Il codice di esempio seguente dimostra come valutare la qualità dei dati per un `DynamicFrame` prima di eseguire una trasformazione `SelectFields`. Lo script verifica che tutte le regole di qualità dei dati siano rispettate prima di tentare la trasformazione.

```
import com.amazonaws.services.glue.GlueContext
import com.amazonaws.services.glue.MappingSpec
import com.amazonaws.services.glue.errors.CallSite
import com.amazonaws.services.glue.util.GlueArgParser
import com.amazonaws.services.glue.util.Job
import com.amazonaws.services.glue.util.JsonOptions
import org.apache.spark.SparkContext
import scala.collection.JavaConverters._
import com.amazonaws.services.glue.dq.EvaluateDataQuality

object GlueApp {
  def main(sysArgs: Array[String]) {
    val spark: SparkContext = new SparkContext()
    val glueContext: GlueContext = new GlueContext(spark)
    // @params: [JOB_NAME]
    val args = GlueArgParser.getResolvedOptions(sysArgs, Seq("JOB_NAME").toArray)
    Job.init(args("JOB_NAME"), glueContext, args.asJava)

    // Create DynamicFrame with data
    val Legislators_Area = glueContext.getCatalogSource(database="legislators",
 tableName="areas_json", transformationContext="S3bucket_node1").getDynamicFrame()

    // Define data quality ruleset
    val DQ_Ruleset = """
      Rules = [ColumnExists "id"]
```

```
    """

    // Evaluate data quality
    val DQ_Results = EvaluateDataQuality.apply(frame=Legislators_Area,
 ruleset=DQ_Ruleset, publishingOptions=JsonOptions("""{"dataQualityEvaluationContext":
 "Legislators_Area", "enableDataQualityMetrics": "true",
 "enableDataQualityResultsPublishing": "true"}"""))
    assert(DQ_Results.filter(_.getField("Outcome").contains("Failed")).count == 0,
 "Failing DQ rules for Legislators_Area caused the job to fail.")

    // Script generated for node Select Fields
    val SelectFields_Results = Legislators_Area.selectFields(paths=Seq("id", "name"),
 transformationContext="Legislators_Area")

    Job.commit()
  }
}
```

AWS GlueScala FloatNode APIs

Pacchetto: com.amazonaws.services.glue.types

FloatNode classe di casi

FloatNode

```
case class FloatNode extends ScalarNode(value, TypeCode.FLOAT)  (
           value : Float )
```

FloatNode campi val

- `ordering`

FloatNode metodi def

```
def equals( other : Any )
```

FillMissingValues classe

Pacchetto: com.amazonaws.services.glue.ml

```
object FillMissingValues
```

Applicazione di def

```
def apply(frame: DynamicFrame,
          missingValuesColumn: String,
          outputColumn: String = "",
          transformationContext: String = "",
          callSite: CallSite = CallSite("Not provided", ""),
          stageThreshold: Long = 0,
          totalThreshold: Long = 0): DynamicFrame
```

Riempie i valori mancanti di un frame dinamico in una colonna specificata e restituisce un frame dinamico con stime in una nuova colonna. Per le righe senza valori mancanti, il valore della colonna specificato viene duplicato nella nuova colonna.

- `frame`— La cartella DynamicFrame in cui inserire i valori mancanti. Obbligatorio.
- `missingValuesColumn` — La colonna contenente valori mancanti (valori `null` e stringhe vuote). Obbligatorio.
- `outputColumn` — Il nome della nuova colonna che conterrà i valori stimati per tutte le righe il cui valore era mancante. Facoltativo; il valore predefinito è il valore di `missingValuesColumn` con suffisso formato da `"_filled"`.
- `transformationContext` — Una stringa univoca usata per identificare le informazioni sullo stato (facoltativo).
- `callSite` — Usato per fornire informazioni sul contesto per la segnalazione degli errori (facoltativo).
- `stageThreshold` — Il numero massimo di errori che si possono verificare nella trasformazione prima che venga arrestata (facoltativo, il valore di default è zero).
- `totalThreshold` — Il numero massimo di errori che si possono verificare in generale prima che l'elaborazione venga arrestata (facoltativo, il valore di default è zero).

Restituisce un nuovo frame dinamico con una colonna aggiuntiva che contiene stime per le righe con valori mancanti e il valore attuale per le altre righe.

FindMatches classe

Pacchetto: com.amazonaws.services.glue.ml

```
object FindMatches
```

Applicazione di def

```
def apply(frame: DynamicFrame,
            transformId: String,
            transformationContext: String = "",
            callSite: CallSite = CallSite("Not provided", ""),
            stageThreshold: Long = 0,
            totalThreshold: Long = 0,
            enforcedMatches: DynamicFrame = null): DynamicFrame,
   computeMatchConfidenceScores: Boolean
```

Trova le corrispondenze in un frame di input e restituisci un nuovo frame con una nuova colonna contenente un ID univoco per gruppo di corrispondenza.

- `frame`— La lingua DynamicFrame in cui trovare le corrispondenze. Obbligatorio.
- `transformId`— Un ID univoco associato alla FindMatches trasformazione da applicare al frame di input. Obbligatorio.
- `transformationContext` — Identificatore per questo `DynamicFrame`. Il `transformationContext` viene usato come chiave per lo stato dei segnalibro di processo che viene mantenuto tra esecuzioni. Opzionale.
- `callSite` — Usato per fornire informazioni sul contesto per la segnalazione degli errori. Questi valori vengono impostati automaticamente durante la chiamata da Python. Opzionale.
- `stageThreshold` — Il numero massimo di record di errore consentiti nel calcolo di questo `DynamicFrame` prima di generare un'eccezione, esclusi i record presenti nell'oggetto `DynamicFrame` precedente. Opzionale. Il valore di default è zero.
- `totalThreshold` — Il numero massimo di record di errore totali prima di generare un'eccezione, inclusi quelli dei frame precedenti. Opzionale. Il valore di default è zero.
- `enforcedMatches` — Il frame per le corrispondenze applicate. Opzionale. Il valore predefinito è `null`.
- `computeMatchConfidenceScores`: un valore booleano che indica se calcolare un punteggio di confidenza per ciascun gruppo di record corrispondenti. Opzionale. Il valore predefinito è false.

Restituisce un nuovo frame dinamico con un identificatore univoco assegnato a ciascun gruppo di record corrispondenti.

FindIncrementalMatches classe

Pacchetto: com.amazonaws.services.glue.ml

```
object FindIncrementalMatches
```

Applicazione di def

```
apply(existingFrame: DynamicFrame,
            incrementalFrame: DynamicFrame,
            transformId: String,
            transformationContext: String = "",
            callSite: CallSite = CallSite("Not provided", ""),
            stageThreshold: Long = 0,
            totalThreshold: Long = 0,
            enforcedMatches: DynamicFrame = null): DynamicFrame,
   computeMatchConfidenceScores: Boolean
```

Trova le corrispondenze in frame incrementali esistenti e restituisci un nuovo frame con una colonna contenente un ID univoco per gruppo di corrispondenza.

- `existingframe` — Un frame esistente a cui è stato assegnato un ID corrispondente per ogni gruppo. Obbligatorio.
- `incrementalframe` — Un frame incrementale utilizzato per trovare corrispondenze rispetto al frame esistente. Obbligatorio.
- `transformId`— Un ID univoco associato alla FindIncrementalMatches trasformazione da applicare ai frame di input. Obbligatorio.
- `transformationContext` — Identificatore per questo `DynamicFrame`. Il `transformationContext` viene usato come chiave per lo stato dei segnalibro di processo che viene mantenuto tra esecuzioni. Opzionale.
- `callSite` — Usato per fornire informazioni sul contesto per la segnalazione degli errori. Questi valori vengono impostati automaticamente durante la chiamata da Python. Opzionale.
- `stageThreshold` — Il numero massimo di record di errore consentiti nel calcolo di questo `DynamicFrame` prima di generare un'eccezione, esclusi i record presenti nell'oggetto `DynamicFrame` precedente. Opzionale. Il valore di default è zero.
- `totalThreshold` — Il numero massimo di record di errore totali prima di generare un'eccezione, inclusi quelli dei frame precedenti. Opzionale. Il valore di default è zero.

- `enforcedMatches` — Il frame per le corrispondenze applicate. Opzionale. Il valore predefinito è `null`.
- `computeMatchConfidenceScores`: un valore booleano che indica se calcolare un punteggio di confidenza per ciascun gruppo di record corrispondenti. Opzionale. Il valore predefinito è false.

Restituisce un nuovo frame dinamico con un identificatore univoco assegnato a ciascun gruppo di record corrispondenti.

AWS GlueScala IntegerNode APIs

Pacchetto: com.amazonaws.services.glue.types

IntegerNode classe di casi

IntegerNode

```
case class IntegerNode extends ScalarNode(value, TypeCode.INT)  (
           value : Int )
```

IntegerNode campi val

- `ordering`

IntegerNode metodi def

```
def equals( other : Any )
```

AWS GlueScala LongNode APIs

Pacchetto: com.amazonaws.services.glue.types

LongNode classe di casi

LongNode

```
case class LongNode extends ScalarNode(value, TypeCode.LONG)  (
           value : Long )
```

LongNode campi val

- `ordering`

### LongNode metodi def

```
def equals( other : Any )
```

## AWS GlueScala MapLikeNode APIs

Pacchetto: com.amazonaws.services.glue.types

### MapLikeNode classe

### MapLikeNode

```
class MapLikeNode extends DynamicNode  (
          value : mutable.Map[String, DynamicNode] )
```

### MapLikeNode metodi def

```
def clear : Unit
```

```
def get( name : String ) : Option[DynamicNode]
```

```
def getValue
```

```
def has( name : String ) : Boolean
```

```
def isEmpty : Boolean
```

```
def put( name : String,
         node : DynamicNode
       ) : Option[DynamicNode]
```

```
def remove( name : String ) : Option[DynamicNode]
```

```
def toIterator : Iterator[(String, DynamicNode)]
```

```
def toJson : String
```

```
def toJson( useQuotes : Boolean ) : String
```

Esempio: considerato questo JSON:

```
{"foo": "bar"}
```

Se `useQuotes == true`, `toJson` genera `{"foo": "bar"}`. Se `useQuotes == false`, `toJson` genera `{foo: bar}` @return.

AWS GlueScala MapNode APIs

Pacchetto: com.amazonaws.services.glue.types

MapNode classe di casi

MapNode

```
case class MapNode extends MapLikeNode(value)  (
           value : mutable.Map[String, DynamicNode] )
```

MapNode metodi def

```
def clone
```

```
def equals( other : Any )
```

```
def hashCode : Int
```

```
def nodeType
```

```
def this
```

AWS GlueScala NullNode APIs

Argomenti

- NullNode classe
- NullNode oggetto case

Pacchetto: com.amazonaws.services.glue.types

NullNode classe

NullNode

```
class NullNode
```

NullNode oggetto case

NullNode

```
case object NullNode extends NullNode
```

AWS GlueScala ObjectNode APIs

Argomenti

- ObjectNode oggetto
- ObjectNode classe case

Pacchetto: com.amazonaws.services.glue.types

ObjectNode oggetto

ObjectNode

```
object ObjectNode
```

ObjectNode metodi def

```
def apply( frameKeys : Set[String],
           v1 : mutable.Map[String, DynamicNode],
           v2 : mutable.Map[String, DynamicNode],
           resolveWith : String
         ) : ObjectNode
```

ObjectNode classe case

ObjectNode

```
case class ObjectNode extends MapLikeNode(value)  (
          val value : mutable.Map[String, DynamicNode] )
```

### ObjectNode metodi def

```
def clone
```

```
def equals( other : Any )
```

```
def hashCode : Int
```

```
def nodeType
```

```
def this
```

## AWS GlueScala ScalarNode APIs

### Argomenti

- ScalarNode classe
- ScalarNode oggetto

## Pacchetto: com.amazonaws.services.glue.types

### ScalarNode classe

### ScalarNode

```
class ScalarNode extends DynamicNode  (
          value : Any,
          scalarType : TypeCode )
```

### ScalarNode metodi def

```
def compare( other : Any,
             operator : String
          ) : Boolean
```

```
def getValue
```

```
def hashCode : Int
```

```
def nodeType
```

```
def toJson
```

### ScalarNode oggetto

### ScalarNode

```
object ScalarNode
```

### ScalarNode metodi def

```
def apply( v : Any ) : DynamicNode
```

```
def compare( tv : Ordered[T],
             other : T,
             operator : String
           ) : Boolean
```

```
def compareAny( v : Any,
                y : Any,
                o : String )
```

```
def withEscapedSpecialCharacters( jsonToEscape : String ) : String
```

## AWS GlueScala ShortNode APIs

### Pacchetto: com.amazonaws.services.glue.types

### ShortNode classe di casi

### ShortNode

```
case class ShortNode extends ScalarNode(value, TypeCode.SHORT)  (
           value : Short )
```

ShortNode campi val

- `ordering`

ShortNode metodi def

```
def equals( other : Any )
```

AWS GlueScala StringNode APIs

Pacchetto: com.amazonaws.services.glue.types

StringNode classe di casi

StringNode

```
case class StringNode extends ScalarNode(value, TypeCode.STRING)  (
           value : String )
```

StringNode campi val

- `ordering`

StringNode metodi def

```
def equals( other : Any )
```

```
def this( value : UTF8String )
```

AWS GlueScala TimestampNode APIs

Pacchetto: com.amazonaws.services.glue.types

TimestampNode classe di casi

TimestampNode

```
case class TimestampNode extends ScalarNode(value, TypeCode.TIMESTAMP)  (
           value : Timestamp )
```

TimestampNode campi val

- `ordering`

TimestampNode metodi def

```
def equals( other : Any )
```

```
def this( value : Long )
```

AWS GlueScala GlueArgParser APIs

Pacchetto: com.amazonaws.services.glue.util

GlueArgParser oggetto

GlueArgParser

```
object GlueArgParser
```

Questo oggetto è rigorosamente coerente con la versione Python di `utils.getResolvedOptions` nel pacchetto `AWSGlueDataplanePython`.

GlueArgParser metodi def

```
def getResolvedOptions( args : Array[String],
                        options : Array[String]
                      ) : Map[String, String]
```

```
def initParser( userOptionsSet : mutable.Set[String] ) : ArgumentParser
```

Example Recupero degli argomenti trasmessi a un processo

Per recuperare gli argomenti del processo, è possibile utilizzare il metodo `getResolvedOptions`. Esamina l'esempio seguente, che recupera un argomento del processo denominato `aws_region`.

```
val args = GlueArgParser.getResolvedOptions(sysArgs,
 Seq("JOB_NAME","aws_region").toArray)
Job.init(args("JOB_NAME"), glueContext, args.asJava)
val region = args("aws_region")
println(region)
```

### AWS Gluelavoro Scala APIs

Pacchetto: com.amazonaws.services.glue.util

### Oggetto del processo

### Processo

```
object Job
```

### Metodi def del processo

```
def commit
```

```
def init( jobName : String,
          glueContext : GlueContext,
          args : java.util.Map[String, String] = Map[String, String]().asJava
        ) : this.type
```

```
def init( jobName : String,
          glueContext : GlueContext,
          endpoint : String,
          args : java.util.Map[String, String]
        ) : this.type
```

```
def isInitialized
```

```
def reset
```

```
def runId
```

# Funzionalità e ottimizzazioni per la programmazione degli script ETL di AWS Glue per Spark

Le sezioni seguenti descrivono le tecniche e i valori che si applicano generalmente alla programmazione ETL (estrazione, trasformazione e caricamento) di AWS Glue per Spark in qualsiasi linguaggio.

Argomenti


- Tipi e opzioni di connessione per ETL in AWS Glue per Spark
- Opzioni del formato dati per input e output in AWS Glue per Spark
- AWS Supporto di Glue Data Catalog per i job SQL di Spark
- Utilizzo di segnalibri di processo
- Utilizzo del rilevamento di dati sensibili all'esterno di AWS Glue Studio
- API Visual Job di AWS Glue


## Tipi e opzioni di connessione per ETL in AWS Glue per Spark

In AWS Glue Spark, vari metodi e trasformazioni PySpark e Scala specificano il tipo di connessione utilizzando un `connectionType` parametro. Specificano le opzioni di connessione utilizzando un parametro `connectionOptions` o `options`.

Il parametro `connectionType` può assumere i valori indicati nella tabella seguente. I valori dei parametri associati `connectionOptions` (o `options`) per ciascun tipo sono documentati nelle sezioni seguenti. Salvo indicazione contraria, i parametri si applicano quando la connessione viene utilizzata come sorgente o sink.

Per il codice di esempio che illustra l'impostazione e l'utilizzo delle opzioni di connessione, consulta la home page per ogni tipo di connessione.

| **`connectionType`** | Si connette a |
| --- | --- |
| dynamodb | Amazon DynamoDB database |
| kinesis | Flusso di dati Amazon Kinesis |
| s3 | Amazon S3 |

| `connectionType` | Si connette a |
| --- | --- |
| documentdb | Amazon DocumentDB (compatibile con MongoDB) database |
| opensearch | OpenSearch Servizio Amazon. |
| redshift | Database Amazon Redshift |
| kafka | Kafka o Amazon Managed Streaming for Apache Kafka |
| azurecosmos | Azure Cosmos per NoSQL. |
| azuresql | Azure SQL. |
| bigquery | Google BigQuery. |
| mongodb | Database MongoDB, incluso MongoDB Atlas. |
| sqlserver | Microsoft SQL Server database (vedere Connessioni JDBC) |
| mysql | MySQL database (vedere Connessioni JDBC) |
| oracle | Oracle database (vedere Connessioni JDBC) |
| postgresql | PostgreSQL database (vedere Connessioni JDBC) |
| saphana | SAP HANA. |
| snowflake | Data lake Snowflake |
| teradata | Teradata Vantage. |
| vertica | Vertica. |
| personalizzato.* | Archivi dati Spark, Athena o JDBC (consulta Valori Custom e Marketplace AWS ConnectionType |
| marketplace.* | Archivi dati Spark, Athena o JDBC (consulta Valori Custom e Marketplace AWS ConnectionType) |

Connessioni a DynamoDB

Puoi usare AWS Glue for Spark per leggere e scrivere su tabelle in DynamoDB in Glue. AWS Ti connetti a DynamoDB utilizzando le autorizzazioni IAM allegate al tuo job Glue. AWS AWS Gluesupporta la scrittura di dati nella tabella DynamoDB di un altro AWS account. Per ulteriori informazioni, consulta the section called "Accesso multi-account in più regioni alle tabelle DynamoDB".

Il connettore DynamoDB originale utilizza oggetti DynamicFrame Glue per lavorare con i dati estratti da DynamoDB. AWS Glue 5.0+ introduce una novità Connettore DynamoDB con supporto Spark DataFrame che fornisce il supporto nativo per Spark. DataFrame

Oltre al connettore AWS Glue DynamoDB ETL, puoi leggere da DynamoDB utilizzando il connettore di esportazione DynamoDB, che richiama una richiesta DynamoDB e la archivia in una posizione Amazon S3 da te `ExportTableToPointInTime` fornita, nel formato DynamoDB JSON. AWS Gluequindi crea un DynamicFrame oggetto leggendo i dati dalla posizione di esportazione di Amazon S3.

Il writer DynamoDB è disponibile nella versione 1.0 di AWS Glue Glue o nelle versioni successive. Il connettore di esportazione DynamoDB per AWS Glue è disponibile nella versione 2.0 di AWS Glue o nelle versioni successive. Il nuovo connettore DynamoDB DataFrame basato su DynamoDB è disponibile nella versione AWS Glue 5.0 o successive.

Per ulteriori informazioni su DynamoDB, consulta la documentazione di Amazon DynamoDB.

> **Note**
>
> Il lettore DynamoDB ETL non supporta filtri o predicati pushdown.

Configurazione delle connessioni a MongoDB

Per connetterti a DynamoDB AWS da Glue, concedi al ruolo IAM associato al AWS tuo job Glue l'autorizzazione a interagire con DynamoDB. Per ulteriori informazioni sulle autorizzazioni necessarie per leggere o scrivere da DynamoDB, consulta Operazioni, risorse e chiavi di condizione per Amazon DynamoDB nella documentazione di IAM.

Nelle seguenti situazioni, potresti aver bisogno di una configurazione aggiuntiva:

- Quando utilizzi il connettore di esportazione DynamoDB, devi configurare IAM in modo che il processo possa richiedere l'esportazione di tabelle DynamoDB. Inoltre, dovrai identificare un

bucket Amazon S3 per l'esportazione e fornire le autorizzazioni appropriate in IAM affinché DynamoDB possa scrivervi e che il tuo job Glue possa leggerlo. AWS Per ulteriori informazioni, consulta Richiesta di esportazione di una tabella in DynamoDB.

- Se il tuo job AWS Glue ha requisiti di connettività Amazon VPC specifici, usa il tipo di connessione `NETWORK` AWS Glue per fornire opzioni di rete. Poiché l'accesso a DynamoDB è autorizzato da IAM, non è necessario un tipo di connessione AWS Glue DynamoDB.

### Lettura e scrittura su DynamoDB

Gli esempi di codice seguenti mostrano come leggere (tramite il connettore ETL) e scrivere tabelle DynamoDB. Mostrano la lettura da una tabella e la scrittura su un'altra tabella.

Python

```
import sys
from pyspark.context import SparkContext
from awsglue.context import GlueContext
from awsglue.job import Job
from awsglue.utils import getResolvedOptions

args = getResolvedOptions(sys.argv, ["JOB_NAME"])
glue_context= GlueContext(SparkContext.getOrCreate())
job = Job(glue_context)
job.init(args["JOB_NAME"], args)

dyf = glue_context.create_dynamic_frame.from_options(
    connection_type="dynamodb",
    connection_options={"dynamodb.input.tableName": test_source,
        "dynamodb.throughput.read.percent": "1.0",
        "dynamodb.splits": "100"
    }
)
print(dyf.getNumPartitions())

glue_context.write_dynamic_frame_from_options(
    frame=dyf,
    connection_type="dynamodb",
    connection_options={"dynamodb.output.tableName": test_sink,
        "dynamodb.throughput.write.percent": "1.0"
    }
)
```

```
job.commit()
```

## Scala

```
import com.amazonaws.services.glue.GlueContext
import com.amazonaws.services.glue.util.GlueArgParser
import com.amazonaws.services.glue.util.Job
import com.amazonaws.services.glue.util.JsonOptions
import com.amazonaws.services.glue.DynamoDbDataSink
import org.apache.spark.SparkContext
import scala.collection.JavaConverters._


object GlueApp {

  def main(sysArgs: Array[String]): Unit = {
    val glueContext = new GlueContext(SparkContext.getOrCreate())
    val args = GlueArgParser.getResolvedOptions(sysArgs, Seq("JOB_NAME").toArray)
    Job.init(args("JOB_NAME"), glueContext, args.asJava)

    val dynamicFrame = glueContext.getSourceWithFormat(
      connectionType = "dynamodb",
      options = JsonOptions(Map(
        "dynamodb.input.tableName" -> test_source,
        "dynamodb.throughput.read.percent" -> "1.0",
        "dynamodb.splits" -> "100"
      ))
    ).getDynamicFrame()

    print(dynamicFrame.getNumPartitions())

    val dynamoDbSink: DynamoDbDataSink =  glueContext.getSinkWithFormat(
      connectionType = "dynamodb",
      options = JsonOptions(Map(
        "dynamodb.output.tableName" -> test_sink,
        "dynamodb.throughput.write.percent" -> "1.0"
      ))
    ).asInstanceOf[DynamoDbDataSink]

    dynamoDbSink.writeDynamicFrame(dynamicFrame)

    Job.commit()
```

```
  }

}
```

Utilizzo del connettore di esportazione DynamoDB

Il connettore di esportazione ha prestazioni migliori rispetto al connettore ETL quando le dimensioni della tabella DynamoDB sono superiori a 80 GB. Inoltre, dato che la richiesta di esportazione è condotta al di fuori dei processi Spark in un processo AWS Glue, puoi abilitare Scalabilità automatica di processi Glue AWS per salvare l'utilizzo della DPU durante la richiesta di esportazione. Con il connettore di esportazione, non è inoltre necessario configurare il numero di divisioni per il parallelismo dell'esecutore Spark o la percentuale di lettura del throughput DynamoDB.

> **Note**
>
> DynamoDB ha requisiti specifici per richiamare le richieste `ExportTableToPointInTime`. Per ulteriori informazioni, consulta Richiesta di esportazione di una tabella in DynamoDB.
> Ad esempio, è necessario abilitare Point-in-Time-Restore (PITR) sulla tabella per utilizzare questo connettore. Il connettore DynamoDB supporta anche la AWS KMS crittografia per le esportazioni DynamoDB verso Amazon S3. L'indicazione della configurazione di sicurezza nella configurazione del job AWS Glue abilita la AWS KMS crittografia per un'esportazione DynamoDB. La chiave KMS deve essere nella stessa regione del bucket Amazon S3.
> Tieni presente che si applicano costi aggiuntivi per l'esportazione DynamoDB e i costi di storage Amazon S3. I dati esportati in Amazon S3 persistono al termine dell'esecuzione di un processo in modo da poterli riutilizzare senza ulteriori esportazioni DynamoDB. Un requisito per l'utilizzo di questo connettore è che il point-in-time ripristino (PITR) sia abilitato per la tabella.
> Il connettore ETL DynamoDB o il connettore di esportazione non supportano filtri o predicati pushdown da applicare all'origine DynamoDB.

Gli esempi di codice seguenti mostrano come leggere (tramite il connettore di esportazione) e stampare il numero di partizioni.

Python

```
import sys
from pyspark.context import SparkContext
```

```
from awsglue.context import GlueContext
from awsglue.job import Job
from awsglue.utils import getResolvedOptions

args = getResolvedOptions(sys.argv, ["JOB_NAME"])
glue_context= GlueContext(SparkContext.getOrCreate())
job = Job(glue_context)
job.init(args["JOB_NAME"], args)

dyf = glue_context.create_dynamic_frame.from_options(
    connection_type="dynamodb",
    connection_options={
        "dynamodb.export": "ddb",
        "dynamodb.tableArn": test_source,
        "dynamodb.s3.bucket": bucket_name,
        "dynamodb.s3.prefix": bucket_prefix,
        "dynamodb.s3.bucketOwner": account_id_of_bucket,
    }
)
print(dyf.getNumPartitions())

job.commit()
```

## Scala

```
import com.amazonaws.services.glue.GlueContext
import com.amazonaws.services.glue.util.GlueArgParser
import com.amazonaws.services.glue.util.Job
import com.amazonaws.services.glue.util.JsonOptions
import com.amazonaws.services.glue.DynamoDbDataSink
import org.apache.spark.SparkContext
import scala.collection.JavaConverters._


object GlueApp {

  def main(sysArgs: Array[String]): Unit = {
    val glueContext = new GlueContext(SparkContext.getOrCreate())
    val args = GlueArgParser.getResolvedOptions(sysArgs, Seq("JOB_NAME").toArray)
    Job.init(args("JOB_NAME"), glueContext, args.asJava)

    val dynamicFrame = glueContext.getSourceWithFormat(
      connectionType = "dynamodb",
```

```
      options = JsonOptions(Map(
        "dynamodb.export" -> "ddb",
        "dynamodb.tableArn" -> test_source,
        "dynamodb.s3.bucket" -> bucket_name,
        "dynamodb.s3.prefix" -> bucket_prefix,
        "dynamodb.s3.bucketOwner" -> account_id_of_bucket,
      ))
    ).getDynamicFrame()

    print(dynamicFrame.getNumPartitions())

    Job.commit()
  }

}
```

Questi esempi mostrano come eseguire la lettura da (tramite il connettore di esportazione) e stampare il numero di partizioni da una tabella AWS Glue Data Catalog con una dynamodb classificazione:

Python

```
import sys
from pyspark.context import SparkContext
from awsglue.context import GlueContext
from awsglue.job import Job
from awsglue.utils import getResolvedOptions

args = getResolvedOptions(sys.argv, ["JOB_NAME"])
glue_context= GlueContext(SparkContext.getOrCreate())
job = Job(glue_context)
job.init(args["JOB_NAME"], args)

dynamicFrame = glue_context.create_dynamic_frame.from_catalog(
        database=catalog_database,
        table_name=catalog_table_name,
        additional_options={
            "dynamodb.export": "ddb",
            "dynamodb.s3.bucket": s3_bucket,
            "dynamodb.s3.prefix": s3_bucket_prefix
        }
    )
```

```
print(dynamicFrame.getNumPartitions())

job.commit()
```

Scala

```
import com.amazonaws.services.glue.GlueContext
import com.amazonaws.services.glue.util.GlueArgParser
import com.amazonaws.services.glue.util.Job
import com.amazonaws.services.glue.util.JsonOptions
import com.amazonaws.services.glue.DynamoDbDataSink
import org.apache.spark.SparkContext
import scala.collection.JavaConverters._


object GlueApp {

  def main(sysArgs: Array[String]): Unit = {
    val glueContext = new GlueContext(SparkContext.getOrCreate())
    val args = GlueArgParser.getResolvedOptions(sysArgs, Seq("JOB_NAME").toArray)
    Job.init(args("JOB_NAME"), glueContext, args.asJava)

    val dynamicFrame = glueContext.getCatalogSource(
        database = catalog_database,
        tableName = catalog_table_name,
        additionalOptions = JsonOptions(Map(
            "dynamodb.export" -> "ddb",
            "dynamodb.s3.bucket" -> s3_bucket,
            "dynamodb.s3.prefix" -> s3_bucket_prefix
        ))
    ).getDynamicFrame()
    print(dynamicFrame.getNumPartitions())
)
```

### Semplificazione dell'utilizzo del JSON di esportazione DynamoDB

Le esportazioni di DynamoDB con il connettore di esportazione DynamoDB AWS Glue possono risultare in file JSON con specifiche strutture annidate. Per ulteriori informazioni, consulta Oggetti dati. AWS Gluefornisce una DynamicFrame trasformazione che può trasformare tali strutture in una easier-to-use forma per le applicazioni a valle.

La trasformazione può essere invocata in uno dei due modi possibili. Puoi impostare l'opzione di connessione `"dynamodb.simplifyDDBJson"` sul valore `"true"` quando effettui una chiamata a un metodo per leggere da DynamoDB. Puoi anche chiamare la trasformazione come metodo disponibile indipendentemente nella libreria AWS Glue.

Prendi in considerazione lo schema seguente generato da un'esportazione DynamoDB:

```
root
|-- Item: struct
|    |-- parentMap: struct
|    |    |-- M: struct
|    |    |    |-- childMap: struct
|    |    |    |    |-- M: struct
|    |    |    |    |    |-- appName: struct
|    |    |    |    |    |    |-- S: string
|    |    |    |    |    |-- packageName: struct
|    |    |    |    |    |    |-- S: string
|    |    |    |    |    |-- updatedAt: struct
|    |    |    |    |    |    |-- N: string
|    |-- strings: struct
|    |    |-- SS: array
|    |    |    |-- element: string
|    |-- numbers: struct
|    |    |-- NS: array
|    |    |    |-- element: string
|    |-- binaries: struct
|    |    |-- BS: array
|    |    |    |-- element: string
|    |-- isDDBJson: struct
|    |    |-- BOOL: boolean
|    |-- nullValue: struct
|    |    |-- NULL: boolean
```

La trasformazione `simplifyDDBJson` semplificherà questo processo in:

```
root
|-- parentMap: struct
|    |-- childMap: struct
|    |    |-- appName: string
|    |    |-- packageName: string
|    |    |-- updatedAt: string
|-- strings: array
```

```
|    |-- element: string
|-- numbers: array
|    |-- element: string
|-- binaries: array
|    |-- element: string
|-- isDDBJson: boolean
|-- nullValue: null
```

> **Note**
>
> `simplifyDDBJson`è disponibile in AWS Glue 3.0 e versioni successive. Per semplificare il JSON di esportazione DynamoDB è disponibile anche la trasformazione `unnestDDBJson`. Incoraggiamo gli utenti a passare da `unnestDDBJson` a `simplifyDDBJson`.

Configurazione del parallelismo nelle operazioni DynamoDB

Per migliorare le prestazioni, è possibile regolare alcuni parametri disponibili per il connettore DynamoDB. Il vostro obiettivo quando regolate i parametri di paralleismo è massimizzare l'uso dei Glue worker forniti. AWS Quindi, se hai bisogno di maggiori prestazioni, ti consigliamo di ampliare il tuo lavoro aumentando il numero di. DPUs

Puoi modificare il parallelismo in un'operazione di lettura di DynamoDB con il parametro `dynamodb.splits` quando utilizzi il connettore ETL. La lettura con il connettore di esportazione non richiede la configurazione del numero di divisioni per il parallelismo dell'esecutore Spark. Puoi modificare il parallelismo in un'operazione di scrittura DynamoDB con `dynamodb.output.numParallelTasks`.

Lettura con il connettore ETL per DynamoDB

Ti consigliamo di calcolare `dynamodb.splits` in base al numero massimo di worker impostato nella configurazione del lavoro e al seguente calcolo `numSlots`. In caso di dimensionamento automatico, il numero effettivo di worker disponibili potrebbe variare al di sotto di tale limite. Per ulteriori informazioni sull'impostazione del numero massimo di worker, consulta Numero di worker (`NumberOfWorkers`) in the section called “Configurazione delle proprietà del processo Spark”.

- `numExecutors = NumberOfWorkers - 1`

  Per il contesto, un esecutore è riservato per il driver Spark; altri esecutori sono utilizzati per elaborare i dati.

- `numSlotsPerExecutor =`

  AWS Glue 3.0 and later versions

  - 4: se `WorkerType` è `G.1X`.
  - 8: se `WorkerType` è `G.2X`.
  - 16: se `WorkerType` è `G.4X`.
  - 32: se `WorkerType` è `G.8X`.

  AWS Glue 2.0 and legacy versions

  - 8: se `WorkerType` è `G.1X`.
  - 16: se `WorkerType` è `G.2X`.
- `numSlots = numSlotsPerExecutor * numExecutors`

Ti consigliamo di impostare `dynamodb.splits` sul numero di slot disponibili, `numSlots`.

Scrittura su DynamoDB

Il parametro `dynamodb.output.numParallelTasks` viene utilizzato per determinare il valore WCU per ogni attività Spark, utilizzando il seguente calcolo:

```
permittedWcuPerTask = ( TableWCU * dynamodb.throughput.write.percent ) / 
dynamodb.output.numParallelTasks
```

Il writer DynamoDB funzionerà al meglio se la configurazione rappresenta accuratamente il numero di attività Spark che scrivono su DynamoDB. In alcuni casi, potrebbe essere necessario sovrascrivere il calcolo predefinito per migliorare le prestazioni di scrittura. Se questo parametro non viene specificato, il valore WCU consentito per i processi Spark verrà calcolato automaticamente mediante la seguente formula:

- 
  - `numPartitions = dynamicframe.getNumPartitions()`
  - `numSlots` (come definito in precedenza in questa sezione)
  - `numParallelTasks = min(numPartitions, numSlots)`
- Esempio 1: DPU=10, =Standard. WorkerType L'input DynamicFrame ha 100 partizioni RDD.
  - `numPartitions = 100`
  - `numExecutors = (10 - 1) * 2 - 1 = 17`
  - `numSlots = 4 * 17 = 68`
  - `numParallelTasks = min(100, 68) = 68`

- Esempio 2: DPU=10, =Standard. WorkerType L'ingresso DynamicFrame ha 20 partizioni RDD.
  - `numPartitions = 20`
  - `numExecutors = (10 - 1) * 2 - 1 = 17`
  - `numSlots = 4 * 17 = 68`
  - `numParallelTasks = min(20, 68) = 20`

> **Note**
>
> I lavori sulle versioni precedenti di AWS Glue e quelli che utilizzano Standard worker richiedono metodi diversi per calcolare il numero di slot. Se hai bisogno di ottimizzare le prestazioni di questi lavori, ti consigliamo di passare alle versioni supportate di AWS Glue.

Indicazioni di riferimento per le opzioni di connessione a DynamoDB

Indica una connessione ad Amazon DynamoDB.

Le opzioni di connessione differiscono per una connessione sorgente e una connessione sink.

"connectionType": "dynamodb" con il connettore ETL come origine

Usa le seguenti opzioni di connessione `"connectionType": "dynamodb"` come sorgente, quando usi il connettore AWS Glue DynamoDB ETL:

- `"dynamodb.input.tableName"`: (Obbligatorio) la tabella DynamoDB da cui leggere.
- `"dynamodb.throughput.read.percent"`: (Facoltativo) percentuale di unità di capacità di lettura (RCU) da usare. Il valore predefinito è "0,5". I valori accettabili vanno da "0,1" a "1,5", inclusi.
  - `0.5` rappresenta la velocità di lettura di default, il che significa che AWS Glue cercherà di consumare metà della capacità di lettura della tabella. Se si aumenta il valore sopra `0.5`, AWS Glue aumenterà il tasso di richiesta; diminuendo il valore al di sotto di `0.5`, riduce la frequenza di richiesta di lettura. La velocità di lettura effettiva varia in base a fattori come la presenza di una distribuzione uniforme delle chiavi nella tabella DynamoDB.
  - Quando la tabella DynamoDB è in modalità on demand, AWS Glue gestisce la capacità di lettura della tabella a 40000. Per esportare una tabella di grandi dimensioni, si consiglia di passare alla modalità su richiesta della tabella DynamoDB.

- `"dynamodb.splits"`: (Facoltativo) Definisce il numero di partizioni applicate a questa tabella DynamoDB durante la lettura. Il valore predefinito è "1". I valori accettabili vanno da "1" a "1,000,000", inclusi.

  1 indica che non c'è parallelismo. Si consiglia vivamente di specificare un valore maggiore per migliorare le prestazioni utilizzando la formula riportata di seguito. Per ulteriori informazioni sull'impostazione corretta di un valore, consulta the section called “Parallelismo di DynamoDB”.
- `"dynamodb.sts.roleArn"`: (Facoltativo) il ruolo IAM ARN da assumere per l'accesso multi-account. Questo parametro è disponibile in AWS Glue 1.0 o versione successiva.
- `"dynamodb.sts.roleSessionName"`: (Facoltativo) nome della sessione STS. L'impostazione predefinita è "glue-dynamodb-read-sts-session». Questo parametro è disponibile in AWS Glue 1.0 o versione successiva.

"connectionType": "dynamodb" con il connettore di esportazione DynamoDB AWS Glue come sorgente

Utilizzare le seguenti opzioni di connessione con «connectionType»: «dynamodb» come origine, quando utilizzi il Connettore di esportazione DynamoDB AWS Glue, disponibile solo per la versione AWS Glue 2.0 in poi:

- `"dynamodb.export"`: (Richiesto) Un valore stringa:
  - Se impostato su ddb abilita il Connettore di esportazione DynamoDB AWS Glue dove un nuovo `ExportTableToPointInTimeRequest` verrà invocato durante il processo AWS Glue. Verrà generata una nuova esportazione con la posizione passata da `dynamodb.s3.bucket` e `dynamodb.s3.prefix`.
  - Se impostato su `s3` abilita il Connettore di esportazione DynamoDB AWS Glue ma salta la creazione di una nuova esportazione DynamoDB e utilizza invece il `dynamodb.s3.bucket` e il `dynamodb.s3.prefix` come posizione di Amazon S3 di un'esportazione precedente di quella tabella.
- `"dynamodb.tableArn"`: (Obbligatorio) la tabella DynamoDB da cui leggere.
- `"dynamodb.unnestDDBJson"`: false per impostazione predefinita (facoltativo). Valori validi: booleani. Se impostato su true, esegue una trasformazione non nidificata della struttura JSON DynamoDB presente nelle esportazioni. È un errore impostare contemporaneamente `"dynamodb.unnestDDBJson"` e `"dynamodb.simplifyDDBJson"` su true. In AWS Glue 3.0 e versioni successive, si consiglia di utilizzare `"dynamodb.simplifyDDBJson"` per un comportamento migliore durante la semplificazione dei tipi di mappe DynamoDB. Per ulteriori

informazioni, consulta the section called “Semplificazione dell'utilizzo del JSON di esportazione DynamoDB”.

- `"dynamodb.simplifyDDBJson"`: false per impostazione predefinita (facoltativo). Valori validi: booleani. Se impostato su true, esegue una trasformazione per semplificare la struttura JSON DynamoDB presente nelle esportazioni. Questa opzione ha lo stesso scopo di `"dynamodb.unnestDDBJson"`, ma fornisce un supporto migliore per i tipi di mappe DynamoDB o per i tipi di mappe annidate nella tabella DynamoDB. Questa opzione è disponibile in AWS Glue 3.0 e versioni successive. È un errore impostare contemporaneamente `"dynamodb.unnestDDBJson"` e `"dynamodb.simplifyDDBJson"` su true. Per ulteriori informazioni, consulta the section called “Semplificazione dell'utilizzo del JSON di esportazione DynamoDB”.
- `"dynamodb.s3.bucket"`: (facoltativo) indica la posizione del bucket Amazon S3 in cui deve essere condotto il processo DynamoDB `ExportTableToPointInTime`. Il formato del file per l'esportazione è DynamoDB JSON.
  - `"dynamodb.s3.prefix"`: (facoltativo) indica la posizione del prefisso di Amazon S3 all'interno del bucket Amazon S3 in cui devono essere archiviati i carichi `ExportTableToPointInTime` di DynamoDB. Se non si specificano i valori `dynamodb.s3.prefix` e `dynamodb.s3.bucket`, per impostazione predefinita sarà utilizzata la posizione della directory temporanea specificata nella configurazione del processo AWS Glue. Per ulteriori informazioni, vedere Parametri speciali utilizzati da AWS Glue.
  - `"dynamodb.s3.bucketOwner"`: indica il proprietario del bucket necessario per l'accesso di Amazon S3 tra account.
- `"dynamodb.sts.roleArn"`: (Facoltativo) L'ARN del ruolo IAM da assumere per l'accesso tra account e più and/or regioni per la tabella DynamoDB. Nota: lo stesso ARN del ruolo IAM verrà utilizzato per accedere alla posizione Amazon S3 specificata per la richiesta `ExportTableToPointInTime`.
- `"dynamodb.sts.roleSessionName"`: (Facoltativo) nome della sessione STS. L'impostazione predefinita è "-session». glue-dynamodb-read-sts
- `"dynamodb.exportTime"` (facoltativo). Valori validi: stringhe che rappresentano istanti ISO-8601. point-in-timeA in cui deve essere effettuata l'esportazione.
- `"dynamodb.sts.region"`: la regione che ospita la tabella DynamoDB da leggere (obbligatorio se si effettua una chiamata tra regioni utilizzando un endpoint regionale).

"connectionType": "dynamodb" con il connettore ETL come sink

Utilizzare le seguenti opzioni di connessione con `"connectionType": "dynamodb"` come sink:

- `"dynamodb.output.tableName"`: (Obbligatorio) la tabella DynamoDB su cui scrivere.
- `"dynamodb.throughput.write.percent"`: (Facoltativo) percentuale di unità di capacità di scrittura (WCU) da usare. Il valore predefinito è "0,5". I valori accettabili vanno da "0,1" a "1,5", inclusi.
  - `0.5` rappresenta la velocità di scrittura di default, il che significa che AWS Glue cercherà di consumare metà della capacità di scrittura della tabella. Se si aumenta il valore sopra 0,5, AWS Glue aumenterà il tasso di richiesta; diminuendo il valore al di sotto di 0,5, riduce la frequenza di richiesta di scrittura. (La velocità di scrittura effettiva varia in base a fattori come la presenza di una distribuzione uniforme delle chiavi nella tabella DynamoDB).
  - Quando la tabella DynamoDB è in modalità on demand, AWS Glue gestisce la capacità di lettura della tabella come `40000`. Per importare una tabella di grandi dimensioni, si consiglia di passare alla modalità su richiesta della tabella DynamoDB.
- `"dynamodb.output.numParallelTasks"`: (Facoltativo) definisce il numero di attività parallele scritte contemporaneamente in DynamoDB. Utilizzato per calcolare WCU permissivo per i processi Spark. Nella maggior parte dei casi, AWS Glue calcolerà un valore predefinito ragionevole per questo valore. Per ulteriori informazioni, consulta the section called “Parallelismo di DynamoDB”.
- `"dynamodb.output.retry"`: (Facoltativo) definisce il numero di tentativi eseguiti quando esiste una `ProvisionedThroughputExceededException` da DynamoDB. Il valore predefinito è "10".
- `"dynamodb.sts.roleArn"`: (facoltativo) il ruolo IAM ARN da assumere per l'accesso multi-account.
- `"dynamodb.sts.roleSessionName"`: (Facoltativo) nome della sessione STS. L'impostazione predefinita è "glue-dynamodb-write-sts-session».

Accesso multi-account in più regioni alle tabelle DynamoDB

I processi ETL AWS Glue supportano sia l'accesso multi-account che in più regioni alle tabelle DynamoDB. I processi ETL AWS Glue I job ETL supportano sia la lettura di dati dalla tabella DynamoDB di un altro AWS account, sia la scrittura di dati nella tabella DynamoDB di un altro AWS account. AWS Gluesupporta anche la lettura da una tabella DynamoDB in un'altra regione e la scrittura in una tabella DynamoDB in un'altra regione. Questa sezione fornisce istruzioni su come configurare l'accesso e fornisce uno script di esempio.

Le procedure descritte in questa sezione fanno riferimento a un tutorial IAM per la creazione di un ruolo IAM e la concessione dell'accesso al ruolo. Il tutorial discute anche l'assunzione di un ruolo, ma qui si utilizzerà invece uno script di processo per assumere il ruolo in AWS Glue. Questo tutorial contiene anche informazioni sulle pratiche generali multi-account. Per ulteriori informazioni, consulta Tutorial: Delegare l'accesso tra AWS account utilizzando i ruoli IAM nella Guida per l'utente IAM.

Creare un ruolo

Segui il passaggio 1 del tutorial per creare un ruolo IAM nell'account A. Quando definisci le autorizzazioni del ruolo, puoi scegliere di allegare policy esistenti come `AmazonDynamoDBReadOnlyAccess` o di consentire il ruolo `AmazonDynamoDBFullAccess` a read/write DynamoDB. L'esempio seguente mostra la creazione di un ruolo denominato `DynamoDBCrossAccessRole`, con la policy di autorizzazione `AmazonDynamoDBFullAccess`.

Concedi autorizzazione per l'accesso al ruolo

Segui la fase 2 del tutorial nella Guida per l'utente di IAM per consentire all'account B di passare al ruolo appena creato. L'esempio seguente crea una nuova policy con l'istruzione riportata di seguito:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "sts:AssumeRole",
    "Resource": "arn:aws:iam::111122223333:role/DynamoDBCrossAccessRole"
  }
}
```

Quindi, puoi allegare questa policy a quella group/role/user che desideri utilizzare per accedere a DynamoDB.

Assumi il ruolo nello script di processo AWS Glue

Ora è possibile accedere all'account B e creare un processo AWS Glue. Per creare un processo, fai riferimento alle istruzioni in Configurazione delle proprietà dei job per i job Spark in AWS Glue.

Nello script del processo è necessario utilizzare il parametro `dynamodb.sts.roleArn` per assumere il ruolo `DynamoDBCrossAccessRole`. Supponendo che questo ruolo consenta di ottenere

le credenziali temporanee, che devono essere utilizzate per accedere a DynamoDB nell'account B, esamina questi script di esempio.

Per una lettura multi-account tra regioni (connettore ETL):

```
import sys
from pyspark.context import SparkContext
from awsglue.context import GlueContext
from awsglue.job import Job
from awsglue.utils import getResolvedOptions

args = getResolvedOptions(sys.argv, ["JOB_NAME"])
glue_context= GlueContext(SparkContext.getOrCreate())
job = Job(glue_context)
job.init(args["JOB_NAME"], args)

dyf = glue_context.create_dynamic_frame_from_options(
    connection_type="dynamodb",
    connection_options={
    "dynamodb.region": "us-east-1",
    "dynamodb.input.tableName": "test_source",
    "dynamodb.sts.roleArn": "<DynamoDBCrossAccessRole's ARN>"
    }
)
dyf.show()
job.commit()
```

Per una lettura multi-account tra regioni (connettore ELT):

```
import sys
from pyspark.context import SparkContext
from awsglue.context import GlueContext
from awsglue.job import Job
from awsglue.utils import getResolvedOptions

args = getResolvedOptions(sys.argv, ["JOB_NAME"])
glue_context= GlueContext(SparkContext.getOrCreate())
job = Job(glue_context)
job.init(args["JOB_NAME"], args)

dyf = glue_context.create_dynamic_frame_from_options(
    connection_type="dynamodb",
    connection_options={
```

```
        "dynamodb.export": "ddb",
        "dynamodb.tableArn": "<test_source ARN>",
        "dynamodb.sts.roleArn": "<DynamoDBCrossAccessRole's ARN>"
    }
)
dyf.show()
job.commit()
```

Per una lettura e scrittura multi-account tra regioni:

```
import sys
from pyspark.context import SparkContext
from awsglue.context import GlueContext
from awsglue.job import Job
from awsglue.utils import getResolvedOptions

args = getResolvedOptions(sys.argv, ["JOB_NAME"])
glue_context= GlueContext(SparkContext.getOrCreate())
job = Job(glue_context)
job.init(args["JOB_NAME"], args)

dyf = glue_context.create_dynamic_frame_from_options(
    connection_type="dynamodb",
    connection_options={
        "dynamodb.region": "us-east-1",
        "dynamodb.input.tableName": "test_source"
    }
)
dyf.show()

glue_context.write_dynamic_frame_from_options(
    frame=dyf,
    connection_type="dynamodb",
    connection_options={
        "dynamodb.region": "us-west-2",
        "dynamodb.output.tableName": "test_sink",
        "dynamodb.sts.roleArn": "<DynamoDBCrossAccessRole's ARN>"
    }
)

job.commit()
```

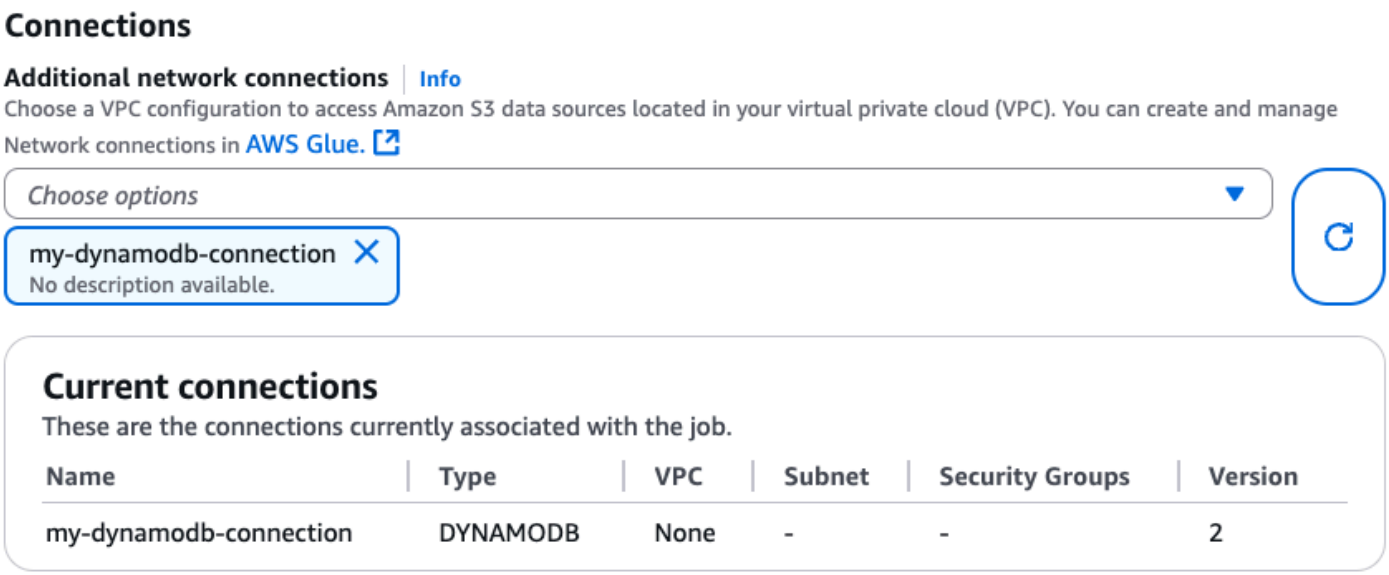
Connettore DynamoDB con supporto Spark DataFrame

Il connettore DynamoDB con supporto DataFrame Spark consente di leggere e scrivere su tabelle in DynamoDB utilizzando le API Spark. DataFrame I passaggi di configurazione del connettore sono gli stessi del connettore e sono disponibili qui. DynamicFrame-based

Per caricare la libreria dei DataFrame-based connettori, assicurati di collegare una connessione DynamoDB al job Glue.

> **Note**
>
> L'interfaccia utente della console Glue attualmente non supporta la creazione di una connessione DynamoDB. Puoi usare Glue CLI (CreateConnection) per creare una connessione DynamoDB:
>
> ```
> aws glue create-connection \
>     --connection-input '{ \
>         "Name": "my-dynamodb-connection", \
>         "ConnectionType": "DYNAMODB", \
>         "ConnectionProperties": {}, \
>         "ValidateCredentials": false, \
>         "ValidateForComputeEnvironments": ["SPARK"] \
>     }'
> ```

Dopo aver creato la connessione DynamoDB, puoi collegarla al tuo lavoro Glue tramite CLI CreateJob(UpdateJob,) o direttamente nella pagina «Dettagli del lavoro»:




Dopo esserti assicurato che una connessione con DYNAMODB Type sia collegata al tuo lavoro Glue, puoi utilizzare le seguenti operazioni di lettura, scrittura ed esportazione dal connettore. DataFrame-based

Lettura e scrittura su DynamoDB con il connettore DataFrame-based

I seguenti esempi di codice mostrano come leggere e scrivere su tabelle DynamoDB tramite il connettore. DataFrame-based Mostrano la lettura da una tabella e la scrittura su un'altra tabella.

Python

```
import sys
from pyspark.context import SparkContext
from awsglue.context import GlueContext
from awsglue.job import Job
from awsglue.utils import getResolvedOptions

args = getResolvedOptions(sys.argv, ["JOB_NAME"])
glue_context= GlueContext(SparkContext.getOrCreate())
spark = glueContext.spark_session
job = Job(glue_context)
job.init(args["JOB_NAME"], args)

# Read from DynamoDB
df = spark.read.format("dynamodb") \
    .option("dynamodb.input.tableName", "test-source") \
```

```
    .option("dynamodb.throughput.read.ratio", "0.5") \
    .option("dynamodb.consistentRead", "false") \
    .load()

print(df.rdd.getNumPartitions())

# Write to DynamoDB
df.write \
  .format("dynamodb") \
  .option("dynamodb.output.tableName", "test-sink") \
  .option("dynamodb.throughput.write.ratio", "0.5") \
  .option("dynamodb.item.size.check.enabled", "true") \
  .save()

job.commit()
```

## Scala

```
import com.amazonaws.services.glue.GlueContext
import com.amazonaws.services.glue.util.GlueArgParser
import com.amazonaws.services.glue.util.Job
import org.apache.spark.SparkContext
import scala.collection.JavaConverters._

object GlueApp {
  def main(sysArgs: Array[String]): Unit = {

    val glueContext = new GlueContext(SparkContext.getOrCreate())
    val spark = glueContext.getSparkSession
    val args = GlueArgParser.getResolvedOptions(sysArgs, Seq("JOB_NAME").toArray)
    Job.init(args("JOB_NAME"), glueContext, args.asJava)

    val df = spark.read
      .format("dynamodb")
      .option("dynamodb.input.tableName", "test-source")
      .option("dynamodb.throughput.read.ratio", "0.5")
      .option("dynamodb.consistentRead", "false")
      .load()

    print(df.rdd.getNumPartitions)

    df.write
      .format("dynamodb")
```

```
      .option("dynamodb.output.tableName", "test-sink")
      .option("dynamodb.throughput.write.ratio", "0.5")
      .option("dynamodb.item.size.check.enabled", "true")
      .save()

    job.commit()
  }
}
```

Utilizzo dell'esportazione DynamoDB tramite il connettore DataFrame-based

L'operazione di esportazione è preferita all'operazione di lettura per tabelle DynamoDB di dimensioni superiori a 80 GB. I seguenti esempi di codice mostrano come leggere da una tabella, esportare in S3 e stampare il numero di partizioni tramite il connettore. DataFrame-based

> **Note**
>
> La funzionalità di esportazione DynamoDB è disponibile tramite l'oggetto Scala. `DynamoDBExport` Gli utenti Python possono accedervi tramite l'interoperabilità JVM di Spark o utilizzare l'SDK AWS per Python (boto3) con l'API DynamoDB. `ExportTableToPointInTime`

Scala

```
import com.amazonaws.services.glue.GlueContext
import com.amazonaws.services.glue.util.{GlueArgParser, Job}
import org.apache.spark.SparkContext
import glue.spark.dynamodb.DynamoDBExport
import scala.collection.JavaConverters._

object GlueApp {
  def main(sysArgs: Array[String]): Unit = {
    val glueContext = new GlueContext(SparkContext.getOrCreate())
    val spark = glueContext.getSparkSession
    val args = GlueArgParser.getResolvedOptions(sysArgs, Seq("JOB_NAME").toArray)
    Job.init(args("JOB_NAME"), glueContext, args.asJava)

    val options = Map(
      "dynamodb.export" -> "ddb",
```

```
      "dynamodb.tableArn" -> "arn:aws:dynamodb:us-east-1:123456789012:table/my-
table",
      "dynamodb.s3.bucket" -> "my-s3-bucket",
      "dynamodb.s3.prefix" -> "my-s3-prefix",
      "dynamodb.simplifyDDBJson" -> "true"
    )
    val df = DynamoDBExport.fullExport(spark, options)

    print(df.rdd.getNumPartitions)
    df.count()

    Job.commit()
  }
}
```

Opzioni di configurazione

Opzioni di lettura

| Opzione | Description | Predefinita |
| --- | --- | --- |
| `dynamodb.input.tableName` | Nome della tabella DynamoDB (obbligatorio) | - |
| `dynamodb.throughput.read` | Le unità di capacità di lettura (RCU) da utilizzare. Se non specificato, `dynamodb.throughput.read.ratio` viene utilizzato per il calcolo. | - |
| `dynamodb.throughput.read.ratio` | Il rapporto tra unità di capacità di lettura (RCU) da utilizzare | 0,5 |
| `dynamodb.table.read.capacity` | La capacità di lettura della tabella su richiesta utilizzata per il calcolo della velocità effettiva. Questo parametro è valido solo nelle tabelle di capacità su richiesta. L'impostazione predefinita sono le unità di lettura a throughput caldo. | - |
| `dynamodb.splits` | Definisce il numero di segmenti utilizzati nelle operazioni di scansione parallela. Se non | - |

| Opzione | Description | Predefinita |
| --- | --- | --- |
| | viene fornito, il connettore calcolerà un valore predefinito ragionevole. | |
| `dynamodb.consistentRead` | Se utilizzare letture fortemente coerenti | FALSE |
| `dynamodb.input.retry` | Definisce quanti tentativi eseguiamo quando esiste un'eccezione riprovabile. | 10 |

Opzioni di scrittura

| Opzione | Description | Predefinita |
| --- | --- | --- |
| `dynamodb.output.tableName` | Nome della tabella DynamoDB (obbligatorio) | - |
| `dynamodb.throughput.write` | Le unità di capacità di scrittura (WCU) da utilizzare. Se non specificato, `dynamodb.throughput.write.ratio` viene utilizzato per il calcolo. | - |
| `dynamodb.throughput.write.ratio` | Il rapporto tra unità di capacità di scrittura (WCU) da utilizzare | 0,5 |
| `dynamodb.table.write.capacity` | La capacità di scrittura della tabella su richiesta utilizzata per il calcolo della velocità effettiva. Questo parametro è valido solo nelle tabelle di capacità su richiesta. L'impostazione predefinita sono le unità di scrittura con throughput caldo. | - |
| `dynamodb.item.size.check.enabled` | Se vero, il connettore calcola la dimensione dell'elemento e interrompe l'operazione se la dimensione supera la dimensione massima, prima di scrivere nella tabella DynamoDB. | TRUE |

| Opzione | Description | Predefinita |
| --- | --- | --- |
| `dynamodb.output.retry` | Definisce il numero di tentativi da eseguire in presenza di un'eccezione riutilizzabile. | 10 |

Opzioni di esportazione

| Opzione | Description | Predefinita |
| --- | --- | --- |
| `dynamodb.export` | Se impostato su, `ddb` abilita il connettore di esportazione AWS Glue DynamoDB dove verrà richiamato un `ExportTableToPointInTimeRequet` nuovo connettore durante il AWS processo Glue. Verrà generata una nuova esportazione con la posizione passata da `dynamodb.s3.bucket` e `dynamodb.s3.prefix` . Se impostato su, `s3` abilita il connettore di esportazione AWS Glue, DynamoDB, ma salta la creazione di una nuova esportazione DynamoDB e utilizza invece l'and `dynamodb.s3.bucket` come posizione Amazon S3 `dynamodb.s3.prefix` del passato esportato da quella tabella. | ddb |
| `dynamodb.tableArn` | La tabella DynamoDB da cui leggere. Obbligatorio se `dynamodb.export` è impostato su `ddb`. | |
| `dynamodb.simplifyDDBJson` | Se impostato su`true`, esegue una trasformazione per semplificare lo schema della struttura JSON di DynamoDB presente nelle esportazioni. | FALSE |
| `dynamodb.s3.bucket` | Il bucket S3 per archiviare i dati temporanei durante l'esportazione in DynamoDB (obbligatorio) | |

| Opzione | Description | Predefinita |
| --- | --- | --- |
| `dynamodb.s3.prefix` | Il prefisso S3 per archiviare dati temporanei durante l'esportazione in DynamoDB | |
| `dynamodb.s3.bucketOwner` | Indicare il proprietario del bucket necessario per l'accesso ad Amazon S3 su più account | |
| `dynamodb.s3.sse.algorithm` | Tipo di crittografia utilizzato nel bucket in cui verranno archiviati i dati temporanei. I valori validi sono `AES256` e `KMS`. | |
| `dynamodb.s3.sse.kmsKeyId` | L'ID della chiave AWS KMS gestita utilizzata per crittografare il bucket S3 in cui verranno archiviati i dati temporanei (se applicabile). | |
| `dynamodb.exportTime` | Un point-in-time in cui deve essere effettuata l'esportazione. Valori validi: stringhe che rappresentano istanti. ISO-8601 | |

Opzioni generali

| Opzione | Description | Predefinita |
| --- | --- | --- |
| `dynamodb.sts.roleArn` | Il ruolo IAM (ARN) da assumere per l'accesso tra account diversi | - |
| `dynamodb.sts.roleSessionName` | Nome della sessione STS | `glue-dynamodb-sts-session` |
| `dynamodb.sts.region` | Regione per il client STS (per l'assunzione di ruoli interregionali) | Uguale all'opzione `region` |

### Connessioni Kinesis

È possibile usare una connessione Kinesis per leggere e scrivere su flussi di dati Amazon Kinesis utilizzando le informazioni archiviate in una tabella del Catalogo dati o fornendo informazioni per accedere direttamente al flusso di dati. Puoi leggere le informazioni da Kinesis in Spark DataFrame, quindi convertirle in un Glue. AWS DynamicFrame Puoi DynamicFrames scrivere su Kinesis in formato JSON. Se accedi direttamente al flusso di dati, utilizza queste opzioni per fornire le informazioni su come accedere al flusso di dati.

Se si utilizza `getCatalogSource` o `create_data_frame_from_catalog` per consumare i registri da una sorgente di streaming Kinesis, il processo avrà le informazioni sul database catalogo dati e sul nome della tabella, e potrà usarle per ottenere alcuni parametri di base per la lettura dalla sorgente di streaming Kinesis. Se si utilizza `getSource`, `getSourceWithFormat`, `createDataFrameFromOptions` o `create_data_frame_from_options`, dovrai specificare questi parametri di base utilizzando le opzioni di connessione descritte qui.

È possibile specificare le opzioni di connessione per Kinesis utilizzando i seguenti argomenti per i metodi specificati nella classe `GlueContext`.

- Scala
  - `connectionOptions`: utilizza con `getSource`, `createDataFrameFromOptions` e `getSink`
  - `additionalOptions`: utilizza con `getCatalogSource`, `getCatalogSink`
  - `options`: utilizza con `getSourceWithFormat`, `getSinkWithFormat`
- Python
  - `connection_options`: utilizza con `create_data_frame_from_options`, `write_dynamic_frame_from_options`
  - `additional_options`: utilizza con `create_data_frame_from_catalog`, `write_dynamic_frame_from_catalog`
  - `options`: utilizza con `getSource`, `getSink`

Per osservazioni e restrizioni sui processi ETL dei flussi di dati, consultare la pagina the section called "Streaming di note e restrizioni ETL".

### Configurazione di Kinesis

Per connetterti a un flusso di dati Kinesis in un job AWS Glue Spark, avrai bisogno di alcuni prerequisiti:

- In caso di lettura, il job AWS Glue deve disporre delle autorizzazioni IAM di livello di accesso Read per il flusso di dati Kinesis.
- In fase di scrittura, il job AWS Glue deve disporre delle autorizzazioni IAM di livello di accesso Write per il flusso di dati Kinesis.

In alcuni casi, è necessario configurare ulteriori prerequisiti:

- Se il tuo job AWS Glue è configurato con connessioni di rete aggiuntive (in genere per connettersi ad altri set di dati) e una di queste connessioni offre opzioni di rete Amazon VPC, questo indirizzerà il tuo lavoro a comunicare tramite Amazon VPC. In questo caso, per comunicare tramite Amazon VPC dovrai configurare anche il flusso di dati Kinesis. È possibile farlo creando un endpoint VPC di interfaccia tra l'Amazon VPC e il flusso di dati Kinesis. Per ulteriori informazioni, consulta la pagina Using Amazon Kinesis Data Streams with Interface VPC Endpoints.
- Quando si specifica un flusso di dati Amazon Kinesis in un altro account, è necessario impostare i ruoli e le policy per consentire l'accesso multi-account. Per ulteriori informazioni, consultare Esempio: lettura da un flusso Kinesis in un account diverso.

Per ulteriori informazioni sui prerequisiti dei processi ETL dei flussi di dati, consultare la pagina the section called “Aggiunta di processi di streaming ETL”.

Esempio: lettura da flussi Kinesis

Esempio: lettura da flussi Kinesis

Usato in combinazione con the section called “forEachBatch”.

Esempio per l'origine di streaming Amazon Kinesis:

```
kinesis_options =
   { "streamARN": "arn:aws:kinesis:us-east-2:777788889999:stream/fromOptionsStream",
     "startingPosition": "TRIM_HORIZON",
     "inferSchema": "true",
     "classification": "json"
   }
data_frame_datasource0 =
 glueContext.create_data_frame.from_options(connection_type="kinesis",
 connection_options=kinesis_options)
```

Esempio: scrittura su flussi Kinesis

Esempio: lettura da flussi Kinesis

Usato in combinazione con the section called "forEachBatch".

Esempio per l'origine di streaming Amazon Kinesis:

```
kinesis_options =
   { "streamARN": "arn:aws:kinesis:us-east-2:777788889999:stream/fromOptionsStream",
     "startingPosition": "TRIM_HORIZON",
     "inferSchema": "true",
     "classification": "json"
   }
data_frame_datasource0 =
 glueContext.create_data_frame.from_options(connection_type="kinesis",
 connection_options=kinesis_options)
```

Indicazioni di riferimento alle opzioni di connessione a Kinesis

Indica le opzioni di connessione ad Amazon Kinesis Data Streams.

Utilizza le seguenti opzioni di connessione per le origini dati in streaming Kinesis:

- `"streamARN"`: (obbligatorio) utilizzato per la lettura/scrittura. L'ARN del flusso di dati di Kinesis.
- `"classification"`: (obbligatorio per la lettura) utilizzato per la lettura. Il formato di file utilizzato dai dati nel record. Obbligatorio, a meno che non sia fornito tramite Catalogo dati.
- `"streamName"`: (facoltativo) utilizzato per la lettura. Il nome di un flusso di dati Kinesis da cui leggere. Usato con `endpointUrl`.
- `"endpointUrl"`: (facoltativo) utilizzato per la lettura. Predefinito: "». https://kinesis.us-east-1.amazonaws.com L' AWS endpoint del flusso Kinesis. Non è necessario modificarlo a meno che non ci si stia connettendo a una regione speciale.
- `"partitionKey"`: (facoltativo) utilizzato per la scrittura. La chiave di partizione di Kinesis utilizzata per la produzione dei record.
- `"delimiter"`: (facoltativo) utilizzato per la lettura. Il separatore di valori utilizzato quando `classification` è CSV. Il valore predefinito è ",".
- `"startingPosition"`: (opzionale) utilizzato per la lettura. La posizione di partenza nel flusso dei dati Kinesis da cui leggere i dati. I valori possibili sono `"latest"`, `"trim_horizon"`, `"earliest"` o una stringa di timestamp in formato UTC con il modello `yyyy-mm-`

ddTHH:MM:SSZ, dove Z rappresenta uno scostamento del fuso orario UTC con un +/- (ad esempio: "2023-04-04T08:00:00-04:00"). Il valore predefinito è "latest". Nota: la stringa Timestamp in formato UTC per "startingPosition" è supportata solo per AWS Glue versione 4.0 o successiva.

- "failOnDataLoss": (facoltativo) non è possibile eseguire il processo se una partizione attiva è mancante o scaduta. Il valore predefinito è "false".
- "awsSTSRoleARN": (facoltativo) utilizzato per la lettura/scrittura. L'Amazon Resource Name (ARN) del ruolo da assumere utilizzando AWS Security Token Service (AWS STS). Questo ruolo deve disporre delle autorizzazioni per descrivere o leggere le operazioni dei registri per il flusso di dati Kinesis. Quando si accede a un flusso di dati in un altro account, è necessario utilizzare questo parametro. Usato in combinazione con "awsSTSSessionName".
- "awsSTSSessionName": (facoltativo) utilizzato per la lettura/scrittura. Un identificatore della sessione che assume il ruolo usando AWS STS. Quando si accede a un flusso di dati in un altro account, è necessario utilizzare questo parametro. Usato in combinazione con "awsSTSRoleARN".
- "awsSTSEndpoint": (Facoltativo) L' AWS STS endpoint da utilizzare quando ci si connette a Kinesis con un ruolo presunto. Ciò consente di utilizzare l' AWS STS endpoint regionale in un VPC, cosa non possibile con l'endpoint globale predefinito.
- "maxFetchTimeInMs": (opzionale) utilizzato per la lettura. Il tempo massimo impiegato affinché l'esecutore del processo legga i record della batch attuale dal flusso di dati Kinesis, specificato in millisecondi (ms). Entro questo periodo è possibile effettuate più chiamate API GetRecords. Il valore predefinito è 1000.
- "maxFetchRecordsPerShard": (opzionale) utilizzato per la lettura. Il numero massimo di record da recuperare per shard nel flusso di dati Kinesis per microbatch. Nota: il client può superare questo limite se il processo di streaming ha già letto i record aggiuntivi da Kinesis (nella stessa chiamata get-records). Se maxFetchRecordsPerShard deve essere rigoroso, allora deve essere un multiplo di maxRecordPerRead. Il valore predefinito è 100000.
- "maxRecordPerRead": (opzionale) utilizzato per la lettura. Il numero massimo di record da recuperare nel flusso di dati Kinesis in ciascuna operazione getRecords. Il valore predefinito è 10000.
- "addIdleTimeBetweenReads": (opzionale) utilizzato per la lettura. Aggiunge un ritardo tra due operazioni consecutive getRecords. Il valore predefinito è "False". Questa opzione è configurabile solo per Glue versione 2.0 e successive.

- `"idleTimeBetweenReadsInMs"`: (opzionale) utilizzato per la lettura. Il ritardo minimo tra due operazioni consecutive `getRecords`, specificato in ms. Il valore predefinito è `1000`. Questa opzione è configurabile solo per Glue versione 2.0 e successive.
- `"describeShardInterval"`: (opzionale) utilizzato per la lettura. L'intervallo di tempo minimo tra due chiamate API `ListShards` affinché lo script consideri il resharding. Per ulteriori informazioni, consultare Strategie per il resharding nella Guida per gli sviluppatori di Amazon Kinesis Data Streams. Il valore predefinito è `1s`.
- `"numRetries"`: (opzionale) utilizzato per la lettura. Il numero massimo di tentativi per le richieste API Kinesis Data Streams. Il valore predefinito è `3`.
- `"retryIntervalMs"`: (opzionale) utilizzato per la lettura. Il periodo di raffreddamento (specificato in ms) prima di riprovare la chiamata API Kinesis Data Streams. Il valore predefinito è `1000`.
- `"maxRetryIntervalMs"`: (opzionale) utilizzato per la lettura. Il periodo di raffreddamento (specificato in ms) tra due tentativi di chiamata API Kinesis Data Streams. Il valore predefinito è `10000`.
- `"avoidEmptyBatches"`: (opzionale) utilizzato per la lettura. Impedisce la creazione di un processo microbatch vuoto controllando la presenza di dati non letti nel flusso dei dati Kinesis prima che il batch venga avviato. Il valore predefinito è `"False"`.
- `"schema"`: (obbligatorio quando inferSchema è impostato su falso) utilizzato per la lettura. Lo schema da utilizzare per elaborare il payload. Se la classificazione è `avro`, lo schema fornito dovrà essere nel formato dello schema Avro. Se la classificazione è `avro`, lo schema fornito dovrà essere nel formato dello schema DDL.

  Di seguito sono riportati alcuni esempi di schema.

  Example in DDL schema format

  ```
  `column1` INT, `column2` STRING , `column3` FLOAT
  ```

  Example in Avro schema format

  ```
  {
    "type":"array",
    "items":
    {
      "type":"record",
      "name":"test",
      "fields":
      [
  ```

```
      {
        "name":"_id",
        "type":"string"
      },
      {
        "name":"index",
        "type":
        [
          "int",
          "string",
          "float"
        ]
      }
    ]
  }
}
```

- `"inferSchema"`: (opzionale) utilizzato per la lettura. Il valore predefinito è "false". Se impostato su "true", lo schema verrà rilevato in fase di runtime dal payload all'interno di `foreachbatch`.
- `"avroSchema"`: (obsoleto) utilizzato per la lettura. Parametro utilizzato per specificare uno schema di dati Avro quando viene utilizzato il formato Avro. Questo parametro è obsoleto. Utilizzo del parametro `schema`.
- `"addRecordTimestamp"`: (opzionale) utilizzato per la lettura. Quando questa opzione è impostata su "true", l'output dei dati conterrà una colonna aggiuntiva denominata "__src_timestamp" che indica l'ora in cui il record corrispondente è stato ricevuto dal flusso. Il valore predefinito è "false". Questa opzione è supportata in AWS Glue versione 4.0 o successive.
- `"emitConsumerLagMetrics"`: (opzionale) utilizzato per la lettura. Quando l'opzione è impostata su «true», per ogni batch emette le metriche relative alla durata compresa tra il record più vecchio ricevuto dallo stream e il momento in cui arriva a. AWS Glue CloudWatch Il nome della metrica è «glue.driver.streaming. maxConsumerLagInMs». Il valore predefinito è "false". Questa opzione è supportata in AWS Glue versione 4.0 o successive.
- `"fanoutConsumerARN"`: (opzionale) utilizzato per la lettura. L'ARN di un consumatore di un flusso Kinesis per il flusso specificato in `streamARN`. Utilizzato per abilitare la modalità di fan-out avanzato per la connessione Kinesis. Per ulteriori informazioni sull'utilizzo di un flusso Kinesis con fan-out avanzato, consultare la pagina the section called “Utilizzo del fan-out avanzato nei processi di flussi di dati Kinesis”.
- `"recordMaxBufferedTime"`: (opzionale) utilizzato per la scrittura. Predefinito: 1000 (ms). Tempo massimo di memorizzazione nel buffer di un record in attesa di essere scritto.

- `"aggregationEnabled"`: (opzionale) utilizzato per la scrittura. Default: true. Speciifica se i record devono essere aggregati prima di inviarli a Kinesis.
- `"aggregationMaxSize"`: (opzionale) utilizzato per la scrittura. Impostazione predefinita: 51200 (byte). Se un record è superiore a questo limite, ignorerà l'aggregatore. Ricorda che Kinesis impone un limite di 50 KB alla dimensione del record. Se imposti questo valore oltre i 50 KB, i record di grandi dimensioni verranno rifiutati da Kinesis.
- `"aggregationMaxCount"`: (facoltativo) utilizzato per la scrittura. Predefinito: 4294967295. Numero massimo di voci da inserire in un record aggregato.
- `"producerRateLimit"`: (facoltativo) utilizzato per la scrittura. Predefinito: 150 (%). Limita la velocità di trasmissione effettiva per partizione inviata da un singolo produttore (ad esempio, il tuo processo), come percentuale del limite di backend.
- `"collectionMaxCount"`: (facoltativo) utilizzato per la scrittura. Predefinito: 500. Numero massimo di articoli da imballare in una PutRecords richiesta.
- `"collectionMaxSize"`: (facoltativo) utilizzato per la scrittura. Impostazione predefinita: 5242880 (byte). Quantità massima di dati da inviare con una PutRecords richiesta.

Utilizzo del fan-out avanzato nei processi di flussi di dati Kinesis

Un consumatore con fan-out avanzato è in grado di ricevere i record da un flusso Kinesis con una velocità di trasmissione effettiva dedicata che può essere superiore a quella dei consumatori tipici. Questo viene fatto ottimizzando il protocollo di trasferimento utilizzato per fornire dati a un consumatore Kinesis, ad esempio il tuo processo. Per ulteriori informazioni sul fan-out avanzato di Kinesis, consulta la documentazione di Kinesis.

Nella modalità fan-out avanzato, le opzioni di connessione `maxRecordPerRead` e `idleTimeBetweenReadsInMs` non sono più valide, poiché tali parametri non sono configurabili quando si utilizza il fan-out avanzato. Le opzioni di configurazione per i nuovi tentativi funzionano come descritto.

Utilizza le seguenti procedure per abilitare e disabilitare il fan-out avanzato per il tuo processo di flussi di dati. Devi registrare un consumatore del flusso per ogni processo che consumerà i dati del flusso.

Per abilitare il consumo con fan-out avanzato nel processo:

1. Registra un consumatore del flusso per il tuo processo utilizzando l'API Kinesis. Segui le istruzioni per registrare un consumatore con fan-out avanzato utilizzando l'API Flusso di dati

Kinesis riportate nella documentazione di Kinesis. Dovrai solo seguire il primo passo: chiamare RegisterStreamConsumer. La tua richiesta dovrebbe restituire un ARN,. *consumerARN*

2. Imposta l'opzione di connessione `fanoutConsumerARN` su *consumerARN* negli argomenti del metodo di connessione.
3. Riavvia il processo.

Per disabilitare il consumo con fan-out avanzato nel processo:

1. Rimuovi l'opzione di connessione `fanoutConsumerARN` dalla chiamata al metodo.
2. Riavvia il processo.
3. Segui le istruzioni per annullare la registrazione di un consumatore riportate nella documentazione di Kinesis. Queste istruzioni si applicano alla console, ma possono essere utilizzate anche per l'API Kinesis. Per ulteriori informazioni sulla cancellazione della registrazione dei consumatori in streaming tramite l'API Kinesis, consulta la documentazione di Kinesis. DeregisterStreamConsumer

### Connessioni Amazon S3

Puoi usare AWS Glue for Spark per leggere e scrivere file in Amazon S3. AWS Glue for Spark supporta immediatamente molti formati di dati comuni archiviati in Amazon S3, tra cui CSV, Avro, JSON, Orc e Parquet. Per ulteriori informazioni sui formati di dati supportati, consulta la pagina the section called "Opzioni del formato dei dati". Ogni formato di dati può supportare un set diverso di funzionalità AWS Glue. Consulta la pagina relativa al formato dei dati per le specifiche del supporto delle funzionalità. Inoltre, puoi leggere e scrivere file con versioni diverse archiviati nei framework di data lake Hudi, Iceberg e Delta Lake. Per ulteriori informazioni sui framework di data lake, consulta la pagina the section called "Framework data lake".

Con AWS Glue puoi partizionare i tuoi oggetti Amazon S3 in una struttura di cartelle durante la scrittura, quindi recuperarli per partizione per migliorare le prestazioni utilizzando una semplice configurazione. È inoltre possibile impostare la configurazione per raggruppare file di piccole dimensioni durante la trasformazione dei dati per migliorare le prestazioni. In Amazon S3 è possibile leggere e scrivere archivi `bzip2` e `gzip`.

Argomenti

- Configurazione delle connessioni S3
- Indicazioni di riferimento alle opzioni di connessione ad Amazon S3

- Sintassi di connessione obsolete per i formati di dati
- Esclusione delle classi di storage Amazon S3
- Gestione delle partizioni per l'output ETL in AWS Glue
- Lettura di file di input in gruppi di grandi dimensioni
- Endpoint Amazon VPC per Amazon S3

### Configurazione delle connessioni S3

Per connetterti ad Amazon S3 con un job AWS Glue with Spark, avrai bisogno di alcuni prerequisiti:

- Il job AWS Glue deve disporre delle autorizzazioni IAM per i bucket Amazon S3 pertinenti.

In alcuni casi, è necessario configurare ulteriori prerequisiti:

- Quando configuri l'accesso multi-account, configura i controlli dell'accesso appropriati nel bucket Amazon S3.
- Per motivi di sicurezza, è possibile scegliere di instradare le richieste Amazon S3 tramite un Amazon VPC. Questo approccio può introdurre problematiche relative alla larghezza di banda e alla disponibilità. Per ulteriori informazioni, consulta the section called “Endpoint Amazon VPC per Amazon S3”.

### Indicazioni di riferimento alle opzioni di connessione ad Amazon S3

Indica una connessione ad Amazon S3.

Poiché Amazon S3 gestisce i file anziché le tabelle, oltre a specificare le proprietà di connessione fornite in questo documento, dovrai specificare una configurazione aggiuntiva sul tipo di file. Queste informazioni vengono specificate tramite le opzioni di formato dei dati. Per ulteriori informazioni sulle opzioni di formato, consulta la pagina the section called “Opzioni del formato dei dati”. È inoltre possibile specificare queste informazioni mediante l'integrazione con Catalogo dati AWS Glue.

Per un esempio della distinzione tra opzioni di connessione e opzioni di formato, prendi in considerazione come il metodo the section called “create_dynamic_frame_from_options” recepisce `connection_type`, `connection_options`, `format` e `format_options`. Questa sezione illustra in modo specifico i parametri forniti a `connection_options`.

Utilizzare le seguenti opzioni di connessione con `"connectionType": "s3"`:

- `"paths"`: (Obbligatorio) un elenco dei percorsi Amazon S3 da cui leggere.
- `"exclusions"`: (facoltativo) una stringa contenente un elenco di JSON di modelli glob in stile Unix da escludere. Ad esempio `"[\"**.pdf\"]"` esclude tutti i file PDF. Per ulteriori informazioni sulla sintassi glob supportata da AWS Glue, vedere Modelli di inclusione e di esclusione.
- `"compressionType"`: o `"compression"`: (facoltativo) specifica il modo in cui i dati sono compressi. Utilizza `"compressionType"` per origini Amazon S3 e `"compression"` per destinazioni Amazon S3. In genere questo non è necessario se i dati hanno un'estensione del file standard. I valori possibili sono `"gzip"` e `"bzip2"`). È possibile che vengano supportati formati di compressione aggiuntivi per formati specifici. Per i dettagli sul supporto delle varie funzionalità, consulta la pagina relativa al formato dei dati.
- `"groupFiles"`: (facoltativo) il raggruppamento di file è attivato per impostazione predefinita quando l'input contiene più di 50.000 file. Per attivare il raggruppamento con meno di 50.000 file, imposta questo parametro su `"inPartition"`. Per disabilitare il raggruppamento in presenza di più di 50.000 file, imposta il parametro su `"none"`.
- `"groupSize"`: (facoltativo) dimensione del gruppo target in byte. Il valore di default viene calcolato in base alla dimensione dei dati di input e alle dimensioni del cluster. Quando sono presenti meno di 50.000 file di input, `"groupFiles"` deve essere impostato su `"inPartition"` per rendere effettiva la modifica.
- `"recurse"`: (facoltativo) se è impostato su "true", legge i file in modo ricorsivo in tutte le sottodirectory dei percorsi specificati.
- `"maxBand"`: (facoltativo, avanzato) questa opzione controlla la durata in millisecondi dopo la quale è probabile che l'elenco `s3` sia coerente. I file con timestamp di modifica che rientrano negli ultimi `maxBand` secondi, vengono tracciati in modo specifico quando si usano `JobBookmarks` per verificare la consistenza finale in Amazon S3. Per la maggior parte degli utenti non è necessario impostare questa opzione. Il valore di default è 900.000 millisecondi o 15 minuti.
- `"maxFilesInBand"`: (Facoltativo, avanzato) questa opzione specifica il numero massimo di file da salvare negli ultimi `maxBand` secondi. Se si supera questo valore, i file aggiuntivi vengono saltati e solo elaborati nella successiva esecuzione del processo. Per la maggior parte degli utenti non è necessario impostare questa opzione.
- `"isFailFast"`: (Facoltativo) questa opzione determina se un processo ETL AWS Glue generi eccezioni di analisi del lettore. Se impostato su `true`, i processi non riescono rapidamente se quattro tentativi del processo Spark non riescono a analizzare correttamente i dati.
- `"catalogPartitionPredicate"`: (facoltativo) utilizzato per la lettura. Il contenuto di una clausola `WHERE` in SQL. Utilizzato per la lettura dalle tabelle di Catalogo dati con una

quantità molto elevata di partizioni. Recupera le partizioni corrispondenti dagli indici di Catalogo dati. Utilizzato con `push_down_predicate`, un'opzione sul metodo the section called "create_dynamic_frame_from_catalog" (e altri metodi simili). Per ulteriori informazioni, consulta the section called "Predicati di partizione di catalogo".

- `"partitionKeys"`: (facoltativo) utilizzato per la scrittura. Una matrice di stringhe di etichette di colonna. AWS Glue partizionerà i dati come specificato da questa configurazione. Per ulteriori informazioni, consulta the section called "Scrittura delle partizioni".
- `"excludeStorageClasses"`: (facoltativo) utilizzato per la lettura. Una serie di stringhe che specificano le classi di storage Amazon S3. AWS Glue escluderà gli oggetti Amazon S3 in base a questa configurazione. Per ulteriori informazioni, consulta the section called "Esclusione delle classi di storage Amazon S3".

Sintassi di connessione obsolete per i formati di dati

È possibile accedere a determinati formati di dati utilizzando una sintassi specifica per il tipo di connessione. Questa sintassi è obsoleta. Pertanto, si consiglia di specificare i formati utilizzando il tipo di connessione `s3` e le opzioni di formato fornite in the section called "Opzioni del formato dei dati".

"connectionType": "Orc"

Indica una connessione a file archiviati in Amazon S3 nel formato Apache Hive Optimized Row Columnar (ORC).

Utilizzare le seguenti opzioni di connessione con `"connectionType": "orc"`:

- `paths`: (Obbligatorio) un elenco dei percorsi Amazon S3 da cui leggere.
- (Altre name/value coppie di opzioni): tutte le opzioni aggiuntive, incluse le opzioni di formattazione, vengono passate direttamente a SparkSQL`DataSource`.

"connectionType": "parquet"

Indica una connessione a file archiviati in Amazon S3 nel formato di file Apache Parquet.

Utilizzare le seguenti opzioni di connessione con `"connectionType": "parquet"`:

- `paths`: (Obbligatorio) un elenco dei percorsi Amazon S3 da cui leggere.

- (Altre name/value coppie di opzioni): tutte le opzioni aggiuntive, incluse le opzioni di formattazione, vengono passate direttamente a SparkSQL`DataSource`.

Esclusione delle classi di storage Amazon S3

Se esegui processi ETL AWS Glue che leggono file o partizioni da Amazon Simple Storage Service (Amazon S3), puoi escludere alcuni tipi di classe di archiviazione Amazon S3.

Le seguenti classi di storage sono disponibili in Amazon S3:

- `STANDARD`: per lo storage generico dei dati a cui si accede di frequente.
- `INTELLIGENT_TIERING`: per i dati di lunga durata con modelli di accesso sconosciuti o modificati.
- `STANDARD_IA` e `ONEZONE_IA`: per i dati esistenti da molto tempo a cui si accede meno frequentemente.
- `GLACIER`, `DEEP_ARCHIVE` e `REDUCED_REDUNDANCY`: per l'archiviazione a lungo termine e la conservazione digitale.

Per ulteriori informazioni, consulta Classi di storage Amazon S3 nella Guida per gli sviluppatori di Amazon S3.

Gli esempi in questa sezione mostrano come escludere le classi di storage `DEEP_ARCHIVE` e `GLACIER`. Queste classi consentono di elencare i file, ma non di leggere i file a meno che non vengano ripristinati. (Per ulteriori informazioni, consulta Ripristino di oggetti archiviati nella Guida per gli sviluppatori di Amazon S3).

Utilizzando le esclusioni della classe di archiviazione, puoi assicurarti che i processi AWS Glue funzionino su tabelle che dispongono di partizioni tra questi livelli di classi di archiviazione. Senza esclusioni, i processi che leggono i dati da questi livelli hanno esito negativo con il seguente errore: AmazonS3Exception: l'operazione non è valida per la classe di storage dell'oggetto.

Esistono diversi modi per filtrare le classi di archiviazione Amazon S3 in AWS Glue.

Argomenti

- Esclusione delle classi di storage Amazon S3 durante la creazione di un frame dinamico
- Esclusione delle classi di storage Amazon S3 in una tabella del catalogo dati

Esclusione delle classi di storage Amazon S3 durante la creazione di un frame dinamico

Per escludere le classi di archiviazione Amazon S3 durante la creazione di un frame dinamico, utilizza `excludeStorageClasses` in `additionalOptions`. AWS Glue utilizza automaticamente la propria implementazione Amazon S3 `Lister` per elencare ed escludere i file corrispondenti alle classi di archiviazione specificate.

I seguenti esempi Python e Scala mostrano come escludere le classi di storage `GLACIER` e `DEEP_ARCHIVE` durante la creazione di un frame dinamico.

Esempio di Python:

```
glueContext.create_dynamic_frame.from_catalog(
    database = "my_database",
    tableName = "my_table_name",
    redshift_tmp_dir = "",
    transformation_ctx = "my_transformation_context",
    additional_options = {
        "excludeStorageClasses" : ["GLACIER", "DEEP_ARCHIVE"]
    }
)
```

Esempio di Scala:

```
val* *df = glueContext.getCatalogSource(
    nameSpace, tableName, "", "my_transformation_context",
    additionalOptions = JsonOptions(
        Map("excludeStorageClasses" -> List("GLACIER", "DEEP_ARCHIVE"))
    )
).getDynamicFrame()
```

Esclusione delle classi di storage Amazon S3 in una tabella del catalogo dati

È possibile specificare le esclusioni delle classi di archiviazione da utilizzare da un processo AWS Glue ETL come parametro di tabella nel AWS Glue Data Catalog. È possibile includere questo parametro nell'`CreateTable`operazione utilizzando AWS Command Line Interface (AWS CLI) o utilizzando l'API a livello di codice. Per ulteriori informazioni, vedere Struttura della tabella e. CreateTable

Puoi anche specificare classi di storage escluse nella console AWS Glue.

Per escludere classi di storage Amazon S3 (console)

1. Accedi a Console di gestione AWS e apri la AWS Glue console all'indirizzo https://console.aws.amazon.com/glue/.
2. Nel riquadro di navigazione a sinistra, scegliere Tables (Tabelle).
3. Scegliere il nome della tabella nell'elenco, quindi selezionare Edit table (Modifica tabella).
4. In Table properties (Proprietà tabella), aggiungere **`excludeStorageClasses`** come una chiave e **`[\"GLACIER\",\"DEEP_ARCHIVE\"]`** come un valore.
5. Scegli Applica.

## Gestione delle partizioni per l'output ETL in AWS Glue

Il partizionamento è una tecnica importante per organizzare i set di dati in modo da poterne eseguire query efficaci. I dati sono organizzati in una struttura gerarchica di directory basata su valori distinti di una o più colonne.

Ad esempio, puoi decidere di partizionare i log di un'applicazione in Amazon Simple Storage Service (Amazon S3) per data, suddivisi per anno, mese e giorno. I file che corrispondono ai dati di un solo giorno vengono quindi posti sotto un prefisso, ad esempio `s3://my_bucket/logs/year=2018/month=01/day=23/`. Sistemi come Amazon Athena, Amazon Redshift Spectrum e ora AWS Glue possono usare queste partizioni per filtrare i dati in base al valore di partizione senza dover leggere tutti i dati sottostanti da Amazon S3.

I crawler non solo deducono i tipi e gli schemi di file, ma identificano anche automaticamente la struttura delle partizioni del set di dati quando popolano il Glue Data Catalog. AWS Le colonne di partizione risultanti sono disponibili per le query nei processi ETL AWS Glue o per motori di query come Amazon Athena.

Dopo aver eseguito il crawling di una tabella, puoi visualizzare le partizioni create dal crawler. Nella console AWS Glue, scegli Applications (Applicazioni) nel pannello di navigazione sinistro. Scegli la tabella creata dal crawler, quindi scegli View Partitions (Visualizza partizioni).

Per i percorsi partizionati di tipo Apache Hive nello stile `key=val`, i crawler popolano automaticamente il nome della colonna usando il nome della chiave. Altrimenti, utilizzano i nomi predefiniti, ad esempio `partition_0`, `partition_1` e così via. È possibile modificare i nomi predefiniti sulla console. A tale scopo, accedi alla tabella. Controlla se gli indici esistono nella scheda Indici. In tal caso, è necessario eliminarli per procedere; potrai ricrearli in seguito utilizzando i nuovi nomi di colonna. Quindi, scegli Modifica schema e modifica i nomi delle colonne delle partizioni.

Negli script ETL, puoi applicare un filtro alle colonne di partizione. Poiché le informazioni sulla partizione sono memorizzate nel catalogo dati, utilizza le chiamate API `from_catalog` per includere le colonne delle partizioni nel `DynamicFrame`. Ad esempio, utilizza `create_dynamic_frame.from_catalog` anziché `create_dynamic_frame.from_options`.

Il partizionamento è una tecnica di ottimizzazione che riduce la scansione dei dati. Per ulteriori informazioni sul processo di identificazione di quando questa tecnica è appropriata, consulta Reduce the amount of data scan nella guida Best practices for performance tuning AWS Glue for Apache Spark jobs su AWS Prescriptive Guidance.

Prefiltraggio con i predicati pushdown

In molti casi, puoi utilizzare un predicato pushdown per filtrare le partizioni senza dover elencare e leggere tutti i file del set di dati. Invece di leggere l'intero set di dati e quindi filtrarlo in un file DynamicFrame, puoi applicare il filtro direttamente sui metadati della partizione nel Data Catalog. Quindi elenchi e leggi solo ciò di cui hai effettivamente bisogno in un. DynamicFrame

Ad esempio, in Python puoi scrivere quanto segue.

```
glue_context.create_dynamic_frame.from_catalog(
    database = "my_S3_data_set",
    table_name = "catalog_data_table",
    push_down_predicate = my_partition_predicate)
```

Questo crea un file DynamicFrame che carica solo le partizioni nel Data Catalog che soddisfano l'espressione del predicato. A seconda di quanto piccolo sia il sottoinsieme di dati che stai caricando, questo può far risparmiare moltissimo tempo nell'elaborazione.

L'espressione del predicato può essere qualsiasi espressione booleana supportata da Spark SQL. Funziona tutto ciò che puoi inserire in una clausola `WHERE` in una query Spark SQL. Ad esempio, l'espressione del predicato `pushDownPredicate = "(year=='2017' and month=='04')"` carica solo le partizioni in nel catalogo dati che hanno sia `year` uguale a 2017 che `month` uguale a 04. Per ulteriori informazioni, consulta la documentazione di Apache Spark SQL, in particolare il riferimento alle funzioni SQL Scala.

Filtraggio lato server utilizzando predicati delle partizioni di catalogo

L'opzione `push_down_predicate` viene applicata dopo aver elencato tutte le partizioni dal catalogo e prima di pubblicare i file da Amazon S3 per tali partizioni. Se per una tabella ci

sono molte partizioni, l'elenco delle partizioni del catalogo può comunque essere soggetto a un sovraccarico di tempo aggiuntivo. Per ovviare a questo sovraccarico, puoi utilizzare il partizionamento lato server con l'`catalogPartitionPredicate`opzione che utilizza gli indici di partizione nel Glue Data Catalog. AWS Questo rende il filtraggio delle partizioni molto più veloce quando sono presenti milioni di partizioni in una tabella. Puoi utilizzare sia `push_down_predicate` che `catalogPartitionPredicate` insieme in `additional_options` se il `catalogPartitionPredicate` richiede sintassi del predicato che non è ancora supportata con gli indici delle partizioni del catalogo.

Python:

```
dynamic_frame = glueContext.create_dynamic_frame.from_catalog(
    database=dbname,
    table_name=tablename,
    transformation_ctx="datasource0",
    push_down_predicate="day>=10 and customer_id like '10%'",
    additional_options={"catalogPartitionPredicate":"year='2021' and month='06'"}
)
```

Scala:

```
val dynamicFrame = glueContext.getCatalogSource(
    database = dbname,
    tableName = tablename,
    transformationContext = "datasource0",
    pushDownPredicate="day>=10 and customer_id like '10%'",
    additionalOptions = JsonOptions("""{
        "catalogPartitionPredicate": "year='2021' and month='06'"}""")
    ).getDynamicFrame()
```

> **Note**
>
> `push_down_predicate` e `catalogPartitionPredicate` usano sintassi diverse. Il primo utilizza la sintassi standard Spark SQL e il secondo utilizza il parser JSQL.

### Scrittura delle partizioni

Per impostazione predefinita, a non è partizionato quando viene scritto DynamicFrame . Tutti i file di output vengono scritti nel livello principale del percorso di output specificato. Fino a poco tempo fa,

l'unico modo per scrivere un file DynamicFrame nelle partizioni era convertirlo in un SQL Spark prima di scriverlo. DataFrame

Tuttavia, DynamicFrames ora supporta il partizionamento nativo utilizzando una sequenza di chiavi, utilizzando l'`partitionKeys`opzione quando si crea un sink. Ad esempio, il codice Python seguente scrive un set di dati in Amazon S3 in formato Parquet, in directory partizionate in base al campo type. Da qui puoi elaborare le partizioni usando altri sistemi, ad esempio Amazon Athena.

```
glue_context.write_dynamic_frame.from_options(
    frame = projectedEvents,
    connection_type = "s3",
    connection_options = {"path": "$outpath", "partitionKeys": ["type"]},
    format = "parquet")
```

Lettura di file di input in gruppi di grandi dimensioni

Puoi impostare le proprietà delle tabelle per permettere a un processo ETL AWS Glue di raggruppare i file quando vengono letti da un datastore Amazon S3. Queste proprietà permettono a ogni attività ETL di leggere un gruppo di file di input in una singola partizione in memoria. Ciò è particolarmente utile quando è presente un numero elevato di file di piccole dimensioni nel datastore Amazon S3. Quando imposti determinate proprietà, indichi ad AWS Glue di raggruppare i file in una partizione dati Amazon S3 e di impostare la dimensione dei gruppi da leggere. Puoi anche impostare queste opzioni durante la lettura da un datastore Amazon S3 con il metodo `create_dynamic_frame.from_options`.

Per abilitare il raggruppamento di file in una tabella, devi impostare coppie di valore chiave nel campo parametri della struttura della tabella. Utilizza la notazione JSON per impostare un valore per il campo parametri della tabella. Per ulteriori informazioni sulle modifiche delle proprietà di una tabella, consulta Visualizzazione e gestione dei dettagli della tabella.

Puoi usare questo metodo per abilitare il raggruppamento per le tabelle nel catalogo dati con i datastore Amazon S3.

groupFiles

Imposta groupFiles su `inPartition` per abilitare il raggruppamento dei file all'interno di una partizione dati Amazon S3. AWS Glue abilita automaticamente il raggruppamento se ci sono più di 50.000 file di input, come nel seguente esempio.

```
'groupFiles': 'inPartition'
```

groupSize

Imposta groupSize (Dimensione gruppo) alla dimensione target dei gruppi in byte. La proprietà groupSize è facoltativa e se non è presente AWS Glue calcola una dimensione per usare tutti i core della CPU nel cluster, riducendo comunque il numero complessivo di attività ETL e di partizioni in memoria.

Ad esempio, di seguito viene impostata la dimensione del gruppo su 1 MB.

```
'groupSize': '1048576'
```

È necessario impostare `groupsize` con il risultato di un calcolo. Ad esempio 1024 * 1024 = 1048576.

recurse

Imposta recurse su `True` per leggere in modo ricorsivo i file in tutte le sottodirectory quando si specifica `paths` come una matrice di percorsi. Non è necessario impostare recurse se `paths` è una matrice di chiavi di oggetti in Amazon S3 o se il formato di input è parquet/orc, come nell'esempio seguente.

```
'recurse':True
```

Se leggi da Amazon S3 utilizzando direttamente il metodo `create_dynamic_frame.from_options`, aggiungi queste opzioni di connessione. Ad esempio, i seguenti tentativi di raggruppare file in gruppi da 1 MB.

```
df = glueContext.create_dynamic_frame.from_options("s3", {'paths': ["s3://s3path/"],
 'recurse':True, 'groupFiles': 'inPartition', 'groupSize': '1048576'}, format="json")
```

> **Note**
>
> `groupFiles`è supportato se DynamicFrames creato dai seguenti formati di dati: csv, ion, GrokLog, json e xml. Questa opzione non è supportata per avro, parquet e orc.

Endpoint Amazon VPC per Amazon S3

Per motivi di sicurezza, molti AWS clienti eseguono le proprie applicazioni all'interno di un ambiente Amazon Virtual Private Cloud (Amazon VPC). Con Amazon VPC, puoi avviare EC2 istanze Amazon in un cloud privato virtuale, logicamente isolato da altre reti, inclusa la rete Internet pubblica. Con un Amazon VPC, puoi controllarne l'intervallo di indirizzi IP, le sottoreti, le tabelle di routing e i gateway di rete, nonché le impostazioni di sicurezza.

> **Note**
>
> Se hai creato il tuo AWS account dopo il 04/12/2013, hai già un VPC predefinito in ogni regione. AWS Puoi iniziare a utilizzare il tuo VPC predefinito subito, senza ulteriori configurazioni.
> Per ulteriori informazioni, consulta VPC e sottoreti predefiniti nella Guida per l'utente di Amazon VPC.

Molti clienti esprimono legittime preoccupazioni in materia di sicurezza e di privacy quando si tratta di inviare e ricevere dati tramite la rete Internet pubblica. I clienti possono risolvere questi problemi usando una rete privata virtuale (VPN) per instradare tutto il traffico di rete Amazon S3 tramite la propria infrastruttura di rete aziendale. Questo approccio tuttavia può introdurre problematiche relative alla larghezza di banda e alla disponibilità.

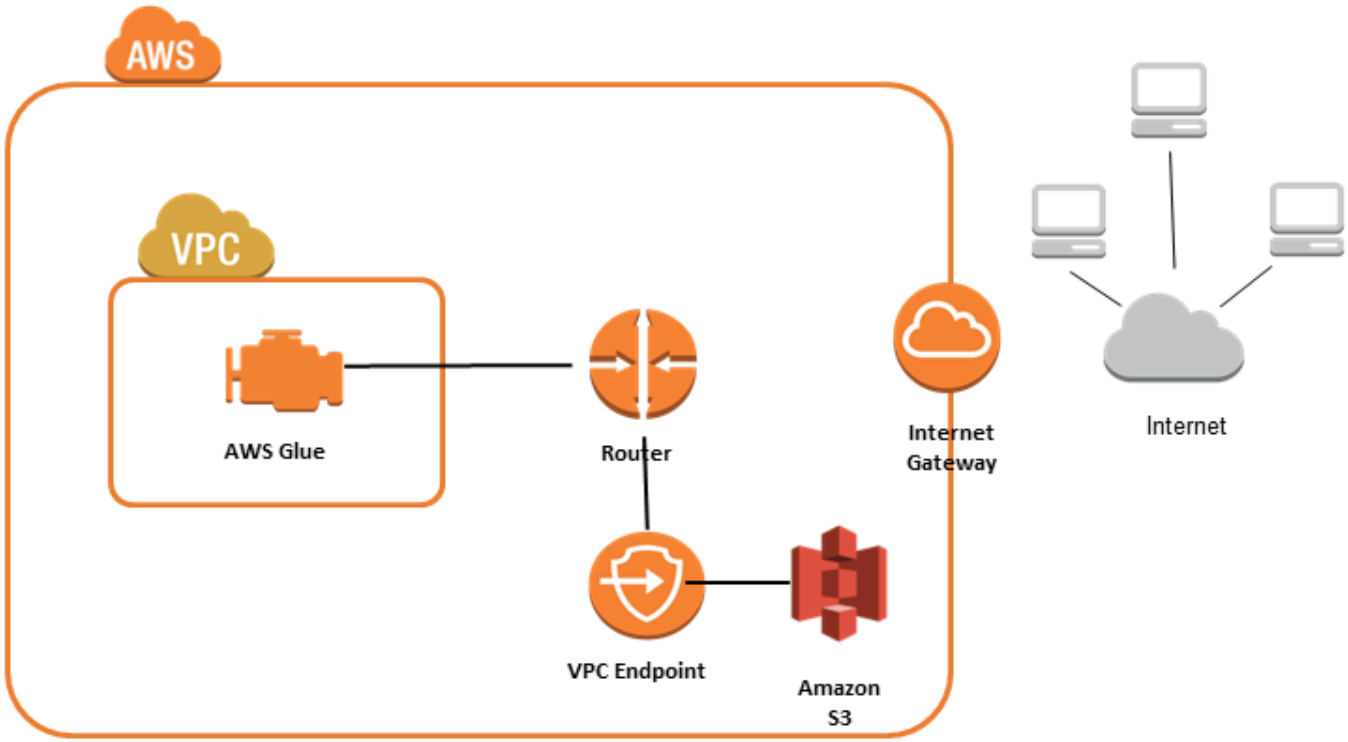
Gli endpoint VPC per Amazon S3 possono mitigare queste difficoltà. Un endpoint VPC per Amazon S3 permette a AWS Glue di usare indirizzi IP privati per accedere ad Amazon S3 senza esposizione nella rete Internet pubblica. AWS Glue non richiede indirizzi IP pubblici e non è necessario disporre di un gateway Internet, un dispositivo NAT o un gateway virtuale privato all'interno del VPC. Puoi utilizzare policy endpoint per controllare l'accesso ad Amazon S3. Il traffico tra il tuo VPC e il AWS servizio non esce dalla rete Amazon.

Quando crei un endpoint VPC per Amazon S3, qualsiasi richiesta a un endpoint Amazon S3 all'interno della regione (ad esempio s3.us-west-2.amazonaws.com) viene instradata verso un endpoint Amazon S3 privato all'interno della rete Amazon. Non è necessario modificare le

applicazioni in esecuzione su EC2 istanze Amazon nel tuo VPC: il nome dell'endpoint rimane lo stesso, ma il percorso verso Amazon S3 rimane interamente all'interno della rete Amazon e non accede alla rete Internet pubblica.

Per ulteriori informazioni sugli endpoint VPC, consulta Endpoint VPC nella Guida per l'utente di Amazon VPC.

Il diagramma seguente mostra in che modo AWS Glue può usare un endpoint VPC per accedere ad Amazon S3.




Per configurare l'accesso ad Amazon S3

1. Accedi Console di gestione AWS e apri la console Amazon VPC all'indirizzo. https://console.aws.amazon.com/vpc/
2. Nel riquadro di navigazione a sinistra, scegli Endpoints (Endpoint).

3. Scegli Create Endpoint (Crea endpoint) e segui la procedura per creare un endpoint VPC Amazon S3 di tipo Gateway.

### Connessioni ad Amazon DocumentDB

Puoi usare AWS Glue for Spark per leggere e scrivere su tabelle in Amazon DocumentDB. Puoi connetterti ad Amazon DocumentDB utilizzando le credenziali archiviate tramite Gestione dei segreti AWS una connessione AWS Glue.

Per ulteriori informazioni su Amazon DocumentDB, consulta la documentazione di Amazon DocumentDB.

> **Note**
>
> I cluster elastici di Amazon DocumentDB non sono attualmente supportati quando si utilizza il connettore AWS Glue. Per ulteriori informazioni sui cluster elastici, consulta la pagina Using Amazon DocumentDB elastic clusters.

### Lettura e scrittura nelle raccolte Amazon DocumentDB

> **Note**
>
> Quando crei un processo ETL a cui si connette Amazon DocumentDB, per la proprietà del processo `Connections`, devi designare un oggetto connessione che specifica il cloud privato virtuale (VPC) in cui Amazon DocumentDB è in esecuzione. Per l'oggetto connessione, il tipo di connessione deve essere `JDBC`, e `JDBC URL` deve essere `mongo://<DocumentDB_host>:27017`.

> **Note**
>
> Questi esempi di codice sono stati sviluppati per AWS Glue 3.0. Per migrare a AWS Glue 4.0, consultathe section called "MongoDB". Il parametro `uri` è cambiato.

> **Note**
>
> Quando si utilizza Amazon DocumentDB, `retryWrites` deve essere impostato su false in determinate situazioni, ad esempio quando il documento scritto specifica `_id`. Per ulteriori informazioni, consulta Differenze funzionali con MongoDB nella documentazione di Amazon DocumentDB.

Il seguente script Python dimostra l'utilizzo di tipi e opzioni di connessione per la lettura e la scrittura su Amazon DocumentDB.

```
import sys
from awsglue.transforms import *
from awsglue.utils import getResolvedOptions
from pyspark.context import SparkContext, SparkConf
from awsglue.context import GlueContext
from awsglue.job import Job
import time

## @params: [JOB_NAME]
args = getResolvedOptions(sys.argv, ['JOB_NAME'])

sc = SparkContext()
glueContext = GlueContext(sc)
spark = glueContext.spark_session

job = Job(glueContext)
job.init(args['JOB_NAME'], args)

output_path = "s3://some_bucket/output/" + str(time.time()) + "/"
documentdb_uri = "mongodb://<mongo-instanced-ip-address>:27017"
documentdb_write_uri = "mongodb://<mongo-instanced-ip-address>:27017"

read_docdb_options = {
    "uri": documentdb_uri,
    "database": "test",
    "collection": "coll",
    "username": "username",
    "password": "1234567890",
    "ssl": "true",
    "ssl.domain_match": "false",
    "partitioner": "MongoSamplePartitioner",
```

```
    "partitionerOptions.partitionSizeMB": "10",
    "partitionerOptions.partitionKey": "_id"
}

write_documentdb_options = {
    "retryWrites": "false",
    "uri": documentdb_write_uri,
    "database": "test",
    "collection": "coll",
    "username": "username",
    "password": "pwd"
}

# Get DynamicFrame from  DocumentDB
dynamic_frame2 =
 glueContext.create_dynamic_frame.from_options(connection_type="documentdb",

 connection_options=read_docdb_options)

# Write DynamicFrame to MongoDB and DocumentDB
glueContext.write_dynamic_frame.from_options(dynamic_frame2,
 connection_type="documentdb",

 connection_options=write_documentdb_options)

job.commit()
```

Il seguente script Scala dimostra l'utilizzo di tipi e opzioni di connessione per la lettura e la scrittura su Amazon DocumentDB.

```
import com.amazonaws.services.glue.GlueContext
import com.amazonaws.services.glue.MappingSpec
import com.amazonaws.services.glue.errors.CallSite
import com.amazonaws.services.glue.util.GlueArgParser
import com.amazonaws.services.glue.util.Job
import com.amazonaws.services.glue.util.JsonOptions
import com.amazonaws.services.glue.DynamicFrame
import org.apache.spark.SparkContext
import scala.collection.JavaConverters._

object GlueApp {
  val DOC_URI: String = "mongodb://<mongo-instanced-ip-address>:27017"
  val DOC_WRITE_URI: String = "mongodb://<mongo-instanced-ip-address>:27017"
```

```
  lazy val documentDBJsonOption = jsonOptions(DOC_URI)
  lazy val writeDocumentDBJsonOption = jsonOptions(DOC_WRITE_URI)
  def main(sysArgs: Array[String]): Unit = {
    val spark: SparkContext = new SparkContext()
    val glueContext: GlueContext = new GlueContext(spark)
    val args = GlueArgParser.getResolvedOptions(sysArgs, Seq("JOB_NAME").toArray)
    Job.init(args("JOB_NAME"), glueContext, args.asJava)

    // Get DynamicFrame from DocumentDB
    val resultFrame2: DynamicFrame = glueContext.getSource("documentdb",
 documentDBJsonOption).getDynamicFrame()

    // Write DynamicFrame to DocumentDB
    glueContext.getSink("documentdb", writeJsonOption).writeDynamicFrame(resultFrame2)

    Job.commit()
  }

  private def jsonOptions(uri: String): JsonOptions = {
    new JsonOptions(
      s"""{"uri": "${uri}",
         |"database":"test",
         |"collection":"coll",
         |"username": "username",
         |"password": "pwd",
         |"ssl":"true",
         |"ssl.domain_match":"false",
         |"partitioner": "MongoSamplePartitioner",
         |"partitionerOptions.partitionSizeMB": "10",
         |"partitionerOptions.partitionKey": "_id"}""".stripMargin)
  }
}
```

Indicazioni di riferimento alle opzioni di connessione ad Amazon DocumentDB

Indica una connessione ad Amazon DocumentDB (con compatibilità MongoDB).

Le opzioni di connessione differiscono per una connessione sorgente e una connessione sink.

"connectionType": "documentdb" come sorgente

Utilizzare le seguenti opzioni di connessione con `"connectionType": "documentdb"` come origine:

- `"uri"`: (obbligatorio) l'host Amazon DocumentDB da cui leggere, formattato come `mongodb://<host>:<port>`.
- `"database"`: (Obbligatorio) il database di Amazon DocumentDB da cui leggere.
- `"collection"`: (Obbligatorio) la raccolta di Amazon DocumentDB da cui leggere.
- `"username"`: (Obbligatorio) il nome utente di Amazon DocumentDB.
- `"password"`: (Obbligatorio) la password di Amazon DocumentDB.
- `"ssl"`: (Obbligatorio se si utilizza SSL) se la connessione utilizza SSL, è necessario includere questa opzione con il valore `"true"`.
- `"ssl.domain_match"`: (Obbligatorio se si utilizza SSL) se la connessione utilizza SSL, è necessario includere questa opzione con il valore `"false"`.
- `"batchSize"`: (Facoltativo): il numero di documenti da restituire per ogni batch, utilizzato all'interno del cursore dei batch interni.
- `"partitioner"`: (Facoltativo): il nome della classe del partizionatore per la lettura dei dati di input da Amazon DocumentDB. Il connettore fornisce i seguenti partizionatori:
  - `MongoDefaultPartitioner`(impostazione predefinita) (non supportato in AWS Glue 4.0)
  - `MongoSamplePartitioner`(Non supportato in AWS Glue 4.0)
  - `MongoShardedPartitioner`
  - `MongoSplitVectorPartitioner`
  - `MongoPaginateByCountPartitioner`
  - `MongoPaginateBySizePartitioner`(Non supportato in AWS Glue 4.0)
- `"partitionerOptions"` ( Facoltativo): opzioni per il partizionatore designato. Per ogni partizionatore sono supportate le seguenti opzioni:
  - `MongoSamplePartitioner`: `partitionKey`, `partitionSizeMB`, `samplesPerPartition`
  - `MongoShardedPartitioner`: `shardkey`
  - `MongoSplitVectorPartitioner`: `partitionKey`, partitionSizeMB
  - `MongoPaginateByCountPartitioner`: `partitionKey`, `numberOfPartitions`
  - `MongoPaginateBySizePartitioner`: `partitionKey`, partitionSizeMB

  Per ulteriori informazioni su queste opzioni, vedere Partitioner Configuration (Configurazione partizionatore) nella documentazione di MongoDB.

"connectionType": "documentdb" come sink

Utilizzare le seguenti opzioni di connessione con `"connectionType": "documentdb"` come sink:

- `"uri"`: (Obbligatorio) l'host Amazon DocumentDB su cui scrivere, formattato come `mongodb://<host>:<port>`.
- `"database"`: (Obbligatorio) il database di Amazon DocumentDB su cui scrivere.
- `"collection"`: (Obbligatorio) la raccolta di Amazon DocumentDB su cui scrivere.
- `"username"`: (Obbligatorio) il nome utente di Amazon DocumentDB.
- `"password"`: (Obbligatorio) la password di Amazon DocumentDB.
- `"extendedBsonTypes"`: (Facoltativo) se il valore è `true`, abilita i tipi BSON estesi durante la scrittura dei dati su Amazon DocumentDB. Il valore predefinito è `true`.
- `"replaceDocument"`: (Facoltativo) Se il valore è `true`, sostituisce l'intero documento quando si salvano set di dati che contengono un campo `_id`. Se il valore è `false`, vengono aggiornati solo i campi del documento che corrispondono ai campi del set di dati. Il valore predefinito è `true`.
- `"maxBatchSize"`: (Facoltativo): la dimensione massima del batch per le operazioni in blocco durante il salvataggio dei dati. Il valore predefinito è 512.
- `"retryWrites"`: (Facoltativo): Riprova automaticamente alcune operazioni di scrittura una sola volta se AWS Glue rileva un errore di rete.

OpenSearch Connessioni di servizio

Puoi usare AWS Glue for Spark per leggere e scrivere su tabelle in OpenSearch Service in AWS Glue 4.0 e versioni successive. Puoi definire cosa leggere dal OpenSearch Servizio con una OpenSearch query. Ti connetti al OpenSearch servizio utilizzando le credenziali di autenticazione di base HTTP archiviate Gestione dei segreti AWS tramite una connessione AWS Glue. Questa funzionalità non è compatibile con OpenSearch Service serverless.

Per ulteriori informazioni su Amazon OpenSearch Service, consulta la documentazione OpenSearch di Amazon Service.

Configurazione delle connessioni OpenSearch al servizio

Per connetterti a OpenSearch Service da AWS Glue, dovrai creare e archiviare le tue credenziali OpenSearch Service in modo Gestione dei segreti AWS segreto, quindi associare quel segreto a una connessione OpenSearch Service AWS Glue.

Prerequisiti:

- Identifica l'endpoint *aosEndpoint* e la porta del dominio da *aosPort* cui desideri leggere o crea la risorsa seguendo le istruzioni nella documentazione di Amazon OpenSearch Service. Per ulteriori informazioni sulla creazione di un dominio, consulta Creazione e gestione di domini Amazon OpenSearch Service nella documentazione di Amazon OpenSearch Service.

  Un endpoint OpenSearch di dominio Amazon Service avrà il seguente modulo predefinito, https://search - *domainName* -*unstructuredIdContent*. *region*.es.amazonaws.com. Per ulteriori informazioni sull'identificazione dell'endpoint del tuo dominio, consulta Creazione e gestione dei domini Amazon OpenSearch Service nella documentazione di Amazon OpenSearch Service.

  Identifica o genera credenziali di autenticazione HTTP di base *aosUser* e *aosPassword* per il tuo dominio.

Per configurare una connessione al OpenSearch servizio:

1. In Gestione dei segreti AWS, crea un segreto utilizzando le tue credenziali OpenSearch di servizio. Per creare un segreto in Secrets Manager, segui il tutorial disponibile in Crea un Gestione dei segreti AWS segreto nella Gestione dei segreti AWS documentazione. Dopo aver creato il segreto, mantieni il nome del segreto *secretName* per il passaggio successivo.

   - Quando selezionate le coppie chiave/valore, create una coppia per la chiave `USERNAME` con il valore. *aosUser*
   - Quando selezionate le coppie chiave/valore, create una coppia per la chiave `PASSWORD` con il valore. *aosPassword*
2. Nella console AWS Glue, crea una connessione seguendo i passaggi riportati di seguitothe section called "Aggiungere una AWS Glue connessione". Dopo aver creato la connessione, mantieni il nome della connessione*connectionName*, per utilizzi futuri in AWS Glue.

   - Quando selezioni un tipo di connessione, seleziona OpenSearch Servizio.
   - Quando selezioni un endpoint di dominio, fornisci*aosEndpoint*.
   - Quando selezioni una porta, fornisci*aosPort*.
   - Quando selezioni un AWS segreto, fornisci*secretName*.

Dopo aver creato una connessione OpenSearch al servizio AWS Glue, dovrai eseguire i seguenti passaggi prima di eseguire il lavoro AWS Glue:

- Concedi al ruolo IAM associato al tuo lavoro AWS Glue il permesso di lettura*secretName*.

- Nella configurazione del lavoro AWS Glue, fornisci *connectionName* una connessione di rete aggiuntiva.

Lettura dagli indici OpenSearch dei servizi

Prerequisiti:

- Un indice dei OpenSearch servizi da cui desideri leggere,*aosIndex*.
- Una connessione AWS Glue OpenSearch Service configurata per fornire informazioni di autenticazione e posizione di rete. Per acquisirla, completa i passaggi della procedura precedente, Per configurare una connessione al OpenSearch servizio. Avrai bisogno del nome della connessione AWS Glue,*connectionName*.

Questo esempio legge un indice da Amazon OpenSearch Service. Dovrai fornire il parametro `pushdown`.

Esempio:

```
opensearch_read = glueContext.create_dynamic_frame.from_options(
    connection_type="opensearch",
    connection_options={
        "connectionName": "connectionName",
        "opensearch.resource": "aosIndex",
        "pushdown": "true",
    }
)
```

Puoi anche fornire una stringa di query per filtrare i risultati restituiti nel tuo DynamicFrame. Sarà necessario configurare `opensearch.query`.

`opensearch.query`può accettare una stringa di parametri di query URL *queryString* o un oggetto JSON DSL di query. *queryObject* Per ulteriori informazioni sulla query DSL, vedere Query DSL nella documentazione. OpenSearch Per fornire una stringa di parametri di query URL, anteponi `?q=` alla query, come faresti con un URL completo. Per fornire un oggetto di query DSL, la stringa evita l'oggetto JSON prima di fornirlo.

Esempio:

```
            queryObject = "{ "query": { "multi_match": { "query": "Sample", "fields":
 [ "sample" ] } } }"
            queryString = "?q=queryString"

            opensearch_read_query = glueContext.create_dynamic_frame.from_options(
    connection_type="opensearch",
    connection_options={
        "connectionName": "connectionName",
        "opensearch.resource": "aosIndex",
        "opensearch.query": queryString,
        "pushdown": "true",
    }
)
```

Per ulteriori informazioni su come creare una query al di fuori della relativa sintassi specifica, vedi Sintassi della stringa di query nella documentazione. OpenSearch

Quando si leggono da OpenSearch raccolte che contengono dati di tipo array, è necessario specificare quali campi sono di tipo array nella chiamata al metodo utilizzando il `opensearch.read.field.as.array.include` parametro.

Ad esempio, durante la lettura del documento seguente incontrerai i campi di array `genre` e `actor`:

```
{
    "_index": "movies",
    "_id": "2",
    "_version": 1,
    "_seq_no": 0,
    "_primary_term": 1,
    "found": true,
    "_source": {
        "director": "Frankenheimer, John",
        "genre": [
            "Drama",
            "Mystery",
            "Thriller",
            "Crime"
        ],
        "year": 1962,
        "actor": [
            "Lansbury, Angela",
            "Sinatra, Frank",
            "Leigh, Janet",
```

```
            "Harvey, Laurence",
            "Silva, Henry",
            "Frees, Paul",
            "Gregory, James",
            "Bissell, Whit",
            "McGiver, John",
            "Parrish, Leslie",
            "Edwards, James",
            "Flowers, Bess",
            "Dhiegh, Khigh",
            "Payne, Julie",
            "Kleeb, Helen",
            "Gray, Joe",
            "Nalder, Reggie",
            "Stevens, Bert",
            "Masters, Michael",
            "Lowell, Tom"
        ],
        "title": "The Manchurian Candidate"
    }
}
```

In questo caso, dovrai includere i nomi dei campi in questione nella chiamata al metodo. Esempio:

```
"opensearch.read.field.as.array.include": "genre,actor"
```

Se il campo dell'array è annidato all'interno della struttura del documento, fai riferimento a esso utilizzando la notazione a punti: `"genre,actor,foo.bar.baz"`. Ciò specificherebbe un array `baz` incluso nel documento di origine tramite il documento incorporato `foo` contenente il documento incorporato `bar`.

Scrittura nelle tabelle OpenSearch dei servizi

Questo esempio scrive informazioni da un servizio esistente DynamicFrame *`dynamicFrame`* a OpenSearch Service. Se l'indice contiene già informazioni, AWS Glue aggiungerà i dati dal tuo DynamicFrame. Dovrai fornire il parametro `pushdown`.

Prerequisiti:

- Una tabella dei OpenSearch servizi su cui scrivere. Avrai bisogno delle informazioni di identificazione per la tabella. Chiamiamo questo*`tableName`*.

- Una connessione AWS Glue OpenSearch Service configurata per fornire informazioni di autenticazione e posizione di rete. Per acquisirla, completa i passaggi della procedura precedente, Per configurare una connessione al OpenSearch servizio. Avrai bisogno del nome della connessione AWS Glue,*connectionName*.

Esempio:

```
glueContext.write_dynamic_frame.from_options(
    frame=dynamicFrame,
    connection_type="opensearch",
    connection_options={
      "connectionName": "connectionName",
      "opensearch.resource": "aosIndex",
    },
)
```

OpenSearch Riferimento all'opzione di connessione al servizio

- `connectionName`: obbligatorio. Utilizzato per la lettura/scrittura. Il nome di una connessione al AWS Glue OpenSearch Service configurata per fornire informazioni di autenticazione e posizione di rete al metodo di connessione utilizzato.
- `opensearch.resource`: obbligatorio. Utilizzato per la lettura/scrittura. Valori validi: nomi degli OpenSearch indici. Il nome dell'indice con cui interagirà il metodo di connessione.
- `opensearch.query`: utilizzato per la lettura. Valori validi: stringa con escape JSON o, quando inizia con `?`, la parte di ricerca di un URL. Una OpenSearch query che filtra ciò che deve essere recuperato durante la lettura. Per ulteriori informazioni sull'utilizzo di questo parametro, consulta la sezione precedente the section called “Leggi dal servizio OpenSearch ”.
- `pushdown` — Richiesto se. Utilizzato per la lettura. Valori validi: booleani. Indica a Spark di passare le query di lettura in OpenSearch modo che il database restituisca solo i documenti pertinenti.
- `opensearch.read.field.as.array.include`: richiesto se si leggono dati di tipo array. Utilizzato per la lettura. Valori validi: elenchi di nomi di campi separati da virgole. Specifica i campi da leggere come matrici dai documenti. OpenSearch Per ulteriori informazioni sull'utilizzo di questo parametro, consulta la sezione precedente the section called “Leggi dal servizio OpenSearch ”.

## Connessioni Redshift

Puoi usare AWS Glue for Spark per leggere e scrivere su tabelle nei database Amazon Redshift. Quando si connette ai database Amazon Redshift, AWS Glue sposta i dati tramite Amazon S3 per ottenere il massimo throughput, utilizzando SQL e comandi Amazon Redshift. `COPY` `UNLOAD` In AWS Glue 4.0 e versioni successive, puoi utilizzare l'integrazione Amazon Redshift per Apache Spark per leggere e scrivere con ottimizzazioni e funzionalità specifiche di Amazon Redshift oltre a quelle disponibili durante la connessione tramite versioni precedenti.

Scopri come AWS Glue sta semplificando più che mai per gli utenti di Amazon Redshift la migrazione a AWS Glue per l'integrazione dei dati senza server e l'ETL.

### Configurazione delle connessioni Redshift

Per utilizzare i cluster Amazon Redshift in AWS Glue, sono necessari alcuni prerequisiti:

- Una directory Amazon S3 da utilizzare per l'archiviazione temporanea durante la lettura e la scrittura sul database.
- Un Amazon VPC che consente la comunicazione tra il cluster Amazon Redshift, il job AWS Glue e la directory Amazon S3.
- Autorizzazioni IAM appropriate sul job AWS Glue e sul cluster Amazon Redshift.

### Configurazione dei ruoli IAM

#### Configurazione del ruolo per il cluster Amazon Redshift

Il tuo cluster Amazon Redshift deve essere in grado di leggere e scrivere su Amazon S3 per integrarsi con AWS Glue jobs. Per consentire ciò, puoi associare i ruoli IAM al cluster Amazon Redshift a cui desideri connetterti. Il tuo ruolo dovrebbe disporre di una policy che consenta la lettura e la scrittura nella tua directory temporanea di Amazon S3. Il tuo ruolo dovrebbe avere un rapporto di fiducia che consenta al servizio `redshift.amazonaws.com` di `AssumeRole`.

#### Associazione di un ruolo IAM ad Amazon Redshift

1. Prerequisiti: un bucket o una directory Amazon S3 utilizzato per l'archiviazione temporanea dei file.
2. Identifica le autorizzazioni Amazon S3 che occorreranno al cluster Amazon Redshift. Quando spostano dati da e verso un cluster Amazon Redshift, i job AWS Glue emettono istruzioni COPY e UNLOAD su Amazon Redshift. Se il tuo job modifica una tabella in Amazon Redshift,

AWS Glue emetterà anche istruzioni CREATE LIBRARY. Per informazioni sulle autorizzazioni specifiche di Amazon S3 necessarie ad Amazon Redshift per eseguire queste istruzioni, consulta la documentazione di Amazon Redshift: Amazon Redshift: Permissions to access other Resources. AWS

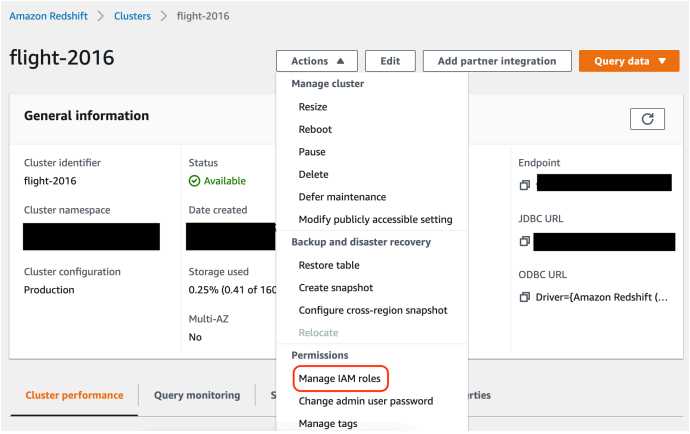
3. Nella console IAM, crea una policy IAM con le autorizzazioni necessarie. Per ulteriori informazioni sulla creazione di una policy, consulta la pagina Creazione di policy IAM.
4. Da IAM, crea un ruolo e un rapporto di fiducia che consenta ad Amazon Redshift di assumere il ruolo. Segui le istruzioni nella documentazione IAM Per creare un ruolo per un servizio (console) AWS

   - Quando ti viene chiesto di scegliere un caso d'uso del AWS servizio, scegli «Redshift - Personalizzabile».
   - Quando ti viene chiesto di collegare una policy, scegli la policy che hai definito in precedenza.

   > **Note**
   >
   > Per ulteriori informazioni sulla configurazione dei ruoli per Amazon Redshift, consulta Autorizzazione di Amazon Redshift ad AWS accedere ad altri servizi per tuo conto nella documentazione di Amazon Redshift.

5. Da Amazon Redshift, associa il ruolo al tuo cluster Amazon Redshift. Segui le istruzioni nella documentazione di Amazon Redshift.

   Seleziona l'opzione evidenziata nella console Amazon Redshift per configurare questa impostazione:



> **Note**
>
> Per impostazione predefinita, i lavori AWS Glue passano le credenziali temporanee di Amazon Redshift create utilizzando il ruolo specificato per eseguire il lavoro. Non è consigliabile utilizzare queste credenziali. Per motivi di sicurezza, queste credenziali scadono dopo 1 ora.

Imposta il ruolo per il lavoro AWS Glue

Il job AWS Glue richiede un ruolo per accedere al bucket Amazon S3. Non sono necessarie le autorizzazioni IAM per il cluster Amazon Redshift, il tuo accesso è controllato dalla connettività in Amazon VPC e dalle credenziali del database.

Configurazione di Amazon VPC

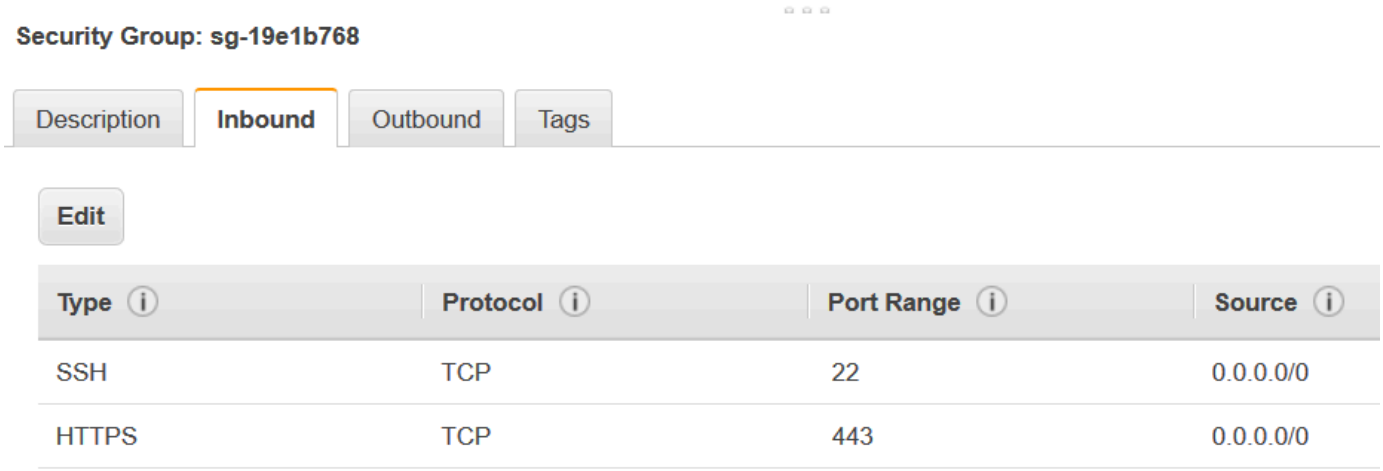
Per configurare l'accesso ai datastore Amazon Redshift

1. Accedi Console di gestione AWS e apri la console Amazon Redshift all'indirizzo. https://console.aws.amazon.com/redshiftv2/
2. Nel pannello di navigazione a sinistra, seleziona Cluster.
3. Seleziona il nome del cluster al quale desideri accedere da AWS Glue.
4. Nella sezione Proprietà cluster scegli un gruppo di sicurezza in Gruppi di sicurezza VPC per permettere l'uso a AWS Glue. Registra il nome del gruppo di sicurezza scelto per riferimenti futuri. La scelta del gruppo di sicurezza apre l'elenco Gruppi di sicurezza della console Amazon EC2.
5. Scegli il gruppo di sicurezza da modificare e passa alla scheda In entrata.
6. Aggiungi una regola autoreferenziale per permettere ai componenti di AWS Glue di comunicare tra loro. In particolare, aggiungi o verifica che sia presente una regola con Tipo `All TCP`, Protocollo `TCP`, Intervallo porte che include tutte le porte e Origine corrispondente al nome del gruppo di sicurezza indicato da ID gruppo.

   La regola in entrata è simile alla seguente:

| Tipo | Protocollo | Intervallo porte | Origine |
| --- | --- | --- | --- |
| Tutte le regole TCP | TCP | 0–65535 | database-security-group |

Esempio:




7. Aggiungi una regola anche per il traffico in uscita. Quindi, apri il traffico in uscita per tutte le porte, ad esempio:

| Tipo | Protocollo | Intervallo porte | Destinazione |
| --- | --- | --- | --- |
| All Traffic | ALL | ALL | 0.0.0.0/0 |

In alternativa, crea una regola autoreferenziale in cui Tipo `All TCP`, Protocollo sta per `TCP` e Intervallo porte include tutte le porte, la cui Destinazione ha lo stesso nome del gruppo di sicurezza dell'ID gruppo. Se usi un endpoint VPC Amazon S3, aggiungi anche una regola HTTPS per l'accesso di Amazon S3. *`s3-prefix-list-id`*È necessario nella regola del gruppo di sicurezza per consentire il traffico dal VPC all'endpoint VPC Amazon S3.

Esempio:

| Tipo | Protocollo | Intervallo porte | Destinazione |
|---|---|---|---|
| Tutte le regole TCP | TCP | 0–65535 | *security-group* |
| HTTPS | TCP | 443 | *s3-prefix-list-id* |

Configura AWS Glue

Dovrai creare una connessione AWS Glue Data Catalog che fornisca informazioni sulla connessione Amazon VPC.

Per configurare la connettività Amazon Redshift (Amazon VPC) a AWS Glue nella console:

1. Crea una connessione a Catalogo dati seguendo i passaggi indicati nella sezione the section called "Aggiungere una AWS Glue connessione". Dopo aver creato la connessione, mantieni il nome della connessione per *connectionName* il passaggio successivo.

   - Quando selezioni un Tipo di connessione, seleziona Amazon Redshift.
   - Quando selezioni un Cluster Redshift, seleziona il tuo cluster in base al nome.
   - Fornisci informazioni di connessione predefinite per un utente Amazon Redshift sul tuo cluster.
   - Le impostazioni di Amazon VPC verranno configurate automaticamente.

   > **Note**
   >
   > Quando crei una connessione Amazon Redshift tramite l'SDK AWS , dovrai fornire manualmente il valore `PhysicalConnectionRequirements` per il tuo Amazon VPC.

2. Nella configurazione del lavoro AWS Glue, fornisci *connectionName* una connessione di rete aggiuntiva.

Esempio: lettura da tabelle Amazon Redshift

È possibile leggere da cluster Amazon Redshift e ambienti Amazon Redshift serverless.

Prerequisiti: una tabella Amazon Redshift da cui desideri leggere. Segui i passaggi della sezione precedente, the section called "Configurazione di Redshift" dopodiché dovresti avere l'URI Amazon S3 per una directory temporanea *temp-s3-dir* e un ruolo IAM,*rs-role-name*, (nell'account*role-account-id*).

Using the Data Catalog

Prerequisiti aggiuntivi: un database Catalogo dati e una tabella dai quali desideri che la tabella Amazon Redshift legga. Per ulteriori informazioni su Catalogo dati, consultare la pagina Rilevamento e catalogazione dei dati. Dopo aver creato una voce per la tabella Amazon Redshift, identificherai la tua connessione con un *redshift-dc-database-name* segno e. *redshift-table-name*

Configurazione: nelle opzioni della funzione identificherai la tabella di Catalogo dati con i parametri `database` e `table_name`. Identificherai la directory temporanea Amazon S3 con `redshift_tmp_dir`. Dovrai inoltre fornire *rs-role-name* l'utilizzo della `aws_iam_role` chiave nel `additional_options` parametro.

```
 glueContext.create_dynamic_frame.from_catalog(
    database = "redshift-dc-database-name",
    table_name = "redshift-table-name",
    redshift_tmp_dir = args["temp-s3-dir"],
    additional_options = {"aws_iam_role": "arn:aws:iam::role-account-id:role/rs-
role-name"})
```

Connecting directly

Prerequisiti aggiuntivi: è necessario il nome della tabella Amazon Redshift (. *redshift-table-name* Avrai bisogno delle informazioni di connessione JDBC per il cluster Amazon Redshift in cui è archiviata quella tabella. Fornirai le informazioni di connessione con*host*, *portredshift-database-name*, *username* e. *password*

Quando lavori con i cluster Amazon Redshift, puoi recuperare le informazioni di connessione dalla console Amazon Redshift. Se utilizzi Amazon Redshift serverless, consulta la sezione Connecting to Amazon Redshift Serverless nella documentazione di Amazon Redshift.

Configurazione: nelle opzioni della funzione identificherai i parametri di connessione con `url`, `dbtable`, `user` e `password`. Identificherai la tua directory temporanea Amazon S3 con `redshift_tmp_dir`. Quando utilizzi `from_options`, puoi specificare il tuo ruolo IAM

utilizzando `aws_iam_role`. La sintassi è simile alla connessione tramite Catalogo dati, ma è necessario inserire i parametri nella mappa `connection_options`.

È una cattiva pratica codificare le password negli script AWS Glue. Valuta la possibilità di archiviare le password Gestione dei segreti AWS e recuperarle nello script con SDK for Python (Boto3).

```
my_conn_options = {
    "url": "jdbc:redshift://host:port/redshift-database-name",
    "dbtable": "redshift-table-name",
    "user": "username",
    "password": "password",
    "redshiftTmpDir": args["temp-s3-dir"],
    "aws_iam_role": "arn:aws:iam::account id:role/rs-role-name"
}

df = glueContext.create_dynamic_frame.from_options("redshift", my_conn_options)
```

Esempio: scrittura su tabelle Amazon Redshift

È possibile scrivere su cluster Amazon Redshift e ambienti Amazon Redshift serverless.

Prerequisiti: un cluster Amazon Redshift e segui i passaggi della the section called "Configurazione di Redshift" sezione precedente, dopodiché dovresti avere l'URI Amazon S3 per una directory temporanea e un ruolo *rs-role-name* IAM*temp-s3-dir*,, (nell'account). *role-account-id* Avrai anche bisogno di un `DynamicFrame` del quale desideri scrivere il contenuto nel database.

Using the Data Catalog

Prerequisiti aggiuntivi: un database Catalogo dati sul quale desideri che scrivano il cluster e la tabella Amazon Redshift. Per ulteriori informazioni su Catalogo dati, consulta la pagina Rilevamento e catalogazione dei dati. Identificherai la tua connessione con *redshift-dc-database-name* e la tabella di destinazione con. *redshift-table-name*

Configurazione: nelle opzioni della funzione identificherai il database di Catalogo dati con il parametro `database`, quindi fornirai la tabella con `table_name`. Identificherai la tua directory temporanea Amazon S3 con `redshift_tmp_dir`. Fornirai anche *rs-role-name* l'utilizzo della `aws_iam_role` chiave nel `additional_options` parametro.

```
 glueContext.write_dynamic_frame.from_catalog(
    frame = input dynamic frame,
    database = "redshift-dc-database-name",
    table_name = "redshift-table-name",
    redshift_tmp_dir = args["temp-s3-dir"],
    additional_options = {"aws_iam_role": "arn:aws:iam::account-id:role/rs-role-
name"})
```

Connecting through a AWS Glue connection

È possibile connettersi ad Amazon Redshift direttamente utilizzando il metodo `write_dynamic_frame.from_options`. Tuttavia, anziché inserire i dettagli di connessione direttamente nello script, puoi fare riferimento ai dettagli di connessione archiviati in una connessione a Catalogo dati con il metodo `from_jdbc_conf`. È possibile eseguire questa operazione senza effettuare il crawling o creare tabelle di Catalogo dati per il database. Per ulteriori informazioni sulle connessioni a Catalogo dati, consulta la pagina Connessione ai dati.

Prerequisiti aggiuntivi: una connessione a Catalogo dati per il database, una tabella Amazon Redshift da cui desideri leggere

Configurazione: identificherai la tua connessione al Data Catalog con*dc-connection-name*. Identificherai il database e la tabella Amazon Redshift con *redshift-table-name* e. *redshift-database-name* Fornirai le informazioni di connessione a Catalogo dati con `catalog_connection` e le informazioni relative ad Amazon Redshift con `dbtable` e `database`. La sintassi è simile alla connessione tramite Catalogo dati, ma è necessario inserire i parametri nella mappa `connection_options`.

```
my_conn_options = {
    "dbtable": "redshift-table-name",
    "database": "redshift-database-name",
    "aws_iam_role": "arn:aws:iam::role-account-id:role/rs-role-name"
}

glueContext.write_dynamic_frame.from_jdbc_conf(
    frame = input dynamic frame,
    catalog_connection = "dc-connection-name",
    connection_options = my_conn_options,
    redshift_tmp_dir = args["temp-s3-dir"])
```

Indicazioni di riferimento alle opzione di connessione ad Amazon Redshift

Le opzioni di connessione di base utilizzate per tutte le connessioni JDBC AWS Glue per configurare informazioni come `url` `user` e `password` sono coerenti per tutti i tipi JDBC. Per ulteriori informazioni sui parametri JDBC standard, consulta la pagina the section called "Parametri di connessione JDBC".

Il tipo di connessione Amazon Redshift richiede alcune opzioni di connessione aggiuntive:

- `"redshiftTmpDir"`: (obbligatorio) il percorso Amazon S3 in cui i dati temporanei possono essere caricati durante la copia dal database.
- `"aws_iam_role"`: (facoltativo) l'ARN di un ruolo IAM. Il job AWS Glue passerà questo ruolo al cluster Amazon Redshift per concedere al cluster le autorizzazioni necessarie per completare le istruzioni del job.

Opzioni di connessione aggiuntive disponibili in AWS Glue 4.0+

Puoi anche passare le opzioni per il nuovo connettore Amazon Redshift tramite le opzioni di connessione AWS Glue. Per un elenco completo delle opzioni di connettori supportate, consulta la sezione Parametri SQL Spark in Integrazione di Amazon Redshift per Apache Spark.

Per comodità, ribadiamo di seguito alcune nuove opzioni:

| Name | Obbligatorio | Predefinita | Description |
|---|---|---|---|
| autopushdown | No | TRUE | Applica il pushdown di predicati e query acquisendo e analizzando i piani logici di Spark per le operazioni SQL. Le operazioni vengono tradotte in una query SQL e quindi eseguite in Amazon Redshift |

| Name | Obbligatorio | Predefinita | Description |
| --- | --- | --- | --- |
| | | | per migliorare le prestazioni. |
| autopushdown.s3_result_cache | No | FALSE | Memorizza nella cache la query SQL per scaricare i dati sulla mappatura dei percorsi di Amazon S3 in memoria, in modo che la stessa query non debba essere eseguita nuovamente nella stessa sessione di Spark. Supportato solo quando `autopushdown` è abilitato. |
| unload_s3_format | No | PARQUET | PARQUET: scarica i risultati della query in formato Parquet.<br><br>TESTO: scarica i risultati della query in formato testo delimitato da barra verticale. |
| sse_kms_key | No | N/D | La chiave AWS SSE-KMS da utilizzare per la crittografia durante `UNLOAD` le operazioni anziché la crittografia predefinita per. AWS |

| Name | Obbligatorio | Predefinita | Description |
| --- | --- | --- | --- |
| extracopyoptions | No | N/D | Un elenco di opzioni ulteriori da aggiunger e al comando `COPY` di Amazon Redshift durante il caricamen to dei dati, come `TRUNCATECOLUMNS` o `MAXERROR n` (per altre opzioni, consulta COPY: parametri facoltativi).<br><br>È importante notare che, poiché queste opzioni vengono aggiunte alla fine del comando `COPY`, è possibile utilizzar e solo le opzioni rilevanti alla fine del comando. Questo dovrebbe coprire la maggior parte dei casi d'uso possibili. |
| cvsnullstring (sperimentale) | No | NULL | Il valore di stringa da scrivere per i valori null quando si utilizza il `tempformat` CSV. Dovrebbe trattarsi di un valore che non è presente nei dati effettivi. |

Questi nuovi parametri possono essere utilizzati nei seguenti modi.

Nuove opzioni per il miglioramento delle prestazioni

Il nuovo connettore introduce alcune nuove opzioni di miglioramento delle prestazioni:

- `autopushdown`: abilitato per impostazione predefinita.
- `autopushdown.s3_result_cache`: disabilitato per impostazione predefinita.
- `unload_s3_format`: PARQUET per impostazione predefinita.

Per informazioni sull'utilizzo di queste opzioni, consultare Integrazione di Amazon Redshift per Apache Spark. Si consiglia di non attivare `autopushdown.s3_result_cache` quando si eseguono operazioni di lettura e scrittura miste perché i risultati memorizzati nella cache potrebbero contenere informazioni obsolete. L'opzione `unload_s3_format` è impostata su PARQUET per impostazione predefinita per il comando UNLOAD per migliorare le prestazioni e ridurre i costi di archiviazione. Per utilizzare il comportamento predefinito del comando UNLOAD, reimposta l'opzione su TEXT.

Nuova opzione di crittografia per la lettura

Per impostazione predefinita, i dati nella cartella temporanea utilizzata da AWS Glue durante la lettura dei dati dalla tabella Amazon Redshift vengono crittografati tramite la crittografia SSE-S3. Per utilizzare le chiavi gestite dal cliente di AWS Key Management Service (AWS KMS) per crittografare i dati, puoi impostare da `("sse_kms_key" # kmsKey)` dove KSMKey proviene l'ID della chiave AWS KMS, anziché l'opzione di impostazione precedente nella versione 3.0. `("extraunloadoptions" # s"ENCRYPTED KMS_KEY_ID '$kmsKey'")` AWS Glue

```
datasource0 = glueContext.create_dynamic_frame.from_catalog(
  database = "database-name",
  table_name = "table-name",
  redshift_tmp_dir = args["TempDir"],
  additional_options = {"sse_kms_key":"<KMS_KEY_ID>"},
  transformation_ctx = "datasource0"
)
```

Supporto dell'URL JDBC basato su IAM

Il nuovo connettore supporta un URL JDBC basato su IAM, quindi non è necessario inserire un segreto or. user/password Con un URL JDBC basato su IAM, il connettore utilizza il ruolo di runtime del processo per accedere all'origine dati Amazon Redshift.

Fase 1: collegamento della seguente politica minima obbligatoria al ruolo di runtime del processo AWS Glue.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "redshift:GetClusterCredentials",
            "Resource": [
                "arn:aws:redshift:us-east-1:111122223333:dbgroup:<cluster name>/
*",
                "arn:aws:redshift:*:111122223333:dbuser:*/*",
                "arn:aws:redshift:us-east-1:111122223333:dbname:<cluster name>/
<database name>"
            ]
        },
        {
            "Sid": "VisualEditor1",
            "Effect": "Allow",
            "Action": "redshift:DescribeClusters",
            "Resource": "*"
        }
    ]
}
```

Fase 2: uso dell'URL JDBC basato su IAM come segue. Specifica una nuova opzione `DbUser` con il nome utente Amazon Redshift con cui ti stai connettendo.

```
conn_options = {
     // IAM-based JDBC URL
    "url": "jdbc:redshift:iam://<cluster name>:<region>/<database name>",
    "dbtable": dbtable,
    "redshiftTmpDir": redshiftTmpDir,
    "aws_iam_role": aws_iam_role,
    "DbUser": "<Redshift User name>" // required for IAM-based JDBC URL
    }
```

```
redshift_write = glueContext.write_dynamic_frame.from_options(
    frame=dyf,
    connection_type="redshift",
    connection_options=conn_options
)

redshift_read = glueContext.create_dynamic_frame.from_options(
    connection_type="redshift",
    connection_options=conn_options
)
```

> **Note**
>
> Un `DynamicFrame` al momento supporta un URL JDBC basato su IAM solo con un `DbUser` nel flusso di lavoro `GlueContext.create_dynamic_frame.from_options`.

Migrazione da AWS Glue versione 3.0 alla versione 4.0

In AWS Glue 4.0, i job ETL hanno accesso a un nuovo connettore Amazon Redshift Spark e a un nuovo driver JDBC con diverse opzioni e configurazioni. Il nuovo connettore e driver Amazon Redshift sono stati progettati per le prestazioni e garantiscono la coerenza transazionale dei dati. Questi prodotti sono illustrati nella documentazione di Amazon Redshift. Per ulteriori informazioni, consulta:

- Integrazione di Amazon Redshift per Apache Spark
- Driver JDBC Amazon Redshift, versione 2.1

Restrizione dei nomi e degli identificatori di tabelle/colonne

Il nuovo connettore e il driver Amazon Redshift Spark hanno un requisito più limitato per il nome della tabella Redshift. Per ulteriori informazioni, consulta Nomi e identificatori per definire il nome della tabella Amazon Redshift. Il flusso di lavoro relativo ai segnalibri del processo potrebbe non funzionare con un nome di tabella che non corrisponde alle regole e con determinati caratteri, ad esempio uno spazio.

Se hai tabelle legacy con nomi non conformi alle regole dei nomi e degli identificatori e riscontri problemi con i segnalibri (processi che rielaborano i vecchi dati delle tabelle Amazon Redshift), ti consigliamo di rinominare le tabelle. Per ulteriori informazioni, consultare Esempi di ALTER TABLE.

Modifica del formato temporale predefinito in Dataframe

Il connettore Spark di AWS Glue versione 3.0 imposta automaticamente `tempformat` su CSV durante la scrittura su Amazon Redshift. Per continuità, in AWS Glue versione 3.0, `DynamicFrame` è ancora impostato su `tempformat` per l'uso di `CSV`. Se in precedenza hai utilizzato Spark Dataframe APIs direttamente con il connettore Amazon Redshift Spark, puoi impostarlo in modo esplicito su CSV nelle `tempformat` opzioni/. `DataframeReader` `Writer` Altrimenti, `tempformat` è impostato su `AVRO` nel nuovo connettore Spark.

Modifica di comportamento: associazione del tipo di dati Amazon Redshift REAL al tipo di dati Spark FLOAT anziché DOUBLE

In AWS Glue versione 3.0, Amazon Redshift `REAL` viene convertito in un tipo `DOUBLE` Spark. Il nuovo connettore Amazon Redshift Spark ha aggiornato il comportamento in modo che il tipo `REAL` Amazon Redshift venga viene convertito e di nuovo dal tipo `FLOAT` Spark. Se si dispone di un caso d'uso precedente in cui si desidera ancora che il tipo `REAL` Amazon Redshift sia mappato a un tipo `DOUBLE` Spark, è possibile utilizzare la seguente soluzione alternativa:

- Per un `DynamicFrame`, mappa il tipo `Float` a un tipo `Double` con `DynamicFrame.ApplyMapping`. Per un `Dataframe`, è necessario usare `cast`.

Esempio di codice:

```
dyf_cast = dyf.apply_mapping([('a', 'long', 'a', 'long'), ('b', 'float', 'b',
 'double')])
```

Gestione del tipo di dati VARBYTE

Quando si lavora con tipi di dati AWS Glue 3.0 e Amazon Redshift, AWS Glue 3.0 converte Amazon `VARBYTE` Redshift in tipo Spark. `STRING` Tuttavia, il connettore Spark Amazon Redshift più recente non supporta il tipo di dati `VARBYTE`. Per aggirare questa limitazione, è possibile creare una vista Redshift che trasformi le colonne `VARBYTE` in un tipo di dati supportato. In seguito, è possibile utilizzare il nuovo connettore per caricare i dati da questa vista anziché dalla tabella originale, in modo da garantire la compatibilità pur mantenendo l'accesso ai dati `VARBYTE`.

Esempio di query Redshift:

```
CREATE VIEW view_name AS SELECT FROM_VARBYTE(varbyte_column, 'hex') FROM table_name
```

## Connessioni Kafka

È possibile usare una connessione Kafka per leggere e scrivere su flussi di dati Kafka utilizzando le informazioni archiviate in una tabella del Catalogo dati o fornendo informazioni per accedere direttamente al flusso di dati. La connessione supporta un cluster Kafka o un cluster Streaming gestito da Amazon per Apache Kafka. Puoi leggere le informazioni di Kafka in uno Spark DataFrame, quindi convertirle in un Glue. AWS DynamicFrame Puoi scrivere su Kafka DynamicFrames in formato JSON. Se si accede direttamente al flusso di dati, utilizzare queste opzioni per fornire le informazioni su come accedere al flusso di dati.

Se si utilizza `getCatalogSource` o `create_data_frame_from_catalog` per consumare i record da una sorgente di streaming Kafka oppure `getCatalogSink` o `write_dynamic_frame_from_catalog` per scrivere i record su Kafka, il processo avrà le informazioni sul database del Catalogo dati e sul nome della tabella, e potrà usarle per ottenere alcuni parametri di base per la lettura dall'origine di streaming Kafka. Se si utilizza `getSource`, `getCatalogSink`, `getSourceWithFormat`, `getSinkWithFormat`, `createDataFrameFromOptions`, `create_data_frame_from_options` o `write_dynamic_frame_from_catalog`, sarà necessario specificare questi parametri di base utilizzando le opzioni di connessione descritte qui.

È possibile specificare le opzioni di connessione per Kafka utilizzando gli argomenti per i metodi specificati nella classe `GlueContext` descritti di seguito.

- Scala
  - `connectionOptions`: utilizza con `getSource`, `createDataFrameFromOptions` e `getSink`
  - `additionalOptions`: utilizza con `getCatalogSource`, `getCatalogSink`
  - `options`: utilizza con `getSourceWithFormat`, `getSinkWithFormat`
- Python
  - `connection_options`: utilizza con `create_data_frame_from_options`, `write_dynamic_frame_from_options`
  - `additional_options`: utilizza con `create_data_frame_from_catalog`, `write_dynamic_frame_from_catalog`
  - `options`: utilizza con `getSource`, `getSink`

Per osservazioni e restrizioni sui processi ETL dei flussi di dati, consultare la pagina the section called “Streaming di note e restrizioni ETL”.

Argomenti


- Configurazione di Kafka
- Esempio: lettura di flussi da Kafka
- Esempio: scrittura in flussi Kafka
- Indicazioni di riferimento alle opzioni di connessione a Kafka


### Configurazione di Kafka

Non ci sono AWS prerequisiti per la connessione agli stream di Kafka disponibili su Internet.

Puoi creare una connessione AWS Glue Kafka per gestire le tue credenziali di connessione. Per ulteriori informazioni, consulta the section called "Creazione di una connessione per un flusso di dati Kafka". Nella configurazione del processo AWS Glue, fornisci *connectionName* una connessione di rete aggiuntiva, quindi, nella chiamata *connectionName* al metodo, fornisci il `connectionName` parametro.

In alcuni casi, è necessario configurare ulteriori prerequisiti:

- Se utilizzi Streaming gestito da Amazon per Apache Kafka con l'autenticazione IAM, avrai bisogno di una configurazione appropriata di IAM.
- Se utilizzi Streaming gestito da Amazon per Apache Kafka con un Amazon VPC, avrai bisogno di una configurazione appropriata di Amazon VPC. Dovrai creare una connessione AWS Glue che fornisca informazioni sulla connessione Amazon VPC. È necessaria la configurazione del lavoro per includere la connessione AWS Glue come connessione di rete aggiuntiva.

Per ulteriori informazioni sui prerequisiti dei processi ETL dei flussi di dati, consulta la pagina the section called "Aggiunta di processi di streaming ETL".

### Esempio: lettura di flussi da Kafka

Usato in combinazione con the section called "forEachBatch".

Esempio per l'origine di streaming Kafka:

```
kafka_options =
    { "connectionName": "ConfluentKafka",
      "topicName": "kafka-auth-topic",
      "startingOffsets": "earliest",
      "inferSchema": "true",
```

```
      "classification": "json"
    }
data_frame_datasource0 =
 glueContext.create_data_frame.from_options(connection_type="kafka",
 connection_options=kafka_options)
```

Esempio: scrittura in flussi Kafka

Esempi di scrittura in Kafka:

Esempio con il metodo `getSink`:

```
data_frame_datasource0 =
glueContext.getSink(
 connectionType="kafka",
 connectionOptions={
  JsonOptions("""{
   "connectionName": "ConfluentKafka",
   "classification": "json",
   "topic": "kafka-auth-topic",
   "typeOfData": "kafka"}
  """)},
 transformationContext="dataframe_ApacheKafka_node1711729173428")
 .getDataFrame()
```

Esempio con il metodo `write_dynamic_frame.from_options`:

```
kafka_options =
    { "connectionName": "ConfluentKafka",
      "topicName": "kafka-auth-topic",
      "classification": "json"
    }
data_frame_datasource0 =
 glueContext.write_dynamic_frame.from_options(connection_type="kafka",
 connection_options=kafka_options)
```

Indicazioni di riferimento alle opzioni di connessione a Kafka

Per la lettura, utilizzare le seguenti opzioni di connessione con `"connectionType": "kafka"`:

- `"bootstrap.servers"`(Obbligatorio) Un elenco di server di bootstrap URLs, ad esempio, come`b-1.vpc-test-2.o4q88o.c6.kafka.us-east-1.amazonaws.com:9094`. Questa

opzione deve essere specificata nella chiamata API o definita nei metadati della tabella in catalogo dati.

- `"security.protocol"`: (obbligatorio) Il protocollo utilizzato per comunicare con i broker. I valori possibili sono `"SSL"` o `"PLAINTEXT"`.
- `"topicName"`: (obbligatorio) un elenco separato da virgole di argomenti a cui iscriversi. Devi specificare solo uno tra `"topicName"`, `"assign"` o `"subscribePattern"`.
- `"assign"`: (obbligatorio) una stringa JSON che specifica il `TopicPartitions` specifico da utilizzare. Devi specificare solo uno tra `"topicName"`, `"assign"` o `"subscribePattern"`.

  Esempio: '{"topicA":[0,1],"topicB":[2,4]}'
- `"subscribePattern"`: (obbligatorio) una stringa regex Java che identifichi l'elenco degli argomenti a cui effettuare la sottoscrizione. Devi specificare solo uno tra `"topicName"`, `"assign"` o `"subscribePattern"`.

  Esempio: 'topic.*'
- `"classification"` (obbligatorio): il formato di file utilizzato dai dati nel record. Obbligatorio, a meno che non sia fornito tramite Catalogo dati.
- `"delimiter"` (opzionale): il separatore di valori utilizzato quando `classification` è CSV. Il valore predefinito è “`,`”.
- `"startingOffsets"`: (opzionale) la posizione di partenza nell'argomento Kafka da cui leggere i dati. I valori possibili sono `"earliest"` o `"latest"`. Il valore predefinito è `"latest"`.
- `"startingTimestamp"`: (Facoltativo, supportato solo per AWS Glue versione 4.0 o successiva) Il timestamp del record nell'argomento Kafka da cui leggere i dati. Il valore possibile è una stringa timestamp in formato UTC nel modello `yyyy-mm-ddTHH:MM:SSZ`, dove `Z` rappresenta un offset del fuso orario UTC con un segno +/- (ad esempio: "2023-04-04T08:00:00-04:00").

  Nota: nell'elenco delle opzioni di connessione dello script di streaming AWS Glue può essere presente solo uno tra 'startingOffsets' o 'startingTimestamp', l'inclusione di entrambe queste proprietà comporterà un errore del lavoro.
- `"endingOffsets"`: (opzionale) il punto di fine di una query batch. I valori possibili sono `"latest"` o una stringa JSON che specifica un offset finale per ogni `TopicPartition`.

  Per la stringa JSON, il formato è `{"topicA":{"0":23,"1":-1},"topicB":{"0":-1}}`. Il valore `-1` come offset rappresenta `"latest"`.
- `"pollTimeoutMs"`: (opzionale) il timeout in millisecondi per il polling dei dati da Kafka negli executor del processo Spark. Il valore predefinito è `600000`.

- `"numRetries"`: (opzionale) i numero di tentativi prima di non riuscire a recuperare gli offset Kafka. Il valore predefinito è `3`.
- `"retryIntervalMs"`: (opzionale) il tempo di attesa in millisecondi prima di riprovare a recuperare gli offset Kafka. Il valore predefinito è `10`.
- `"maxOffsetsPerTrigger"`: (opzionale) il limite di velocità sul numero massimo di offset elaborati per intervallo di trigger. Il numero totale di offset specificato viene suddiviso proporzionalmente tra `topicPartitions` di diversi volumi. Il valore di default è null, il che significa che il consumer legge tutti gli offset fino all'ultimo offset noto.
- `"minPartitions"`: (opzionale) il numero minimo desiderato di partizioni da leggere da Kafka. Il valore di default è null, il che significa che il numero di partizioni Spark è uguale al numero di partizioni Kafka.
- `"includeHeaders"`: (opzionale) indica se includere le intestazioni Kafka. Quando l'opzione è impostata su "true", l'output dei dati conterrà una colonna aggiuntiva denominata "glue_streaming_kafka_headers" con tipo `Array[Struct(key: String, value: String)]`. Il valore di default è "false". Questa opzione è disponibile in AWS Glue versione 3.0 o successive.
- `"schema"`: (obbligatorio quando inferSchema è impostato su false) lo schema da utilizzare per elaborare il payload. Se la classificazione è `avro`, lo schema fornito dovrà essere nel formato dello schema Avro. Se la classificazione è `avro`, lo schema fornito dovrà essere nel formato dello schema DDL.

  Di seguito sono riportati alcuni esempi di schema.

  Example in DDL schema format

  ```
  'column1' INT, 'column2' STRING , 'column3' FLOAT
  ```

  Example in Avro schema format

  ```
  {
  "type":"array",
  "items":
  {
  "type":"record",
  "name":"test",
  "fields":
  [
    {
      "name":"_id",
  ```

```
    "type":"string"
  },
  {
    "name":"index",
    "type":
    [
      "int",
      "string",
      "float"
    ]
  }
]
}
}
```

- `"inferSchema"`: (facoltativo) il valore di default è "false". Se impostato su "true", lo schema verrà rilevato in fase di runtime dal payload all'interno di `foreachbatch`.
- `"avroSchema"`: (obsoleto) parametro utilizzato per specificare uno schema di dati Avro quando viene utilizzato il formato Avro. Questo parametro è obsoleto. Utilizzo del parametro `schema`.
- `"addRecordTimestamp"`: (opzionale) Quando questa opzione è impostata su "true", l'output dei dati conterrà una colonna aggiuntiva denominata "__src_timestamp" che indica l'ora in cui il record corrispondente è stato ricevuto dall'argomento. Il valore predefinito è "false". Questa opzione è supportata in AWS Glue versione 4.0 o successive.
- `"emitConsumerLagMetrics"`: (Facoltativo) Quando l'opzione è impostata su «true», per ogni batch, emette le metriche relative alla durata compresa tra il record più vecchio ricevuto dall'argomento e l'ora in cui arriva. AWS Glue CloudWatch Il nome della metrica è «glue.driver.streaming. maxConsumerLagInMs». Il valore predefinito è "false". Questa opzione è supportata in AWS Glue versione 4.0 o successive.

Per la scrittura, utilizzare le seguenti opzioni di connessione con `"connectionType": "kafka"`:

- `"connectionName"`(Obbligatorio) Nome della connessione AWS Glue utilizzata per connettersi al cluster Kafka (simile al sorgente Kafka).
- `"topic"` (obbligatorio) Se esiste una colonna di argomento, il relativo valore viene utilizzato come argomento quando si scrive la riga specifica in Kafka, a meno che non sia impostata l'opzione di configurazione dell'argomento. In altre parole, l'opzione di configurazione `topic` sovrascrive la colonna dell'argomento.

- `"partition"` (opzionale) Se viene specificato un numero di partizione valido, `partition` verrà utilizzato per l'invio del record.

  Se non viene specificata alcuna partizione ma è presente `key`, verrà scelta una partizione utilizzando un hash della chiave.

  Se non sono presenti né `key` né `partition`, verrà scelta una partizione in base al partizionamento permanente delle modifiche quando nella partizione vengono prodotti almeno batch.size byte.
- `"key"` (opzionale) Utilizzato per il partizionamento se `partition` è null.
- `"classification"` (opzionale) Il formato di file utilizzato dai dati nel record. Supportiamo solo JSON, CSV e Avro.

  Con il formato Avro, possiamo fornire un AvroSchema personalizzato con cui serializzare, ma bisogna considerare che tale schema deve essere fornito anche nell'origine per la deserializzazione. Altrimenti, per impostazione predefinita utilizza Apache per la serializzazione. AvroSchema

Inoltre, è possibile eseguire il fine-tuning del sink Kafka in base alle esigenze aggiornando i parametri di configurazione del produttore di Kafka. Da notare che non esiste un elenco delle opzioni di connessione consentite, tutte le coppie chiave-valore vengono mantenute nel sink così come sono.

Tuttavia, esiste un piccolo elenco di opzioni di rifiuto che non avranno effetto. Per ulteriori informazioni, consultare le configurazioni specifiche di Kafka.

Connessioni Azure Cosmos DB

Puoi usare AWS Glue for Spark per leggere e scrivere su contenitori esistenti in Azure Cosmos DB usando l'API NoSQL in Glue 4.0 e versioni successive. AWS È possibile definire cosa leggere da Azure Cosmos DB con una query SQL. Ti connetti ad Azure Cosmos DB usando una chiave di Azure Cosmos DB archiviata tramite Gestione dei segreti AWS una connessione Glue. AWS

Per altre informazioni su Azure Cosmos DB per NoSQL, consulta la documentazione di Azure.

Configurazione delle connessioni Azure Cosmos DB

Per connetterti ad Azure Cosmos DB da AWS Glue, dovrai creare e archiviare la tua chiave Azure Cosmos DB in un luogo Gestione dei segreti AWS segreto, quindi associare quel segreto a una connessione Azure Cosmos DB Glue. AWS

Prerequisiti:

- In Azure, dovrai identificare o generare una chiave di Azure Cosmos DB da usare con AWS Glue,. `cosmosKey` Per altre informazioni, consulta Accesso sicuro ai dati in Azure Cosmos DB nella documentazione di Azure.

Per configurare una connessione ad Azure Cosmos DB:

1. Nel Gestione dei segreti AWS, crea un segreto usando la tua chiave Azure Cosmos DB. Per creare un segreto in Secrets Manager, segui il tutorial disponibile in Crea un Gestione dei segreti AWS segreto nella Gestione dei segreti AWS documentazione. Dopo aver creato il segreto, mantieni il nome del segreto *`secretName`* per il passaggio successivo.
   - Quando selezionate le coppie chiave/valore, create una coppia per la chiave `spark.cosmos.accountKey` con il valore. *`cosmosKey`*
2. Nella console AWS Glue, crea una connessione seguendo i passaggi riportati di seguitothe section called "Aggiungere una AWS Glue connessione". Dopo aver creato la connessione, mantieni il nome della connessione*`connectionName`*, per utilizzi futuri in AWS Glue.
   - In Tipo di connessione, seleziona Azure Cosmos DB.
   - Quando selezioni un AWS segreto, fornisci*`secretName`*.

Dopo aver creato una connessione AWS Glue Azure Cosmos DB, dovrai eseguire i seguenti passaggi prima di eseguire il lavoro AWS Glue:

- Concedi al ruolo IAM associato al tuo lavoro AWS Glue il permesso di lettura*`secretName`*.
- Nella configurazione del lavoro AWS Glue, fornisci *`connectionName`* una connessione di rete aggiuntiva.

Lettura da container Azure Cosmos DB per NoSQL

Prerequisiti:

- Un container Azure Cosmos DB per NoSQL da cui desideri leggere. Avrai bisogno delle informazioni di identificazione per il container.

Un container Azure Cosmos per NoSQL è identificato dal database e dal container. È necessario fornire i nomi del database e del contenitore quando ci si connette all'API di Azure Cosmos for NoSQL. *cosmosDBName* *cosmosContainerName*

- Una connessione AWS Glue Azure Cosmos DB configurata per fornire informazioni di autenticazione e posizione della rete. Per l'acquisizione, completa i passaggi della procedura precedente, Per configurare una connessione ad Azure Cosmos DB. Avrai bisogno del nome della connessione AWS Glue,*connectionName*.

Esempio:

```
azurecosmos_read = glueContext.create_dynamic_frame.from_options(
    connection_type="azurecosmos",
    connection_options={
    "connectionName": connectionName,
    "spark.cosmos.database": cosmosDBName,
    "spark.cosmos.container": cosmosContainerName,
    }
)
```

Puoi anche fornire una query SQL SELECT per filtrare i risultati restituiti al tuo DynamicFrame. Sarà necessario configurare `query`.

Esempio:

```
azurecosmos_read_query = glueContext.create_dynamic_frame.from_options(
    connection_type="azurecosmos",
    connection_options={
        "connectionName": "connectionName",
        "spark.cosmos.database": cosmosDBName,
        "spark.cosmos.container": cosmosContainerName,
        "spark.cosmos.read.customQuery": "query"
    }
)
```

Scrittura su container Azure Cosmos DB per NoSQL

Questo esempio scrive informazioni da un database esistente DynamicFrame *dynamicFrame* ad Azure Cosmos DB. Se il contenitore contiene già informazioni, AWS Glue aggiungerà i dati dal tuo

DynamicFrame. Se le informazioni nel contanier hanno uno schema diverso da quello scritto, si verificheranno degli errori.

Prerequisiti:

- Una tabella di Azure Cosmos DB su cui scrivere. Avrai bisogno delle informazioni di identificazione per il container. È necessario creare il container prima di chiamare il metodo di connessione.

  Un container Azure Cosmos per NoSQL è identificato dal database e dal container. È necessario fornire i nomi del database e del contenitore quando ci si connette all'API di Azure Cosmos for NoSQL. *cosmosDBName* *cosmosContainerName*
- Una connessione AWS Glue Azure Cosmos DB configurata per fornire informazioni di autenticazione e posizione della rete. Per l'acquisizione, completa i passaggi della procedura precedente, Per configurare una connessione ad Azure Cosmos DB. Avrai bisogno del nome della connessione AWS Glue,*connectionName*.

Esempio:

```
azurecosmos_write = glueContext.write_dynamic_frame.from_options(
    frame=dynamicFrame,
    connection_type="azurecosmos",
    connection_options={
    "connectionName": connectionName,
    "spark.cosmos.database": cosmosDBName,
    "spark.cosmos.container": cosmosContainerName
)
```

Indicazioni di riferimento alle opzioni di connessione ad Azure Cosmos DB

- `connectionName`: obbligatorio. Utilizzato per la lettura/scrittura. Il nome di una connessione AWS Glue Azure Cosmos DB configurata per fornire informazioni di autenticazione e posizione di rete al metodo di connessione.
- `spark.cosmos.database`: obbligatorio. Utilizzato per la lettura/scrittura. Valori validi: nomi di database. Nome del database di Azure Cosmos DB per NoSQL.
- `spark.cosmos.container`: obbligatorio. Utilizzato per la lettura/scrittura. Valori validi: nomi dei contanier. Nome del contanier di Azure Cosmos DB per NoSQL.
- `spark.cosmos.read.customQuery`: utilizzato per la lettura. Valori validi: query SELECT SQL. Query personalizzata per selezionare i documenti da leggere.

Connessioni Azure SQL

Puoi usare AWS Glue for Spark per leggere e scrivere su tabelle su istanze gestite di Azure SQL in AWS Glue 4.0 e versioni successive. È possibile definire cosa leggere da Azure SQL con una query SQL. Ti connetti ad Azure SQL usando le credenziali utente e password archiviate Gestione dei segreti AWS tramite una connessione AWS Glue.

Per altre informazioni su Azure SQL, consulta la documentazione di Azure SQL.

Configurazione delle connessioni Azure SQL

Per connetterti ad Azure SQL da AWS Glue, dovrai creare e archiviare le tue credenziali SQL di Azure in un Gestione dei segreti AWS segreto, quindi associare quel segreto a una connessione Azure SQL AWS Glue.

Per configurare una connessione ad Azure SQL:

1. Nel Gestione dei segreti AWS, crea un segreto usando le tue credenziali SQL di Azure. Per creare un segreto in Secrets Manager, segui il tutorial disponibile in Crea un Gestione dei segreti AWS segreto nella Gestione dei segreti AWS documentazione. Dopo aver creato il segreto, mantieni il nome del segreto *secretName* per il passaggio successivo.
   - Quando selezionate le coppie chiave/valore, create una coppia per la chiave `user` con il valore. *azuresqlUsername*
   - Quando selezionate le coppie chiave/valore, create una coppia per la chiave `password` con il valore. *azuresqlPassword*
2. Nella console AWS Glue, crea una connessione seguendo i passaggi riportati di seguitothe section called "Aggiungere una AWS Glue connessione". Dopo aver creato la connessione, mantieni il nome della connessione*connectionName*, per utilizzi futuri in AWS Glue.
   - In Tipo di connessione, seleziona Azure SQL.
   - Quando fornisci l'URL SQL di Azure, fornisci un URL di endpoint JDBC.

     L'elenco deve essere nel seguente formato:
     `jdbc:sqlserver://`*databaseServerName*`:`*databasePort*`;databaseName=`*azuresqlDBnam*

     AWS Glue richiede le seguenti proprietà URL:
     - `databaseName`: un database predefinito in Azure SQL a cui connettersi.

Per altre informazioni su JDBC URLs for Azure SQL Managed Instances, consulta la documentazione di Microsoft.

- Quando selezioni un AWS segreto, fornisci. *secretName*

Dopo aver creato una connessione SQL di AWS Glue Azure, dovrai eseguire i seguenti passaggi prima di eseguire il lavoro AWS Glue:

- Concedi al ruolo IAM associato al tuo lavoro AWS Glue il permesso di lettura*secretName*.
- Nella configurazione del lavoro AWS Glue, fornisci *connectionName* una connessione di rete aggiuntiva.

Lettura da tabelle SQL di Azure

Prerequisiti:

- Una tabella Azure SQL da cui si desidera leggere. Avrai bisogno delle informazioni di identificazione per la tabella *databaseName* e*tableIdentifier*.

  Una tabella SQL di Azure è identificata dal database, dallo schema e dal nome. È necessario fornire il nome del database e della tabella durante la connessione ad Azure SQL. È inoltre necessario fornire lo schema se diverso da quello predefinito, "pubblico". Il database viene fornito tramite una proprietà URL in*connectionName*, lo schema e il nome della tabella tramitedbtable.
- Una connessione AWS Glue Azure SQL configurata per fornire informazioni di autenticazione. Completa i passaggi della procedura precedente, Per configurare una connessione ad Azure SQL per configurare le informazioni di autenticazione. Avrai bisogno del nome della connessione AWS Glue,*connectionName*.

Esempio:

```
azuresql_read_table = glueContext.create_dynamic_frame.from_options(
    connection_type="azuresql",
    connection_options={
        "connectionName": "connectionName",
        "dbtable": "tableIdentifier"
    }
)
```

Puoi anche fornire una query SQL SELECT per filtrare i risultati restituiti al tuo DynamicFrame. Sarà necessario configurare `query`.

Esempio:

```
azuresql_read_query = glueContext.create_dynamic_frame.from_options(
    connection_type="azuresql",
    connection_options={
        "connectionName": "connectionName",
        "query": "query"
    }
)
```

Scrittura su tabelle SQL di Azure

Questo esempio scrive informazioni da un SQL esistente DynamicFrame *dynamicFrame* ad Azure. Se la tabella contiene già informazioni, AWS Glue aggiungerà i dati dal tuo DynamicFrame.

Prerequisiti:

- Una tabella di Azure SQL su cui scrivere. Avrai bisogno delle informazioni di identificazione per la tabella, *databaseName* e*tableIdentifier*.

  Una tabella SQL di Azure è identificata dal database, dallo schema e dal nome. È necessario fornire il nome del database e della tabella durante la connessione ad Azure SQL. È inoltre necessario fornire lo schema se diverso da quello predefinito, "pubblico". Il database viene fornito tramite una proprietà URL in*connectionName*, lo schema e il nome della tabella tramitedbtable.
- Informazioni di autenticazione SQL di Azure. Completa i passaggi della procedura precedente, Per configurare una connessione ad Azure SQL per configurare le informazioni di autenticazione. Avrai bisogno del nome della connessione AWS Glue,*connectionName*.

Esempio:

```
azuresql_write = glueContext.write_dynamic_frame.from_options(
    connection_type="azuresql",
    connection_options={
        "connectionName": "connectionName",
        "dbtable": "tableIdentifier"
    }
)
```

Indicazioni di riferimento alle opzioni di connessione ad Azure SQL

- `connectionName`: obbligatorio. Utilizzato per la lettura/scrittura. Il nome di una connessione SQL di AWS Glue Azure configurata per fornire informazioni di autenticazione al metodo di connessione.
- `databaseName`: utilizzato per la lettura/scrittura. Valori validi: nomi di database di Azure SQL. Il nome del database in Azure SQL a cui connettersi.
- `dbtable` — Richiesto per la scrittura, richiesto per la lettura a meno che non `query` sia fornito. Usato per le combinazioni di Read/Write. Valid Values: Names of Azure SQL tables, or period separated schema/table nomi. Utilizzato per specificare la tabella e lo schema che identificano la tabella a cui connettersi. Lo schema predefinito è "pubblico". Se la tabella rientra in uno schema non predefinito, fornisci queste informazioni nel modulo *`schemaName`*`.`*`tableName`*.
- `query`: utilizzato per la lettura. Una query Transact-SQL SELECT che definisce cosa recuperare durante la lettura da Azure SQL. Per ulteriori informazioni, consulta la documentazione di Microsoft.

BigQuery connessioni

Puoi utilizzare AWS Glue for Spark per leggere e scrivere su tabelle in Google BigQuery in AWS Glue 4.0 e versioni successive. Puoi leggere da BigQuery una query SQL di Google. Ti connetti BigQuery utilizzando le credenziali archiviate Gestione dei segreti AWS tramite una connessione AWS Glue.

Per ulteriori informazioni su Google BigQuery, consulta il BigQuery sito Web di Google Cloud.

Configurazione delle connessioni BigQuery

Per connetterti a Google BigQuery da AWS Glue, dovrai creare e archiviare le tue credenziali di Google Cloud Platform in modo Gestione dei segreti AWS segreto, quindi associare quel segreto a una connessione Google BigQuery AWS Glue.

Per configurare una connessione a BigQuery:

1. In Google Cloud Platform, crea e identifica le risorse pertinenti:
   - Crea o identifica un progetto GCP contenente BigQuery le tabelle a cui desideri connetterti.
   - Abilita l' BigQuery API. Per ulteriori informazioni, consulta Utilizzare l'API BigQuery Storage Read per leggere i dati delle tabelle.
2. In Google Cloud Platform, crea ed esporta le credenziali dell'account del servizio:

   È possibile utilizzare la procedura guidata per le BigQuery credenziali per velocizzare questo passaggio: Creazione di credenziali.

Per creare un account di servizio in GCP, segui il tutorial disponibile in Creazione di account di servizio.

- Quando selezionate il progetto, selezionate il progetto contenente la tabella. BigQuery
- Quando selezioni i ruoli GCP IAM per il tuo account di servizio, aggiungi o crea un ruolo che conceda le autorizzazioni appropriate per eseguire BigQuery lavori di lettura, scrittura o creazione BigQuery di tabelle.

Per creare le credenziali per il tuo account di servizio, segui il tutorial disponibile in Creazione della chiave di un account di servizio.

- Quando selezioni il tipo di chiave, seleziona JSON.

Ora dovresti avere scaricato un file JSON con le credenziali per il tuo account di servizio. La schermata visualizzata dovrebbe risultare simile a quella nell'immagine seguente:

```
{
  "type": "service_account",
  "project_id": "*****",
  "private_key_id": "*****",
  "private_key": "*****",
  "client_email": "*****",
  "client_id": "*****",
  "auth_uri": "https://accounts.google.com/o/oauth2/auth",
  "token_uri": "https://oauth2.googleapis.com/token",
  "auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",
  "client_x509_cert_url": "*****",
  "universe_domain": "googleapis.com"
}
```

3. Carica il file JSON delle credenziali in una posizione Amazon S3 adeguatamente sicura. Conserva il percorso del file, *s3secretpath* per le fasi future.
4. Nel Gestione dei segreti AWS, crea un segreto utilizzando le tue credenziali di Google Cloud Platform. Per creare un segreto in Secrets Manager, segui il tutorial disponibile in Crea un Gestione dei segreti AWS segreto nella Gestione dei segreti AWS documentazione. Dopo aver creato il segreto, mantieni il nome del segreto *secretName* per il passaggio successivo.

   Quando create Key/value coppie, specificate chiavi e valori come segue:

- Per le `client_id` chiavi `token_uri` `client_x509_cert_urlprivate_key_id`,`project_id`,`universe_domain`,,`auth_provider_` specificate i valori corrispondenti nel file JSON scaricato.
- Per `spark.hadoop.google.cloud.auth.service.account.json.keyfile` chiave, specificare il. *`s3secretpath`*

5. Nel AWS Glue Data Catalog, crea una connessione seguendo i passaggi riportati di seguitothe section called "Aggiungere una AWS Glue connessione". Dopo aver creato la connessione, conservate il nome della connessione per il passaggio successivo. *`connectionName`*

   - Quando selezioni un tipo di connessione, seleziona Google BigQuery.
   - Quando selezioni un AWS segreto, fornisci*`secretName`*.

6. Concedi al ruolo IAM associato al tuo lavoro AWS Glue il permesso di lettura*`secretName`*.
7. Nella configurazione del lavoro AWS Glue, fornisci *`connectionName`* una connessione di rete aggiuntiva.

Leggere dalle BigQuery tabelle

Prerequisiti:

- Una BigQuery tabella da cui vorresti leggere. Avrai bisogno dei nomi delle BigQuery tabelle e dei set di dati, nel modulo`[dataset].[table]`. Chiamiamo questo*`tableName`*.
- Il progetto di fatturazione per il BigQuery tavolo. Avrai bisogno del nome del progetto,*`parentProject`*. Se non esiste un progetto padre di fatturazione, utilizza il progetto contenente la tabella.
- BigQuery informazioni di autenticazione. Completa i passaggi per gestire le credenziali di connessione con AWS Glue per configurare le informazioni di autenticazione. Avrai bisogno del nome della connessione AWS Glue,*`connectionName`*.

Esempio:

```
bigquery_read = glueContext.create_dynamic_frame.from_options(
    connection_type="bigquery",
    connection_options={
        "connectionName": "connectionName",
        "parentProject": "parentProject",
        "sourceType": "table",
```

```
        "table": "tableName",
    }
```

Puoi anche fornire una query, per filtrare i risultati restituiti al tuo DynamicFrame. Sarà necessario configurare `query`, `sourceType`, `viewsEnabled` e `materializationDataset`.

Esempio:

Prerequisiti aggiuntivi:

Dovrai creare o identificare un BigQuery set di dati*materializationDataset*, in cui BigQuery scrivere viste materializzate per le tue query.

Dovrai concedere le autorizzazioni GCP IAM appropriate al tuo account di servizio per creare tabelle. *materializationDataset*

```
glueContext.create_dynamic_frame.from_options(
            connection_type="bigquery",
            connection_options={
                "connectionName": "connectionName",
                "materializationDataset": materializationDataset,
                "parentProject": "parentProject",
                "viewsEnabled": "true",
                "sourceType": "query",
                "query": "select * from bqtest.test"
            }
        )
```

Scrittura su tabelle BigQuery

Questo esempio scrive direttamente sul BigQuery servizio. BigQuery supporta anche il metodo di scrittura «indiretto». Per ulteriori informazioni sulla configurazione di scritture indirette, consulta la pagina the section called "Utilizzo della scrittura indiretta con Google BigQuery".

Prerequisiti:

- Una BigQuery tabella su cui desideri scrivere. Avrai bisogno dei nomi delle BigQuery tabelle e dei set di dati, nel modulo`[dataset].[table]`. È possibile anche fornire un nuovo nome di tabella che verrà creato automaticamente. Chiamiamo questo*tableName*.

- Il progetto di fatturazione per il BigQuery tavolo. Avrai bisogno del nome del progetto,*parentProject*. Se non esiste un progetto padre di fatturazione, utilizza il progetto contenente la tabella.
- BigQuery informazioni di autenticazione. Completa i passaggi per gestire le credenziali di connessione con AWS Glue per configurare le informazioni di autenticazione. Avrai bisogno del nome della connessione AWS Glue,*connectionName*.

Esempio:

```
bigquery_write = glueContext.write_dynamic_frame.from_options(
    frame=frameToWrite,
    connection_type="bigquery",
    connection_options={
        "connectionName": "connectionName",
        "parentProject": "parentProject",
        "writeMethod": "direct",
        "table": "tableName",
    }
)
```

BigQuery riferimento all'opzione di connessione

- `project`: Predefinita: impostazione predefinita dell'account del servizio Google Cloud. Utilizzato per la lettura/scrittura. Il nome di un progetto Google Cloud associato alla tua tabella.
- `table`: (obbligatorio) utilizzato per la lettura/scrittura. Il nome della BigQuery tabella nel formato`[[project:]dataset.]`.
- `dataset`: obbligatorio quando non è definito tramite l'opzione `table`. Utilizzato per la lettura/scrittura. Il nome del set di dati contenente la BigQuery tabella.
- `parentProject`: Predefinita: impostazione predefinita dell'account del servizio Google Cloud. Utilizzato per la lettura/scrittura. Il nome di un progetto Google Cloud associato al `project` utilizzato per la fatturazione.
- `sourceType`: utilizzato per la lettura. Richiesto durante la lettura. Valori validi:`table`, `query` Indica a AWS Glue se leggerai per tabella o per query.
- `materializationDataset`: utilizzato per la lettura. Valori validi: stringhe. Il nome di un BigQuery set di dati utilizzato per memorizzare le materializzazioni per le viste.

- `viewsEnabled`: utilizzato per la lettura. Valore predefinito: false. Valori validi: vero, falso. Configura se BigQuery utilizzerà le viste.
- `query`: utilizzato per la lettura. Usato quando `viewsEnabled` è vero. Una query DQL di GoogleSQL.
- `temporaryGcsBucket`: utilizzato per la scrittura. Obbligatorio quando `writeMethod` è impostato sull'impostazione predefinita (`indirect`). Nome di un bucket di Google Cloud Storage utilizzato per archiviare una forma intermedia dei dati durante la scrittura su. BigQuery
- `writeMethod`: valore predefinito: `indirect`. Valori validi: `direct`, `indirect`. Utilizzato per la scrittura. Specifica il metodo utilizzato per scrivere i dati.
  - Se impostato su`direct`, il connettore scriverà utilizzando l'API BigQuery Storage Write.
  - Se impostato su`indirect`, il connettore scriverà su Google Cloud Storage, quindi lo trasferirà su BigQuery Using a Load. L'account del servizio Google Cloud avrà bisogno delle autorizzazioni GCS appropriate.

Utilizzo della scrittura indiretta con Google BigQuery

Questo esempio utilizza la scrittura indiretta, che scrive i dati su Google Cloud Storage e li copia su Google BigQuery.

Prerequisiti:

Avrai bisogno di un bucket temporaneo di Google Cloud Storage,*`temporaryBucket`*.

Il ruolo GCP IAM per l'account di servizio GCP di AWS Glue richiederà le autorizzazioni GCS appropriate per l'accesso. *`temporaryBucket`*

Configurazione aggiuntiva:

Per configurare la scrittura indiretta con: BigQuery

1. Valuta the section called “Configurazione BigQuery” e individua o scarica nuovamente il file JSON delle credenziali GCP. *`secretName`*Identify, il Gestione dei segreti AWS segreto della connessione Google BigQuery AWS Glue utilizzata nel tuo lavoro.
2. Carica il file JSON delle credenziali in una posizione Amazon S3 adeguatamente sicura. Conserva il percorso del file, *`s3secretpath`* per le fasi future.
3. Modifica*`secretName`*, aggiungendo la `spark.hadoop.google.cloud.auth.service.account.json.keyfile` chiave. Impostare il valore su *`s3secretpath`*.

4. Concedi al tuo lavoro AWS Glue Job le autorizzazioni di accesso ad Amazon S3 IAM. *s3secretpath*

Ora puoi fornire la posizione temporanea del bucket GCS al tuo metodo di scrittura. Non è necessario fornire il `writeMethod`, poiché `indirect` in passato è stata l'impostazione predefinita.

```
bigquery_write = glueContext.write_dynamic_frame.from_options(
    frame=frameToWrite,
    connection_type="bigquery",
    connection_options={
        "connectionName": "connectionName",
        "parentProject": "parentProject",
        "temporaryGcsBucket": "temporaryBucket",
        "table": "tableName",
    }
)
```

Connessioni JDBC

Alcuni tipi di database, tipicamente relazionali, supportano la connessione tramite lo standard JDBC. Per ulteriori informazioni su JDBC, consulta la documentazione dell'API Java JDBC. AWS Glue supporta nativamente la connessione a determinati database tramite i relativi connettori JDBC: le librerie JDBC sono fornite nei job Glue Spark. AWS Quando ci si connette a questi tipi di database utilizzando le librerie do AWS Glue, si ha accesso a un set standard di opzioni.

I valori JDBC connectionType includono quanto segue:

- `"connectionType": "sqlserver"`: designa una connessione a un database Microsoft SQL Server.
- `"connectionType": "mysql"`: designa una connessione al database MySQL.
- `"connectionType": "oracle"`: designa una connessione a un database Oracle.
- `"connectionType": "postgresql"`: designa una connessione al database PostgreSQL.
- `"connectionType": "redshift"`: designa una connessione a un database Amazon Redshift. Per ulteriori informazioni, consulta the section called "Connessioni Redshift".

La tabella seguente elenca le versioni dei driver JDBC supportate da AWS Glue.

| Prodotto | Versioni dei driver JDBC per Glue 5.1 | Versioni dei driver JDBC per Glue 5.0 | Versioni del driver JDBC per Glue 4.0 | Versioni del driver JDBC per Glue 3.0 | Versioni del driver JDBC per Glue 0.9, 1.0, 2.0 |
|---|---|---|---|---|---|
| Microsoft SQL Server | 10.2.0 | 10.2,0 | 9,40 | 7.x | 6.x |
| MySQL | 8,0,33 | 8,0,33 | 8.0.23 | 8.0.23 | 5.1 |
| Banca dati Oracle | 23.3.0.23.09 | 23,3,023,09 | 21,7 | 21,1 | 11.2 |
| PostgreSQL | 42,7,3 | 42,7,3 | 42,36 | 4,2,18 | 42.1.x |
| Amazon Redshift* | redshift-jdbc42-2.1.0.29 | redshift-jdbc42-2.1.0.29 | redshift-jdbc42-2.1.0.16 | redshift-jdbc41-1.2.12.1017 | redshift-jdbc41-1.2.12.1017 |

* Per il tipo di connessione Amazon Redshift, tutte le altre name/value coppie di opzioni incluse nelle opzioni di connessione per una connessione JDBC, incluse le opzioni di formattazione, vengono passate direttamente allo SparkSQL sottostante. DataSource Nei lavori AWS Glue with Spark in AWS Glue 4.0 e versioni successive, il connettore nativo AWS Glue per Amazon Redshift utilizza l'integrazione Amazon Redshift per Apache Spark. Per ulteriori informazioni, consulta la pagina Integrazione di Amazon Redshift per Apache Spark. Nelle versioni precedenti, consulta la sezione Amazon Redshift data source for Spark.

Per configurare Amazon VPC per la connessione ai datastore Amazon RDS tramite JDBC, consulta la pagina the section called "Impostazione di Amazon VPC per connettersi ai data store Amazon RDS".

> **Note**
>
> AWS I Glue job vengono associati solo a una sottorete durante un'esecuzione. Ciò potrebbe influire sulla capacità di connettersi a più origini dati tramite lo stesso processo. Questo comportamento non è limitato alle origini JDBC.

Argomenti


- Indicazioni di riferimento alle opzioni di connessione a JDBC
- Utilizzo di sampleQuery
- Utilizzo di un driver JDBC personalizzato
- Lettura in parallelo dalle tabelle JDBC
- Impostazione di Amazon VPC per le connessioni JDBC ai data store Amazon RDS su JDBC da AWS Glue


## Indicazioni di riferimento alle opzioni di connessione a JDBC

Se hai già definito una connessione AWS JDBC Glue, puoi riutilizzare le proprietà di configurazione in essa definite, come: url, user e password; quindi non devi specificarle nel codice come opzioni di connessione. Questa funzionalità è disponibile in AWS Glue 3.0 e versioni successive. A tale scopo, utilizza le seguenti proprietà di connessione:

- `"useConnectionProperties"`: impostare su “true” per indicare che si desidera utilizzare la configurazione da una connessione.
- `"connectionName"`: inserire il nome della connessione da cui recuperare la configurazione; la connessione deve essere definita nella stessa regione del processo.

Utilizzare queste opzioni di connessione con connessioni JDBC:

- `"url"`: (obbligatorio) l'URL JDBC per il database.
- `"dbtable"`: (obbligatorio) la tabella database da cui leggere. Per gli archivi dati JDBC che supportano schemi all'interno di un database, specificare `schema.table-name`. Se non viene fornito alcuno schema, viene usato lo schema “pubblico” predefinito.
- `"user"`: (obbligatorio) il nome utente da usare per la connessione.
- `"password"`: (obbligatorio) la password da usare per la connessione.
- (Opzionale) Le seguenti opzioni consentono di fornire un driver JDBC personalizzato. Usa queste opzioni se devi usare un driver che AWS Glue non supporta nativamente.

  I processi ETL possono utilizzare versioni di driver JDBC diverse per l'origine dati e la destinazione, anche se l'origine e la destinazione sono lo stesso prodotto di database. In questo modo è possibile eseguire la migrazione dei dati tra database di origine e di destinazione con versioni

diverse. Per utilizzare queste opzioni, è necessario innanzitutto caricare il file JAR del driver JDBC su Amazon S3.

  - `"customJdbcDriverS3Path"`: il percorso Amazon S3 del driver JDBC personalizzato.
  - `"customJdbcDriverClassName"`: il nome della classe del driver JDBC.

- `"bulkSize"`: (opzionale) utilizzato per configurare inserti paralleli per accelerare i carichi di massa nelle destinazioni JDBC. Specifica un valore intero per il grado di parallelismo da utilizzare durante la scrittura o l'inserimento di dati. Questa opzione è utile per migliorare le prestazioni delle scritture in database come Arch User Repository (AUR).
- `"hashfield"`: (opzionale) una stringa utilizzata per specificare il nome di una colonna nella tabella JDBC da utilizzare per dividere i dati in partizioni durante la lettura da tabelle JDBC in parallelo. Fornire “hashfield” O “hashexpression”. Per ulteriori informazioni, consultare the section called “Lettura in parallelo da JDBC”.
- `"hashexpression"`: (opzionale) una clausola SQL select che restituisce un numero intero. Utilizzata per suddividere i dati presenti in una tabella JDBC in partizioni durante la lettura da tabelle JDBC in parallelo. Fornire “hashfield” O “hashexpression”. Per ulteriori informazioni, consultare the section called “Lettura in parallelo da JDBC”.
- `"hashpartitions"`: (opzionale) un numero intero positivo. Utilizzato per specificare il numero di letture parallele della tabella JDBC durante la lettura da tabelle JDBC in parallelo. Impostazione predefinita: 7. Per ulteriori informazioni, consultare the section called “Lettura in parallelo da JDBC”.
- `"sampleQuery"`: (opzionale) un'istruzione di query SQL personalizzata. Utilizzata per specificare un sottoinsieme di informazioni in una tabella per recuperare un campione del contenuto della tabella. Se configurato indipendentemente dai dati, può essere meno efficiente dei DynamicFrame metodi e causare timeout o errori di esaurimento della memoria. Per ulteriori informazioni, consulta the section called “Utilizzo di sampleQuery”.
- `"enablePartitioningForSampleQuery"`: (opzionale) un valore booleano. Valore predefinito: false. Utilizzato per abilitare la lettura da tabelle JDBC in parallelo durante la specificazione della `sampleQuery`. Se impostato su true, **`sampleQuery`** deve terminare con «where» o «and» affinché AWS Glue aggiunga le condizioni di partizionamento. Per ulteriori informazioni, consulta the section called “Utilizzo di sampleQuery”.
- `"sampleSize"`: (opzionale) un numero intero positivo. Limita il numero di righe restituite dalla query di esempio. Funziona solo quando `enablePartitioningForSampleQuery` è true. Se il partizionamento non è abilitato, è invece necessario aggiungere direttamente `"limit x"` nella `sampleQuery` per limitare le dimensioni. Per ulteriori informazioni, consultare the section called “Utilizzo di sampleQuery”.

Utilizzo di sampleQuery

Questa sezione illustra come utilizzare `sampleQuery`, `sampleSize` e `enablePartitioningForSampleQuery`.

`sampleQuery` può essere un modo efficace per campionare alcune righe del set di dati. Per impostazione predefinita, la query viene eseguita da un singolo esecutore. Se configurato indipendentemente dai dati, può essere meno efficiente dei DynamicFrame metodi e causare timeout o errori di esaurimento della memoria. Nelle pipeline ETL, l'esecuzione diretta di query SQL sul database sottostante è in genere necessaria solo per ottimizzare le prestazioni. Se si sta cercando di visualizzare in anteprima alcune righe del tuo set di dati, prendere in considerazione l'utilizzo di the section called "show". Se stai cercando di trasformare il tuo set di dati utilizzando SQL, prendi in considerazione l'utilizzo the section called "toDF" per definire una trasformazione SparkSQL rispetto ai tuoi dati in DataFrame un modulo.

Anche se la query può manipolare diverse tabelle, `dbtable` resta obbligatoria.

Utilizzo di SampleQuery per recuperare un campione della tabella

Quando si utilizza il comportamento SampleQuery predefinito per recuperare un campione dei dati, AWS Glue non prevede un throughput sostanziale, quindi esegue la query su un singolo esecutore. Per limitare i dati forniti e non causare problemi di prestazioni, consigliamo di fornire a SQL una clausola `LIMIT`.

Example Usare SampleQuery senza partizionamento

Il codice di esempio seguente mostra come utilizzare `sampleQuery` senza partizionamento.

```
//A full sql query statement.
val query = "select name from $tableName where age > 0 limit 1"
val connectionOptions = JsonOptions(Map(
    "url" -> url,
    "dbtable" -> tableName,
    "user" -> user,
    "password" -> password,
    "sampleQuery" -> query ))
val dyf = glueContext.getSource("mysql", connectionOptions)
          .getDynamicFrame()
```

Utilizzo di SampleQuery con set di dati più grandi

Se si sta leggendo un set di dati di grandi dimensioni, potrebbe essere necessario abilitare il partizionamento JDBC per interrogare una tabella in parallelo. Per ulteriori informazioni, consultare the section called "Lettura in parallelo da JDBC". Per utilizzare `sampleQuery` con partizionamento JDBC, impostare `enablePartitioningForSampleQuery` su true. L'attivazione di questa funzionalità richiede di apportare alcune modifiche alla `sampleQuery`.

Quando si utilizza il partizionamento JDBC con`sampleQuery`, la query deve terminare con «where» o «and» affinché AWS Glue possa aggiungere le condizioni di partizionamento.

Se si desidera limitare i risultati di SampleQuery durante la lettura da tabelle JDBC in parallelo, impostare il parametro `"sampleSize"` anziché specificare una clausola `LIMIT`.

Example Usare SampleQuery con partizionamento JDBC

Il codice di esempio seguente mostra come utilizzare `sampleQuery` con partizionamento JDBC.

```
//note that the query should end with "where" or "and" if use with JDBC partitioning.
val query = "select name from $tableName where age > 0 and"

//Enable JDBC partitioning by setting hashfield.
//to use sampleQuery with partitioning, set enablePartitioningForSampleQuery.
//use sampleSize to limit the size of returned data.
val connectionOptions = JsonOptions(Map(
    "url" -> url,
    "dbtable" -> tableName,
    "user" -> user,
    "password" -> password,
    "hashfield" -> primaryKey,
    "sampleQuery" -> query,
    "enablePartitioningForSampleQuery" -> true,
    "sampleSize" -> "1" ))
val dyf = glueContext.getSource("mysql", connectionOptions)
          .getDynamicFrame()
```

Note e restrizioni:

Le query di esempio non possono essere utilizzate insieme ai segnalibri del processo. Lo stato del segnalibro verrà ignorato quando viene fornita la configurazione per entrambi.

Utilizzo di un driver JDBC personalizzato

Negli esempi di codice riportati di seguito viene illustrato come leggere e scrivere nei database JDBC con driver JDBC personalizzati. Essi dimostrano la lettura da una versione di un prodotto di database e la scrittura a una versione successiva dello stesso prodotto.

Python

```
import sys
from awsglue.transforms import *
from awsglue.utils import getResolvedOptions
from pyspark.context import SparkContext, SparkConf
from awsglue.context import GlueContext
from awsglue.job import Job
import time
from pyspark.sql.types import StructType, StructField, IntegerType, StringType

sc = SparkContext()
glueContext = GlueContext(sc)
spark = glueContext.spark_session

# Construct JDBC connection options
connection_mysql5_options = {
    "url": "jdbc:mysql://<jdbc-host-name>:3306/db",
    "dbtable": "test",
    "user": "admin",
    "password": "pwd"}

connection_mysql8_options = {
    "url": "jdbc:mysql://<jdbc-host-name>:3306/db",
    "dbtable": "test",
    "user": "admin",
    "password": "pwd",
    "customJdbcDriverS3Path": "s3://amzn-s3-demo-bucket/mysql-connector-
java-8.0.17.jar",
    "customJdbcDriverClassName": "com.mysql.cj.jdbc.Driver"}

connection_oracle11_options = {
    "url": "jdbc:oracle:thin:@//<jdbc-host-name>:1521/ORCL",
    "dbtable": "test",
    "user": "admin",
    "password": "pwd"}
```

```
connection_oracle18_options = {
    "url": "jdbc:oracle:thin:@//<jdbc-host-name>:1521/ORCL",
    "dbtable": "test",
    "user": "admin",
    "password": "pwd",
    "customJdbcDriverS3Path": "s3://amzn-s3-demo-bucket/ojdbc10.jar",
    "customJdbcDriverClassName": "oracle.jdbc.OracleDriver"}

# Read from JDBC databases with custom driver
df_mysql8 = glueContext.create_dynamic_frame.from_options(connection_type="mysql",

 connection_options=connection_mysql8_options)

# Read DynamicFrame from MySQL 5 and write to MySQL 8
df_mysql5 = glueContext.create_dynamic_frame.from_options(connection_type="mysql",

 connection_options=connection_mysql5_options)
glueContext.write_from_options(frame_or_dfc=df_mysql5, connection_type="mysql",
                               connection_options=connection_mysql8_options)

# Read DynamicFrame from Oracle 11 and write to Oracle 18
df_oracle11 =
 glueContext.create_dynamic_frame.from_options(connection_type="oracle",

 connection_options=connection_oracle11_options)
glueContext.write_from_options(frame_or_dfc=df_oracle11, connection_type="oracle",
                               connection_options=connection_oracle18_options)
```

## Scala

```
import com.amazonaws.services.glue.GlueContext
import com.amazonaws.services.glue.MappingSpec
import com.amazonaws.services.glue.errors.CallSite
import com.amazonaws.services.glue.util.GlueArgParser
import com.amazonaws.services.glue.util.Job
import com.amazonaws.services.glue.util.JsonOptions
import com.amazonaws.services.glue.DynamicFrame
import org.apache.spark.SparkContext
import scala.collection.JavaConverters._


object GlueApp {
```

```
  val MYSQL_5_URI: String = "jdbc:mysql://<jdbc-host-name>:3306/db"
  val MYSQL_8_URI: String = "jdbc:mysql://<jdbc-host-name>:3306/db"
  val ORACLE_11_URI: String = "jdbc:oracle:thin:@//<jdbc-host-name>:1521/ORCL"
  val ORACLE_18_URI: String = "jdbc:oracle:thin:@//<jdbc-host-name>:1521/ORCL"

  // Construct JDBC connection options
  lazy val mysql5JsonOption = jsonOptions(MYSQL_5_URI)
  lazy val mysql8JsonOption = customJDBCDriverJsonOptions(MYSQL_8_URI, "s3://amzn-
s3-demo-bucket/mysql-connector-java-8.0.17.jar", "com.mysql.cj.jdbc.Driver")
  lazy val oracle11JsonOption = jsonOptions(ORACLE_11_URI)
  lazy val oracle18JsonOption = customJDBCDriverJsonOptions(ORACLE_18_URI, "s3://
amzn-s3-demo-bucket/ojdbc10.jar", "oracle.jdbc.OracleDriver")

  def main(sysArgs: Array[String]): Unit = {
    val spark: SparkContext = new SparkContext()
    val glueContext: GlueContext = new GlueContext(spark)
    val args = GlueArgParser.getResolvedOptions(sysArgs, Seq("JOB_NAME").toArray)
    Job.init(args("JOB_NAME"), glueContext, args.asJava)

    // Read from JDBC database with custom driver
    val df_mysql8: DynamicFrame = glueContext.getSource("mysql",
 mysql8JsonOption).getDynamicFrame()

    // Read DynamicFrame from MySQL 5 and write to MySQL 8
    val df_mysql5: DynamicFrame = glueContext.getSource("mysql",
 mysql5JsonOption).getDynamicFrame()
    glueContext.getSink("mysql", mysql8JsonOption).writeDynamicFrame(df_mysql5)

    // Read DynamicFrame from Oracle 11 and write to Oracle 18
    val df_oracle11: DynamicFrame = glueContext.getSource("oracle",
 oracle11JsonOption).getDynamicFrame()
    glueContext.getSink("oracle", oracle18JsonOption).writeDynamicFrame(df_oracle11)

    Job.commit()
  }

  private def jsonOptions(url: String): JsonOptions = {
    new JsonOptions(
      s"""{"url": "${url}",
         |"dbtable":"test",
         |"user": "admin",
         |"password": "pwd"}""".stripMargin)
  }
```

```
  private def customJDBCDriverJsonOptions(url: String, customJdbcDriverS3Path:
 String, customJdbcDriverClassName: String): JsonOptions = {
    new JsonOptions(
      s"""{"url": "${url}",
         |"dbtable":"test",
         |"user": "admin",
         |"password": "pwd",
         |"customJdbcDriverS3Path": "${customJdbcDriverS3Path}",
         |"customJdbcDriverClassName" :
 "${customJdbcDriverClassName}"}""".stripMargin)
  }
}
```

Lettura in parallelo dalle tabelle JDBC

Puoi impostare le proprietà della tabella JDBC per permettere a AWS Glue di leggere i dati in parallelo. Quando imposti determinate proprietà, indichi a AWS Glue di eseguire query SQL parallele su partizioni logiche dei dati. Puoi controllare il partizionamento impostando un campo hash o un'espressione hash. Puoi anche controllare il numero di operazioni di lettura parallele usate per accedere ai dati.

La lettura delle tabelle JDBC in parallelo è una tecnica di ottimizzazione che può migliorare le prestazioni. Per ulteriori informazioni sul processo di identificazione di quando questa tecnica è appropriata, consulta Reduce the amount of data scan nella guida sulle migliori pratiche per l'ottimizzazione delle prestazioni per i lavori di Apache Spark su AWS Glue AWS Prescriptive Guidance.

Per abilitare le letture parallele, puoi impostare coppie chiave/valore nel campo dei parametri della struttura della tabella. Utilizza la notazione JSON per impostare un valore per il campo parametri della tabella. Per ulteriori informazioni sulle modifiche delle proprietà di una tabella, consulta Visualizzazione e gestione dei dettagli della tabella. Puoi anche abilitare le letture parallele chiamando i metodi ETL (Extract, Transform and Load, estrazione, trasformazione e caricamento) `create_dynamic_frame_from_options` e `create_dynamic_frame_from_catalog`. Per ulteriori informazioni sulla definizione delle opzioni in questi metodi, consulta from_options e from_catalog.

Puoi usare questo metodo per le tabelle JDBC, ovvero la maggior parte delle tabelle i cui dati dai base costituiscono un datastore JDBC. Queste proprietà vengono ignorate quando viene eseguita la lettura delle tabelle Amazon Redshift e Amazon S3.

hashfield

Imposta `hashfield` sul nome di una colonna nella tabella JDBC da usare per dividere i dati in partizioni. Per ottenere risultati ottimali, questa colonna deve avere una distribuzione uniforme dei valori per distribuire i dati tra le partizioni. Questa colonna può contenere qualsiasi tipo di dati. AWS Glue genera query non sovrapposte eseguite in parallelo per leggere i dati partizionati in base a questa colonna. Ad esempio, se i dati sono distribuiti in modo uniforme in base al mese, è possibile usare la colonna `month` per leggere ogni mese di dati in parallelo.

```
'hashfield': 'month'
```

AWS Glue crea una query per l'hashing del valore di campo a un numero di partizione ed esegue la query per tutte le partizioni in parallelo. Per usare la query personalizzata per partizionare la lettura di una tabella, fornisci un oggetto `hashexpression` al posto di un oggetto `hashfield`.

hashexpression

Imposta `hashexpression` su un'espressione SQL (conforme alla grammatica del motore di database JDBC) che restituisce un numero intero. Un'espressione semplice è il nome di qualsiasi colonna numerica nella tabella. AWS Glue genera query SQL per leggere i dati JDBC in parallelo usando `hashexpression` nella clausola `WHERE` per partizionare i dati.

Ad esempio, è possibile usare la colonna numerica `customerID` per leggere i dati partizionati in base a un numero cliente.

```
'hashexpression': 'customerID'
```

Per fare in modo che AWS Glue controlli il partizionamento, fornisci un oggetto `hashfield` al posto di un oggetto `hashexpression`.

hashpartitions

Imposta `hashpartitions` sul numero di letture parallele della tabella JDBC. Se questa proprietà non viene impostata, il valore predefinito è 7.

Imposta, ad esempio, il numero di letture parallele su `5`, in modo che AWS Glue legga i dati con cinque query (o meno).

```
'hashpartitions': '5'
```

Impostazione di Amazon VPC per le connessioni JDBC ai data store Amazon RDS su JDBC da AWS Glue

Quando si utilizza JDBC per connettersi ai database in Amazon RDS, è necessario eseguire una configurazione aggiuntiva. Per permettere la comunicazione tra i componenti AWS Glue e Amazon RDS, è necessario configurare l'accesso ai datastore Amazon RDS in Amazon VPC. Per permettere la comunicazione dei componenti di AWS Glue, specifica un gruppo di sicurezza con una regola per il traffico in entrata autoreferenziale per tutte le porte TCP. Creando una regola autoreferenziale, si può limitare l'origine allo stesso gruppo di sicurezza nel VPC. Una regola autoreferenziale non aprirà il VPC a tutte le reti. Il gruppo di sicurezza predefinito per il tuo VPC potrebbe già avere una regola autoreferenziata in entrata per ALL Traffic.

Per configurare l'accesso tra AWS Glue e gli archivi dati Amazon RDS

1. Accedi a Console di gestione AWS e apri la console Amazon RDS all'indirizzo https://console.aws.amazon.com/rds/.
2. Nella console Amazon RDS, identificare i gruppi di sicurezza utilizzati per controllare l'accesso al database Amazon RDS.

   Nel riquadro di navigazione a sinistra, scegliere Database, poi selezionare l'istanza a cui ci si desidera connettersi dall'elenco nel riquadro principale.

   Nella pagina dei dettagli del database, individuare i gruppi di sicurezza di VPC nella scheda Connettività e sicurezza.
3. In base all'architettura di rete, identificate quale gruppo di sicurezza associato è meglio modificare per consentire l'accesso al servizio AWS Glue. Salva il suo nome, *`database-security-group`* per riferimenti futuri. Se non esiste un gruppo di sicurezza appropriato, seguire le istruzioni per Fornire l'accesso all'istanza DB nel VPC creando un gruppo di sicurezza nella documentazione relativa ad Amazon RDS.
4. Accedi a Console di gestione AWS e apri la console Amazon VPC all'indirizzo. https://console.aws.amazon.com/vpc/
5. Nella console Amazon VPC, identifica come eseguire l'aggiornamento. *`database-security-group`*

Nel riquadro di navigazione a sinistra, scegli Gruppi di sicurezza, quindi seleziona *database-security-group* dall'elenco nel riquadro principale.

6. Identifica l'ID del gruppo di sicurezza per*database-security-group*,*database-sg-id*. Salvarlo per utilizzarlo in futuro.

   Nella pagina dei dettagli del gruppo di sicurezza, individuare l'ID del gruppo di sicurezza.

7. Modifica le regole in entrata per*database-security-group*, aggiungi una regola di autoreferenziazione per consentire ai AWS Glue componenti di comunicare. In particolare, aggiungi o conferma l'esistenza di una regola in cui Type è**All TCP**, Protocol è**TCP**, Port Range include tutte le porte e Source è. *database-sg-id* Verificare che il gruppo di sicurezza inserito per Origine sia lo stesso del gruppo di sicurezza in fase di modifica.

   Nella pagina dei dettagli del gruppo di sicurezza, selezionare Modifica regole in entrata.

   La regola in entrata è simile alla seguente:

| Tipo | Protocollo | Intervallo porte | Origine |
|---|---|---|---|
| Tutte le regole TCP | TCP | 0–65535 | *database-sg-id* |

8. Aggiungere le regole per il traffico in uscita.

   Nella pagina dei dettagli del gruppo di sicurezza, selezionare Modifica regole in uscita.

   Se il gruppo di sicurezza abilita tutto il traffico in uscita, non sono necessarie regole separate. Esempio:

| Tipo | Protocollo | Intervallo porte | Destinazione |
|---|---|---|---|
| All Traffic | ALL | ALL | 0.0.0.0/0 |

   Se l'architettura di rete è progettata per limitare il traffico in uscita, creare le seguenti regole in uscita:

   Crea una regola autoreferenziale in cui Type è`All TCP`, Protocol è`TCP`, Port Range include tutte le porte e Destination è. *database-sg-id* Verificare che il gruppo di sicurezza inserito per Destinazione sia lo stesso del gruppo di sicurezza in fase di modifica.

Se si utilizza un endpoint VPC Amazon S3, aggiungere una regola HTTPS per permettere il traffico dal VPC verso Amazon S3. Crea una regola in cui Type èHTTPS, Protocol isTCP, Port Range è 443 e Destination è l'ID dell'elenco di prefissi gestiti per l'endpoint gateway Amazon S3,. *s3-prefix-list-id* Per ulteriori informazioni sugli elenchi di prefissi e sugli endpoint gateway Amazon S3, consultare Endpoint gateway per Amazon S3 nella documentazione di Amazon VPC.

Ad esempio:

| Tipo | Protocollo | Intervallo porte | Destinazione |
|---|---|---|---|
| Tutte le regole TCP | TCP | 0–65535 | *database-sg-id* |
| HTTPS | TCP | 443 | *s3-prefix-list-id* |

Connessioni MongoDB

Puoi usare AWS Glue for Spark per leggere e scrivere su tabelle in MongoDB e MongoDB Atlas in Glue 4.0 e versioni successive AWS . È possibile connettersi a MongoDB utilizzando le credenziali di nome utente e password archiviate tramite una connessione Glue. Gestione dei segreti AWS AWS

Per ulteriori informazioni su MongoDB, consulta la documentazione di MongoDB.

Configurazione delle connessioni MongoDB

Per connetterti a MongoDB AWS da Glue, avrai bisogno delle tue credenziali MongoDB e. *mongodbUser* *mongodbPass*

Per connetterti a MongoDB AWS da Glue, potresti aver bisogno di alcuni prerequisiti:

- Se la tua istanza MongoDB si trova in un Amazon VPC, configura Amazon VPC per consentire al job Glue di comunicare con l'istanza MongoDB senza che AWS il traffico attraversi la rete Internet pubblica.

  In Amazon VPC, identifica o crea un VPC, una sottorete e un gruppo di sicurezza che AWS Glue utilizzerà durante l'esecuzione del lavoro. Inoltre, assicurati che Amazon VPC sia configurato per consentire il traffico di rete tra l'istanza MongoDB e questa posizione. In base al layout della rete,

ciò potrebbe richiedere modifiche alle regole del gruppo di sicurezza, alla rete ACLs, ai gateway NAT e alle connessioni peering.

Puoi quindi procedere alla configurazione di AWS Glue per l'uso con MongoDB.

Per configurare una connessione a MongoDB:

1. Facoltativamente Gestione dei segreti AWS, crea un segreto usando le tue credenziali MongoDB. Per creare un segreto in Secrets Manager, segui il tutorial disponibile in Crea un Gestione dei segreti AWS segreto nella Gestione dei segreti AWS documentazione. Dopo aver creato il segreto, mantieni il nome del segreto *secretName* per il passaggio successivo.

   - Quando selezionate le coppie chiave/valore, create una coppia per la chiave `username` con il valore. *mongodbUser*

     Quando selezionate le coppie chiave/valore, create una coppia per la chiave `password` con il valore. *mongodbPass*

2. Nella console AWS Glue, crea una connessione seguendo i passaggi riportati di seguitothe section called "Aggiungere una AWS Glue connessione". Dopo aver creato la connessione, mantieni il nome della connessione*connectionName*, per utilizzi futuri in AWS Glue.

   - Quando selezioni un tipo di connessione, seleziona MongoDB o MongoDB Atlas.
   - Quando selezioni l'URL MongoDB o URL MongoDB Atlas, fornisci il nome host dell'istanza MongoDB.

     Un URL MongoDB viene fornito nel formato `mongodb://`*mongoHost*`:`*mongoPort*`/`*mongoDBname*.

     Un URL MongoDB Atlas viene fornito nel formato `mongodb +srv://`*mongoHost*`/`*mongoDBname*.
   - Se hai scelto di creare un segreto di Secrets Manager, scegli il tipo di Gestione dei segreti AWS credenziale.

     Quindi, in AWS Secret fornisci*secretName*.
   - Se scegli di fornire nome utente e password, fornisci *mongodbUser* e*mongodbPass*.

3. Nelle seguenti situazioni, potresti aver bisogno di una configurazione aggiuntiva:

   - Per le istanze MongoDB ospitate su AWS un Amazon VPC

- Dovrai fornire le informazioni di connessione Amazon VPC alla connessione AWS Glue che definisce le tue credenziali di sicurezza MongoDB. Durante la creazione o l'aggiornamento della connessione, imposta VPC, sottorete e Gruppi di sicurezza nelle opzioni di rete.

Dopo aver creato una connessione AWS Glue MongoDB, dovrai eseguire le seguenti azioni prima di chiamare il tuo metodo di connessione:

- Se hai scelto di creare un segreto di Secrets Manager, concedi al ruolo IAM associato al tuo lavoro AWS Glue il permesso di lettura*secretName*.
- Nella configurazione del lavoro AWS Glue, fornisci *connectionName* una connessione di rete aggiuntiva.

Per utilizzare la tua connessione AWS Glue MongoDB in AWS Glue for Spark, fornisci `connectionName` l'opzione nella chiamata al metodo di connessione. In alternativa, puoi seguire i passaggi the section called "Integrazione con MongoDB" per utilizzare la connessione insieme al AWS Glue Data Catalog.

Lettura da MongoDB utilizzando una connessione Glue AWS

Prerequisiti:

- Una raccolta MongoDB da cui desideri leggere. Avrai bisogno delle informazioni di identificazione per la raccolta.

  Una raccolta MongoDB è identificata da un nome di database e da un nome di raccolta,. *mongodbName* *mongodbCollection*
- Una connessione AWS Glue MongoDB configurata per fornire informazioni di autenticazione. Completa i passaggi della procedura precedente, Per configurare una connessione a MongoDB per configurare le informazioni di autenticazione. Avrai bisogno del nome della connessione AWS Glue,*connectionName*.

Esempio:

```
mongodb_read = glueContext.create_dynamic_frame.from_options(
    connection_type="mongodb",
    connection_options={
        "connectionName": "connectionName",
        "database": "mongodbName",
```

```
        "collection": "mongodbCollection",
        "partitioner":
 "com.mongodb.spark.sql.connector.read.partitioner.SinglePartitionPartitioner",
        "partitionerOptions.partitionSizeMB": "10",
        "partitionerOptions.partitionKey": "_id",
        "disableUpdateUri": "false",
    }
)
```

Scrittura su tabelle MongoDB

Questo esempio scrive informazioni da un file esistente DynamicFrame *dynamicFrame* a MongoDB.

Prerequisiti:

- Una raccolta MongoDB su cui desideri scrivere. Avrai bisogno delle informazioni di identificazione per la raccolta.

  Una raccolta MongoDB è identificata da un nome di database e da un nome di raccolta,. *mongodbName* *mongodbCollection*
- Una connessione AWS Glue MongoDB configurata per fornire informazioni di autenticazione. Completa i passaggi della procedura precedente, Per configurare una connessione a MongoDB per configurare le informazioni di autenticazione. Avrai bisogno del nome della connessione AWS Glue,*connectionName*.

Esempio:

```
glueContext.write_dynamic_frame.from_options(
    frame=dynamicFrame,
    connection_type="mongodb",
    connection_options={
        "connectionName": "connectionName",
        "database": "mongodbName",
        "collection": "mongodbCollection",
        "disableUpdateUri": "false",
        "retryWrites": "false",
    },
)
```

Lettura e scrittura in tabelle su tabelle MongoDB

Questo esempio scrive informazioni da un file esistente DynamicFrame *dynamicFrame* a MongoDB.

Prerequisiti:

- Una raccolta MongoDB da cui desideri leggere. Avrai bisogno delle informazioni di identificazione per la raccolta.

  Una raccolta MongoDB su cui desideri scrivere. Avrai bisogno delle informazioni di identificazione per la raccolta.

  Una raccolta MongoDB è identificata da un nome di database e da un nome di raccolta,,. *mongodbName* *mongodbCollection*
- Informazioni di autenticazione MongoDB e. *mongodbUser* *mongodbPassword*

Esempio:

Python

```
import sys
from awsglue.transforms import *
from awsglue.utils import getResolvedOptions
from pyspark.context import SparkContext, SparkConf
from awsglue.context import GlueContext
from awsglue.job import Job
import time

## @params: [JOB_NAME]
args = getResolvedOptions(sys.argv, ['JOB_NAME'])

sc = SparkContext()
glueContext = GlueContext(sc)
spark = glueContext.spark_session

job = Job(glueContext)
job.init(args['JOB_NAME'], args)

output_path = "s3://some_bucket/output/" + str(time.time()) + "/"
mongo_uri = "mongodb://<mongo-instanced-ip-address>:27017"
mongo_ssl_uri = "mongodb://<mongo-instanced-ip-address>:27017"
write_uri = "mongodb://<mongo-instanced-ip-address>:27017"
```

```
read_mongo_options = {
    "uri": mongo_uri,
    "database": "mongodbName",
    "collection": "mongodbCollection",
    "username": "mongodbUsername",
    "password": "mongodbPassword",
    "partitioner": "MongoSamplePartitioner",
    "partitionerOptions.partitionSizeMB": "10",
    "partitionerOptions.partitionKey": "_id"}

ssl_mongo_options = {
    "uri": mongo_ssl_uri,
    "database": "mongodbName",
    "collection": "mongodbCollection",
    "ssl": "true",
    "ssl.domain_match": "false"
}

write_mongo_options = {
    "uri": write_uri,
    "database": "mongodbName",
    "collection": "mongodbCollection",
    "username": "mongodbUsername",
    "password": "mongodbPassword",
}

# Get DynamicFrame from MongoDB
dynamic_frame =
 glueContext.create_dynamic_frame.from_options(connection_type="mongodb",

 connection_options=read_mongo_options)

# Write DynamicFrame to MongoDB
glueContext.write_dynamic_frame.from_options(dynamicFrame,
 connection_type="mongodb", connection_options=write_mongo_options)

job.commit()
```

## Scala

```
import com.amazonaws.services.glue.GlueContext
import com.amazonaws.services.glue.MappingSpec
```

```
import com.amazonaws.services.glue.errors.CallSite
import com.amazonaws.services.glue.util.GlueArgParser
import com.amazonaws.services.glue.util.Job
import com.amazonaws.services.glue.util.JsonOptions
import com.amazonaws.services.glue.DynamicFrame
import org.apache.spark.SparkContext
import scala.collection.JavaConverters._

object GlueApp {
  val DEFAULT_URI: String = "mongodb://<mongo-instanced-ip-address>:27017"
  val WRITE_URI: String = "mongodb://<mongo-instanced-ip-address>:27017"
  lazy val defaultJsonOption = jsonOptions(DEFAULT_URI)
  lazy val writeJsonOption = jsonOptions(WRITE_URI)
  def main(sysArgs: Array[String]): Unit = {
    val spark: SparkContext = new SparkContext()
    val glueContext: GlueContext = new GlueContext(spark)
    val args = GlueArgParser.getResolvedOptions(sysArgs, Seq("JOB_NAME").toArray)
    Job.init(args("JOB_NAME"), glueContext, args.asJava)

    // Get DynamicFrame from MongoDB
    val dynamicFrame: DynamicFrame = glueContext.getSource("mongodb",
 defaultJsonOption).getDynamicFrame()

    // Write DynamicFrame to MongoDB
    glueContext.getSink("mongodb", writeJsonOption).writeDynamicFrame(dynamicFrame)

    Job.commit()
  }

  private def jsonOptions(uri: String): JsonOptions = {
    new JsonOptions(
      s"""{"uri": "${uri}",
         |"database":"mongodbName",
         |"collection":"mongodbCollection",
         |"username": "mongodbUsername",
         |"password": "mongodbPassword",
         |"ssl":"true",
         |"ssl.domain_match":"false",
         |"partitioner": "MongoSamplePartitioner",
         |"partitionerOptions.partitionSizeMB": "10",
         |"partitionerOptions.partitionKey": "_id"}""".stripMargin)
  }
}
```

Indicazioni di riferimento alle opzioni di connessione a MongoDB

Designa una connessione a MongoDB. Le opzioni di connessione differiscono per una connessione sorgente e una connessione sink.

Queste proprietà di connessione sono condivise tra le connessioni di origine e sink:

- `connectionName`: utilizzato per la lettura/scrittura. Il nome di una connessione AWS Glue MongoDB configurata per fornire informazioni di autenticazione e di rete al metodo di connessione. Quando una connessione AWS Glue è configurata come descritto nella sezione precedentethe section called "Configurazione di MongoDB", la fornitura `connectionName` sostituirà la necessità di fornire le `"uri"` opzioni di `"password"` connessione `"username"` e.
- `"uri"`: (Obbligatorio) L'host MongoDB da cui leggere, formattato come `mongodb://<host>:<port>`. Utilizzato nelle versioni AWS Glue precedenti a AWS Glue 4.0.
- `"connection.uri"`: (Obbligatorio) L'host MongoDB da cui leggere, formattato come `mongodb://<host>:<port>`. Utilizzato in AWS Glue 4.0 e versioni successive.
- `"username"`: (Obbligatorio) Il nome utente MongoDB.
- `"password"`: (Obbligatorio) La password di MongoDB.
- `"database"`: (Obbligatorio) Il database MongoDB da cui leggere. Questa opzione può anche essere trasmessa in `additional_options` quando si chiama `glue_context.create_dynamic_frame_from_catalog` nello script di processo.
- `"collection"`: (Obbligatorio) La raccolta MongoDB da cui leggere. Questa opzione può anche essere trasmessa in `additional_options` quando si chiama `glue_context.create_dynamic_frame_from_catalog` nello script di processo.

"connectionType": "mongodb" come sorgente

Utilizzare le seguenti opzioni di connessione con `"connectionType": "mongodb"` come origine:

- `"ssl"`: (Facoltativo) Se il valore è `true`, avvia una connessione SSL. Il valore predefinito è `false`.
- `"ssl.domain_match"`: (Facoltativo) Se i valori `true` e `ssl` sono `true`, viene eseguito il controllo della corrispondenza del dominio. Il valore predefinito è `true`.
- `"batchSize"`: (Facoltativo): il numero di documenti da restituire per ogni batch, utilizzato all'interno del cursore dei batch interni.

- `"partitioner"`: (Facoltativo): il nome della classe del partizionatore per la lettura dei dati di input da MongoDB. Il connettore fornisce i seguenti partizionatori:
  - `MongoDefaultPartitioner`(impostazione predefinita) (non supportato in AWS Glue 4.0)
  - `MongoSamplePartitioner`(Richiede MongoDB 3.2 o successivo) (non supportato in AWS Glue 4.0)
  - `MongoShardedPartitioner`(Non supportato in AWS Glue 4.0)
  - `MongoSplitVectorPartitioner`(Non supportato in AWS Glue 4.0)
  - `MongoPaginateByCountPartitioner`(Non supportato in AWS Glue 4.0)
  - `MongoPaginateBySizePartitioner`(Non supportato in AWS Glue 4.0)
  - `com.mongodb.spark.sql.connector.read.partitioner.SinglePartitionPartitioner`
  - `com.mongodb.spark.sql.connector.read.partitioner.ShardedPartitioner`
  - `com.mongodb.spark.sql.connector.read.partitioner.PaginateIntoPartitionsPartit`
- `"partitionerOptions"` ( Facoltativo): opzioni per il partizionatore designato. Per ogni partizionatore sono supportate le seguenti opzioni:
  - `MongoSamplePartitioner`: `partitionKey`, `partitionSizeMB`, `samplesPerPartition`
  - `MongoShardedPartitioner`: `shardkey`
  - `MongoSplitVectorPartitioner`: `partitionKey`, `partitionSizeMB`
  - `MongoPaginateByCountPartitioner`: `partitionKey`, `numberOfPartitions`
  - `MongoPaginateBySizePartitioner`: `partitionKey`, `partitionSizeMB`

  Per ulteriori informazioni su queste opzioni, vedere Partitioner Configuration (Configurazione partizionatore) nella documentazione di MongoDB.

"connectionType": "mongodb" come sink

Utilizzare le seguenti opzioni di connessione con `"connectionType": "mongodb"` come sink:

- `"ssl"`: (Facoltativo) Se il valore è `true`, avvia una connessione SSL. Il valore predefinito è `false`.
- `"ssl.domain_match"`: (Facoltativo) Se i valori `true` e `ssl` sono `true`, viene eseguito il controllo della corrispondenza del dominio. Il valore predefinito è `true`.
- `"extendedBsonTypes"`: (Facoltativo) Se il valore è`true`, permette i tipi BSON estesi durante la scrittura di dati su MongoDB. Il valore predefinito è `true`.

- `"replaceDocument"`: (Facoltativo) Se il valore è `true`, sostituisce l'intero documento quando si salvano set di dati che contengono un campo `_id`. Se il valore è `false`, vengono aggiornati solo i campi del documento che corrispondono ai campi del set di dati. Il valore predefinito è `true`.
- `"maxBatchSize"`: (Facoltativo): la dimensione massima del batch per le operazioni in blocco durante il salvataggio dei dati. Il valore predefinito è 512.
- `"retryWrites"`: (Facoltativo): Riprova automaticamente alcune operazioni di scrittura una sola volta se AWS Glue rileva un errore di rete.

### Connessioni SAP HANA

Puoi usare AWS Glue for Spark per leggere e scrivere su tabelle in SAP HANA in AWS Glue 4.0 e versioni successive. È possibile definire cosa leggere da SAP HANA con una query SQL. Ti connetti a SAP HANA utilizzando le credenziali JDBC archiviate tramite Gestione dei segreti AWS una connessione AWS Glue SAP HANA.

Per ulteriori informazioni sulle porte SAP HANA JDBC, consulta la documentazione SAP HANA.

#### Configurazione delle connessioni SAP HANA

Per connetterti a SAP HANA da AWS Glue, dovrai creare e archiviare le tue credenziali SAP HANA in un Gestione dei segreti AWS segreto, quindi associare quel segreto a una connessione SAP HANA Glue. AWS Dovrai configurare la connettività di rete tra il tuo servizio SAP HANA e AWS Glue.

Per connetterti a SAP HANA, potrebbero essere necessari alcuni prerequisiti:

- Se il tuo servizio SAP HANA si trova in un Amazon VPC, configura Amazon VPC per consentire al job AWS Glue di comunicare con il servizio SAP HANA senza che il traffico attraversi la rete Internet pubblica.

  In Amazon VPC, identifica o crea un VPC, una sottorete e un gruppo di sicurezza che AWS Glue utilizzerà durante l'esecuzione del lavoro. Inoltre, assicurati che Amazon VPC sia configurato per consentire il traffico di rete tra l'endpoint SAP HANA e questa posizione. Il tuo processo dovrà stabilire una connessione TCP con la tua porta SAP HANA JDBC. Per ulteriori informazioni sulle porte SAP HANA, consulta la documentazione SAP HANA. In base al layout della rete, ciò potrebbe richiedere modifiche alle regole del gruppo di sicurezza, alla rete ACLs, ai gateway NAT e alle connessioni peering.
- Non ci sono prerequisiti aggiuntivi se l'endpoint SAP HANA è accessibile a Internet.

Per configurare una connessione a SAP HANA:

1. Nel Gestione dei segreti AWS, crea un segreto utilizzando le tue credenziali SAP HANA. Per creare un segreto in Secrets Manager, segui il tutorial disponibile in Crea un Gestione dei segreti AWS segreto nella Gestione dei segreti AWS documentazione. Dopo aver creato il segreto, mantieni il nome del segreto *secretName* per il passaggio successivo.

   - Quando selezionate le coppie chiave/valore, create una coppia per la chiave `username/USERNAME` con il valore. *saphanaUsername*
   - Quando selezionate le coppie chiave/valore, create una coppia per la chiave `password/PASSWORD` con il valore. *saphanaPassword*

2. Nella console AWS Glue, crea una connessione seguendo i passaggi riportati di seguitothe section called "Aggiungere una AWS Glue connessione". Dopo aver creato la connessione, mantieni il nome della connessione*connectionName*, per utilizzi futuri in AWS Glue.

   - In Tipo di connessione, seleziona SAP HANA.
   - Quando fornisci l'URL SAP HANA, fornisci l'URL per la tua istanza.

     SAP HANA JDBC URLs sono nel formato `jdbc:sap://`*saphanaHostname*`:`*saphanaPort*`/?`*databaseName*`=`*saphanaDBname*`,`*Paramete*

     AWS Glue richiede i seguenti parametri URL JDBC:

     - `databaseName`: un database predefinito in SAP HANA a cui connettersi.
   - Quando selezioni un AWS segreto, fornisci. *secretName*

Dopo aver creato una connessione AWS Glue SAP HANA, dovrai eseguire i seguenti passaggi prima di eseguire il lavoro AWS Glue:

- Concedi al ruolo IAM associato al tuo lavoro AWS Glue il permesso di lettura*secretName*.
- Nella configurazione del lavoro AWS Glue, fornisci *connectionName* una connessione di rete aggiuntiva.

Lettura da tabelle SAP HANA

Prerequisiti:

- Una tabella SAP HANA da cui si desidera leggere. Avrai bisogno delle informazioni di identificazione per la tabella.

  Una tabella può essere specificata con un nome di tabella SAP HANA e di schema, nel modulo *schemaName*.*tableName*. Il nome dello schema e il separatore "." non sono necessari se la tabella si trova nello schema predefinito, "pubblico". Chiama questo*tableIdentifier*. Il database viene fornito come parametro URL JDBC in `connectionName`.
- Una connessione AWS Glue SAP HANA configurata per fornire informazioni di autenticazione. Completa i passaggi della procedura precedente, Per configurare una connessione a SAP HANA per configurare le informazioni di autenticazione. Avrai bisogno del nome della connessione AWS Glue,*connectionName*.

Esempio:

```
saphana_read_table = glueContext.create_dynamic_frame.from_options(
    connection_type="saphana",
    connection_options={
        "connectionName": "connectionName",
        "dbtable": "tableIdentifier",
    }
)
```

Puoi anche fornire una query SQL SELECT per filtrare i risultati restituiti al tuo DynamicFrame. Sarà necessario configurare `query`.

Esempio:

```
saphana_read_query = glueContext.create_dynamic_frame.from_options(
    connection_type="saphana",
    connection_options={
        "connectionName": "connectionName",
        "query": "query"
    }
)
```

Scrittura su tabelle SAP HANA

Questo esempio scrive informazioni da un sistema esistente DynamicFrame *dynamicFrame* a SAP HANA. Se la tabella contiene già informazioni, AWS Glue genererà un errore.

Prerequisiti:

- Una tabella SAP HANA su cui scrivere.

  Una tabella può essere specificata con un nome di tabella SAP HANA e di schema, nel modulo *schemaName*.*tableName*. Il nome dello schema e il separatore "." non sono necessari se la tabella si trova nello schema predefinito, "pubblico". Chiama questo*tableIdentifier*. Il database viene fornito come parametro URL JDBC in `connectionName`.
- Informazioni di autenticazione SAP HANA. Completa i passaggi della procedura precedente, Per configurare una connessione a SAP HANA per configurare le informazioni di autenticazione. Avrai bisogno del nome della connessione AWS Glue,*connectionName*.

Esempio:

```
options = {
    "connectionName": "connectionName",
    "dbtable": 'tableIdentifier'
}

    saphana_write = glueContext.write_dynamic_frame.from_options(
        frame=dynamicFrame,
        connection_type="saphana",
        connection_options=options
)
```

Indicazioni di riferimento alle opzioni di connessione a SAP HANA

- `connectionName`: obbligatorio. Utilizzato per la lettura/scrittura. Il nome di una connessione AWS Glue SAP HANA configurata per fornire informazioni di autenticazione e di rete al metodo di connessione.
- `databaseName`: utilizzato per la lettura/scrittura. Valori validi: nomi dei database in SAP HANA. Nome del database a cui connettersi.
- `dbtable` — Richiesto per la scrittura, richiesto per la lettura a meno che non `query` sia fornito. Utilizzato per la lettura/scrittura. Valori validi: contenuto di una clausola SAP HANA SQL FROM. Identifica una tabella in SAP HANA a cui connettersi. È inoltre possibile fornire un codice SQL diverso dal nome della tabella, ad esempio una sottoquery. Per ulteriori informazioni, consulta la clausola From nella documentazione di SAP HANA.

- `query`: utilizzato per la lettura. Una query SAP HANA SQL SELECT che definisce cosa recuperare durante la lettura da SAP HANA.

### Connessioni Snowflake

Puoi usare AWS Glue for Spark per leggere e scrivere su tabelle in Snowflake in AWS Glue 4.0 e versioni successive. È possibile leggere da Snowflake con una query SQL. Puoi connetterti a Snowflake utilizzando uno dei tre metodi seguenti: autenticazione di base (usando nome utente e password), autenticazione o OAuth autenticazione tramite coppia di chiavi. Puoi fare riferimento alle credenziali Snowflake archiviate tramite le connessioni Gestione dei segreti AWS AWS Glue Data. Connessione dati Le credenziali Snowflake per AWS Glue for Spark vengono archiviate separatamente dalle credenziali Snowflake di Data Catalog per i crawler. È necessario scegliere un tipo di connessione `SNOWFLAKE` e non un tipo di connessione `JDBC`configurato per la connessione a Snowflake.

Per ulteriori informazioni su Snowflake, consulta il sito Web di Snowflake. Per ulteriori informazioni su Snowflake on AWS, consulta Snowflake Data Warehouse on Amazon Web Services.

#### Configurazione delle connessioni Snowflake

Non ci sono AWS prerequisiti per la connessione ai database Snowflake disponibili tramite Internet.

Facoltativamente, puoi eseguire la seguente configurazione per gestire le credenziali di connessione con AWS Glue.

Per gestire le credenziali di connessione con AWS Glue

1. Nel Gestione dei segreti AWS, crea un segreto usando le tue credenziali Snowflake. Per creare un segreto in Secrets Manager, segui il tutorial disponibile in Crea un Gestione dei segreti AWS segreto nella Gestione dei segreti AWS documentazione. Dopo aver creato il segreto, mantieni il nome del segreto *`secretName`* per il passaggio successivo.

   - Per OAuth l'autenticazione:
     - Quando selezionate coppie chiave/valore, create una coppia per *`snowflakeUser`* con la chiave `sfUser`
     - Quando selezionate coppie chiave/valore, create una coppia per con la chiave *`OAUTH_CLIENT_SECRET`* `USER_MANAGED_CLIENT_APPLICATION_CLIENT_SECRET`
   - Per l'autenticazione tramite coppia di chiavi:

    - Quando selezionate coppie chiave/valore, create una coppia per con la chiave *snowflakeUser* `sfUser`
    - Quando selezionate coppie chiave/valore, create una coppia per con la chiave *private key* `pem_private_key`
  - Per l'autenticazione di base:
    - Quando selezionate coppie chiave/valore, create una coppia per *snowflakeUser* con la chiave `USERNAME`
    - Quando selezionate coppie chiave/valore, create una coppia per con la chiave *snowflakePassword* `PASSWORD`
  - Quando si selezionano le coppie chiave/valore, è possibile fornire la chiave `sfWarehouse` al proprio warehouse Snowflake.
  - Quando si selezionano coppie chiave/valore, è possibile indicare proprietà di connessione Snowflake aggiuntive utilizzando i nomi delle proprietà Spark corrispondenti come chiavi. Tra le proprietà supportate vi sono:
    - `sfDatabase`: nome del database Snowflake
    - `sfSchema`: nome dello schema Snowflake
    - `sfRole`: nome del ruolo Snowflake
2. Nella console AWS Glue Studio, crea una connessione scegliendo Connessioni dati, quindi Crea connessione. Seguire la procedura guidata di connessione per completare il processo:
   - Al momento di scegliere un'Origine dati, selezionare Snowflake, poi Avanti.
   - Inserire i dettagli della connessione, come l'host e la porta. Al momento di inserire l'URL Snowflake dell'host, indicare l'URL dell'istanza Snowflake. Generalmente, l'URL utilizza un nome host nel modulo *account_identifier*`.snowflakecomputing.com`. Tuttavia, il formato dell'URL può variare a seconda del tipo di account Snowflake (ad esempio AWS, Azure o Snowflake-hosted).
   - Al momento di selezionare il ruolo di servizio IAM, scegliere un'opzione dal menu a discesa. Questo è il ruolo IAM del tuo account che verrà utilizzato per accedere Gestione dei segreti AWS e assegnare l'IP se viene specificato VPC.
   - Quando selezioni un AWS segreto, fornisci. *secretName*
3. Nel passaggio successivo della procedura guidata, impostare le proprietà della connessione Snowflake.
4. Nel passaggio finale della procedura guidata, rivedere le impostazioni e completare il processo per creare la connessione.

Per Snowflake ospitato su AWS un Amazon VPC, potresti aver bisogno di quanto segue:

- Per Snowflake è necessaria una configurazione Amazon VPC appropriata. Per ulteriori informazioni su come configurare il tuo Amazon VPC, consultare la sezione AWS PrivateLink & Snowflake nella documentazione di Snowflake.
- Avrai bisogno di una configurazione Amazon VPC appropriata per AWS Glue. the section called "Configurazione degli endpoint VPC (AWS PrivateLink) per AWS Glue ".
- Dovrai creare una connessione AWS Glue Data Catalog che fornisca le informazioni di connessione Amazon VPC (oltre all'ID di un Gestione dei segreti AWS segreto che definisce le tue credenziali di sicurezza Snowflake). L'URL cambierà durante l'utilizzo AWS PrivateLink, come descritto nella documentazione Snowflake collegata in un elemento precedente.
- È necessario che la configurazione del processo includa la connessione a Catalogo dati come Connessione di rete aggiuntiva.

Lettura dalle tabelle Snowflake

Prerequisiti: una tabella Snowflake da cui desideri leggere. Avrai bisogno del nome della tabella Snowflake,. *tableName* Se l'utente Snowflake non dispone di un set di namespace predefinito, saranno necessari il nome del database Snowflake e il nome dello schema. *databaseName* *schemaName* Inoltre, se il tuo utente Snowflake non dispone di un set di warehouse predefinito, avrai bisogno di un nome di warehouse. *warehouseName* Per selezionare la connessione di rete aggiuntiva a cui connettersi, verrà `connectionName` utilizzato il parametro.

```
snowflake_read = glueContext.create_dynamic_frame.from_options(
  connection_type="snowflake",
  connection_options={
        "connectionName": "connectionName",
        "dbtable": "tableName",
        "sfDatabase": "databaseName",
        "sfSchema": "schemaName",
        "sfWarehouse": "warehouseName",
    }
)
```

Inoltre, puoi utilizzare i parametri `autopushdown` e `query` per leggere una parte di una tabella Snowflake. Questo può essere molto più efficiente rispetto al filtraggio dei risultati dopo che sono stati caricati in Spark. Prendiamo in esame un esempio in cui tutte le vendite sono archiviate nella stessa tabella, ma è necessario analizzare solo le vendite di un determinato negozio nei giorni festivi. Se tali

informazioni sono archiviate nella tabella, è possibile utilizzare il predicato pushdown per recuperare i risultati come segue:

```
snowflake_node = glueContext.create_dynamic_frame.from_options(
    connection_type="snowflake",
    connection_options={
        "autopushdown": "on",
        "query": "select * from sales where store='1' and IsHoliday='TRUE'",
        "connectionName": "snowflake-glue-conn",
        "sfDatabase": "databaseName",
        "sfSchema": "schemaName",
        "sfWarehouse": "warehouseName",
    }
)
```

Scrittura su tabelle Snowflake

Prerequisiti: un database Snowflake su cui scrivere. Avrai bisogno di un nome di tabella attuale o desiderato,*tableName*. Se l'utente Snowflake non dispone di un set di namespace predefinito, saranno necessari il nome del database Snowflake e il nome dello schema. *databaseName* *schemaName* Inoltre, se il tuo utente Snowflake non dispone di un set di warehouse predefinito, avrai bisogno di un nome di warehouse. *warehouseName* Per selezionare la connessione di rete aggiuntiva a cui connettersi, verrà `connectionName` utilizzato il parametro.

```
glueContext.write_dynamic_frame.from_options(
    connection_type="snowflake",
    connection_options={
        "connectionName": "connectionName",
        "dbtable": "tableName",
        "sfDatabase": "databaseName",
        "sfSchema": "schemaName",
        "sfWarehouse": "warehouseName",
    },
)
```

Indicazioni di riferimento alle opzioni di connessione a Snowflake

Il tipo di connessione Snowflake accetta le seguenti opzioni di connessione:

È possibile recuperare alcuni dei parametri di questa sezione da una connessione AWS Glue (`sfUrlsfUser`,,`sfPassword`), nel qual caso non è necessario fornirli. È possibile farlo fornendo il parametro `connectionName`.

È possibile recuperare i parametri di connessione dai Gestione dei segreti AWS segreti utilizzando il `secretId` parametro. Quando si utilizza Secrets Manager, è possibile recuperare automaticamente le seguenti proprietà Spark, se presenti nel segreto:

- `sfUser` (utilizzando la chiave `USERNAME` o `sfUser`)
- `sfPassword`(utilizzando la chiave `PASSWORD` o`sfPassword`, quando si utilizza l'autenticazione di base)
- `sfWarehouse` (utilizzando la chiave `sfWarehouse`)
- `sfDatabase` (utilizzando la chiave `sfDatabase`)
- `sfSchema` (utilizzando la chiave `sfSchema`)
- `sfRole` (utilizzando la chiave `sfRole`)
- `pem_private_key`(utilizzando la chiave`pem_private_key`, quando si utilizza l'autenticazione a coppia di chiavi)
- `USER_MANAGED_CLIENT_APPLICATION_CLIENT_SECRET`(quando si utilizza l'autenticazione OAuth )

Ordine di precedenza delle proprietà: quando la stessa proprietà viene specificata in più posizioni, AWS Glue utilizza il seguente ordine di precedenza (dal più alto al più basso):

1. Opzioni di connessione indicate esplicitamente nel codice del processo
2. Proprietà di connessione Glue
3. Gestione dei segreti AWS valori segreti (quando `secretId` specificato)
4. Impostazioni predefinite dell'utente Snowflake

Per la connessione a Snowflake generalmente vengono utilizzati i seguenti parametri.

- `sfDatabase`: obbligatorio se in Snowflake non è impostato un valore predefinito per l'utente. Utilizzato per la lettura/scrittura. Il database da utilizzare per la sessione dopo la connessione.
- `sfSchema`: obbligatorio se in Snowflake non è impostato un valore predefinito per l'utente. Utilizzato per la lettura/scrittura. Lo schema da utilizzare per la sessione dopo la connessione.
- `sfWarehouse`: obbligatorio se in Snowflake non è impostato un valore predefinito per l'utente. Utilizzato per la lettura/scrittura. Il warehouse virtuale predefinito da utilizzare per la sessione dopo la connessione.

- `sfRole`: obbligatorio se in Snowflake non è impostato un valore predefinito per l'utente. Utilizzato per la lettura/scrittura. Il ruolo di sicurezza predefinito da utilizzare per la sessione dopo la connessione.
- `sfUrl`: (obbligatorio) utilizzato per la lettura/scrittura. Specifica il nome host del tuo account nel seguente formato: *`account_identifier`*`.snowflakecomputing.com`. Per ulteriori informazioni sugli identificatori di account, consulta la pagina Account Identifiers nella documentazione di Snowflake.
- `sfUser`: (obbligatorio) utilizzato per la lettura/scrittura. Il nome di accesso per l'utente Snowflake.
- `sfPassword`— (Richiesto quando si utilizza l'autenticazione di base) Utilizzato per la lettura/scrittura. La password per l'utente Snowflake.
- `dbtable`: obbligatorio quando si lavora con tabelle complete. Utilizzato per la lettura/scrittura. Il nome della tabella da leggere o la tabella in cui vengono scritti i dati. Durante la lettura, vengono recuperate tutte le colonne e i record.
- `pem_private_key`— (Richiesto quando si utilizza l'autenticazione a coppia di chiavi) Utilizzato per la lettura/scrittura. Una stringa di chiave privata non crittografata con codifica b64. La chiave privata per l'utente Snowflake. È comune copiare tale chiave da un file PEM. Per ulteriori informazioni, consulta Autenticazione e rotazione delle coppie di chiavi nella documentazione di Snowflake.
- `USER_MANAGED_CLIENT_APPLICATION_CLIENT_SECRET`— (Richiesto quando si utilizza OAuth l'autenticazione) Utilizzato per operazioni di lettura e scrittura. Questo valore corrisponde a OAUTH_CLIENT_SECRET, che può essere ottenuto dall'integrazione di sicurezza Snowflake configurata per abilitare l'autenticazione basata sull'account. OAuth Per maggiori dettagli, consulta la documentazione sulla configurazione dell'integrazione di OAuth sicurezza Snowflake - Configura Snowflake per client personalizzati. OAuth
- `query`: obbligatorio durante la lettura con una query. Utilizzato per la lettura. La query esatta (istruzione `SELECT`) da eseguire

Le seguenti opzioni vengono utilizzate per configurare comportamenti specifici durante il processo di connessione a Snowflake.

- `preactions`: utilizzato per la lettura/scrittura. Valori validi: elenco di istruzioni SQL separato da punto e virgola in formato stringa. Le istruzioni SQL vengono eseguite prima del trasferimento dei dati tra AWS Glue e Snowflake. Se un'istruzione contiene `%s`, `%s` viene sostituito con il nome della tabella a cui si fa riferimento per l'operazione.

- `postactions`: utilizzato per la lettura/scrittura. Le istruzioni SQL vengono eseguite dopo il trasferimento dei dati tra AWS Glue e Snowflake. Se un'istruzione contiene `%s`, `%s` viene sostituito con il nome della tabella a cui si fa riferimento per l'operazione.
- `autopushdown`: valore predefinito: `"on"`. Valori validi: `"on"`, `"off"`. Questo parametro controlla se il pushdown automatico delle query è abilitato. Se il pushdown è abilitato, quando su Spark viene eseguita una query, se una parte di essa può essere "trasferita" al server Snowflake, viene sottoposta a pushdown. Ciò migliora le prestazioni di alcune query. Per sapere se la tua query può essere spostata verso il basso, consulta la sezione Pushdown nella documentazione di Snowflake.

Inoltre, alcune delle opzioni disponibili sul connettore Snowflake Spark potrebbero essere supportate in Glue. AWS Per ulteriori informazioni sulle opzioni disponibili sul connettore Snowflake Spark, consulta la sezione Setting Configuration Options for the Connector nella documentazione di Snowflake.

Metodi di autenticazione per Snowflake

AWS Glue supporta i seguenti metodi di autenticazione per la connessione a Snowflake:

- Autenticazione di base: Fornitura `sfUser` e parametri`sfPassword`.
- Autenticazione con coppia di chiavi: Fornitura `sfUser` e `pem_private_key` parametri. Quando si utilizza l'autenticazione chiave-coppia, il parametro `sfPassword` non è richiesto.
- OAuth autenticazione: il connettore Snowflake supporta il tipo di concessione AUTHORIZATION_CODE per richiedere l'accesso ai dati Snowflake. Questo tipo di concessione viene definito «a 3 gambe OAuth», in quanto prevede il reindirizzamento degli utenti a un server di autorizzazione di terze parti dove possono autenticare e approvare l'accesso. Questo metodo viene utilizzato quando si crea una connessione tramite la AWS Glue Console.
  - Prerequisito: per utilizzare questo metodo di autenticazione, assicuratevi che la seguente configurazione sia completa:
    - Configura Snowflake OAuth per un client personalizzato seguendo la documentazione ufficiale di Snowflake: Configura Snowflake per client personalizzati. OAuth
    - Imposta l'URI di reindirizzamento corretto durante la creazione dell'integrazione di sicurezza Snowflake. Ad esempio: se state creando la connessione nella regione DUB (eu-west-1), l'URI di reindirizzamento dovrebbe essere: `https://eu-west-1.console.aws.amazon.com/gluestudio/oauth`
    - Dopo aver creato l'integrazione di sicurezza, conservate le seguenti informazioni da utilizzare durante la creazione della connessione Glue:

    - OAUTH_CLIENT_ID: questo valore deve essere fornito come ID client dell'applicazione User Managed Client nella pagina di creazione della connessione Glue.
    - OAUTH_CLIENT_SECRET: questo valore deve essere memorizzato nel Secret utilizzato per la connessione, sotto la chiave AWS USER_MANAGED_CLIENT_APPLICATION_CLIENT_SECRET.
- OAuth Ambiti: (Facoltativo) Definisce le autorizzazioni o i livelli di accesso specifici richiesti all'account Snowflake. Ad esempio, un ambito potrebbe limitare l'accesso a una particolare risorsa o operazione.
    - Questo valore può essere specificato nel formato seguente: `session:role:Snowflake_Role_Name`
    - Ad esempio: `session:role:ANALYST_ROLE`
- URL del codice di autorizzazione: (obbligatorio) L'endpoint a cui l'utente viene reindirizzato per accedere e concedere l'autorizzazione.
    - Ad esempio: `https://host/oauth/authorize`
- URL del token di autorizzazione: (obbligatorio) L'endpoint utilizzato per scambiare il codice di autorizzazione con un token di accesso.
    - Ad esempio: `https://host/oauth/token-request`
- User Managed Client Application Client Id: (Obbligatorio) L'identificatore univoco per l'applicazione OAuth client registrata in Snowflake
- AWS Segreto: (Obbligatorio) Si riferisce a un segreto di AWS Secrets Manager contenente le seguenti coppie chiave-valore:
    - sfUser - Il nome utente di Snowflake
    - USER_MANAGED_CLIENT_APPLICATION_CLIENT_SECRET - Il segreto del client associato all'applicazione client OAuth

Tutti e tre i metodi di autenticazione sono completamente supportati e possono essere configurati utilizzando qualsiasi combinazione di opzioni di connessione, connessioni Glue o Gestione dei segreti AWS segreti.

Limitazioni del connettore Snowflake

La connessione a Snowflake con AWS Glue for Spark è soggetta alle seguenti limitazioni.

- Questo connettore non supporta i segnalibri di processo. Per ulteriori informazioni sui segnalibri di processo, consultare the section called "Monitoraggio dei dati elaborati mediante segnalibri di processo".
- Questo connettore non supporta la lettura e la scrittura di Snowflake tramite tabelle nel AWS Glue Data Catalog utilizzando i `create_dynamic_frame.from_catalog` metodi and. `write_dynamic_frame.from_catalog`
- Questo connettore supporta l'autenticazione di base, l'autenticazione tramite coppia di chiavi e l'autenticazione. OAuth Altri metodi di autenticazione (come SAML) non sono attualmente supportati.
- Questo connettore non è supportato nei processi di flussi di dati.
- Questo connettore supporta le query basate su istruzioni `SELECT` per il recupero di informazioni, ad esempio con il parametro `query`. Altri tipi di query (ad esempio istruzioni DML, `SHOW` o `DESC`) non sono supportati.
- Snowflake limita la dimensione del testo della query (ad esempio istruzioni SQL) inviato tramite i client Snowflake a 1 MB per istruzione. Per ulteriori informazioni, consulta la pagina Limits on Query Text Size.

### Connessioni Teradata Vantage

È possibile utilizzare AWS Glue for Spark per leggere e scrivere su tabelle esistenti in Teradata Vantage in AWS Glue 4.0 e versioni successive. È possibile definire cosa leggere da Teradata con una query SQL. È possibile connettersi a Teradata utilizzando le credenziali di nome utente e password memorizzate tramite Gestione dei segreti AWS una connessione AWS Glue.

Per ulteriori informazioni su Teradata, consulta la documentazione di Teradata.

#### Configurazione delle connessioni Teradata

Per connetterti a Teradata da AWS Glue, dovrai creare e archiviare le tue credenziali Teradata in un luogo Gestione dei segreti AWS segreto, quindi associare quel segreto a una connessione Glue Teradata. AWS Se la tua istanza Teradata si trova in un Amazon VPC, dovrai anche fornire opzioni di rete alla tua connessione AWS Glue Teradata.

Per connettersi a Teradata da AWS Glue, potrebbero essere necessari alcuni prerequisiti:

- Se accedi al tuo ambiente Teradata tramite Amazon VPC, configura Amazon VPC per consentire al tuo job AWS Glue di comunicare con l'ambiente Teradata. Sconsigliamo l'accesso all'ambiente Teradata tramite la rete Internet pubblica.

In Amazon VPC, identifica o crea un VPC, una sottorete e un gruppo di sicurezza che AWS Glue utilizzerà durante l'esecuzione del lavoro. Inoltre, assicurati che Amazon VPC sia configurato per consentire il traffico di rete tra l'istanza Teradata e questa posizione. Il tuo processo dovrà stabilire una connessione TCP con la tua porta del client Teradata. Per ulteriori informazioni sulle porte Teradata, consulta la documentazione di Teradata.

In base al layout di rete, la connettività VPC sicura potrebbe richiedere modifiche ad Amazon VPC e ad altri servizi di rete. Per ulteriori informazioni sulla AWS connettività, consulta le opzioni di AWS connettività nella documentazione di Teradata.

Per configurare una connessione AWS Glue Teradata:

1. Nella configurazione Teradata, identifica o crea un utente e una password con cui AWS Glue si connetterà, *`teradataUser`* e. *`teradataPassword`* Per ulteriori informazioni, consulta Vantage Security Overview nella documentazione di Teradata.
2. Nel Gestione dei segreti AWS, crea un segreto usando le tue credenziali Teradata. Per creare un segreto in Secrets Manager, segui il tutorial disponibile in Crea un Gestione dei segreti AWS segreto nella Gestione dei segreti AWS documentazione. Dopo aver creato il segreto, mantieni il nome del segreto *`secretName`* per il passaggio successivo.
   - Quando selezionate le coppie chiave/valore, create una coppia per la chiave `user` con il valore. *`teradataUsername`*
   - Quando selezionate le coppie chiave/valore, create una coppia per la chiave `password` con il valore. *`teradataPassword`*
3. Nella console AWS Glue, crea una connessione seguendo i passaggi riportati di seguitothe section called "Aggiungere una AWS Glue connessione". Dopo aver creato la connessione, mantieni il nome della connessione per il passaggio successivo. *`connectionName`*
   - In Tipo di connessione, seleziona Snowflake.
   - Quando fornisci JDBC URL, fornisci l'URL per la tua istanza. Puoi anche codificare determinati parametri di connessione, separati da virgole, nel tuo URL JDBC. L'URL deve rispettare il seguente formato: `jdbc:teradata://`*`teradataHostname`*`/`*`ParameterName`*`=`*`ParameterValue`*`,`*`ParameterName`*

     I parametri URL supportati includono:

     - `DATABASE`: nome del database sull'host a cui accedere per impostazione predefinita.

    - DBS_PORT: la porta del database, utilizzata con una porta non standard.
    - Quando selezioni un tipo di credenziale, seleziona Gestione dei segreti AWS, quindi imposta AWS Segreto *secretName* su.
4. Nelle seguenti situazioni, potresti aver bisogno di una configurazione aggiuntiva:
    - Per le istanze Teradata ospitate su AWS un Amazon VPC
        - Dovrai fornire le informazioni di connessione Amazon VPC alla connessione AWS Glue che definisce le tue credenziali di sicurezza Teradata. Durante la creazione o l'aggiornamento della connessione, imposta VPC, sottorete e Gruppi di sicurezza nelle opzioni di rete.

Dopo aver creato una connessione AWS Glue Teradata, dovrai eseguire i seguenti passaggi prima di chiamare il metodo di connessione.

- Concedi al ruolo IAM associato al tuo lavoro AWS Glue il permesso di lettura*secretName*.
- Nella configurazione del lavoro AWS Glue, fornisci *connectionName* una connessione di rete aggiuntiva.

Lettura da Teradata

Prerequisiti:

- Una tabella Teradata da cui si desidera leggere. Avrai bisogno del nome della tabella,*tableName*.
- Una connessione AWS Glue Teradata configurata per fornire informazioni di autenticazione. Completa i passaggi Per configurare una connessione a Teradata per configurare le informazioni di autenticazione. Avrai bisogno del nome della connessione AWS Glue,*connectionName*.

Esempio:

```
teradata_read_table = glueContext.create_dynamic_frame.from_options(
    connection_type="teradata",
    connection_options={
        "connectionName": "connectionName",
        "dbtable": "tableName"
    }
)
```

Puoi anche fornire una query SQL SELECT per filtrare i risultati restituiti al tuo DynamicFrame. Sarà necessario configurare `query`.

Esempio:

```
teradata_read_query = glueContext.create_dynamic_frame.from_options(
    connection_type="teradata",
    connection_options={
        "connectionName": "connectionName",
        "query": "query"
    }
)
```

Scrittura su tabelle Teradata

Prerequisiti: una tabella Teradata su cui scrivere,. *`tableName`* È necessario creare la tabella prima di chiamare il metodo di connessione.

Esempio:

```
teradata_write = glueContext.write_dynamic_frame.from_options(
    connection_type="teradata",
    connection_options={
        "connectionName": "connectionName",
        "dbtable": "tableName"
    }
)
```

Indicazioni di riferimento alle opzioni di connessione a Teradata

- `connectionName`: obbligatorio. Utilizzato per la lettura/scrittura. Il nome di una connessione AWS Glue Teradata configurata per fornire informazioni di autenticazione e di rete al metodo di connessione utilizzato.
- `dbtable` — Richiesto per la scrittura, richiesto per la lettura a meno che non `query` sia fornito. Utilizzato per la lettura/scrittura. Il nome di una tabella con cui interagirà il metodo di connessione.
- `query`: utilizzato per la lettura. Una query SELECT SQL che definisce cosa recuperare durante la lettura da Teradata.

## Connessioni Teradata Vantage NOS

La connessione Teradata NOS (Native Object Store) è una nuova connessione per Teradata Vantage che sfrutta la query Teradata WRITE_NOS per leggere dalle tabelle esistenti e la query READ_NOS per scrivere nelle tabelle. Queste query usano Amazon S3 come directory di staging e, pertanto, il connettore Teradata NOS è più veloce del connettore Teradata esistente (basato su JDBC), specialmente nella gestione di grandi quantità di dati.

È possibile utilizzare la connessione Teradata NOS in Spark AWS Glue per leggere e scrivere su tabelle esistenti in Teradata Vantage nella versione 5.0 e successive. AWS Glue È possibile definire cosa leggere da Teradata con una query SQL. È possibile connettersi a Teradata utilizzando le credenziali di nome utente e password memorizzate tramite una connessione. Gestione dei segreti AWS AWS Glue

Per ulteriori informazioni su Teradata, consulta la documentazione di Teradata.

Argomenti


- Creazione di una connessione Teradata NOS
- Lettura da tabelle Teradata
- Scrittura su tabelle Teradata
- Indicazioni di riferimento alle opzioni di connessione a Teradata
- Fornisci opzioni nell'interfaccia utente di AWS Glue Visual ETL


### Creazione di una connessione Teradata NOS

Per connettersi a Teradata NOS da AWS Glue, è necessario creare e archiviare le credenziali Teradata in un luogo Gestione dei segreti AWS segreto, quindi associare tale segreto a una connessione Teradata NOS. AWS Glue Se l'istanza Teradata si trova in un Amazon VPC, sarà inoltre necessario fornire opzioni di rete alla connessione Teradata NOS. AWS Glue

Prerequisiti:

- Se accedi al tuo ambiente Teradata tramite Amazon VPC, configura Amazon VPC per consentire al tuo AWS Glue job di comunicare con l'ambiente Teradata. Sconsigliamo l'accesso all'ambiente Teradata tramite la rete Internet pubblica.
- In Amazon VPC, identifica o crea un VPC, una sottorete e un gruppo di sicurezza da utilizzare durante l'esecuzione del AWS Glue lavoro. Inoltre, assicurati che Amazon VPC sia configurato per consentire il traffico di rete tra l'istanza Teradata e questa posizione. Il tuo processo dovrà stabilire

una connessione TCP con la tua porta del client Teradata. Per maggiori informazioni sulle porte Teradata, consulta i Gruppi di sicurezza per Teradata Vantage.

- In base al layout di rete, la connettività VPC sicura potrebbe richiedere modifiche ad Amazon VPC e ad altri servizi di rete. Per ulteriori informazioni sulla AWS connettività, consulta Opzioni di AWS connettività nella documentazione di Teradata.

Per configurare una connessione AWS Glue Teradata NOS:

1. Nella configurazione Teradata, identifica o crea un *teradataUsername* e *teradataPassword* AWS Glue con cui ti connetterai. Per ulteriori informazioni, consulta Panoramica sulla sicurezza di Vantage nella documentazione di Teradata.
2. Nel Gestione dei segreti AWS, crea un segreto usando le tue credenziali Teradata. Per creare un segreto in Gestione dei segreti AWS, segui il tutorial disponibile in Creare un Gestione dei segreti AWS segreto nella documentazione. Gestione dei segreti AWS Dopo aver creato il segreto, mantieni il nome del segreto *secretName* per il passaggio successivo.
   - Quando selezioni Key/value le coppie, crea una coppia per la chiave USERNAME con il valore*teradataUsername*.
   - Quando selezionate Key/value le coppie, create una coppia per la chiave PASSWORD con il valore*teradataPassword*.
3. Nella AWS Glue console, crea una connessione seguendo i passaggi descritti in Aggiungere una AWS Glue connessione. Dopo aver creato la connessione, mantieni il nome della connessione per il passaggio successivo. *connectionName*
   - In Tipo di connessione, seleziona Teradata Vantage NOS.
   - Quando fornisci JDBC URL, fornisci l'URL per la tua istanza. Puoi anche codificare determinati parametri di connessione, separati da virgole, nel tuo URL JDBC. L'URL deve essere conforme al seguente formato: `jdbc:teradata://teradataHostname/ParameterName=ParameterValue,ParameterName=ParameterValue` .
   - I parametri URL supportati includono:
     - `DATABASE`: nome del database sull'host a cui accedere per impostazione predefinita.
     - `DBS_PORT`: la porta del database, utilizzata con una porta non standard.
   - Quando selezioni un tipo di credenziale, seleziona Gestione dei segreti AWS, quindi imposta AWS Segreto su. *secretName*
4. Nelle seguenti situazioni, potresti aver bisogno di una configurazione aggiuntiva:

- Per le istanze Teradata ospitate su AWS un Amazon VPC, dovrai fornire le informazioni di connessione Amazon VPC alla connessione che definisce le tue credenziali di sicurezza Teradata AWS Glue . Durante la creazione o l'aggiornamento della connessione, imposta VPC, sottorete e Gruppi di sicurezza nelle opzioni di rete.

Dopo aver creato una connessione AWS Glue Teradata Vantage NOS, dovrai eseguire i seguenti passaggi prima di chiamare il metodo di connessione.

1. Concedi il permesso di lettura al ruolo IAM associato al tuo AWS Glue lavoro. *secretName*
2. Nella configurazione del tuo AWS Glue lavoro, fornisci *connectionName* come connessione di rete aggiuntiva in Connessioni.

Lettura da tabelle Teradata

Prerequisiti:

- Una tabella Teradata da cui si desidera leggere. Avrai bisogno del nome della tabella,*tableName*.
- L'ambiente Teradata ha accesso in scrittura al percorso Amazon S3 specificato `staging_fs_url` dall'opzione,. *stagingFsUrl*
- Il ruolo IAM associato al AWS Glue job ha accesso in scrittura alla posizione Amazon S3 specificata dall'`staging_fs_url`opzione.
- Una connessione AWS Glue Teradata NOS configurata per fornire informazioni di autenticazione. Completa i passaggi descritti in Per configurare una connessione AWS Glue Teradata NOS: per configurare le tue informazioni di autenticazione. Avrai bisogno del nome della connessione,. AWS Glue *connectionName*

Esempio:

```
teradata_read_table = glueContext.create_dynamic_frame.from_options(
    connection_type= "teradatanos",
    connection_options={
        "connectionName": "connectionName",
        "dbtable": "tableName",
        "staging_fs_url": "stagingFsUrl"
    }
)
```

Puoi anche fornire una query SQL SELECT, per filtrare i risultati restituiti al tuo DynamicFrame. Dovrai configurare la query. Se configuri sia dbTable che la query, il connettore non riesce a leggere i dati. Esempio:

```
teradata_read_query = glueContext.create_dynamic_frame.from_options(
    connection_type="teradatanos",
    connection_options={
        "connectionName": "connectionName",
        "query": "query",
        "staging_fs_url": "stagingFsUrl"
    }
)
```

Inoltre, è possibile utilizzare l' DataFrame API Spark per leggere le tabelle Teradata. Esempio:

```
options = {
    "url": "JDBC_URL",
    "dbtable": "tableName",
    "user": "teradataUsername", # or use "username" as key here
    "password": "teradataPassword",
    "staging_fs_url": "stagingFsUrl"
}
teradata_read_table = spark.read.format("teradatanos").option(**options).load()
```

Scrittura su tabelle Teradata

Prerequisiti

- Una tabella Teradata su cui desideri scrivere:. *tableName*
- L'ambiente Teradata ha accesso in lettura alla posizione Amazon S3 specificata `staging_fs_url` dall'opzione,. *stagingFsUrl*
- Il ruolo IAM associato al AWS Glue job ha accesso in scrittura alla posizione Amazon S3 specificata dall'`staging_fs_url`opzione.
- Una connessione AWS Glue Teradata configurata per fornire informazioni di autenticazione. Completa i passaggi descritti in Per configurare una connessione AWS Glue Teradata NOS: per configurare le tue informazioni di autenticazione. Avrai bisogno del nome della connessione,. AWS Glue *connectionName*

  Esempio:

```
teradata_write = glueContext.write_dynamic_frame.from_options(
    frame=dynamicFrame,
    connection_type= "teradatanos",
    connection_options={
        "connectionName": "connectionName",
        "dbtable": "tableName",
        "staging_fs_url": "stagingFsUrl"
    }
)
```

Indicazioni di riferimento alle opzioni di connessione a Teradata

Opzioni di connessione e funzionamento:

- `connectionName`: obbligatorio. Utilizzato per la lettura/scrittura. Il nome di una connessione AWS Glue Teradata configurata per fornire informazioni di autenticazione e di rete al metodo di connessione utilizzato.
- `staging_fs_url`: obbligatorio. Utilizzato per la lettura/scrittura. Una posizione scrivibile in Amazon S3, da usare per i dati scaricati durante la lettura da Teradata e per i dati Parquet da caricare in Redshift durante la scrittura su Teradata. Il bucket S3 deve trovarsi nella stessa regione della regione dei lavori. AWS Glue
- `dbtable` — Richiesto per la scrittura, richiesto per la lettura a meno che non `query` sia fornito. Utilizzato per la lettura/scrittura. Il nome di una tabella con cui interagirà il metodo di connessione.
- `query`: utilizzato per la lettura. Una query SELECT SQL che definisce cosa recuperare durante la lettura da Teradata. Non è possibile passare se è specificata l'opzione `dbtable`.
- `clean_staging_s3_dir`— Facoltativo. Utilizzato per la lettura/scrittura. Se impostato su true, pulisce gli oggetti Amazon S3 di staging dopo una lettura o una scrittura. Il valore di default è true.
- `pre_actions`— Facoltativo. Utilizzato per la scrittura. Elenco separato da punti e virgola dei comandi SQL eseguiti prima del trasferimento dei dati tra Spark e Teradata Vantage.
- `post_actions`— Facoltativo. Utilizzato per la scrittura. Elenco separato da punti e virgola dei comandi SQL eseguiti dopo il trasferimento dei dati tra Spark e Teradata Vantage.
- `truncate`— Facoltativo. Utilizzato per la scrittura. Se è impostato su true, il connettore tronca la tabella durante la scrittura in modalità di sovrascrittura. Se è impostato su false, il connettore elimina la tabella durante la scrittura in modalità di sovrascrittura. Il valore predefinito è false.

- `create_table_script`— Facoltativo. Utilizzato per la scrittura. Istruzione SQL per creare una tabella durante la scrittura su Teradata Vantage. Utile quando si vuole creare una tabella con metadati personalizzati (ad esempio tabella SET o CREATE MULTISET o modificare l'indice primario). Notare che il nome della tabella utilizzato nello script di creazione della tabella deve corrispondere al nome della tabella specificato nell'opzione `dbtable`.
- `partition_size_in_mb`— Facoltativo. Utilizzato per la lettura. Dimensione massima di una partizione Spark in megabyte durante la lettura di oggetti Amazon S3 di staging. Il valore predefinito è 128.

È possibile fornire opzioni avanzate durante la creazione di un nodo Teradata. Queste opzioni sono le stesse disponibili durante la programmazione AWS Glue per gli script Spark.

Per informazioni, consulta the section called "Connessioni Teradata Vantage".

Opzioni di autorizzazione:

Di seguito sono elencate le opzioni utilizzate per fornire le credenziali dell' AWS account utilizzate dal connettore per accedere al bucket di staging Amazon S3. Puoi scegliere di (1) non fornire alcuna opzione di autorizzazione e utilizzare credenziali temporanee generate dal tuo ruolo di AWS Glue esecuzione; oppure (2) fornire un oggetto di autorizzazione creato da `auth_object` te; oppure (3) specificare se si utilizzano credenziali a lungo termine oppure fornire e `aws_access_key_id and aws_secret_access_key` se si utilizzano credenziali temporanee. `aws_access_key aws_secret_access_key aws_session_token`

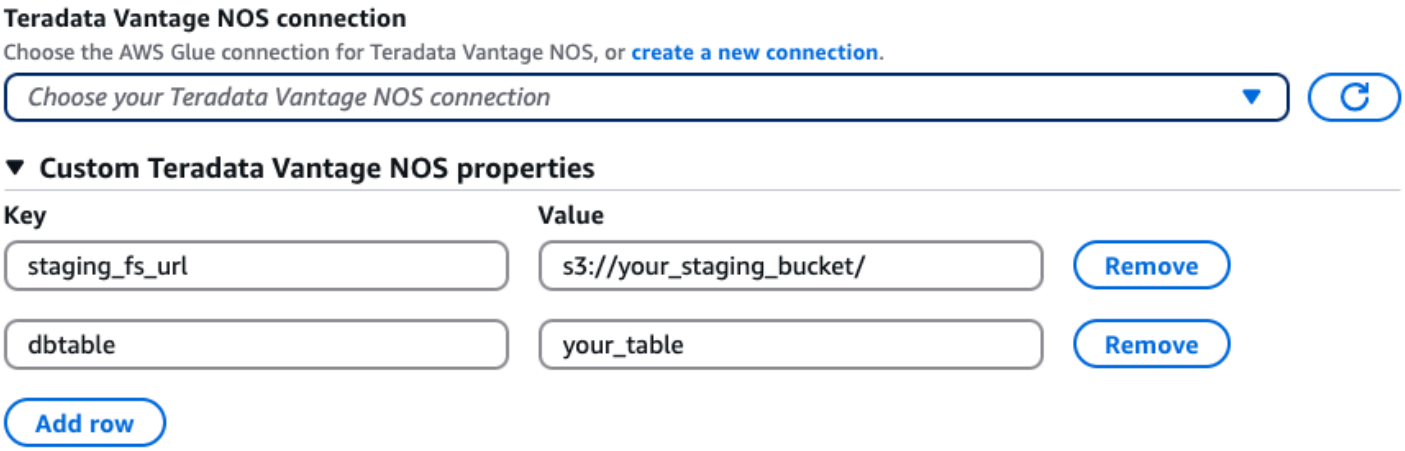
- `auth_object` : Opzionale. Usata per l'accesso al bucket Amazon S3 di staging Amazon S3. Una stringa di oggetti di autorizzazione creata nell'istanza Teradata. Se fornita, il connettore userà questo oggetto di autorizzazione per accedere al bucket Amazon S3 di staging. Se non viene fornita `aws_access_key_id` e `aws_secret_access_key` inoltre non viene fornita, una credenziale temporanea verrà recuperata dal ruolo di AWS Glue esecuzione e utilizzata dal connettore. L' AWS account associato a questo oggetto di autorizzazione deve trovarsi nella stessa regione dei tuoi AWS Glue job e del tuo bucket di staging Amazon S3 o deve essere configurato con cross-account trust.
- `aws_access_key_id` : Opzionale. Usata per l'accesso al bucket Amazon S3 di staging Amazon S3. Parte di una credenziale di sicurezza dell' AWS account. Se `auth_object` non viene fornito e `aws_access_key_id` viene fornito con `aws_secret_access_key`, il connettore li userà per accedere al bucket Amazon S3 di staging. L' AWS account associato a questa chiave di accesso

deve trovarsi nella stessa regione in cui AWS Glue lavori e nel bucket di staging Amazon S3 o deve essere configurato con cross-account trust.

- `aws_secret_access_key` : Opzionale. Usata per l'accesso al bucket Amazon S3 di staging Amazon S3. Parte di una credenziale di sicurezza dell' AWS account. Se `auth_object` non viene fornito e `aws_secret_access_key` viene fornito con `aws_access_key_id`, il connettore li userà per accedere al bucket Amazon S3 di staging. L' AWS account associato a questa chiave segreta deve trovarsi nella stessa regione del tuo AWS Glue job e del tuo bucket di staging Amazon S3 o deve essere configurato con cross-account trust.
- `aws_session_token` : Opzionale. Usata per l'accesso al bucket Amazon S3 di staging Amazon S3. Parte di una credenziale di sicurezza temporanea AWS dell'account. Dovrebbe essere fornita con `aws_access_key_id` e `aws_secret_access_key`.

### Fornisci opzioni nell'interfaccia utente di AWS Glue Visual ETL

È possibile fornire tutte le opzioni sopra indicate nell'interfaccia utente del processo ETL visivo. Per quanto riguarda l'opzione ConnectionName, è necessario selezionarla dall'elenco a discesa delle connessioni Teradata Vantage NOS. Per tutte le altre opzioni, bisogna fornirle tramite le proprietà personalizzate di Teradata Vantage NOS come coppie chiave-valore.




### Connessioni Vertica

Puoi usare AWS Glue for Spark per leggere e scrivere su tabelle in Vertica in AWS Glue 4.0 e versioni successive. È possibile definire cosa leggere da Vertica con una query SQL. Ti connetti a Vertica utilizzando le credenziali di nome utente e password memorizzate Gestione dei segreti AWS tramite una connessione AWS Glue.

Per ulteriori informazioni su Vertica, consulta la documentazione di Vertica.

Configurazione delle connessioni Vertica

Per connetterti a Vertica da AWS Glue, dovrai creare e archiviare le tue credenziali Vertica in un luogo Gestione dei segreti AWS segreto, quindi associare quel segreto a una connessione Vertica Glue. AWS Se la tua istanza Vertica si trova in un Amazon VPC, dovrai anche fornire opzioni di rete alla tua connessione AWS Glue Vertica. Ti servirà un bucket o una cartella Amazon S3 da utilizzare per l'archiviazione temporanea durante la lettura e la scrittura sul database.

Per connetterti a Vertica da AWS Glue, avrai bisogno di alcuni prerequisiti:

- Un bucket o una cartella Amazon S3 da utilizzare per l'archiviazione temporanea durante la lettura e la scrittura sul database, a cui si fa riferimento da. `tempS3Path`

  > **Note**
  >
  > Quando si utilizza Vertica nelle anteprime dei dati di lavoro di AWS Glue, i file temporanei potrebbero non essere rimossi automaticamente da. `tempS3Path` Per garantire la rimozione dei file temporanei, interrompi direttamente la sessione di anteprima dei dati scegliendo Termina sessione nel riquadro Anteprima dei dati.
  > Se non sei in grado di terminare direttamente la sessione di anteprima dei dati, valuta la possibilità di impostare la configurazione del ciclo di vita di Amazon S3 per rimuovere i dati obsoleti. Consigliamo di rimuovere i dati più vecchi di 49 ore, in base al runtime massimo del processo in aggiunta a un margine. Per ulteriori informazioni sulla configurazione del ciclo di vita di Amazon S3, consulta Gestione del ciclo di vita dello storage nella documentazione di Amazon S3.

- Una policy IAM con le autorizzazioni appropriate per il tuo percorso Amazon S3 che puoi associare al tuo ruolo lavorativo in AWS Glue.
- Se la tua istanza Vertica si trova in un Amazon VPC, configura Amazon VPC per consentire al job AWS Glue di comunicare con l'istanza Vertica senza che il traffico attraversi la rete Internet pubblica.

  In Amazon VPC, identifica o crea un VPC, una sottorete e un gruppo di sicurezza che AWS Glue utilizzerà durante l'esecuzione del lavoro. Inoltre, assicurati che Amazon VPC sia configurato per consentire il traffico di rete tra l'istanza Vertica e questa posizione. Il tuo processo dovrà stabilire una connessione TCP con la tua porta del client Vertica, (per impostazione predefinita, 5433). In base al layout della rete, ciò potrebbe richiedere modifiche alle regole del gruppo di sicurezza, alla rete ACLs, ai gateway NAT e alle connessioni peering.

È quindi possibile procedere alla configurazione di AWS Glue per l'uso con Vertica.

Per configurare una connessione a Vertica:

1. Nel Gestione dei segreti AWS, crea un segreto utilizzando le tue credenziali Vertica, e. *verticaUsername* *verticaPassword* Per creare un segreto in Secrets Manager, segui il tutorial disponibile in Crea un Gestione dei segreti AWS segreto nella Gestione dei segreti AWS documentazione. Dopo aver creato il segreto, mantieni il nome del segreto *secretName* per il passaggio successivo.

   - Quando selezionate le coppie chiave/valore, create una coppia per la chiave `user` con il valore. *verticaUsername*
   - Quando selezionate coppie chiave/valore, create una coppia per la chiave `password` con il valore. *verticaPassword*

2. Nella console AWS Glue, crea una connessione seguendo i passaggi riportati di seguitothe section called "Aggiungere una AWS Glue connessione". Dopo aver creato la connessione, mantieni il nome della connessione per il passaggio successivo. *connectionName*

   - In Tipo di connessione, seleziona Vertica.
   - In Host Vertica, fornisci il nome host dell'installazione Vertica.
   - In Porta Vertica, indica la porta tramite cui è disponibile l'installazione di Vertica.
   - Quando selezioni un AWS segreto, fornisci*secretName*.

3. Nelle seguenti situazioni, potresti aver bisogno di una configurazione aggiuntiva:

   - Per le istanze Vertica ospitate su un AWS Amazon VPC
     - Fornisci le informazioni di connessione Amazon VPC alla connessione AWS Glue che definisce le tue credenziali di sicurezza Vertica. Durante la creazione o l'aggiornamento della connessione, imposta VPC, sottorete e Gruppi di sicurezza nelle opzioni di rete.

Dopo aver creato una connessione AWS Glue Vertica, dovrai eseguire i seguenti passaggi prima di chiamare il tuo metodo di connessione.

- Concedi le autorizzazioni per il ruolo IAM associato al tuo lavoro AWS Glue a*tempS3Path*.
- Concedi al ruolo IAM associato al tuo lavoro AWS Glue il permesso di lettura*secretName*.
- Nella configurazione del lavoro AWS Glue, fornisci *connectionName* una connessione di rete aggiuntiva.

Lettura da Vertica

Prerequisiti:

- Una tabella Vertica da cui si desidera leggere. Avrai bisogno del nome del database Vertica *dbName* e del nome della tabella,*tableName*.
- Una connessione AWS Glue Vertica configurata per fornire informazioni di autenticazione. Completa i passaggi della procedura precedente, Per configurare una connessione a Vertica per configurare le informazioni di autenticazione. Avrai bisogno del nome della connessione AWS Glue,*connectionName*.
- Un bucket o una cartella Amazon S3 da utilizzare per lo storage temporaneo, menzionato in precedenza. Avrai bisogno del nome,*tempS3Path*. Dovrai connetterti a questa posizione utilizzando il protocollo `s3a`.

Esempio:

```
dynamicFrame = glueContext.create_dynamic_frame.from_options(
    connection_type="vertica",
    connection_options={
        "connectionName": "connectionName",
        "staging_fs_url": "s3a://tempS3Path",
        "db": "dbName",
        "table": "tableName",
    }
)
```

Puoi anche fornire una query SQL SELECT, per filtrare i risultati restituiti DynamicFrame o per accedere a un set di dati da più tabelle.

Esempio:

```
dynamicFrame = glueContext.create_dynamic_frame.from_options(
    connection_type="vertica",
    connection_options={
        "connectionName": "connectionName",
        "staging_fs_url": "s3a://tempS3Path",
        "db": "dbName",
        "query": "select * FROM tableName",
    },
)
```

Scrittura su tabelle Vertica

Questo esempio scrive informazioni da un file esistente DynamicFrame *dynamicFrame* a Vertica. Se la tabella contiene già informazioni, AWS Glue aggiungerà i dati dal tuo DynamicFrame.

Prerequisiti:

- Un nome di tabella attuale o desiderato su cui scrivere. *tableName* Avrai anche bisogno del nome del database Vertica corrispondente,*dbName*.
- Una connessione AWS Glue Vertica configurata per fornire informazioni di autenticazione. Completa i passaggi della procedura precedente, Per configurare una connessione a Vertica per configurare le informazioni di autenticazione. Avrai bisogno del nome della connessione AWS Glue,*connectionName*.
- Un bucket o una cartella Amazon S3 da utilizzare per lo storage temporaneo, menzionato in precedenza. Avrai bisogno del nome,*tempS3Path*. Dovrai connetterti a questa posizione utilizzando il protocollo `s3a`.

Esempio:

```
glueContext.write_dynamic_frame.from_options(
    frame=dynamicFrame,
    connection_type="vertica",
    connection_options={
        "connectionName": "connectionName",
        "staging_fs_url": "s3a://tempS3Path",
        "db": "dbName",
        "table": "tableName",
    }
)
```

Indicazioni di riferimento alle opzioni di connessione a Vertica

- `connectionName`: obbligatorio. Utilizzato per la lettura/scrittura. Il nome di una connessione AWS Glue Vertica configurata per fornire informazioni di autenticazione e di rete al metodo di connessione utilizzato.
- db: obbligatorio. Utilizzato per la lettura/scrittura. Il nome dell'indice in Vertica con cui interagirà il metodo di connessione.
- dbSchema — Obbligatorio se necessario per identificare la tabella. Utilizzato per la lettura/scrittura. Default: `public`. Il nome di uno schema con cui interagirà il metodo di connessione.

- `table` — Richiesto per la scrittura, richiesto per la lettura a meno che non `query` sia fornito. Utilizzato per la lettura/scrittura. Il nome di una tabella con cui interagirà il metodo di connessione.
- `query`: utilizzato per la lettura. Una query SELECT SQL che definisce cosa recuperare durante la lettura da Teradata.
- `staging_fs_url`: obbligatorio. Utilizzato per la lettura/scrittura. Valori validi:. `s3a` URLs L'URL di un bucket o di una cartella Amazon S3 da utilizzare per l'archiviazione temporanea.

DataFrame opzioni per ETL in AWS Glue 5.0 for Spark

A DataFrame è un set di dati organizzato in colonne denominate simili a una tabella e supporta operazioni in stile funzionale (map/reduce/filter/etc.) e operazioni SQL (select, project, aggregate).

Per creare un file DataFrame per un'origine dati supportata da Glue, sono necessari i seguenti requisiti:

- Connettore dell'origine dati `ClassName`
- Connessione dell'origine dati `Options`

Allo stesso modo, per scrivere DataFrame a su un data sink supportato da Glue, sono necessari gli stessi:

- Connettore del data sink `ClassName`
- Connessione del data sink `Options`

Tieni presente che le funzionalità di AWS Glue come i segnalibri di lavoro e DynamicFrame le opzioni come non `connectionName` sono supportate in DataFrame. Per maggiori dettagli sulle operazioni supportate DataFrame e sulle operazioni supportate, consulta la documentazione di Spark per. DataFrame

Specificare il connettore ClassName

Per specificare il `ClassName` tipo di dati source/sink, utilizzate l'`.format`opzione per fornire il connettore corrispondente `ClassName` che definisce i dati source/sink.

Connettori JDBC

Per i connettori JDBC, specificare `jdbc` come valore dell'opzione `.format` e indicare il driver JDBC `ClassName` nell'opzione `driver`.

```
df = spark.read.format("jdbc").option("driver", "<DATA SOURCE JDBC DRIVER
 CLASSNAME>")...

df.write.format("jdbc").option("driver", "<DATA SINK JDBC DRIVER CLASSNAME>")...
```

La tabella seguente elenca il driver JDBC `ClassName` dell'origine dati supportata in AWS Glue for. DataFrames

| Origine dati | Autista ClassName |
|---|---|
| PostgreSQL | org.PostgreSQL.Driver |
| Oracle | oracle.jdbc.driver. OracleDriver |
| SQLServer | com.microsoft.sqlserver.jdbc. SQLServerDriver |
| MySQL | com.mysql.jdbc. Driver |
| SAP Hana | com.sap.db.jdbc. Driver |
| Teradata | com.teradata.jdbc. TeraDriver |

Connettori Spark

Per i connettori Spark, specificare il `ClassName` del connettore come valore dell'opzione `.format`.

```
df = spark.read.format("<DATA SOURCE CONNECTOR CLASSNAME>")...

df.write.format("<DATA SINK CONNECTOR CLASSNAME>")...
```

La tabella seguente elenca il connettore Spark `ClassName` dell'origine dati supportata in AWS Glue for DataFrames.

| Origine dati | ClassName |
|---|---|
| MongoDB/DocumentDB | glue.spark.mongodb |
| Redshift | io.github.spark_redshift_community.spark.redshift |

| | |
|---|---|
| AzureCosmos | cosmos.oltp |
| AzureSQL | com.microsoft.sqlserver.jdbc.spark |
| BigQuery | com.google.cloud.spark.bigquery |
| OpenSearch | org.opensearch.spark.sql |
| Snowflake | net.snowflake.spark.snowflake |
| Vertica | com.vertica.spark.datasource. VerticaSource |

Specificare le opzioni di connessione

Per specificare `Options` la connessione a un dato source/sink, usa `.option(<KEY>, <VALUE>)` per fornire opzioni individuali o `.options(<MAP>)` per fornire più opzioni come mappa chiave-valore.

Ogni dato source/sink supporta il proprio set di connessioni. `Options` Per i dettagli sui dati disponibili`Options`, consulta la documentazione pubblica relativa allo specifico connettore source/sink Spark di dati elencata nella tabella seguente.

- JDBC
- MongoDB/DocumentDB
- Redshift
- AzureCosmos
- AzureSQL
- BigQuery
- OpenSearch
- Snowflake
- Vertica

Esempi

I seguenti esempi leggono da PostgreSQL e scrivono in: SnowFlake

Python

Esempio:

```
from awsglue.context import GlueContext
from pyspark.sql import SparkSession

spark = SparkSession.builder.getOrCreate()

dataSourceClassName = "jdbc"
dataSourceOptions = {
  "driver": "org.postgresql.Driver",
  "url": "<url>",
  "user": "<user>",
  "password": "<password>",
  "dbtable": "<dbtable>",
}

dataframe = spark.read.format(className).options(**options).load()

dataSinkClassName = "net.snowflake.spark.snowflake"
dataSinkOptions = {
  "sfUrl": "<url>",
  "sfUsername": "<username>",
  "sfPassword": "<password>",
  "sfDatabase" -> "<database>",
  "sfSchema" -> "<schema>",
  "sfWarehouse" -> "<warehouse>"
}

dataframe.write.format(dataSinkClassName).options(**dataSinkOptions).save()
```

Scala

Esempio:

```
import org.apache.spark.sql.SparkSession

val spark = SparkSession.builder().getOrCreate()

val dataSourceClassName = "jdbc"
val dataSourceOptions = Map(
  "driver" -> "org.postgresql.Driver",
  "url" -> "<url>",
  "user" -> "<user>",
```

```
  "password" -> "<password>",
  "dbtable" -> "<dbtable>"
)

val dataframe =
 spark.read.format(dataSourceClassName).options(dataSourceOptions).load()

val dataSinkClassName = "net.snowflake.spark.snowflake"
val dataSinkOptions = Map(
  "sfUrl" -> "<url>",
  "sfUsername" -> "<username>",
  "sfPassword" -> "<password>",
  "sfDatabase" -> "<database>",
  "sfSchema" -> "<schema>",
  "sfWarehouse" -> "<warehouse>"
)

dataframe.write.format(dataSinkClassName).options(dataSinkOptions).save()
```

Valori Custom e Marketplace AWS ConnectionType

Questi sono i seguenti:

- `"connectionType": "marketplace.athena"`: designa una connessione a un archivio dati Amazon Athena. La connessione utilizza un connettore di Marketplace AWS.
- `"connectionType": "marketplace.spark"`: designa una connessione a un archivio dati Apache Spark. La connessione utilizza un connettore di Marketplace AWS.
- `"connectionType": "marketplace.jdbc"`: designa una connessione a un archivio dati JDBC. La connessione utilizza un connettore di Marketplace AWS.
- `"connectionType": "custom.athena"`: designa una connessione a un archivio dati Amazon Athena. La connessione utilizza un connettore personalizzato che va caricato in AWS Glue Studio.
- `"connectionType": "custom.spark"`: designa una connessione a un archivio dati Apache Spark. La connessione utilizza un connettore personalizzato che va caricato in AWS Glue Studio.
- `"connectionType": "custom.jdbc"`: designa una connessione a un archivio dati JDBC. La connessione utilizza un connettore personalizzato che va caricato in AWS Glue Studio.

Opzioni di connessione per il tipo custom.jdbc o marketplace.jdbc

- `className`— String, obbligatorio, il nome completo della classe Java del driver JDBC (ad esempio,). `com.mysql.cj.jdbc.Driver` È necessario specificare questa opzione quando si utilizza un connettore JDBC o personalizzato. Marketplace AWS Se si omette questa opzione, il processo potrebbe fallire con un `No suitable driver` errore, in particolare quando si utilizzano URL JDBC con parametri.
- `connectionName`: stringa, obbligatorio, nome della connessione associata al connettore.
- `url`: stringa, obbligatorio, URL JDBC con segnaposto (`${}`) che vengono utilizzati per creare la connessione all'origine dati. Il segnaposto `${secretKey}` viene sostituito con il segreto con lo stesso nome in Gestione dei segreti AWS. Per ulteriori informazioni sulla creazione dell'URL, fare riferimento alla documentazione dell'archivio dati.
- `secretId` o `user/password`: stringa, obbligatorio, utilizzato per recuperare le credenziali per l'URL.
- `dbTable` o `query`: stringa, obbligatorio, la tabella o la query SQL da cui ottenere i dati. Puoi specificare `dbTable` o `query`, ma non entrambi.
- `partitionColumn`: stringa, facoltativo, il nome di una colonna intera utilizzata per il partizionamento. Questa opzione funziona solo quando è inclusa con `lowerBound`, `upperBound` e `numPartitions`. Questa opzione funziona allo stesso modo del lettore Spark SQL JDBC. Per ulteriori informazioni, consulta JDBC To Other Databases nella Apache Spark SQL and Datasets Guide. DataFrames

  I valori `lowerBound` e `upperBound` vengono utilizzati per decidere lo stride della partizione, non per filtrare le righe nella tabella. Tutte le righe della tabella vengono partizionate e restituite.

  > **Note**
  >
  > Quando si utilizza una query anziché un nome di tabella, è necessario verificare che la query funzioni con la condizione di partizionamento specificata. Ad esempio:
  >
  > - Se il formato della query è `"SELECT col1 FROM table1"`, testa la query aggiungendo una clausola `WHERE` alla fine della query che utilizza la colonna della partizione.
  > - Se il formato della query è `"SELECT col1 FROM table1 WHERE col2=val"`, testa la query estendendo la clausola `WHERE` con `AND` e un'espressione che utilizza la colonna della partizione.

- `lowerBound`: intero, facoltativo, il valore minimo di `partitionColumn` che viene utilizzato per decidere lo stride della partizione.
- `upperBound`: intero, facoltativo, il valore massimo di `partitionColumn` che viene utilizzato per decidere lo stride della partizione.
- `numPartitions`: intero, facoltativo, il numero di partizioni. Questo valore, insieme a `lowerBound` (incluso) e `upperBound` (escluso), forma stride di partizione per espressioni con le clausole `WHERE` generate che vengono utilizzate per dividere la `partitionColumn`.

  > **Important**
  >
  > Presta attenzione al numero di partizioni perché troppe partizioni potrebbero causare problemi nei sistemi di database esterni.

- `filterPredicate`: stringa, opzionale, clausola condizione extra per filtrare i dati dall'origine. Ad esempio:

  ```
  BillingCity='Mountain View'
  ```

  Quando si utilizza una query anziché un nome di table, è necessario verificare che la query funzioni con il `filterPredicate` specificato. Ad esempio:

  - Se il formato della query è `"SELECT col1 FROM table1"`, testa la query aggiungendo una clausola `WHERE` alla fine della query che utilizza il predicato filtro.
  - Se il formato della query è `"SELECT col1 FROM table1 WHERE col2=val"`, testa la query estendendo la clausola `WHERE` con `AND` e un'espressione che utilizza il predicato filtro.

- `dataTypeMapping`: dizionario, opzionale, mappatura del tipo di dati personalizzata che crea una mappatura da un tipo di dati JDBC a un tipo di dati Glue. Ad esempio, l'opzione `"dataTypeMapping":{"FLOAT":"STRING"}` mappa i campi di dati di tipo JDBC `FLOAT` nel tipo Java `String` chiamando il metodo `ResultSet.getString()` del driver e lo usa per costruire registri di AWS Glue. L'oggetto `ResultSet` viene implementato da ciascun driver, quindi il comportamento è specifico del driver utilizzato. Consulta la documentazione relativa al driver JDBC per capire come il driver esegue le conversioni.
- I tipi di dati AWS Glue correntemente supportati sono:
  - DATE
  - STRING
  - TIMESTAMP

- INT
- FLOAT
- LONG
- BIGDECIMAL
- BYTE
- SHORT
- DOUBLE

I tipi di dati JDBC supportati sono Java8 java.sql.types.

Le mappature di default dei tipi di dati (da JDBC a AWS Glue) sono:

- DATE -> DATE
- VARCHAR -> STRING
- CHAR -> STRING
- LONGNVARCHAR -> STRING
- TIMESTAMP -> TIMESTAMP
- INTEGER -> INT
- FLOAT -> FLOAT
- REAL -> FLOAT
- BIT -> BOOLEAN
- BOOLEAN -> BOOLEAN
- BIGINT -> LONG
- DECIMAL -> BIGDECIMAL
- NUMERIC -> BIGDECIMAL
- TINYINT -> SHORT
- SMALLINT -> SHORT
- DOUBLE -> DOUBLE

Se si utilizza un mapping del tipo di dati personalizzato con l'opzione `dataTypeMapping`, è possibile sovrascrivere una mappatura di default del tipo di dati. Sono interessati solo i tipi di dati JDBC elencati nell'opzione `dataTypeMapping`; per tutti gli altri tipi di dati JDBC viene utilizzata la mappatura di default. Se necessario, è possibile aggiungere mappature per tipi di dati JDBC

aggiuntivi. Se un tipo di dati JDBC non è incluso nella mappatura di default o in una mappatura personalizzata, per impostazione predefinita viene convertito nel tipo di dati `STRING` AWS Glue.

Il seguente esempio di codice Python mostra come leggere dai database JDBC con driver JDBC. Marketplace AWS Mostra la lettura da un database e la scrittura in una posizione S3.

```
    import sys
    from awsglue.transforms import *
    from awsglue.utils import getResolvedOptions
    from pyspark.context import SparkContext
    from awsglue.context import GlueContext
    from awsglue.job import Job

    ## @params: [JOB_NAME]
    args = getResolvedOptions(sys.argv, ['JOB_NAME'])

    sc = SparkContext()
    glueContext = GlueContext(sc)
    spark = glueContext.spark_session
    job = Job(glueContext)
    job.init(args['JOB_NAME'], args)
    ## @type: DataSource
    ## @args: [connection_type = "marketplace.jdbc", connection_options =
     {"dataTypeMapping":{"INTEGER":"STRING"},"upperBound":"200","query":"select id,
       name, department from department where id < 200","numPartitions":"4",
       "partitionColumn":"id","lowerBound":"0","connectionName":"test-connection-
jdbc"},
        transformation_ctx = "DataSource0"]
    ## @return: DataSource0
    ## @inputs: []
    DataSource0 = glueContext.create_dynamic_frame.from_options(connection_type =
      "marketplace.jdbc", connection_options = {"dataTypeMapping":{"INTEGER":"STRING"},
      "upperBound":"200","query":"select id, name, department from department where
       id < 200","numPartitions":"4","partitionColumn":"id","lowerBound":"0",
       "connectionName":"test-connection-jdbc"}, transformation_ctx = "DataSource0")
    ## @type: ApplyMapping
    ## @args: [mappings = [("department", "string", "department", "string"), ("name",
 "string",
      "name", "string"), ("id", "int", "id", "int")], transformation_ctx =
 "Transform0"]
    ## @return: Transform0
    ## @inputs: [frame = DataSource0]
```

```
    Transform0 = ApplyMapping.apply(frame = DataSource0, mappings = [("department",
"string",
     "department", "string"), ("name", "string", "name", "string"), ("id", "int",
"id", "int")],
      transformation_ctx = "Transform0")
   ## @type: DataSink
   ## @args: [connection_type = "s3", format = "json", connection_options = {"path":
    "s3://<S3 path>/", "partitionKeys": []}, transformation_ctx = "DataSink0"]
   ## @return: DataSink0
   ## @inputs: [frame = Transform0]
   DataSink0 = glueContext.write_dynamic_frame.from_options(frame = Transform0,
     connection_type = "s3", format = "json", connection_options = {"path":
     "s3://<S3 path>/", "partitionKeys": []}, transformation_ctx = "DataSink0")
   job.commit()
```

Opzioni di connessione per il tipo custom.athena o marketplace.athena

- `className` – Stringa, obbligatorio, nome della classe driver. Quando si utilizza il Athena-CloudWatch connettore, il valore di questo parametro è il prefisso del nome della classe (ad esempio,). `"com.amazonaws.athena.connectors"` Il Athena-CloudWatch connettore è composto da due classi: un gestore di metadati e un gestore di record. Se si fornisce qui il prefisso comune, l'API carica le classi corrette in base a tale prefisso.
- `tableName`— Stringa, obbligatoria, il nome del flusso di CloudWatch log da leggere. In questo frammento di codice viene utilizzato il nome della vista speciale `all_log_streams`, il che significa che il frame di dati dinamico restituito conterrà i dati di tutti i flussi di log nel gruppo di log.
- `schemaName`— String, obbligatorio, il nome del gruppo di CloudWatch log da cui leggere. Ad esempio, `/aws-glue/jobs/output`.
- `connectionName` – Stringa, obbligatorio, nome della connessione associata al connettore.

Per ulteriori opzioni per questo connettore, consulta il file README di Amazon Athena CloudWatch Connector su. GitHub

Il seguente esempio di codice Python mostra come leggere da un archivio dati Athena utilizzando un connettore Marketplace AWS . Mostra la lettura da Athena e la scrittura in una posizione S3.

```
    import sys
    from awsglue.transforms import *
    from awsglue.utils import getResolvedOptions
    from pyspark.context import SparkContext
    from awsglue.context import GlueContext
```

```
    from awsglue.job import Job

    ## @params: [JOB_NAME]
    args = getResolvedOptions(sys.argv, ['JOB_NAME'])

    sc = SparkContext()
    glueContext = GlueContext(sc)
    spark = glueContext.spark_session
    job = Job(glueContext)
    job.init(args['JOB_NAME'], args)
    ## @type: DataSource
    ## @args: [connection_type = "marketplace.athena", connection_options =
     {"tableName":"all_log_streams","schemaName":"/aws-glue/jobs/output",
      "connectionName":"test-connection-athena"}, transformation_ctx = "DataSource0"]
    ## @return: DataSource0
    ## @inputs: []
    DataSource0 = glueContext.create_dynamic_frame.from_options(connection_type =
      "marketplace.athena", connection_options = {"tableName":"all_log_streams",,
      "schemaName":"/aws-glue/jobs/output","connectionName":
      "test-connection-athena"}, transformation_ctx = "DataSource0")
    ## @type: ApplyMapping
    ## @args: [mappings = [("department", "string", "department", "string"), ("name",
"string",
     "name", "string"), ("id", "int", "id", "int")], transformation_ctx =
"Transform0"]
    ## @return: Transform0
    ## @inputs: [frame = DataSource0]
    Transform0 = ApplyMapping.apply(frame = DataSource0, mappings = [("department",
"string",
     "department", "string"), ("name", "string", "name", "string"), ("id", "int",
"id", "int")],
      transformation_ctx = "Transform0")
    ## @type: DataSink
    ## @args: [connection_type = "s3", format = "json", connection_options = {"path":
     "s3://<S3 path>/", "partitionKeys": []}, transformation_ctx = "DataSink0"]
    ## @return: DataSink0
    ## @inputs: [frame = Transform0]
    DataSink0 = glueContext.write_dynamic_frame.from_options(frame = Transform0,
      connection_type = "s3", format = "json", connection_options = {"path":
      "s3://<S3 path>/", "partitionKeys": []}, transformation_ctx = "DataSink0")
    job.commit()
```

Opzioni di connessione per il tipo custom.spark o marketplace.spark

- `className`: stringa, obbligatorio, nome della classe del connettore.
- `secretId`: stringa, facoltativo, utilizzato per recuperare le credenziali per la connessione del connettore.
- `connectionName` – Stringa, obbligatorio, nome della connessione associata al connettore.
- Altre opzioni dipendono dall'archivio dati. Ad esempio, le opzioni di OpenSearch configurazione iniziano con il prefissoes, come descritto nella documentazione di Elasticsearch for Apache Hadoop. Le connessioni Spark a Snowflake utilizzano opzioni come `sfUser` e `sfPassword`, come descritto in Using the Spark Connector nella guida Connecting to Snowflake.

Il seguente esempio di codice Python mostra come leggere da un archivio OpenSearch dati utilizzando una `marketplace.spark` connessione.

```
    import sys
    from awsglue.transforms import *
    from awsglue.utils import getResolvedOptions
    from pyspark.context import SparkContext
    from awsglue.context import GlueContext
    from awsglue.job import Job

    ## @params: [JOB_NAME]
    args = getResolvedOptions(sys.argv, ['JOB_NAME'])

    sc = SparkContext()
    glueContext = GlueContext(sc)
    spark = glueContext.spark_session
    job = Job(glueContext)
    job.init(args['JOB_NAME'], args)
    ## @type: DataSource
    ## @args: [connection_type = "marketplace.spark", connection_options =
{"path":"test",
      "es.nodes.wan.only":"true","es.nodes":"https://<AWS endpoint>",
      "connectionName":"test-spark-es","es.port":"443"}, transformation_ctx =
"DataSource0"]
    ## @return: DataSource0
    ## @inputs: []
    DataSource0 = glueContext.create_dynamic_frame.from_options(connection_type =
      "marketplace.spark", connection_options = {"path":"test","es.nodes.wan.only":
      "true","es.nodes":"https://<AWS endpoint>","connectionName":
```

```
    "test-spark-es","es.port":"443"}, transformation_ctx = "DataSource0")
## @type: DataSink
## @args: [connection_type = "s3", format = "json", connection_options = {"path":
     "s3://<S3 path>/", "partitionKeys": []}, transformation_ctx = "DataSink0"]
## @return: DataSink0
## @inputs: [frame = DataSource0]
DataSink0 = glueContext.write_dynamic_frame.from_options(frame = DataSource0,
   connection_type = "s3", format = "json", connection_options = {"path":
   "s3://<S3 path>/", "partitionKeys": []}, transformation_ctx = "DataSink0")
job.commit()
```

Opzioni generali

Le opzioni in questa sezione sono fornite come connettore `connection_options`, ma non si applicano specificamente a tale connettore.

I seguenti parametri vengono generalmente utilizzati per la configurazione dei segnalibri. Possono applicarsi ai flussi di lavoro Amazon S3 o JDBC. Per ulteriori informazioni, consulta the section called "Utilizzo di segnalibri di processo".

- `jobBookmarkKeys`: un array di nomi di colonna.
- `jobBookmarkKeysSortOrder`: una stringa che definisce come confrontare i valori in base all'ordinamento. Valori validi: `"asc"`, `"desc"`.
- `useS3ListImplementation`: utilizzato per gestire le prestazioni della memoria quando si elencano i contenuti dei bucket Amazon S3. Per ulteriori informazioni, consulta Ottimizzare la gestione della memoria in AWS Glue.

## Opzioni del formato dati per input e output in AWS Glue per Spark

Queste pagine offrono informazioni sul supporto delle funzionalità e sui parametri di configurazione per i formati di dati supportati da AWS Glue per Spark. Consulta quanto riportato di seguito per una descrizione dell'uso e dell'applicabilità di queste informazioni.

Supporto delle funzionalità per tutti i formati di dati in AWS Glue

Ogni formato di dati può supportare diverse funzionalità di AWS Glue. Le funzioni comuni indicate di seguito possono essere supportate o meno in base al tipo di formato. Consulta la documentazione relativa al formato dati per capire come sfruttare le nostre funzioni per soddisfare i tuoi requisiti.

| | |
|---|---|
| Lettura | AWS Glue è in grado di riconoscere e interpret are questo formato di dati senza risorse aggiuntive, come i connettori. |
| Scrittura | AWS Glue può scrivere dati in questo formato senza risorse aggiuntive. Puoi includere librerie di terzi nel tuo processo e utilizzare funzioni standard di Apache Spark per scrivere i dati, come con altri ambienti Spark. Per ulteriori informazioni sull'inclusione di librerie, consulta the section called "Librerie Python". |
| Lettura in streamin | AWS Glue è in grado di riconoscere e interpret are questo formato di dati da un flusso di messaggi Apache Kafka, Amazon Managed Streaming for Apache Kafka o Amazon Kinesis. Prevediamo che i flussi presentino i dati in un formato coerente, quindi vengano letti come `DataFrames` . |
| Gruppo di file piccoli | AWS Glue può raggruppare i file per il lavoro in batch inviato a ciascun nodo durante l'esecuzi one delle trasformazioni di AWS Glue. Ciò può migliorare significativamente le prestazioni per carichi di lavoro che implicano grandi quantità di file piccoli. Per ulteriori informazioni, consulta the section called "Raggruppamento dei file di input". |
| Segnalib i di processc | AWS Glue è in grado di monitorare l'avanzam ento delle trasformazioni che eseguono lo stesso lavoro sullo stesso set di dati in tutte le esecuzioni di lavoro con i segnalibri dei lavori. Ciò può migliorare le prestazioni per carichi di lavoro che implicano set di dati in cui occorre operare solo su nuovi dati dall'ultima esecuzion e del processo. Per ulteriori informazioni, |

consulta the section called "Monitoraggio dei dati elaborati mediante segnalibri di processo".

Parametri utilizzati per interagire con i formati di dati in AWS Glue

Alcuni tipi di connessione AWS Glue supportano più `format` tipi, pertanto è necessario specificare informazioni sul formato dei dati con un `format_options` oggetto quando si utilizzano metodi come`GlueContext.write_dynamic_frame.from_options`.

- `s3`— Per ulteriori informazioni, vedere Tipi di connessione e opzioni per ETL in AWS Glue:Parametri di connessione di S3. Puoi anche visualizzare la documentazione relativa ai metodi che facilitano questo tipo di connessione: the section called "create_dynamic_frame_from_options" e the section called "write_dynamic_frame_from_options" in Python e i metodi Scala corrispondenti the section called "getSourceWithFormato" e the section called "getSinkWithFormato".

- `kinesis`— Per ulteriori informazioni, vedere Tipi di connessione e opzioni per ETL in AWS Glue:Parametri di connessione Kinesis. Puoi anche visualizzare la documentazione relativa ai metodi che facilitano questo tipo di connessione: the section called "create_data_frame_from_options" e il metodo Scala corrispondente the section called "createDataFrameFromOptions".
- `kafka`— Per ulteriori informazioni, vedere Tipi di connessione e opzioni per ETL in AWS Glue:Parametri di connessione Kafka. Puoi anche visualizzare la documentazione relativa ai metodi che facilitano questo tipo di connessione: the section called "create_data_frame_from_options" e il metodo Scala corrispondente the section called "createDataFrameFromOptions".

Alcuni tipi di connessione non richiedono `format_options`. Ad esempio, nell'utilizzo normale, una connessione JDBC a un database relazionale recupera i dati in un formato dati tabulare coerente. Pertanto, la lettura da una connessione JDBC non richiedere `format_options`.

Alcuni metodi per la lettura e la scrittura di dati in Glue non richiedono `format_options`. Ad esempio, utilizzando `GlueContext.create_dynamic_frame.from_catalog` con i crawler AWS Glue. I crawler determinano la forma dei dati. Quando si utilizzano i crawler, un classificatore AWS Glue esaminerà i dati per prendere decisioni intelligenti su come rappresentare il formato dei dati. Quindi memorizzerà una rappresentazione dei dati nel AWS Glue Data Catalog, che può essere utilizzata all'interno di uno script AWS Glue ETL per recuperare i dati con il

`GlueContext.create_dynamic_frame.from_catalog` metodo. I crawler eliminano la necessità di specificare manualmente informazioni sul formato dati.

Per i lavori che accedono a tabelle AWS Lake Formation gestite, AWS Glue supporta la lettura e la scrittura di tutti i formati supportati dalle tabelle governate da Lake Formation. Per l'elenco corrente dei formati supportati per le tabelle AWS Lake Formation governate, consulta Notes and Restrictions for Governed Tables nella AWS Lake Formation Developer Guide.

> **Note**
>
> Per scrivere Apache Parquet, AWS Glue ETL supporta solo la scrittura su una tabella governata, specificando un'opzione per un tipo di scritture Parquet personalizzata ottimizzata per Dynamic Frames. Quando scrivi su una tabella governata con il formato `parquet`, è necessario aggiungere la chiave `useGlueParquetWriter` con un valore di `true` nei parametri della tabella.

Argomenti


- Utilizzo del formato CSV in AWS Glue
- Utilizzo del formato Parquet in AWS Glue
- Utilizzo del formato XML in AWS Glue
- Utilizzo del formato Avro in AWS Glue
- Utilizzo del formato GrokLog in Glue AWS
- Utilizzo del formato Ion in AWS Glue
- Utilizzo del formato JSON in AWS Glue
- Utilizzo del formato ORC in AWS Glue
- Utilizzo di framework di data lake con AWS Glue processi ETL
- Riferimento alla configurazione condivisa


## Utilizzo del formato CSV in AWS Glue

AWS Glue recupera i dati dalle fonti e li scrive su destinazioni archiviate e trasportate in vari formati di dati. Se i dati vengono archiviati o trasportati nel formato dati CSV, questo documento presenta le funzionalità disponibili per l'utilizzo dei dati in Glue. AWS

AWS Glue supporta l'utilizzo del formato con valori separati da virgole (CSV). Questo formato è un formato di dati minimo basato su righe. CSVs spesso non sono strettamente conformi a uno standard, ma puoi fare riferimento a RFC 4180 e RFC 7111 per ulteriori informazioni.

Puoi usare AWS Glue per leggere CSVs da Amazon S3 e da fonti di streaming, nonché CSVs scrivere su Amazon S3. Puoi leggere e scrivere gli archivi `bzip` e `gzip` contenenti file CSV da S3. Puoi configurare il comportamento di compressione sul Parametri di connessione di S3 invece che nella configurazione discussa in questa pagina.

La tabella seguente mostra quali funzioni comuni di AWS Glue supportano l'opzione di formato CSV.

| Lettura | Scrittura | Lettura in streaming | Gruppo di file piccoli | Segnalibri di processo |
|---|---|---|---|---|
| Supportata | Supportato | Supportato | Supportato | Supportata |

Esempio: lettura di file CSV o cartelle da S3

Prerequisiti: avrai bisogno dei percorsi S3 (`s3path`) nei file CSV o nelle cartelle che desideri leggere.

Configurazione : nelle opzioni della funzione, specifica `format="csv"`. In `connection_options`, utilizzare la chiave `paths` per specificare `s3path`. Puoi configurare il modo in cui il reader interagisce con S3 in `connection_options`. Per i dettagli, vedi Tipi di connessione e opzioni per ETL in AWS Glue:Parametri di connessione di S3. Puoi configurare il modo in cui il reader interpreta i file CSV nel tuo `format_options`. Per maggiori dettagli, consulta .Riferimento alla configurazione CSV.

Il seguente script AWS Glue ETL mostra il processo di lettura di file o cartelle CSV da S3.

Forniamo un reader CSV personalizzato con ottimizzazioni delle prestazioni per i flussi di lavoro comuni attraverso la chiave di configurazione `optimizePerformance`. Per determinare se questo reader è adatto al tuo carico di lavoro, consulta the section called "Utilizzo di un reader CSV ottimizzato".

Python

Per questo esempio, utilizzare il metodo create_dynamic_frame.from_options.

```
# Example: Read CSV from S3
```

```
# For show, we handle a CSV with a header row.  Set the withHeader option.
# Consider whether optimizePerformance is right for your workflow.

from pyspark.context import SparkContext
from awsglue.context import GlueContext

sc = SparkContext.getOrCreate()
glueContext = GlueContext(sc)
spark = glueContext.spark_session

dynamicFrame = glueContext.create_dynamic_frame.from_options(
    connection_type="s3",
    connection_options={"paths": ["s3://s3path"]},
    format="csv",
    format_options={
        "withHeader": True,
        # "optimizePerformance": True,
    },
)
```

Puoi anche usarlo DataFrames in uno script (). `pyspark.sql.DataFrame`

```
dataFrame = spark.read\
    .format("csv")\
    .option("header", "true")\
    .load("s3://s3path")
```

Scala

Per questo esempio, utilizzate l'operazione getSourceWithFormat.

```
// Example: Read CSV from S3
// For show, we handle a CSV with a header row.  Set the withHeader option.
// Consider whether optimizePerformance is right for your workflow.

import com.amazonaws.services.glue.util.JsonOptions
import com.amazonaws.services.glue.{DynamicFrame, GlueContext}
import org.apache.spark.SparkContext

object GlueApp {
  def main(sysArgs: Array[String]): Unit = {
    val spark: SparkContext = new SparkContext()
    val glueContext: GlueContext = new GlueContext(spark)
```

```
    val dynamicFrame = glueContext.getSourceWithFormat(
      formatOptions=JsonOptions("""{"withHeader": true}"""),
      connectionType="s3",
      format="csv",
      options=JsonOptions("""{"paths": ["s3://s3path"], "recurse": true}""")
    ).getDynamicFrame()
  }
}
```

È inoltre possibile utilizzare DataFrames in uno script (`org.apache.spark.sql.DataFrame`).

```
val dataFrame = spark.read
  .option("header","true")
  .format("csv")
  .load("s3://s3path“)
```

Esempio: scrittura di file e cartelle CSV su S3

Prerequisiti: è necessario un comando inizializzato DataFrame (`dataFrame`) o un DynamicFrame (`dynamicFrame`). È necessario anche del tuo percorso di output S3 previsto, `s3path`.

Configurazione: nelle opzioni della funzione, specifica `format="csv"`. In `connection_options`, utilizzare la chiave `paths` per specificare `s3path`. Puoi configurare il modo in cui il writer interagisce con S3 in `connection_options`. Per i dettagli, vedi Tipi di connessione e opzioni per ETL in AWS Glue:Parametri di connessione di S3. Puoi configurare il modo in cui l'operazione scrive il contenuto dei file in `format_options`. Per maggiori dettagli, consulta .Riferimento alla configurazione CSV. Il seguente script ETL di AWS Glue mostra il processo di scrittura di file o cartelle CSV da S3.

Python

Per questo esempio, utilizza il metodo write_dynamic_frame.from_options.

```
# Example: Write CSV to S3
# For show, customize how we write string type values.  Set quoteChar to -1 so our
 values are not quoted.

from pyspark.context import SparkContext
from awsglue.context import GlueContext
```

```
sc = SparkContext.getOrCreate()
glueContext = GlueContext(sc)

glueContext.write_dynamic_frame.from_options(
    frame=dynamicFrame,
    connection_type="s3",
    connection_options={"path": "s3://s3path"},
    format="csv",
    format_options={
        "quoteChar": -1,
    },
)
```

Puoi anche usare DataFrames in uno script (`pyspark.sql.DataFrame`).

```
dataFrame.write\
    .format("csv")\
    .option("quote", None)\
    .mode("append")\
    .save("s3://s3path")
```

Scala

Per questo esempio, utilizzate il metodo getSinkWithFormat.

```
// Example: Write CSV to S3
// For show, customize how we write string type values. Set quoteChar to -1 so our
 values are not quoted.

import com.amazonaws.services.glue.util.JsonOptions
import com.amazonaws.services.glue.{DynamicFrame, GlueContext}
import org.apache.spark.SparkContext

object GlueApp {
  def main(sysArgs: Array[String]): Unit = {
    val spark: SparkContext = new SparkContext()
    val glueContext: GlueContext = new GlueContext(spark)

    glueContext.getSinkWithFormat(
        connectionType="s3",
        options=JsonOptions("""{"path": "s3://s3path"}"""),
```

```
        format="csv"
    ).writeDynamicFrame(dynamicFrame)
  }
}
```

È inoltre possibile utilizzare DataFrames in uno script (`org.apache.spark.sql.DataFrame`).

```
dataFrame.write
    .format("csv")
    .option("quote", null)
    .mode("Append")
    .save("s3://s3path")
```

Riferimento alla configurazione CSV

Puoi usare quanto segue `format_options` ovunque le librerie AWS Glue lo specifichinoformat="csv":

- `separator`: specifica il carattere delimitatore. L'impostazione predefinita è una virgola, ma è possibile specificare qualsiasi altro carattere.
  - Tipo: testo, Valore predefinito: `","`
- `escaper`: specifica un carattere da utilizzare per l'escape. Questa opzione viene utilizzata solo durante la lettura di file CSV e non durante la scrittura. Se questa opzione è abilitata, il carattere immediatamente seguente viene usato così come è, ad eccezione di un piccolo set di caratteri escape ben noti (`\n`, `\r`, `\t` e `\0`).
  - Tipo: testo, Valore predefinito: nessuno
- `quoteChar`: specifica il carattere da usare per le virgolette. Per impostazione predefinita, vengono usate le virgolette doppie. Imposta questo valore su `-1` per disattivare completamente le virgolette.
  - Tipo: testo, Valore predefinito: `'"'`
- `multiLine`: specifica se un singolo record può estendersi su più righe. Ciò può accadere quando un campo contiene un carattere di nuova riga tra virgolette. Imposta questa opzione su `True` se i registri si estendono su più righe. L'abilitazione di `multiLine` potrebbe ridurre le prestazioni perché richiede una divisione dei file più cauta durante l'analisi.
  - Tipo: booleano, Valore predefinito: `false`
- `withHeader`: specifica se trattare la prima riga come intestazione. Questa opzione può essere usata nella classe `DynamicFrameReader`.

  - Tipo: booleano, Valore predefinito: `false`
- `writeHeader`: specifica se scrivere l'intestazione nell'output. Questa opzione può essere usata nella classe `DynamicFrameWriter`.
  - Tipo: booleano, Valore predefinito: `true`
- `skipFirst`: specifica se ignorare la prima riga di dati.
  - Tipo: booleano, Valore predefinito: `false`
- `optimizePerformance`: specifica se utilizzare il reader CSV SIMD avanzato insieme ai formati di memoria colonnare basati su Apache Arrow. Disponibile solo in AWS Glue versione 3.0 o successiva.
  - Tipo: booleano, Valore predefinito: `false`
- `strictCheckForQuoting`— Durante la scrittura CSVs, Glue può aggiungere virgolette ai valori che interpreta come stringhe. Questo viene fatto per evitare ambiguità in ciò che viene scritto. Per risparmiare tempo nella decisione su cosa scrivere, Glue può utilizzare le virgolette in determinate situazioni in cui in realtà non sono necessarie. L'attivazione di un controllo rigoroso eseguirà un calcolo più intensivo e ricorrerà alle virgolette solo quando strettamente necessario. Disponibile solo in AWS Glue versione 3.0 o successiva.
  - Tipo: booleano, Valore predefinito: `false`

Ottimizza le prestazioni di lettura con il reader CSV SIMD vettorizzato

AWS Glue versione 3.0 aggiunge un lettore CSV ottimizzato che può velocizzare significativamente le prestazioni complessive del lavoro rispetto ai reader CSV basati su righe.

Il reader ottimizzato:

- Usa le istruzioni SIMD della CPU per leggere dal disco
- Scrive immediatamente i record in memoria in un formato colonnare (Apache Arrow)
- Divide i record in batch

Ciò consente di risparmiare tempo di elaborazione quando i record verranno raggruppati in batch o convertiti in un formato colonnare in un secondo momento. Alcuni esempi sono quando si modificano schemi o si recuperano i dati in base alla colonna.

Per utilizzare il reader ottimizzato, imposta `"optimizePerformance"` su `true` nelle `format_options` o nella proprietà della tabella.

```
glueContext.create_dynamic_frame.from_options(
    frame = datasource1,
    connection_type = "s3",
    connection_options = {"paths": ["s3://s3path"]},
    format = "csv",
    format_options={
        "optimizePerformance": True,
        "separator": ","
        },
    transformation_ctx = "datasink2")
```

Limitazioni per il reader CSV vettorizzato

Tieni presente le seguenti limitazioni del reader CSV vettorizzato:

- Non supporta il le opzioni di formato `multiLine` e `escaper`. Utilizza l'`escaper` predefinito del carattere doppia virgoletta `'"'`. Quando queste opzioni sono impostate, AWS Glue torna automaticamente a utilizzare il reader CSV basato su righe.
- Non supporta la creazione di un con DynamicFrame . ChoiceType
- Non supporta la creazione di record DynamicFrame di errori.
- Non supporta la lettura di file CSV con caratteri multibyte, come i caratteri giapponesi o cinesi.

## Utilizzo del formato Parquet in AWS Glue

AWS Glue recupera i dati dalle fonti e li scrive su destinazioni archiviate e trasportate in vari formati di dati. Se i dati vengono archiviati o trasportati nel formato dati Parquet, questo documento presenta le funzionalità disponibili per l'utilizzo dei dati in AWS Glue.

AWS Glue supporta l'utilizzo del formato Parquet. Questo formato è un formato dati basato su colonne orientato alle prestazioni. Per un'introduzione al formato da parte dell'autorità standard, consulta Panoramica della documentazione di Apache Parquet.

Puoi usare AWS Glue per leggere file Parquet da Amazon S3 e da sorgenti di streaming, nonché scrivere file Parquet su Amazon S3. Puoi leggere e scrivere gli archivi `bzip` e `gzip` contenenti file Parquet da S3. Puoi configurare il comportamento di compressione sul Parametri di connessione di S3 invece che nella configurazione discussa in questa pagina.

La tabella seguente mostra quali funzioni comuni di AWS Glue supportano l'opzione di formato Parquet.

| Lettura | Scrittura | Lettura in streaming | Gruppo di file piccoli | Segnalibri di processo |
|---|---|---|---|---|
| Supportata | Supportato | Supportata | Non supportato. | Supportato* |

* Supportato nella versione AWS Glue 1.0+

Esempio: lettura di file Parquet o cartelle da S3

Prerequisiti: avrai bisogno dei percorsi S3 (`s3path`) nei file Parquet o nelle cartelle che desideri leggere.

Configurazione: nelle opzioni della funzione, specifica `format="parquet"`. Nelle tue `connection_options`, utilizza la chiave `paths` per specificare `s3path`.

Puoi configurare il modo in cui il reader interagisce con S3 in `connection_options`. Per i dettagli, vedi Tipi di connessione e opzioni per ETL in AWS Glue:Parametri di connessione di S3.

Puoi configurare il modo in cui il reader interpreta i file Parquet nel tuo `format_options`. Per maggiori dettagli, consulta Riferimento alla configurazione Parquet.

Il seguente script AWS Glue ETL mostra il processo di lettura dei file o delle cartelle Parquet da S3:

Python

Per questo esempio, utilizzare il metodo create_dynamic_frame.from_options.

```
# Example: Read Parquet from S3

from pyspark.context import SparkContext
from awsglue.context import GlueContext

sc = SparkContext.getOrCreate()
glueContext = GlueContext(sc)
spark = glueContext.spark_session

dynamicFrame = glueContext.create_dynamic_frame.from_options(
    connection_type = "s3",
    connection_options = {"paths": ["s3://s3path/"]},
    format = "parquet"
)
```

Puoi anche usarlo DataFrames in uno script ()`pyspark.sql.DataFrame`.

```
dataFrame = spark.read.parquet("s3://s3path/")
```

Scala

Per questo esempio, utilizzate il metodo getSourceWithFormat.

```
// Example: Read Parquet from S3

import com.amazonaws.services.glue.util.JsonOptions
import com.amazonaws.services.glue.{DynamicFrame, GlueContext}
import org.apache.spark.SparkContext

object GlueApp {
  def main(sysArgs: Array[String]): Unit = {
    val spark: SparkContext = new SparkContext()
    val glueContext: GlueContext = new GlueContext(spark)

    val dynamicFrame = glueContext.getSourceWithFormat(
      connectionType="s3",
      format="parquet",
      options=JsonOptions("""{"paths": ["s3://s3path"]}""")
    ).getDynamicFrame()
  }
}
```

È inoltre possibile utilizzare DataFrames in uno script (`org.apache.spark.sql.DataFrame`).

```
spark.read.parquet("s3://s3path/")
```

Esempio: scrittura di file e cartelle Parquet su S3

Prerequisiti: è necessario un DataFrame (`dataFrame`) o DynamicFrame (`dynamicFrame`) inizializzato. È necessario anche del tuo percorso di output S3 previsto, `s3path`.

Configurazione: nelle opzioni della funzione, specifica `format="parquet"`. In `connection_options`, utilizzare la chiave `paths` per specificare `s3path`.

È possibile modificare ulteriormente il modo in cui il writer interagisce con S3 nelle `connection_options`. Per i dettagli, vedi Tipi di connessione e opzioni per ETL in AWS Glue:Parametri di connessione di S3. Puoi configurare il modo in cui l'operazione scrive il contenuto dei file in `format_options`. Per maggiori dettagli, consulta .Riferimento alla configurazione Parquet.

Il seguente script AWS Glue ETL mostra il processo di scrittura di file e cartelle Parquet su S3.

Forniamo uno scrittore Parquet personalizzato con ottimizzazioni delle prestazioni DynamicFrames, tramite la `useGlueParquetWriter` chiave di configurazione. Per determinare se questo writer è adatto al tuo carico di lavoro, consulta Scrittore Glue Parquet.

Python

Per questo esempio, utilizza il metodo write_dynamic_frame.from_options.

```
# Example: Write Parquet to S3
# Consider whether useGlueParquetWriter is right for your workflow.

from pyspark.context import SparkContext
from awsglue.context import GlueContext

sc = SparkContext.getOrCreate()
glueContext = GlueContext(sc)

glueContext.write_dynamic_frame.from_options(
    frame=dynamicFrame,
    connection_type="s3",
    format="parquet",
    connection_options={
        "path": "s3://s3path",
    },
    format_options={
        # "useGlueParquetWriter": True,
    },
)
```

Puoi anche usarlo DataFrames in uno script (`pyspark.sql.DataFrame`).

```
df.write.parquet("s3://s3path/")
```

Scala

Per questo esempio, utilizzate il metodo getSinkWithFormat.

```
// Example: Write Parquet to S3
// Consider whether useGlueParquetWriter is right for your workflow.

import com.amazonaws.services.glue.util.JsonOptions
import com.amazonaws.services.glue.{DynamicFrame, GlueContext}
import org.apache.spark.SparkContext

object GlueApp {
  def main(sysArgs: Array[String]): Unit = {
    val spark: SparkContext = new SparkContext()
    val glueContext: GlueContext = new GlueContext(spark)

    glueContext.getSinkWithFormat(
        connectionType="s3",
        options=JsonOptions("""{"path": "s3://s3path"}"""),
        format="parquet"
    ).writeDynamicFrame(dynamicFrame)
  }
}
```

È inoltre possibile utilizzare DataFrames in uno script (`org.apache.spark.sql.DataFrame`).

```
df.write.parquet("s3://s3path/")
```

Riferimento alla configurazione Parquet

Puoi usare quanto segue `format_options` ovunque le librerie AWS Glue lo specifichino`format="parquet"`:

- `useGlueParquetWriter`— Specifica l'uso di un writer Parquet personalizzato con ottimizzazioni delle prestazioni per DynamicFrame i flussi di lavoro. Per informazioni dettagliate sull'utilizzo, consulta Writer Glue Parquet.
  - Tipo: booleano, Valore predefinito: `false`
- `compression`: specifica il codec di compressione utilizzato. I valori sono pienamente compatibili con `org.apache.parquet.hadoop.metadata.CompressionCodecName`.
  - Tipo: testo enumerato, Valore predefinito: `"snappy"`

  - Valori: `"uncompressed"`, `"snappy"`, `"gzip"` e `"lzo"`
- `blockSize`: specifica la dimensione in byte di un gruppo di righe memorizzate nel buffer in memoria. Utilizzi questo valore per ottimizzare le prestazioni. Le dimensioni dovrebbero dividersi esattamente in un numero di megabyte.
  - Tipo: numerico, Valore predefinito: `134217728`
  - Il valore predefinito è 128 MB.
- `pageSize`: specifica le dimensioni in byte di una pagina. Utilizzi questo valore per ottimizzare le prestazioni. Una pagina è l'unità più piccola che deve essere letta interamente per accedere a un singolo record.
  - Tipo: numerico, Valore predefinito: `1048576`
  - Il valore predefinito è 1 MB.

> **Note**
>
> Inoltre, tutte le opzioni accettate dal codice SparkSQL sottostante possono essere passate tramite il parametro mappa `connection_options`. Ad esempio, puoi impostare una configurazione Spark come mergeSchema per il reader Spark di AWS Glue per unire lo schema di tutti i file.

Ottimizza le prestazioni di scrittura con AWS Glue Parquet writer

> **Note**
>
> Storicamente si accedeva al writer AWS Glue Parquet tramite il tipo di `glueparquet` formato. Questo schema di accesso non è più raccomandato. Utilizza invece il tipo `parquet` con `useGlueParquetWriter` abilitato.

Il AWS masterizzatore Glue Parquet presenta miglioramenti delle prestazioni che consentono una scrittura più rapida dei file Parquet. Il writer tradizionale calcola uno schema prima della scrittura. Il formato Parquet non memorizza lo schema in un modo recuperabile rapidamente, quindi questa operazione potrebbe richiedere del tempo. Con lo scrittore AWS Glue Parquet, non è necessario uno schema precalcolato. Quando arrivano i dati, il writer calcola e modifica lo schema in modo dinamico.

Quando specifichi `useGlueParquetWriter`, tieni presente le seguenti limitazioni:

- Il writer supporta solo l'evoluzione dello schema, come l'aggiunta o la rimozione di colonne, ma non la modifica dei tipi di colonna, ad esempio con `ResolveChoice`.
- Lo scrittore non supporta la scrittura vuota, ad esempio DataFrames per scrivere un file contenente solo uno schema. Quando si esegue l'integrazione con AWS Glue Data Catalog tramite impostazione`enableUpdateCatalog=True`, il tentativo di scrivere uno spazio vuoto non DataFrame aggiornerà il Data Catalog. Il nome di una tabella nel Catalogo dati.

Se la trasformazione non richiede queste limitazioni, l'attivazione dello scrittore AWS Glue Parquet dovrebbe aumentare le prestazioni.

Utilizzo del formato XML in AWS Glue

AWS Glue recupera i dati dalle fonti e li scrive su destinazioni archiviate e trasportate in vari formati di dati. Se i dati vengono archiviati o trasportati nel formato di dati XML, questo documento presenta le funzionalità disponibili per l'utilizzo dei dati in AWS Glue.

AWS Glue supporta l'utilizzo del formato XML. Questo formato rappresenta strutture dati altamente configurabili e rigidamente definite che non sono basate su righe o colonne. XML è altamente standardizzato. Per un'introduzione al formato da parte dell'autorità di standard, consulta Informazioni essenziali su XML.

Puoi usare AWS Glue per leggere file XML da Amazon S3 `bzip` e `gzip` archivi contenenti file XML. Puoi configurare il comportamento di compressione sul Parametri di connessione di S3 invece che nella configurazione discussa in questa pagina.

La tabella seguente mostra quali funzioni comuni di AWS Glue supportano l'opzione di formato XML.

| Lettura | Scrittura | Lettura in streaming | Gruppo di file piccoli | Segnalibri di processo |
|---|---|---|---|---|
| Supportata | Non supportato. | Non supportato. | Supportata | Supportata |

Esempio: lettura di XML da S3

Il reader XML assume un nome di tag XML. Esamina gli elementi con quel tag all'interno del relativo input per dedurre uno schema e compila un DynamicFrame con i valori corrispondenti. La funzionalità AWS Glue XML si comporta in modo simile alla sorgente dati XML per Apache Spark. Potresti essere

in grado di ottenere informazioni sul comportamento di base confrontando questo reader con la documentazione di quel progetto.

Prerequisiti: avrai bisogno dei percorsi S3 (`s3path`) nei file XML o nelle cartelle che desideri leggere e di alcune informazioni sul file XML. Avrai bisogno anche del tag per l'elemento XML che desideri leggere, `xmlTag`.

Configurazione: nelle opzioni della funzione, specifica `format="xml"`. In `connection_options`, utilizza la chiave `paths` per specificare `s3path`. Puoi configurare il modo in cui il reader interagisce con S3 in `connection_options`. Per i dettagli, vedi Tipi di connessione e opzioni per ETL in AWS Glue:Parametri di connessione di S3. In `format_options`, utilizza la chiave `rowTag` per specificare `xmlTag`. Puoi configurare il modo in cui il reader interpreta i file XML nel tuo `format_options`. Per maggiori dettagli, consulta Riferimento alla configurazione XML.

Il seguente script ETL di AWS Glue mostra il processo di lettura di file o cartelle XML da S3.

Python

Per questo esempio, utilizza il metodo create_dynamic_frame.from_options.

```
# Example: Read XML from S3
# Set the rowTag option to configure the reader.

from awsglue.context import GlueContext
from pyspark.context import SparkContext

sc = SparkContext.getOrCreate()
glueContext = GlueContext(sc)

dynamicFrame = glueContext.create_dynamic_frame.from_options(
    connection_type="s3",
    connection_options={"paths": ["s3://s3path"]},
    format="xml",
    format_options={"rowTag": "xmlTag"},
)
```

Puoi anche usare DataFrames in uno script (`pyspark.sql.DataFrame`).

```
dataFrame = spark.read\
    .format("xml")\
    .option("rowTag", "xmlTag")\
```

```
    .load("s3://s3path")
```

Scala

Per questo esempio, utilizzate l'operazione getSourceWithFormat.

```
// Example: Read XML from S3
// Set the rowTag option to configure the reader.

import com.amazonaws.services.glue.util.JsonOptions
import com.amazonaws.services.glue.GlueContext
import org.apache.spark.sql.SparkSession

val glueContext = new GlueContext(SparkContext.getOrCreate())
val sparkSession: SparkSession = glueContext.getSparkSession

object GlueApp {
  def main(sysArgs: Array[String]): Unit = {
    val dynamicFrame = glueContext.getSourceWithFormat(
      formatOptions=JsonOptions("""{"rowTag": "xmlTag"}"""),
      connectionType="s3",
      format="xml",
      options=JsonOptions("""{"paths": ["s3://s3path"], "recurse": true}""")
    ).getDynamicFrame()
}
```

È inoltre possibile utilizzare DataFrames in uno script (`org.apache.spark.sql.DataFrame`).

```
val dataFrame = spark.read
  .option("rowTag", "xmlTag")
  .format("xml")
  .load("s3://s3path“)
```

Riferimento alla configurazione XML

Puoi usare quanto segue `format_options` ovunque le librerie AWS Glue lo specifichino`format="xml"`:

- `rowTag`: specifica il tag XML nel file da trattare come riga. I tag di riga non possono essere con chiusura automatica.

  - Tipo: testo, obbligatorio
- `encoding`: specifica la codifica dei caratteri. Può essere il nome o l'alias di un Charset supportato dal nostro ambiente di runtime. Non forniamo garanzie specifiche sul supporto della codifica, ma le codifiche principali dovrebbero funzionare.
  - Tipo: testo, Valore predefinito: `"UTF-8"`
- `excludeAttribute`: specifica se escludere o meno gli attributi negli elementi.
  - Tipo: booleano, Valore predefinito: `false`
- `treatEmptyValuesAsNulls`: specifica se trattare uno spazio vuoto come valore null.
  - Tipo: booleano, Valore predefinito: `false`
- `attributePrefix`: un prefisso per gli attributi, per differenziarli dal testo degli elementi figlio. Questo prefisso viene usato per i nomi di campi.
  - Tipo: testo, Valore predefinito: `"_"`
- `valueTag`: il tag usato per un valore quando ci sono attributi nell'elemento che non hanno elementi figlio.
  - Tipo: testo, Valore predefinito: `"_VALUE"`
- `ignoreSurroundingSpaces`: specifica se lo spazio vuoto intorno ai valori deve essere ignorato.
  - Tipo: booleano, Valore predefinito: `false`
- `withSchema`: contiene lo schema previsto, in situazioni in cui si desidera sovrascrivere lo schema dedotto. Se non utilizzi questa opzione, AWS Glue deduce lo schema dai dati XML.
  - Tipo: testo, Valore predefinito: non applicabile
  - Il valore deve essere un oggetto JSON che rappresenta un `StructType`.

Specifica manuale dello schema XML

Esempio di schema XML manuale

Questo è un esempio dell'utilizzo di `withSchema` per specificare lo schema per i dati XML.

```
from awsglue.gluetypes import *

schema = StructType([
  Field("id", IntegerType()),
  Field("name", StringType()),
  Field("nested", StructType([
    Field("x", IntegerType()),
```

```
    Field("y", StringType()),
    Field("z", ChoiceType([IntegerType(), StringType()]))
  ]))
])

datasource0 = create_dynamic_frame_from_options(
    connection_type,
    connection_options={"paths": ["s3://xml_bucket/someprefix"]},
    format="xml",
    format_options={"withSchema": json.dumps(schema.jsonValue())},
    transformation_ctx = ""
)
```

Utilizzo del formato Avro in AWS Glue

AWS Glue recupera i dati dalle fonti e li scrive su destinazioni archiviate e trasportate in vari formati di dati. Se i dati vengono archiviati o trasportati nel formato dati Avro, questo documento presenta le funzionalità disponibili per l'utilizzo dei dati in Glue. AWS

AWS Glue supporta l'utilizzo del formato Avro. Questo formato è un formato dati basato su righe orientato alle prestazioni. Per un'introduzione al formato da parte dell'autorità degli standard, consulta la Documentazione di Apache Avro 1.8.2.

Puoi usare AWS Glue per leggere file Avro da Amazon S3 e da sorgenti di streaming, nonché scrivere file Avro su Amazon S3. Puoi leggere e scrivere archivi `bzip2` e `gzip` contenenti file Avro da S3. Inoltre, è possibile scrivere archivi `deflate`, `snappy` e `xz` contenenti file Avro. Puoi configurare il comportamento di compressione sul Parametri di connessione di S3 invece che nella configurazione discussa in questa pagina.

La tabella seguente mostra quali operazioni comuni di AWS Glue supportano l'opzione di formato Avro.

| Lettura | Scrittura | Lettura in streaming | Gruppo di file piccoli | Segnalibri di processo |
|---|---|---|---|---|
| Supportata | Supportata | Supportato* | Non supportato. | Supportata |

* Supportato con restrizioni. Per ulteriori informazioni, consulta the section called "Note e restrizioni per le origini di streaming Avro".

Esempio: lettura di cartelle o file Avro da S3

Prerequisiti: occorrono i percorsi S3 (`s3path`) nelle cartelle o nei file Avro da leggere.

Configurazione: nelle opzioni della funzione, specifica `format="avro"`. In `connection_options`, utilizzare la chiave `paths` per specificare `s3path`. Puoi configurare il modo in cui il reader interagisce con S3 in `connection_options`. Per i dettagli, vedi Opzioni di formato dei dati per ingressi e uscite ETL in AWS Glue:. the section called "Parametri di connessione di S3" Puoi configurare la modalità con cui il reader interpreta i file Avro CSV in `format_options`. Per i dettagli, consulta Documentazione di riferimento della configurazione Avro.

Il seguente script AWS Glue ETL mostra il processo di lettura dei file o delle cartelle Avro da S3:

Python

Per questo esempio, utilizzare il metodo create_dynamic_frame.from_options.

```
from pyspark.context import SparkContext
from awsglue.context import GlueContext

sc = SparkContext.getOrCreate()
glueContext = GlueContext(sc)

dynamicFrame = glueContext.create_dynamic_frame.from_options(
    connection_type="s3",
    connection_options={"paths": ["s3://s3path"]},
    format="avro"
)
```

Scala

Per questo esempio, utilizzate l'operazione Format. getSourceWith

```
import com.amazonaws.services.glue.util.JsonOptions
import com.amazonaws.services.glue.GlueContext
import org.apache.spark.sql.SparkContext

object GlueApp {
  def main(sysArgs: Array[String]): Unit = {
    val spark: SparkContext = new SparkContext()
    val glueContext: GlueContext = new GlueContext(spark)

    val dynamicFrame = glueContext.getSourceWithFormat(
```

```
      connectionType="s3",
      format="avro",
      options=JsonOptions("""{"paths": ["s3://s3path"]}""")
    ).getDynamicFrame()
  }
```

Esempio: scrittura di file e cartelle Avro in S3

Prerequisiti: è necessario un DataFrame (`dataFrame`) o DynamicFrame (`dynamicFrame`) inizializzato. È necessario anche del tuo percorso di output S3 previsto, `s3path`.

Configurazione: nelle opzioni della funzione, specifica `format="avro"`. Nelle tue `connection_options`, utilizza la chiave `paths` per specificare `s3path`. È possibile modificare ulteriormente il modo in cui il writer interagisce con S3 nelle `connection_options`. Per i dettagli, vedi Opzioni di formato dei dati per ingressi e uscite ETL in AWS Glue:. the section called "Parametri di connessione di S3" Puoi modificare la modalità con cui il writer interpreta i file Avro in `format_options`. Per i dettagli, consulta Documentazione di riferimento della configurazione Avro.

Il seguente script AWS Glue ETL mostra il processo di scrittura di file o cartelle Avro su S3.

Python

Per questo esempio, utilizza il metodo write_dynamic_frame.from_options.

```
from pyspark.context import SparkContext
from awsglue.context import GlueContext

sc = SparkContext.getOrCreate()
glueContext = GlueContext(sc)

glueContext.write_dynamic_frame.from_options(
    frame=dynamicFrame,
    connection_type="s3",
    format="avro",
    connection_options={
        "path": "s3://s3path"
    }
)
```

Scala

Per questo esempio, utilizzate il metodo Format. getSinkWith

```
import com.amazonaws.services.glue.util.JsonOptions
import com.amazonaws.services.glue.{DynamicFrame, GlueContext}
import org.apache.spark.SparkContext

object GlueApp {
  def main(sysArgs: Array[String]): Unit = {
    val spark: SparkContext = new SparkContext()
    val glueContext: GlueContext = new GlueContext(spark)

    glueContext.getSinkWithFormat(
      connectionType="s3",
      options=JsonOptions("""{"path": "s3://s3path"}"""),
      format="avro"
    ).writeDynamicFrame(dynamicFrame)
  }
}
```

Documentazione di riferimento della configurazione Avro

È possibile utilizzare i seguenti `format_options` valori ovunque specifichino le librerie AWS Glue`format="avro"`:

- `version`— Speciifica la versione del reader/writer formato Apache Avro da supportare. Il valore predefinito è 1.7. Puoi specificare `format_options={"version": “1.8”}` per abilitare la lettura e la scrittura del tipo logico Avro. Per ulteriori informazioni, consulta Specifiche di Apache Avro 1.7.7 e Specifiche di Apache Avro 1.8.2.

  Il connettore Apache Avro 1.8 supporta le seguenti conversioni di tipo logico:

Per le operazioni di lettura: questa tabella mostra la conversione tra il tipo di dati Avro (tipo logico e tipo primitivo Avro) e il tipo di dati AWS Glue `DynamicFrame` per le operazioni di lettura Avro 1.7 e 1.8.

| Tipo di dati Avro: Tipo logico | Tipo di dati Avro: Tipi primitivi .NET | GlueDynamicFrame Tipo di dati: Avro Reader 1.7 | GlueDynamicFrame Tipo di dati: Avro Reader 1.8 |
|---|---|---|---|
| Decimale | bytes | BINARY | Decimale |

| Tipo di dati Avro:<br><br>Tipo logico | Tipo di dati Avro:<br><br>Tipi primitivi .NET | GlueDynamicFrame Tipo di dati:<br><br>Avro Reader 1.7 | GlueDynamicFrame Tipo di dati:<br><br>Avro Reader 1.8 |
|---|---|---|---|
| Decimale | fisso | BINARY | Decimale |
| Data | int | INT | Data |
| Tempo (millisecondi) | int | INT | INT |
| Tempo (microsecondi) | Long | LONG | LONG |
| Timestamp (millisecondi) | Long | LONG | Time stamp |
| Timestamp (microsecondi) | Long | LONG | LONG |
| Durata (non un tipo logico) | fisso di 12 | BINARY | BINARY |

Per le operazioni di scrittura: questa tabella mostra la conversione tra il tipo di dati AWS Glue `DynamicFrame` e il tipo di dati Avro per le operazioni di scrittura Avro 1.7 e 1.8.

| Tipo di dati AWS Glue **`DynamicFrame`** | Tipo di dati Avro:<br><br>Avro Writer 1.7 | Tipo di dati Avro:<br><br>Avro Writer 1.8 |
|---|---|---|
| Decimale | Stringa | decimal |
| Data | Stringa | data |
| Time stamp | Stringa | timestamp-micros |

Supporto Avro Spark DataFrame

Per utilizzare Avro dall' DataFrame API Spark, devi installare il plug-in Spark Avro per la versione Spark corrispondente. La versione di Spark disponibile nel tuo job è determinata dalla versione di

AWS Glue in uso. Per ulteriori informazioni sulle versioni di Spark, consulta the section called "AWS Glue versioni". Questo plugin è gestito da Apache; non forniamo garanzie specifiche sul supporto.

In AWS Glue 2.0, usa la versione 2.4.3 del plugin Spark Avro. Puoi trovare questo JAR su Maven Central, consulta org.apache.spark:spark-avro_2.12:2.4.3.

In AWS Glue 3.0, usa la versione 3.1.1 del plugin Spark Avro. Puoi trovare questo JAR su Maven Central, consulta org.apache.spark:spark-avro_2.12:3.1.1.

Per includere elementi aggiuntivi JARs in un lavoro AWS Glue ETL, utilizzate il parametro `--extra-jars` job. Per ulteriori informazioni sui parametri di processo, consulta the section called "Parametri Job". Puoi configurare questo parametro anche nella Console di gestione AWS.

Utilizzo del formato GrokLog in Glue AWS

AWS Glue recupera i dati dalle fonti e li scrive su destinazioni archiviate e trasportate in vari formati di dati. Se i dati vengono archiviati o trasportati in un formato di testo semplice strutturato in modo lasco, questo documento presenta le funzionalità disponibili per l'utilizzo dei dati nei modelli AWS Glue through Grok.

AWS Glue supporta utilizzando modelli Grok. I modelli Grok sono simili a gruppi di acquisizione di espressioni regolari. Riconoscono modelli di sequenze di caratteri in un file di testo semplice e forniscono loro un tipo e uno scopo. In AWS Glue, il loro scopo principale è leggere i log. Per un'introduzione a Grok da parte degli autori, consulta Documentazione di riferimento di Logstash: plugin per filtri Grok.

| Lettura | Scrittura | Lettura in streaming | Gruppo di file piccoli | Segnalibri di processo |
|---|---|---|---|---|
| Supportata | Non applicabile | Supportata | Supportata | Non supportato. |

Documentazione di riferimento sulla configurazione grokLog

È possibile usare i valori di `format_options` seguenti con `format="grokLog"`:

- `logFormat`: specifica il pattern grok corrispondente al formato del log.
- `customPatterns`: specifica ulteriori pattern Grok usati.
- `MISSING`: specifica il segnale da usare per identificare i valori mancanti. Il valore predefinito è `'-'`.

- `LineCount`: specifica il numero di righe in ogni record di log. Il valore predefinito è `'1'` e attualmente sono supportati solo record a riga singola.
- `StrictMode`: valore booleano che specifica se la modalità strict è abilitata. In modalità strict, il lettore non esegue la conversione o il ripristino automatico dei tipi. Il valore predefinito è `"false"`.

### Utilizzo del formato Ion in AWS Glue

AWS Glue recupera i dati dalle fonti e li scrive su destinazioni archiviate e trasportate in vari formati di dati. Se i dati vengono archiviati o trasportati nel formato dati Ion, questo documento presenta le funzionalità disponibili per l'utilizzo dei dati in AWS Glue.

AWS Glue supporta l'utilizzo del formato Ion. Questo formato rappresenta strutture di dati (che non sono basate su righe o colonne) in rappresentazioni binarie e di testo semplice intercambiabili. Per un'introduzione al formato da parte degli autori, consulta Amazon Ion. Per ulteriori informazioni consulta la specifica Amazon Ion.

Puoi usare AWS Glue per leggere file Ion da Amazon S3. Puoi leggere e scrivere archivi `bzip` e `gzip` contenenti file Ion da S3. Puoi configurare il comportamento di compressione sul Parametri di connessione di S3 invece che nella configurazione discussa in questa pagina.

La tabella seguente mostra quali operazioni comuni di AWS Glue supportano l'opzione di formato Ion.

| Lettura | Scrittura | Lettura in streaming | Gruppo di file piccoli | Segnalibri di processo |
|---|---|---|---|---|
| Supportata | Non supportato. | Non supportato. | Supportata | Non supportato. |

### Esempio: lettura di cartelle e file Ion da S3

Prerequisiti: occorreranno i percorsi S3 (`s3path`) nelle cartelle o nei file Ion da leggere.

Configurazione: nelle opzioni della funzione, specifica `format="json"`. Nelle tue `connection_options`, utilizza la chiave `paths` per specificare `s3path`. Puoi configurare il modo in cui il reader interagisce con S3 in `connection_options`. Per i dettagli, vedi Tipi di connessione e opzioni per ETL in AWS Glue:the section called "Parametri di connessione di S3".

Il seguente script AWS Glue ETL mostra il processo di lettura di file o cartelle Ion da S3:

Python

Per questo esempio, utilizzare il metodo create_dynamic_frame.from_options.

```
# Example: Read ION from S3

from pyspark.context import SparkContext
from awsglue.context import GlueContext

sc = SparkContext.getOrCreate()
glueContext = GlueContext(sc)

dynamicFrame = glueContext.create_dynamic_frame.from_options(
    connection_type="s3",
    connection_options={"paths": ["s3://s3path"]},
    format="ion"
)
```

Scala

Per questo esempio, utilizzate l'operazione getSourceWithFormat.

```
// Example: Read ION from S3

import com.amazonaws.services.glue.util.JsonOptions
import com.amazonaws.services.glue.GlueContext
import org.apache.spark.SparkContext

object GlueApp {
  def main(sysArgs: Array[String]): Unit = {
    val spark: SparkContext = new SparkContext()
    val glueContext: GlueContext = new GlueContext(spark)

    val dynamicFrame = glueContext.getSourceWithFormat(
      connectionType="s3",
      format="ion",
      options=JsonOptions("""{"paths": ["s3://s3path"], "recurse": true}""")
    ).getDynamicFrame()
  }
}
```

Documentazione di riferimento della configurazione Ion

Non ci sono valori di `format_options` per `format="ion"`.

Utilizzo del formato JSON in AWS Glue

AWS Glue recupera i dati dalle fonti e li scrive su destinazioni archiviate e trasportate in vari formati di dati. Se i dati vengono archiviati o trasportati nel formato dati JSON, questo documento presenta le funzionalità disponibili per l'utilizzo dei dati in Glue. AWS

AWS Glue supporta l'utilizzo del formato JSON. Questo formato rappresenta strutture di dati con forma coerente ma contenuti flessibili, che non sono basate su righe o colonne. JSON è definito tramite standard paralleli emessi da diverse autorità, una delle quali è ECMA-404. Per un'introduzione al formato da una fonte di riferimento comune, consulta Introduzione a JSON.

Puoi usare AWS Glue per leggere file JSON da Amazon S3 `gzip` e file JSON compressi. `bzip` Puoi configurare il comportamento di compressione sul Parametri di connessione di S3 invece che nella configurazione discussa in questa pagina.

| Lettura | Scrittura | Lettura in streaming | Gruppo di file piccoli | Segnalibri di processo | |
|---|---|---|---|---|---|
| Supportata | Supportato | Supportato | Supportato | Supportata | |

Esempio: lettura di cartelle o file JSON da S3

Prerequisiti: occorrono i percorsi S3 (`s3path`) nelle cartelle o nei file JSON da leggere.

Configurazione: nelle opzioni della funzione, specifica `format="json"`. Nelle tue `connection_options`, utilizza la chiave `paths` per specificare `s3path`. Puoi modificare ulteriormente la modalità con cui l'operazione di lettura attraversa s3 nelle opzioni di connessione; consulta the section called "Parametri di connessione di S3" per dettagli. Puoi configurare la modalità con cui il reader interpreta i file JSON in `format_options`. Per i dettagli, consulta la Documentazione di riferimento della configurazione JSON.

Il seguente script AWS Glue ETL mostra il processo di lettura di file o cartelle JSON da S3:

Python

Per questo esempio, utilizzare il metodo create_dynamic_frame.from_options.

```
# Example: Read JSON from S3
# For show, we handle a nested JSON file that we can limit with the JsonPath
 parameter
# For show, we also handle a JSON where a single entry spans multiple lines
# Consider whether optimizePerformance is right for your workflow.

from pyspark.context import SparkContext
from awsglue.context import GlueContext

sc = SparkContext.getOrCreate()
glueContext = GlueContext(sc)
spark = glueContext.spark_session

dynamicFrame = glueContext.create_dynamic_frame.from_options(
    connection_type="s3",
    connection_options={"paths": ["s3://s3path"]},
    format="json",
    format_options={
        "jsonPath": "$.id",
        "multiline": True,
        # "optimizePerformance": True, -> not compatible with jsonPath, multiline
    }
)
```

Puoi anche usarlo DataFrames in uno script (). `pyspark.sql.DataFrame`

```
dataFrame = spark.read\
    .option("multiline", "true")\
    .json("s3://s3path")
```

### Scala

Per questo esempio, utilizzate l'operazione getSourceWithFormat.

```
// Example: Read JSON from S3
// For show, we handle a nested JSON file that we can limit with the JsonPath
 parameter
// For show, we also handle a JSON where a single entry spans multiple lines
// Consider whether optimizePerformance is right for your workflow.

import com.amazonaws.services.glue.util.JsonOptions
import com.amazonaws.services.glue.{DynamicFrame, GlueContext}
import org.apache.spark.SparkContext
```

```
object GlueApp {
  def main(sysArgs: Array[String]): Unit = {
    val spark: SparkContext = new SparkContext()
    val glueContext: GlueContext = new GlueContext(spark)

    val dynamicFrame = glueContext.getSourceWithFormat(
      formatOptions=JsonOptions("""{"jsonPath": "$.id", "multiline": true,
 "optimizePerformance":false}"""),
      connectionType="s3",
      format="json",
      options=JsonOptions("""{"paths": ["s3://s3path"], "recurse": true}""")
    ).getDynamicFrame()
  }
}
```

È inoltre possibile utilizzare DataFrames in uno script (`pyspark.sql.DataFrame`).

```
val dataFrame = spark.read
    .option("multiline", "true")
    .json("s3://s3path")
```

Esempio: scrittura di file e cartelle JSON su S3

Prerequisiti: è necessario un DataFrame (`dataFrame`) o DynamicFrame (`dynamicFrame`) inizializzato. È necessario anche del tuo percorso di output S3 previsto, `s3path`.

Configurazione: nelle opzioni della funzione, specifica `format="json"`. In `connection_options`, utilizzare la chiave `paths` per specificare `s3path`. È possibile modificare ulteriormente il modo in cui il writer interagisce con S3 nelle `connection_options`. Per i dettagli, vedi Opzioni di formato dei dati per ingressi e uscite ETL in AWS Glue:. the section called "Parametri di connessione di S3" È possibile configurare la modalità con cui il writer interpreta i file JSON in `format_options`. Per i dettagli, consultare la Documentazione di riferimento della configurazione JSON.

Il seguente script AWS Glue ETL mostra il processo di scrittura di file o cartelle JSON da S3:

Python

Per questo esempio, utilizza il metodo write_dynamic_frame.from_options.

```
# Example: Write JSON to S3
```

```
from pyspark.context import SparkContext
from awsglue.context import GlueContext

sc = SparkContext.getOrCreate()
glueContext = GlueContext(sc)

glueContext.write_dynamic_frame.from_options(
    frame=dynamicFrame,
    connection_type="s3",
    connection_options={"path": "s3://s3path"},
    format="json"
)
```

Puoi anche usarlo DataFrames in uno script (). `pyspark.sql.DataFrame`

```
df.write.json("s3://s3path/")
```

Scala

Per questo esempio, utilizzate il metodo getSinkWithFormat.

```
// Example: Write JSON to S3

import com.amazonaws.services.glue.util.JsonOptions
import com.amazonaws.services.glue.{DynamicFrame, GlueContext}
import org.apache.spark.SparkContext

object GlueApp {
  def main(sysArgs: Array[String]): Unit = {
    val spark: SparkContext = new SparkContext()
    val glueContext: GlueContext = new GlueContext(spark)

    glueContext.getSinkWithFormat(
        connectionType="s3",
        options=JsonOptions("""{"path": "s3://s3path"}"""),
        format="json"
    ).writeDynamicFrame(dynamicFrame)
  }
}
```

È inoltre possibile utilizzare DataFrames in uno script (`pyspark.sql.DataFrame`).

```
df.write.json("s3://s3path")
```

Documentazione di riferimento della configurazione JSON

È possibile usare i valori di `format_options` seguenti con `format="json"`:

- `jsonPath`— Un'JsonPathespressione che identifica un oggetto da leggere nei record. È particolarmente utile quando un file contiene registri annidati in una matrice esterna. Ad esempio, l' JsonPath espressione seguente si rivolge al `id` campo di un oggetto JSON.

  ```
  format="json", format_options={"jsonPath": "$.id"}
  ```

- `multiline`: un valore booleano che specifica se un singolo registro può estendersi su più righe. Ciò può accadere quando un campo contiene un carattere di nuova riga tra virgolette. Imposta questa opzione su `"true"` se i registri si estendono su più righe. Il valore di default è `"false"`, che consente una divisione dei file più netta durante l'analisi.
- `optimizePerformance`: valore booleano che specifica se utilizzare il lettore JSON SIMD avanzato insieme ai formati di memoria colonnare basati su Apache Arrow. Disponibile solo in AWS Glue 3.0. Non compatibile con `multiline` o `jsonPath`. Fornendo una di queste opzioni, AWS Glue tornerà al lettore standard.
- `withSchema`: un valore di stringa che specifica uno schema di tabella nel formato descritto in the section called "Specifica dello schema XML". Utilizzato solo con `optimizePerformance` durante la lettura da connessioni non di catalogo.

Utilizzo del lettore JSON SIMD vettorizzato con formato colonnare Apache Arrow

La versione AWS Glue 3.0 aggiunge un lettore vettorizzato per i dati JSON. Funziona 2 volte più velocemente in determinate condizioni, rispetto al lettore standard. Questo lettore presenta alcune limitazioni di cui gli utenti dovrebbero essere consapevoli prima dell'uso, documentate in questa sezione.

Per utilizzare il lettore ottimizzato, impostare `"optimizePerformance"` a True nella `format_options` o nella proprietà della tabella. È necessario anche fornire `withSchema` a meno che non venga letto dal catalogo. `withSchema` prevede un input come descritto nel the section called "Specifica dello schema XML"

```
// Read from S3 data source
glueContext.create_dynamic_frame.from_options(
    connection_type = "s3",
    connection_options = {"paths": ["s3://s3path"]},
    format = "json",
    format_options={
        "optimizePerformance": True,
        "withSchema": SchemaString
        })

// Read from catalog table
glueContext.create_dynamic_frame.from_catalog(
    database = database,
    table_name = table,
    additional_options = {
    // The vectorized reader for JSON can read your schema from a catalog table
 property.
        "optimizePerformance": True,
        })
```

Per ulteriori informazioni sull'edificio a *SchemaString* nella libreria AWS Glue, vederethe section called "Tipi".

Limitazioni per il lettore CSV vettorizzato

Nota i seguenti limiti:

- Gli elementi JSON con oggetti nidificati o valori di array non sono supportati. Se fornito, AWS Glue tornerà al lettore standard.
- È necessario fornire uno schema, dal catalogo o con `withSchema`.
- Non compatibile con `multiline` o `jsonPath`. Fornendo una di queste opzioni, AWS Glue tornerà al lettore standard.
- Fornire registri di input che non corrispondono allo schema di input provocherà il fallimento del lettore.
- I registri di errori non verranno creati.
- Non supporta file JSON con caratteri multibyte (come caratteri giapponesi o cinesi).

Utilizzo del formato ORC in AWS Glue

AWS Glue recupera i dati dalle fonti e li scrive su destinazioni archiviate e trasportate in vari formati di dati. Se i dati vengono archiviati o trasportati nel formato dati ORC, questo documento presenta le funzionalità disponibili per l'utilizzo dei dati in Glue. AWS

AWS Glue supporta l'utilizzo del formato ORC. Questo formato è un formato dati basato su colonne orientato alle prestazioni. Per un'introduzione al formato da parte dell'autorità degli standard, consulta Apache Orc.

Puoi usare AWS Glue per leggere file ORC da Amazon S3 e da sorgenti di streaming, nonché scrivere file ORC su Amazon S3. Puoi leggere e scrivere archivi `bzip` e `gzip` contenenti file ORC da S3. Puoi configurare il comportamento di compressione sul Parametri di connessione di S3 invece che nella configurazione discussa in questa pagina.

La tabella seguente mostra quali operazioni comuni di AWS Glue supportano l'opzione di formato ORC.

| Lettura | Scrittura | Lettura in streaming | Gruppo di file piccoli | Segnalibri di processo |
|---|---|---|---|---|
| Supportata | Supportato | Supportata | Non supportato. | Supportato* |

* Supportato nella versione AWS Glue 1.0+

Esempio: lettura di cartelle o file ORC da S3

Prerequisiti: occorreranno i percorsi S3 (`s3path`) nelle cartelle o nei file ORC da leggere.

Configurazione: nelle opzioni della funzione, specifica `format="orc"`. Nelle tue `connection_options`, utilizza la chiave `paths` per specificare `s3path`. Puoi configurare il modo in cui il reader interagisce con S3 in `connection_options`. Per i dettagli, vedi Tipi di connessione e opzioni per ETL in AWS Glue:the section called "Parametri di connessione di S3".

Il seguente script AWS Glue ETL mostra il processo di lettura di file o cartelle ORC da S3:

Python

Per questo esempio, utilizzare il metodo create_dynamic_frame.from_options.

```
from pyspark.context import SparkContext
```

```
from awsglue.context import GlueContext

sc = SparkContext.getOrCreate()
glueContext = GlueContext(sc)

dynamicFrame = glueContext.create_dynamic_frame.from_options(
    connection_type="s3",
    connection_options={"paths": ["s3://s3path"]},
    format="orc"
)
```

Puoi anche usarlo DataFrames in uno script (). `pyspark.sql.DataFrame`

```
dataFrame = spark.read\
    .orc("s3://s3path")
```

Scala

Per questo esempio, utilizzate l'operazione getSourceWithFormat.

```
import com.amazonaws.services.glue.util.JsonOptions
import com.amazonaws.services.glue.GlueContext
import org.apache.spark.sql.SparkContext

object GlueApp {
  def main(sysArgs: Array[String]): Unit = {
    val spark: SparkContext = new SparkContext()
    val glueContext: GlueContext = new GlueContext(spark)

    val dynamicFrame = glueContext.getSourceWithFormat(
      connectionType="s3",
      format="orc",
      options=JsonOptions("""{"paths": ["s3://s3path"]}""")
    ).getDynamicFrame()
  }
}
```

È inoltre possibile utilizzare DataFrames in uno script (`pyspark.sql.DataFrame`).

```
val dataFrame = spark.read
    .orc("s3://s3path")
```

Esempio: scrittura di cartelle e file ORC in S3

Prerequisiti: è necessario un DataFrame (`dataFrame`) o DynamicFrame (`dynamicFrame`) inizializzato. È necessario anche del tuo percorso di output S3 previsto, `s3path`.

Configurazione: nelle opzioni della funzione, specifica `format="orc"`. Nelle opzioni di connessione, usa la chiave `paths` per specificare `s3path`. È possibile modificare ulteriormente il modo in cui il writer interagisce con S3 nelle `connection_options`. Per i dettagli, vedi Opzioni di formato dei dati per ingressi e uscite ETL in AWS Glue:. the section called "Parametri di connessione di S3" L'esempio di codice seguente mostra il processo:

Python

Per questo esempio, utilizza il metodo write_dynamic_frame.from_options.

```
from pyspark.context import SparkContext
from awsglue.context import GlueContext

sc = SparkContext.getOrCreate()
glueContext = GlueContext(sc)

glueContext.write_dynamic_frame.from_options(
    frame=dynamicFrame,
    connection_type="s3",
    format="orc",
    connection_options={
        "path": "s3://s3path"
    }
)
```

Puoi anche usare DataFrames in uno script (). `pyspark.sql.DataFrame`

```
df.write.orc("s3://s3path/")
```

Scala

Per questo esempio, utilizzate il metodo getSinkWithFormat.

```
import com.amazonaws.services.glue.util.JsonOptions
import com.amazonaws.services.glue.{DynamicFrame, GlueContext}
import org.apache.spark.SparkContext
```

```
object GlueApp {
  def main(sysArgs: Array[String]): Unit = {
    val spark: SparkContext = new SparkContext()
    val glueContext: GlueContext = new GlueContext(spark)

    glueContext.getSinkWithFormat(
      connectionType="s3",
      options=JsonOptions("""{"path": "s3://s3path"}"""),
      format="orc"
    ).writeDynamicFrame(dynamicFrame)
  }
}
```

È inoltre possibile utilizzare DataFrames in uno script (`pyspark.sql.DataFrame`).

```
df.write.orc("s3://s3path/")
```

Riferimento alla configurazione XML

Non ci sono valori di `format_options` per `format="orc"`. Tutte le opzioni accettate dal codice SparkSQL sottostante possono tuttavia essere passate tramite il parametro mappa `connection_options`.

Utilizzo di framework di data lake con AWS Glue processi ETL

I framework di data lake open source semplificano l'elaborazione incrementale dei dati per i file archiviati in data lake basati su Amazon S3. AWS Glue 3.0 e versioni successive supportano i seguenti framework di data lake open source:

- Apache Hudi
- Linux Foundation Delta Lake
- Apache Iceberg

Forniamo supporto nativo per questi framework in modo che sia possibile leggere e scrivere i dati archiviati in Amazon S3 in modo coerente dal punto di vista transazionale. Non è necessario installare un connettore separato o completare passaggi di configurazione aggiuntivi per utilizzare questi framework nei processi ETL di AWS Glue .

Quando gestisci i set di dati tramite AWS Glue Data Catalog, puoi utilizzare AWS Glue metodi per leggere e scrivere tabelle di data lake con Spark. DataFrames Puoi anche leggere e scrivere dati Amazon S3 utilizzando l'API DataFrame Spark.

Questo video illustra le basi del funzionamento di Apache Hudi, Apache Iceberg e Delta Lake. Scoprirai come inserire, aggiornare ed eliminare i dati nel tuo data lake e come funziona ciascuno di questi framework.

Argomenti


- Limitazioni
- Utilizzo del framework Hudi in AWS Glue
- Utilizzo del framework Delta Lake in AWS Glue
- Utilizzo del framework Iceberg in AWS Glue


## Limitazioni

Considera le seguenti limitazioni prima di utilizzare i framework di data lake con. AWS Glue

- I seguenti AWS Glue `GlueContext` metodi DynamicFrame non supportano la lettura e la scrittura di tabelle del framework Data Lake. Utilizza invece i `GlueContext` metodi DataFrame per l' DataFrame API Spark.
  - `create_dynamic_frame.from_catalog`
  - `write_dynamic_frame.from_catalog`
  - `getDynamicFrame`
  - `writeDynamicFrame`
- I seguenti `GlueContext` metodi DataFrame sono supportati con il controllo dei permessi di Lake Formation:
  - `create_data_frame.from_catalog`
  - `write_data_frame.from_catalog`
  - `getDataFrame`
  - `writeDataFrame`
- Il raggruppamento di file di piccole dimensioni non è supportato.
- I segnalibri dei processi non sono supportati.
- Apache Hudi 0.10.1 per AWS Glue 3.0 non supporta le tabelle Hudi Merge on Read (MoR).

- `ALTER TABLE … RENAME TO`non è disponibile per Apache Iceberg 0.13.1 for 3.0. AWS Glue

Limitazioni per le tabelle in formato data lake gestite dalle autorizzazioni di Lake Formation

I formati data lake sono integrati con AWS Glue ETL tramite le autorizzazioni di Lake Formation. La creazione di un DynamicFrame utilizzo non `create_dynamic_frame` è supportata. Per maggiori informazioni, consulta i seguenti esempi:

- Esempio: lettura e scrittura della tabella Iceberg con il controllo delle autorizzazioni di Lake Formation
- Esempio: lettura e scrittura della tabella Hudi con il controllo delle autorizzazioni di Lake Formation
- Esempio: lettura e scrittura della tabella Delta Lake con il controllo delle autorizzazioni di Lake Formation

> **Note**
>
> L'integrazione con AWS Glue ETL tramite le autorizzazioni Lake Formation per Apache Hudi, Apache Iceberg e Delta Lake è supportata solo nella versione 4.0. AWS Glue

Apache Iceberg ha la migliore integrazione con AWS Glue ETL tramite le autorizzazioni di Lake Formation. Supporta quasi tutte le operazioni e include il supporto per SQL.

Hudi supporta la maggior parte delle operazioni di base, ad eccezione di quelle amministrative. Queste opzioni generalmente vengono eseguite tramite la scrittura di dataframe e specificate tramite `additional_options`. È necessario utilizzare per creare AWS Glue APIs DataFrames per le proprie operazioni poiché SparkSQL non è supportato.

Delta Lake supporta solo la lettura, l'aggiunta e la sovrascrittura dei dati delle tabelle. Delta Lake richiede l'uso delle proprie librerie per poter eseguire varie attività come gli aggiornamenti.

Le seguenti funzionalità non sono disponibili per le tabelle Iceberg gestite dai permessi di Lake Formation.

- Compattazione tramite ETL AWS Glue
- Supporto Spark SQL tramite ETL AWS Glue

Di seguito, sono riportate le limitazioni delle tabelle Hudi gestite dai permessi di Lake Formation:

- Rimozione di file orfani

Di seguito, sono riportate le limitazioni delle tabelle Delta Lake gestite dai permessi di Lake Formation:

- Tutte le funzionalità diverse dall'inserimento e dalla lettura dalle tabelle Delta Lake.

### Utilizzo del framework Hudi in AWS Glue

AWS Glue 3.0 e versioni successive supportano il framework Apache Hudi per i data lake. Hudi è un framework di archiviazione di data lake open source che semplifica l'elaborazione incrementale dei dati e lo sviluppo di pipeline di dati. Questo argomento descrive le funzionalità disponibili per l'utilizzo dei dati in AWS Glue quando si trasportano o si archiviano i dati in una tabella Hudi. Per ulteriori informazioni su Hudi, consulta la documentazione ufficiale di Apache Hudi.

Puoi usare AWS Glue per eseguire operazioni di lettura e scrittura sulle tabelle Hudi in Amazon S3 o lavorare con le tabelle Hudi utilizzando il AWS Glue Data Catalog. Sono supportate anche operazioni aggiuntive, tra cui inserimento, aggiornamento e tutte le operazioni di Apache Spark.

> **Note**
>
> L'implementazione di Apache Hudi 0.15.0 in AWS Glue 5.0 ripristina internamente HUDI-7001. Non mostra la regressione relativa alla generazione Complex Key quando la chiave di registrazione è costituita da un singolo campo. Tuttavia, questo comportamento è diverso da OSS Apache Hudi 0.15.0.
> Apache Hudi 0.10.1 per AWS Glue 3.0 non supporta le tabelle Hudi Merge on Read (MoR).

La tabella seguente elenca la versione Hudi inclusa in ogni versione di AWS Glue.

| AWS Versione Glue | Versione Hudi supportata |
|---|---|
| 5.1 | 1.0.2 |
| 5.0 | 0.15.0 |
| 4.0 | 0.12.1 |

| AWS Versione Glue | Versione Hudi supportata |
| --- | --- |
| 3.0 | 0,10,1 |

Per ulteriori informazioni sui framework di data lake supportati da AWS Glue, consulta. Utilizzo di framework di data lake con AWS Glue processi ETL

Abilitazione di Hudi

Per abilitare Hudi for AWS Glue, completa le seguenti attività:

- Specifica `hudi` come valore per i parametri del processo `--datalake-formats`. Per ulteriori informazioni, consulta Utilizzo dei parametri del lavoro nei lavori AWS Glue.
- Crea una chiave denominata `--conf` per il tuo lavoro AWS Glue e impostala sul seguente valore. In alternativa, puoi impostare la seguente configurazione usando `SparkConf` nel tuo script. Queste impostazioni consentono ad Apache Spark di gestire correttamente le tabelle Hudi.

```
spark.serializer=org.apache.spark.serializer.KryoSerializer
```

- Il supporto delle autorizzazioni Lake Formation per Hudi è abilitato di default per AWS Glue 4.0. Non è necessaria alcuna configurazione aggiuntiva per le tabelle reading/writing Hudi registrate in Lake Formation. Per leggere una tabella Hudi registrata, il ruolo IAM di AWS Glue job deve disporre dell'autorizzazione SELECT. Per scrivere su una tabella Hudi registrata, il ruolo IAM di AWS Glue job deve avere l'autorizzazione SUPER. Per ulteriori informazioni sulla gestione delle autorizzazioni di Lake Formation, consulta Concessione e revoca delle autorizzazioni del catalogo dati.

Utilizzo di una versione differente di Hudi

Per utilizzare una versione di Hudi non supportata da AWS Glue, specificate i vostri file JAR Hudi utilizzando il parametro `--extra-jars` job. Non includere `hudi` come valore per il parametro del processo `--datalake-formats`. Se si utilizza AWS Glue 5.0 o versioni successive, è necessario impostare `--user-jars-first true` il parametro del lavoro.

Esempio: scrivere una tabella Hudi su Amazon S3 e registrarla nel AWS Glue Data Catalog

Questo script di esempio dimostra come scrivere una tabella Hudi su Amazon S3 e registrarla nel AWS Glue Data Catalog. Per registrare la tabella, viene utilizzato lo strumento Hive Sync di Hudi.

> **Note**
>
> Questo esempio richiede di impostare il parametro `--enable-glue-datacatalog` job per utilizzare AWS Glue Data Catalog come metastore Apache Spark Hive. Per ulteriori informazioni, consulta Utilizzo dei parametri del lavoro nei lavori AWS Glue.

### Python

```
# Example: Create a Hudi table from a DataFrame
# and register the table to Glue Data Catalog

additional_options={
    "hoodie.table.name": "<your_table_name>",
    "hoodie.database.name": "<your_database_name>",
    "hoodie.datasource.write.storage.type": "COPY_ON_WRITE",
    "hoodie.datasource.write.operation": "upsert",
    "hoodie.datasource.write.recordkey.field": "<your_recordkey_field>",
    "hoodie.datasource.write.precombine.field": "<your_precombine_field>",
    "hoodie.datasource.write.partitionpath.field": "<your_partitionkey_field>",
    "hoodie.datasource.write.hive_style_partitioning": "true",
    "hoodie.datasource.hive_sync.enable": "true",
    "hoodie.datasource.hive_sync.database": "<your_database_name>",
    "hoodie.datasource.hive_sync.table": "<your_table_name>",
    "hoodie.datasource.hive_sync.partition_fields": "<your_partitionkey_field>",
    "hoodie.datasource.hive_sync.partition_extractor_class":
 "org.apache.hudi.hive.MultiPartKeysValueExtractor",
    "hoodie.datasource.hive_sync.use_jdbc": "false",
    "hoodie.datasource.hive_sync.mode": "hms",
    "path": "s3://<s3Path/>"
}

dataFrame.write.format("hudi") \
    .options(**additional_options) \
    .mode("overwrite") \
    .save()
```

### Scala

```
// Example: Example: Create a Hudi table from a DataFrame
// and register the table to Glue Data Catalog
```

```
val additionalOptions = Map(
  "hoodie.table.name" -> "<your_table_name>",
  "hoodie.database.name" -> "<your_database_name>",
  "hoodie.datasource.write.storage.type" -> "COPY_ON_WRITE",
  "hoodie.datasource.write.operation" -> "upsert",
  "hoodie.datasource.write.recordkey.field" -> "<your_recordkey_field>",
  "hoodie.datasource.write.precombine.field" -> "<your_precombine_field>",
  "hoodie.datasource.write.partitionpath.field" -> "<your_partitionkey_field>",
  "hoodie.datasource.write.hive_style_partitioning" -> "true",
  "hoodie.datasource.hive_sync.enable" -> "true",
  "hoodie.datasource.hive_sync.database" -> "<your_database_name>",
  "hoodie.datasource.hive_sync.table" -> "<your_table_name>",
  "hoodie.datasource.hive_sync.partition_fields" -> "<your_partitionkey_field>",
  "hoodie.datasource.hive_sync.partition_extractor_class" ->
 "org.apache.hudi.hive.MultiPartKeysValueExtractor",
  "hoodie.datasource.hive_sync.use_jdbc" -> "false",
  "hoodie.datasource.hive_sync.mode" -> "hms",
  "path" -> "s3://<s3Path/>")

dataFrame.write.format("hudi")
  .options(additionalOptions)
  .mode("append")
  .save()
```

Esempio: leggere una tabella Hudi da Amazon S3 utilizzando il AWS Glue Data Catalog

In questo esempio viene letta la tabella Hudi che hai creato in Esempio: scrivere una tabella Hudi su Amazon S3 e registrarla nel AWS Glue Data Catalog da Amazon S3.

> **Note**
>
> Questo esempio richiede di impostare il parametro `--enable-glue-datacatalog` job per utilizzare AWS Glue Data Catalog come metastore Apache Spark Hive. Per ulteriori informazioni, consulta Utilizzo dei parametri del lavoro nei lavori AWS Glue.

Python

Per questo esempio, usa il metodo `GlueContext.create_data_frame.from_catalog()`.

```
# Example: Read a Hudi table from Glue Data Catalog
```

```
from awsglue.context import GlueContext
from pyspark.context import SparkContext

sc = SparkContext()
glueContext = GlueContext(sc)

dataFrame = glueContext.create_data_frame.from_catalog(
    database = "<your_database_name>",
    table_name = "<your_table_name>"
)
```

Scala

Per questo esempio, usa il metodo getCatalogSource.

```
// Example: Read a Hudi table from Glue Data Catalog

import com.amazonaws.services.glue.GlueContext
import org.apache.spark.SparkContext

object GlueApp {
  def main(sysArgs: Array[String]): Unit = {
    val spark: SparkContext = new SparkContext()
    val glueContext: GlueContext = new GlueContext(spark)

    val dataFrame = glueContext.getCatalogSource(
      database = "<your_database_name>",
      tableName = "<your_table_name>"
    ).getDataFrame()
  }
}
```

Esempio: aggiornamento e inserimento di un **`DataFrame`** in una tabella Hudi in Amazon S3

Questo esempio utilizza il AWS Glue Data Catalog per inserire un DataFrame nella tabella Hudi in Esempio: scrivere una tabella Hudi su Amazon S3 e registrarla nel AWS Glue Data Catalog cui è stato creato.

> **Note**
>
> Questo esempio richiede di impostare il parametro `--enable-glue-datacatalog` job per utilizzare AWS Glue Data Catalog come metastore Apache Spark Hive. Per ulteriori informazioni, consulta Utilizzo dei parametri del lavoro nei lavori AWS Glue.

### Python

Per questo esempio, usa il metodo `GlueContext.write_data_frame.from_catalog()`.

```
# Example: Upsert a Hudi table from Glue Data Catalog

from awsglue.context import GlueContext
from pyspark.context import SparkContext

sc = SparkContext()
glueContext = GlueContext(sc)

glueContext.write_data_frame.from_catalog(
    frame = dataFrame,
    database = "<your_database_name>",
    table_name = "<your_table_name>",
    additional_options={
        "hoodie.table.name": "<your_table_name>",
        "hoodie.database.name": "<your_database_name>",
        "hoodie.datasource.write.storage.type": "COPY_ON_WRITE",
        "hoodie.datasource.write.operation": "upsert",
        "hoodie.datasource.write.recordkey.field": "<your_recordkey_field>",
        "hoodie.datasource.write.precombine.field": "<your_precombine_field>",
        "hoodie.datasource.write.partitionpath.field": "<your_partitionkey_field>",
        "hoodie.datasource.write.hive_style_partitioning": "true",
        "hoodie.datasource.hive_sync.enable": "true",
        "hoodie.datasource.hive_sync.database": "<your_database_name>",
        "hoodie.datasource.hive_sync.table": "<your_table_name>",
        "hoodie.datasource.hive_sync.partition_fields": "<your_partitionkey_field>",
        "hoodie.datasource.hive_sync.partition_extractor_class":
 "org.apache.hudi.hive.MultiPartKeysValueExtractor",
        "hoodie.datasource.hive_sync.use_jdbc": "false",
        "hoodie.datasource.hive_sync.mode": "hms"
    }
)
```

Scala

Per questo esempio, usa il metodo getCatalogSink.

```
// Example: Upsert a Hudi table from Glue Data Catalog

import com.amazonaws.services.glue.GlueContext
import com.amazonaws.services.glue.util.JsonOptions
import org.apacke.spark.SparkContext

object GlueApp {
  def main(sysArgs: Array[String]): Unit = {
    val spark: SparkContext = new SparkContext()
    val glueContext: GlueContext = new GlueContext(spark)
    glueContext.getCatalogSink("<your_database_name>", "<your_table_name>",
      additionalOptions = JsonOptions(Map(
        "hoodie.table.name" -> "<your_table_name>",
        "hoodie.database.name" -> "<your_database_name>",
        "hoodie.datasource.write.storage.type" -> "COPY_ON_WRITE",
        "hoodie.datasource.write.operation" -> "upsert",
        "hoodie.datasource.write.recordkey.field" -> "<your_recordkey_field>",
        "hoodie.datasource.write.precombine.field" -> "<your_precombine_field>",
        "hoodie.datasource.write.partitionpath.field" ->
 "<your_partitionkey_field>",
        "hoodie.datasource.write.hive_style_partitioning" -> "true",
        "hoodie.datasource.hive_sync.enable" -> "true",
        "hoodie.datasource.hive_sync.database" -> "<your_database_name>",
        "hoodie.datasource.hive_sync.table" -> "<your_table_name>",
        "hoodie.datasource.hive_sync.partition_fields" ->
 "<your_partitionkey_field>",
        "hoodie.datasource.hive_sync.partition_extractor_class" ->
 "org.apache.hudi.hive.MultiPartKeysValueExtractor",
        "hoodie.datasource.hive_sync.use_jdbc" -> "false",
        "hoodie.datasource.hive_sync.mode" -> "hms"
      )))
      .writeDataFrame(dataFrame, glueContext)
  }
}
```

Esempio: lettura di una tabella Hudi da Amazon S3 tramite Spark

Questo esempio legge una tabella Hudi da Amazon S3 utilizzando l'API Spark. DataFrame

Python

```
# Example: Read a Hudi table from S3 using a Spark DataFrame

dataFrame = spark.read.format("hudi").load("s3://<s3path/>")
```

Scala

```
// Example: Read a Hudi table from S3 using a Spark DataFrame

val dataFrame = spark.read.format("hudi").load("s3://<s3path/>")
```

Esempio: scrittura di una tabella Hudi su Amazon S3 tramite Spark

In questo esempio viene scritta una tabella Hudi su Amazon S3 tramite Spark.

Python

```
# Example: Write a Hudi table to S3 using a Spark DataFrame

dataFrame.write.format("hudi") \
    .options(**additional_options) \
    .mode("overwrite") \
    .save("s3://<s3Path/>")
```

Scala

```
// Example: Write a Hudi table to S3 using a Spark DataFrame

dataFrame.write.format("hudi")
  .options(additionalOptions)
  .mode("overwrite")
  .save("s3://<s3path/>")
```

Esempio: lettura e scrittura della tabella Hudi con il controllo delle autorizzazioni di Lake Formation

Questo esempio legge da e scrive su una tabella Hudi con il controllo delle autorizzazioni di Lake Formation.

1. Crea una tabella Hudi e registrala in Lake Formation.

a. Per abilitare il controllo delle autorizzazioni di Lake Formation, devi prima registrare il percorso della tabella Amazon S3 su Lake Formation. Per ulteriori informazioni, consulta la pagina Registrazione di una posizione Amazon S3. Puoi registrarlo dalla console di Lake Formation o utilizzando la AWS CLI:

```
aws lakeformation register-resource --resource-arn arn:aws:s3:::<s3-bucket>/<s3-
folder> --use-service-linked-role --region <REGION>
```

Una volta registrata una posizione Amazon S3, qualsiasi tabella AWS Glue che punta alla posizione (o a una delle sue sedi secondarie) restituirà il valore del `IsRegisteredWithLakeFormation` parametro come true nella chiamata. `GetTable`

b. Crea una tabella Hudi che punti al percorso registrato di Amazon S3 tramite l'API Spark DataFrame:

```
hudi_options = {
    'hoodie.table.name': table_name,
    'hoodie.database.name': database_name,
    'hoodie.datasource.write.storage.type': 'COPY_ON_WRITE',
    'hoodie.datasource.write.recordkey.field': 'product_id',
    'hoodie.datasource.write.table.name': table_name,
    'hoodie.datasource.write.operation': 'upsert',
    'hoodie.datasource.write.precombine.field': 'updated_at',
    'hoodie.datasource.write.hive_style_partitioning': 'true',
    'hoodie.upsert.shuffle.parallelism': 2,
    'hoodie.insert.shuffle.parallelism': 2,
    'path': <S3_TABLE_LOCATION>,
    'hoodie.datasource.hive_sync.enable': 'true',
    'hoodie.datasource.hive_sync.database': database_name,
    'hoodie.datasource.hive_sync.table': table_name,
    'hoodie.datasource.hive_sync.use_jdbc': 'false',
    'hoodie.datasource.hive_sync.mode': 'hms'
}

df_products.write.format("hudi")  \
    .options(**hudi_options)  \
    .mode("overwrite")  \
    .save()
```

2. Concedi a Lake Formation l'autorizzazione per il ruolo IAM di AWS Glue job. Puoi concedere le autorizzazioni dalla console di Lake Formation o utilizzando la AWS CLI. Per ulteriori informazioni,

consulta la pagina Concessione delle autorizzazioni alla tabella tramite la console di Lake Formation e il metodo delle risorse denominate

3. Leggi la tabella Hudi registrata in Lake Formation. Il codice equivale a leggere una tabella Hudi non registrata. Tieni presente che il ruolo IAM di AWS Glue job deve disporre dell'autorizzazione SELECT affinché la lettura abbia esito positivo.

```
val dataFrame = glueContext.getCatalogSource(
     database = "<your_database_name>",
     tableName = "<your_table_name>"
   ).getDataFrame()
```

4. Scrivi sulla tabella Hudi registrata in Lake Formation. Il codice equivale a scrivere su una tabella Hudi non registrata. Tieni presente che il ruolo IAM di AWS Glue job deve disporre dell'autorizzazione SUPER affinché la scrittura abbia esito positivo.

```
glueContext.getCatalogSink("<your_database_name>", "<your_table_name>",
      additionalOptions = JsonOptions(Map(
        "hoodie.table.name" -> "<your_table_name>",
        "hoodie.database.name" -> "<your_database_name>",
        "hoodie.datasource.write.storage.type" -> "COPY_ON_WRITE",
        "hoodie.datasource.write.operation" -> "<write_operation>",
        "hoodie.datasource.write.recordkey.field" -> "<your_recordkey_field>",
        "hoodie.datasource.write.precombine.field" -> "<your_precombine_field>",
        "hoodie.datasource.write.partitionpath.field" -> "<your_partitionkey_field>",
        "hoodie.datasource.write.hive_style_partitioning" -> "true",
        "hoodie.datasource.hive_sync.enable" -> "true",
        "hoodie.datasource.hive_sync.database" -> "<your_database_name>",
        "hoodie.datasource.hive_sync.table" -> "<your_table_name>",
        "hoodie.datasource.hive_sync.partition_fields" ->
 "<your_partitionkey_field>",
        "hoodie.datasource.hive_sync.partition_extractor_class" ->
 "org.apache.hudi.hive.MultiPartKeysValueExtractor",
        "hoodie.datasource.hive_sync.use_jdbc" -> "false",
        "hoodie.datasource.hive_sync.mode" -> "hms"
      )))
      .writeDataFrame(dataFrame, glueContext)
```

## Utilizzo del framework Delta Lake in AWS Glue

AWS Glue 3.0 e versioni successive supportano il framework Linux Foundation Delta Lake. Delta Lake è un framework di archiviazione di data lake open source che consente di eseguire transazioni ACID, scalare la gestione dei metadati e unificare lo streaming e l'elaborazione dei dati in batch. Questo argomento descrive le funzionalità disponibili per l'utilizzo dei dati in AWS Glue durante il trasporto o l'archiviazione dei dati in una tabella Delta Lake. Per saperne di più su Delta Lake, consulta la documentazione ufficiale di Delta Lake.

Puoi usare AWS Glue per eseguire operazioni di lettura e scrittura sulle tabelle Delta Lake in Amazon S3 o lavorare con le tabelle Delta Lake utilizzando il AWS Glue Data Catalog. Sono supportate anche operazioni aggiuntive come inserimento, aggiornamento e letture e scritture in batch di tabelle. Quando usi le tabelle Delta Lake, hai anche la possibilità di utilizzare metodi della libreria Python di Delta Lake come `DeltaTable.forPath`. Per ulteriori informazioni sulla libreria Python di Delta Lake, consultare la documentazione Python di Delta Lake.

La tabella seguente elenca la versione di Delta Lake inclusa in ogni versione di AWS Glue.

| AWS Versione Glue | Versione Delta Lake supportata |
|---|---|
| 5.1 | 3.3.2 |
| 5.0 | 3.3.0 |
| 4.0 | 2.1.0 |
| 3.0 | 1.0.0 |

Per ulteriori informazioni sui framework di data lake supportati da AWS Glue, consulta. Utilizzo di framework di data lake con AWS Glue processi ETL

### Attivazione di Delta Lake for AWS Glue

Per abilitare Delta Lake for AWS Glue, completa le seguenti attività:

- Specifica `delta` come valore per i parametri del processo `--datalake-formats`. Per ulteriori informazioni, consulta Utilizzo dei parametri del lavoro nei lavori AWS Glue.

- Crea una chiave denominata `--conf` per il tuo lavoro AWS Glue e impostala sul seguente valore. In alternativa, puoi impostare la seguente configurazione usando `SparkConf` nel tuo script. Queste impostazioni consentono ad Apache Spark di gestire correttamente le tabelle Delta Lake.

```
spark.sql.extensions=io.delta.sql.DeltaSparkSessionExtension --conf
 spark.sql.catalog.spark_catalog=org.apache.spark.sql.delta.catalog.DeltaCatalog --
conf
 spark.delta.logStore.class=org.apache.spark.sql.delta.storage.S3SingleDriverLogStore
```

- Il supporto delle autorizzazioni Lake Formation per le tabelle Delta è abilitato di default per AWS Glue 4.0. Non è necessaria alcuna configurazione aggiuntiva reading/writing per le tabelle Delta registrate da Lake Formation. Per leggere una tabella Delta registrata, il ruolo IAM di AWS Glue job deve disporre dell'autorizzazione SELECT. Per scrivere su una tabella Delta registrata, il ruolo IAM di AWS Glue job deve disporre dell'autorizzazione SUPER. Per ulteriori informazioni sulla gestione delle autorizzazioni di Lake Formation, consulta Concessione e revoca delle autorizzazioni del catalogo dati.

Utilizzo di una versione differente di Delta Lake

Per utilizzare una versione di Delta Lake non supportata da AWS Glue, specifica i tuoi file JAR Delta Lake utilizzando il parametro `--extra-jars` job. Non includere `delta` come valore per il parametro del processo `--datalake-formats`. Se si utilizza AWS Glue 5.0 o versioni successive, è necessario impostare `--user-jars-first true` il parametro del lavoro. Per utilizzare la libreria Python Delta Lake in questo caso, è necessario specificare i file JAR della libreria utilizzando il parametro del processo `--extra-py-files`. La libreria Python è contenuta nei file JAR di Delta Lake.

Esempio: scrivere una tabella Delta Lake su Amazon S3 e registrarla nel AWS Glue Data Catalog

Il seguente script AWS Glue ETL dimostra come scrivere una tabella Delta Lake su Amazon S3 e registrarla nel AWS Glue Data Catalog.

Python

```
# Example: Create a Delta Lake table from a DataFrame
# and register the table to Glue Data Catalog

additional_options = {
    "path": "s3://<s3Path>"
}
```

```
dataFrame.write \
    .format("delta") \
    .options(**additional_options) \
    .mode("append") \
    .partitionBy("<your_partitionkey_field>") \
    .saveAsTable("<your_database_name>.<your_table_name>")
```

Scala

```
// Example: Example: Create a Delta Lake table from a DataFrame
// and register the table to Glue Data Catalog

val additional_options = Map(
  "path" -> "s3://<s3Path>"
)
dataFrame.write.format("delta")
  .options(additional_options)
  .mode("append")
  .partitionBy("<your_partitionkey_field>")
  .saveAsTable("<your_database_name>.<your_table_name>")
```

Esempio: leggere una tabella Delta Lake da Amazon S3 utilizzando il AWS Glue Data Catalog

Il seguente script AWS Glue ETL legge la tabella Delta Lake in cui è stata creata. Esempio: scrivere una tabella Delta Lake su Amazon S3 e registrarla nel AWS Glue Data Catalog

Python

Per questo esempio, utilizza il metodo create_data_frame.from_catalog.

```
# Example: Read a Delta Lake table from Glue Data Catalog

from awsglue.context import GlueContext
from pyspark.context import SparkContext

sc = SparkContext()
glueContext = GlueContext(sc)

df = glueContext.create_data_frame.from_catalog(
    database="<your_database_name>",
    table_name="<your_table_name>",
    additional_options=additional_options
```

```
)
```

Scala

Per questo esempio, usa il metodo getCatalogSource.

```
// Example: Read a Delta Lake table from Glue Data Catalog

import com.amazonaws.services.glue.GlueContext
import org.apacke.spark.SparkContext

object GlueApp {
  def main(sysArgs: Array[String]): Unit = {
    val spark: SparkContext = new SparkContext()
    val glueContext: GlueContext = new GlueContext(spark)
    val df = glueContext.getCatalogSource("<your_database_name>",
 "<your_table_name>",
      additionalOptions = additionalOptions)
      .getDataFrame()
  }
}
```

Esempio: inserire un **`DataFrame`** in una tabella Delta Lake in Amazon S3 utilizzando il AWS Glue Data Catalog

Questo esempio inserisce i dati nella tabella Delta Lake creata in Esempio: scrivere una tabella Delta Lake su Amazon S3 e registrarla nel AWS Glue Data Catalog.

> **Note**
>
> Questo esempio richiede di impostare il parametro `--enable-glue-datacatalog` job per utilizzare AWS Glue Data Catalog come metastore Apache Spark Hive. Per ulteriori informazioni, consulta Utilizzo dei parametri del lavoro nei lavori AWS Glue.

Python

Per questo esempio, utilizza il metodo write_data_frame.from_catalog.

```
# Example: Insert into a Delta Lake table in S3 using Glue Data Catalog
```

```
from awsglue.context import GlueContext
from pyspark.context import SparkContext

sc = SparkContext()
glueContext = GlueContext(sc)

glueContext.write_data_frame.from_catalog(
    frame=dataFrame,
    database="<your_database_name>",
    table_name="<your_table_name>",
    additional_options=additional_options
)
```

Scala

Per questo esempio, usa il metodo getCatalogSink.

```
// Example: Insert into a Delta Lake table in S3 using Glue Data Catalog

import com.amazonaws.services.glue.GlueContext
import org.apacke.spark.SparkContext

object GlueApp {
  def main(sysArgs: Array[String]): Unit = {
    val spark: SparkContext = new SparkContext()
    val glueContext: GlueContext = new GlueContext(spark)
    glueContext.getCatalogSink("<your_database_name>", "<your_table_name>",
      additionalOptions = additionalOptions)
      .writeDataFrame(dataFrame, glueContext)
  }
}
```

Esempio: lettura di una tabella Delta Lake da Amazon S3 tramite l'API Spark

In questo esempio viene letta una tabella Delta Lake da Amazon S3 tramite l'API Spark.

Python

```
# Example: Read a Delta Lake table from S3 using a Spark DataFrame

dataFrame = spark.read.format("delta").load("s3://<s3path/>")
```

Scala

```
// Example: Read a Delta Lake table from S3 using a Spark DataFrame

val dataFrame = spark.read.format("delta").load("s3://<s3path/>")
```

Esempio: scrittura di una tabella Delta Lake su Amazon S3 tramite Spark

In questo esempio viene scritta una tabella Delta Lake su Amazon S3 tramite Spark.

Python

```
# Example: Write a Delta Lake table to S3 using a Spark DataFrame

dataFrame.write.format("delta") \
    .options(**additional_options) \
    .mode("overwrite") \
    .partitionBy("<your_partitionkey_field>") 
    .save("s3://<s3Path>")
```

Scala

```
// Example: Write a Delta Lake table to S3 using a Spark DataFrame

dataFrame.write.format("delta")
  .options(additionalOptions)
  .mode("overwrite")
  .partitionBy("<your_partitionkey_field>")
  .save("s3://<s3path/>")
```

Esempio: lettura e scrittura della tabella Delta Lake con il controllo delle autorizzazioni di Lake Formation

Questo esempio legge da e scrive su una tabella Delta Lake con il controllo delle autorizzazioni di Lake Formation.

1. Crea una tabella Delta e registrala in Lake Formation
   a. Per abilitare il controllo delle autorizzazioni di Lake Formation, devi prima registrare il percorso della tabella Amazon S3 su Lake Formation. Per ulteriori informazioni, consulta la pagina

Registrazione di una posizione Amazon S3. Puoi registrarlo dalla console di Lake Formation o utilizzando la AWS CLI:

```
aws lakeformation register-resource --resource-arn arn:aws:s3:::<s3-bucket>/<s3-folder> --use-service-linked-role --region <REGION>
```

Una volta registrata una posizione Amazon S3, qualsiasi tabella AWS Glue che punta alla posizione (o a una delle sue sedi secondarie) restituirà il valore del `IsRegisteredWithLakeFormation` parametro come true nella chiamata. `GetTable`

b. Crea una tabella Delta che punti al percorso registrato di Amazon S3 tramite Spark:

> **Note**
>
> Di seguito vengono mostrati gli esempi Python.

```
dataFrame.write \
 .format("delta") \
 .mode("overwrite") \
 .partitionBy("<your_partitionkey_field>") \
 .save("s3://<the_s3_path>")
```

Dopo aver scritto i dati su Amazon S3, usa il crawler AWS Glue per creare una nuova tabella del catalogo Delta. Per ulteriori informazioni, consulta Introduzione al supporto per tabelle Delta Lake nativo con i crawler AWS Glue.

Puoi anche creare la tabella manualmente tramite l'`CreateTable`API AWS Glue.

2. Concedi a Lake Formation l'autorizzazione per il ruolo IAM di AWS Glue job. Puoi concedere le autorizzazioni dalla console di Lake Formation o utilizzando la AWS CLI. Per ulteriori informazioni, consulta la pagina Concessione delle autorizzazioni alla tabella tramite la console di Lake Formation e il metodo delle risorse denominate
3. Leggi la tabella Delta registrata in Lake Formation. Il codice equivale a leggere una tabella Delta non registrata. Tieni presente che il ruolo IAM di AWS Glue job deve disporre dell'autorizzazione SELECT affinché la lettura abbia esito positivo.

```
# Example: Read a Delta Lake table from Glue Data Catalog
```

```
df = glueContext.create_data_frame.from_catalog(
    database="<your_database_name>",
    table_name="<your_table_name>",
    additional_options=additional_options
)
```

4. Scrivi sulla tabella Delta registrata in Lake Formation. Il codice equivale a scrivere su una tabella Delta non registrata. Tieni presente che il ruolo IAM di AWS Glue job deve disporre dell'autorizzazione SUPER affinché la scrittura abbia esito positivo.

   Per impostazione predefinita, AWS Glue utilizza `Append` come SaveMode. È possibile modificarlo impostando l'opzione saveMode in `additional_options`. Per informazioni sul supporto saveMode nelle tabelle Delta, consulta Scrivi su una tabella.

   ```
   glueContext.write_data_frame.from_catalog(
       frame=dataFrame,
       database="<your_database_name>",
       table_name="<your_table_name>",
       additional_options=additional_options
   )
   ```

Utilizzo del framework Iceberg in AWS Glue

AWS Glue 3.0 e versioni successive supportano il framework Apache Iceberg per i data lake. Iceberg fornisce un formato di tabella ad alte prestazioni che funziona proprio come una tabella SQL. Questo argomento descrive le funzionalità disponibili per l'utilizzo dei dati in AWS Glue quando si trasportano o si archiviano i dati in una tabella Iceberg. Per ulteriori informazioni su Iceberg, consulta la documentazione ufficiale di Apache Iceberg.

Puoi usare AWS Glue per eseguire operazioni di lettura e scrittura sulle tabelle Iceberg in Amazon S3 o lavorare con le tabelle Iceberg utilizzando il AWS Glue Data Catalog. Sono supportate anche operazioni aggiuntive, tra cui inserimenti e tutte le query Spark e scritture Spark. L'aggiornamento non è supportato per le tabelle Iceberg.

> **Note**
>
> `ALTER TABLE … RENAME TO` non è disponibile per Apache Iceberg 0.13.1 per AWS Glue 3.0.

La tabella seguente elenca la versione di Iceberg inclusa in ogni versione di AWS Glue.

| AWS Versione Glue | Versione Iceberg supportata |
|---|---|
| 5.1 | 1.10.0 |
| 5.0 | 1.7.1 |
| 4.0 | 1.0.0 |
| 3.0 | 0.13.1 |

Per ulteriori informazioni sui framework di data lake supportati da AWS Glue, consulta. Utilizzo di framework di data lake con AWS Glue processi ETL

Abilitazione del framework Iceberg

Per abilitare Iceberg for AWS Glue, completa le seguenti attività:

- Specifica `iceberg` come valore per i parametri del processo `--datalake-formats`. Per ulteriori informazioni, consulta Utilizzo dei parametri del lavoro nei lavori AWS Glue.
- Crea una chiave denominata `--conf` per il tuo lavoro AWS Glue e impostala sul seguente valore. In alternativa, puoi impostare la seguente configurazione usando `SparkConf` nel tuo script. Queste impostazioni consentono ad Apache Spark di gestire correttamente le tabelle Iceberg.

```
spark.sql.extensions=org.apache.iceberg.spark.extensions.IcebergSparkSessionExtensions
--conf spark.sql.catalog.glue_catalog=org.apache.iceberg.spark.SparkCatalog
--conf spark.sql.catalog.glue_catalog.warehouse=s3://<your-warehouse-dir>/
--conf spark.sql.catalog.glue_catalog.catalog-
impl=org.apache.iceberg.aws.glue.GlueCatalog
--conf spark.sql.catalog.glue_catalog.io-impl=org.apache.iceberg.aws.s3.S3FileIO
```

  Se stai leggendo o scrivendo su tabelle Iceberg registrate con Lake Formation, segui le indicazioni contenute the section called "Lake Formation per FGAC" in AWS Glue 5.0 e versioni successive. In AWS Glue 4.0, aggiungi la seguente configurazione per abilitare il supporto di Lake Formation.

```
--conf spark.sql.catalog.glue_catalog.glue.lakeformation-enabled=true
--conf spark.sql.catalog.glue_catalog.glue.id=<table-catalog-id>
```

Se usi AWS Glue 3.0 con Iceberg 0.13.1, devi impostare le seguenti configurazioni aggiuntive per utilizzare Amazon DynamoDB lock manager e garantire una transazione atomica. AWS Glue 4.0 o versione successiva utilizza il blocco ottimistico per impostazione predefinita. Per ulteriori informazioni, consulta Iceberg AWS Integrations nella documentazione ufficiale di Apache Iceberg.

```
--conf spark.sql.catalog.glue_catalog.lock-
impl=org.apache.iceberg.aws.glue.DynamoLockManager
--conf spark.sql.catalog.glue_catalog.lock.table=<your-dynamodb-table-name>
```

Utilizzo di una versione differente di Iceberg

Per utilizzare una versione di Iceberg non supportata da AWS Glue, specifica i tuoi file JAR Iceberg utilizzando il parametro `--extra-jars` job. Non includere `iceberg` come valore per il parametro `--datalake-formats`. Se si utilizza AWS Glue 5.0 o versioni successive, è necessario impostare `--user-jars-first true` il parametro del lavoro.

Abilitazione della crittografia per le tabelle Iceberg

> **Note**
>
> Le tabelle Iceberg dispongono di meccanismi propri per abilitare la crittografia lato server. È necessario abilitare questa configurazione oltre alla configurazione di sicurezza di AWS Glue.

Per abilitare la crittografia lato server sulle tabelle Iceberg, consulta le indicazioni contenute nella documentazione di Iceberg.

Aggiungere la configurazione Spark per Iceberg su più Regioni

Per aggiungere una configurazione spark aggiuntiva per l'accesso alle tabelle interregionali di Iceberg con il AWS Glue Data Catalog AWS Lake Formation, procedi nel seguente modo:

1. Creare un punto di accesso multi-regione.
2. Configurare le seguenti proprietà Spark:

```
-----
    --conf spark.sql.catalog.my_catalog.s3.use-arn-region-enabled=true \
    --conf spark.sql.catalog.{CATALOG}.s3.access-points.bucket1",
 "arn:aws:s3::<account-id>:accesspoint/<mrap-id>.mrap \
```

```
    --conf spark.sql.catalog.{CATALOG}.s3.access-points.bucket2",
 "arn:aws:s3::<account-id>:accesspoint/<mrap-id>.mrap
-----
```

Esempio: scrivere una tabella Iceberg su Amazon S3 e registrarla nel AWS Glue Data Catalog

Questo script di esempio dimostra come scrivere una tabella Iceberg su Amazon S3. L'esempio utilizza Iceberg AWS Integrations per registrare la tabella nel AWS Glue Data Catalog.

Python

```
# Example: Create an Iceberg table from a DataFrame
# and register the table to Glue Data Catalog

dataFrame.createOrReplaceTempView("tmp_<your_table_name>")

query = f"""
CREATE TABLE glue_catalog.<your_database_name>.<your_table_name>
USING iceberg
TBLPROPERTIES ("format-version"="2")
AS SELECT * FROM tmp_<your_table_name>
"""
spark.sql(query)
```

Scala

```
// Example: Example: Create an Iceberg table from a DataFrame
// and register the table to Glue Data Catalog

dataFrame.createOrReplaceTempView("tmp_<your_table_name>")

val query = """CREATE TABLE glue_catalog.<your_database_name>.<your_table_name>
USING iceberg
TBLPROPERTIES ("format-version"="2")
AS SELECT * FROM tmp_<your_table_name>
"""
spark.sql(query)
```

In alternativa, è possibile scrivere una tabella Iceberg su Amazon S3 e catalogo dati tramite metodi Spark.

Prerequisiti: è necessario fornire un catalogo per l'utilizzo della libreria Iceberg. Quando si utilizza il AWS Glue Data Catalog, AWS Glue lo rende semplice. Il AWS Glue Data Catalog è preconfigurato per essere utilizzato dalle librerie Spark come. `glue_catalog` Le tabelle del Data Catalog sono identificate da a *`databaseName`* e a. *`tableName`* Per ulteriori informazioni sul AWS Glue Data Catalog, vedereRilevamento e catalogazione dei dati.

Se non utilizzi il AWS Glue Data Catalog, dovrai fornire un catalogo tramite Spark APIs. Per ulteriori informazioni, consulta la pagina Spark Configuration nella documentazione di Spark.

In questo esempio viene scritta una tabella Iceberg in Amazon S3 e il catalogo dati tramite Spark.

Python

```
# Example: Write an Iceberg table to S3 on the Glue Data Catalog

# Create (equivalent to CREATE TABLE AS SELECT)
dataFrame.writeTo("glue_catalog.databaseName.tableName") \
    .tableProperty("format-version", "2") \
    .create()

# Append (equivalent to INSERT INTO)
dataFrame.writeTo("glue_catalog.databaseName.tableName") \
    .tableProperty("format-version", "2") \
    .append()
```

Scala

```
// Example: Write an Iceberg table to S3 on the Glue Data Catalog

// Create (equivalent to CREATE TABLE AS SELECT)
dataFrame.writeTo("glue_catalog.databaseName.tableName")
    .tableProperty("format-version", "2")
    .create()

// Append (equivalent to INSERT INTO)
dataFrame.writeTo("glue_catalog.databaseName.tableName")
    .tableProperty("format-version", "2")
    .append()
```

### Esempio: leggere una tabella Iceberg da Amazon S3 utilizzando il AWS Glue Data Catalog

Questo esempio legge la tabella Iceberg che creata in Esempio: scrivere una tabella Iceberg su Amazon S3 e registrarla nel AWS Glue Data Catalog.

Python

Per questo esempio, usa il metodo `GlueContext.create_data_frame.from_catalog()`.

```
# Example: Read an Iceberg table from Glue Data Catalog

from awsglue.context import GlueContext
from pyspark.context import SparkContext

sc = SparkContext()
glueContext = GlueContext(sc)

df = glueContext.create_data_frame.from_catalog(
    database="<your_database_name>",
    table_name="<your_table_name>",
    additional_options=additional_options
)
```

Scala

Per questo esempio, usa il metodo getCatalogSource.

```
// Example: Read an Iceberg table from Glue Data Catalog

import com.amazonaws.services.glue.GlueContext
import org.apacke.spark.SparkContext

object GlueApp {
  def main(sysArgs: Array[String]): Unit = {
    val spark: SparkContext = new SparkContext()
    val glueContext: GlueContext = new GlueContext(spark)
    val df = glueContext.getCatalogSource("<your_database_name>",
 "<your_table_name>",
      additionalOptions = additionalOptions)
      .getDataFrame()
  }
}
```

Esempio: inserire un **`DataFrame`** in una tabella Iceberg in Amazon S3 utilizzando il AWS Glue Data Catalog

Questo esempio inserisce i dati nella tabella Iceberg creata in Esempio: scrivere una tabella Iceberg su Amazon S3 e registrarla nel AWS Glue Data Catalog.

> **Note**
>
> Questo esempio richiede di impostare il parametro `--enable-glue-datacatalog` job per utilizzare AWS Glue Data Catalog come metastore Apache Spark Hive. Per ulteriori informazioni, consulta Utilizzo dei parametri del lavoro nei lavori AWS Glue.

Python

Per questo esempio, usa il metodo `GlueContext.write_data_frame.from_catalog()`.

```
# Example: Insert into an Iceberg table from Glue Data Catalog

from awsglue.context import GlueContext
from pyspark.context import SparkContext

sc = SparkContext()
glueContext = GlueContext(sc)

glueContext.write_data_frame.from_catalog(
    frame=dataFrame,
    database="<your_database_name>",
    table_name="<your_table_name>",
    additional_options=additional_options
)
```

Scala

Per questo esempio, usa il metodo getCatalogSink.

```
// Example: Insert into an Iceberg table from Glue Data Catalog

import com.amazonaws.services.glue.GlueContext
import org.apacke.spark.SparkContext
```

```
object GlueApp {
  def main(sysArgs: Array[String]): Unit = {
    val spark: SparkContext = new SparkContext()
    val glueContext: GlueContext = new GlueContext(spark)
    glueContext.getCatalogSink("<your_database_name>", "<your_table_name>",
      additionalOptions = additionalOptions)
      .writeDataFrame(dataFrame, glueContext)
  }
}
```

Esempio: lettura di una tabella Iceberg da Amazon S3 tramite Spark

Prerequisiti: è necessario fornire un catalogo per l'utilizzo della libreria Iceberg. Quando si utilizza il AWS Glue Data Catalog, AWS Glue lo rende semplice. Il AWS Glue Data Catalog è preconfigurato per essere utilizzato dalle librerie Spark come. `glue_catalog` Le tabelle del Data Catalog sono identificate da a *`databaseName`* e a. *`tableName`* Per ulteriori informazioni sul AWS Glue Data Catalog, vedereRilevamento e catalogazione dei dati.

Se non utilizzi il AWS Glue Data Catalog, dovrai fornire un catalogo tramite Spark APIs. Per ulteriori informazioni, consulta la pagina Spark Configuration nella documentazione di Spark.

In questo esempio viene letta una tabella Iceberg in Amazon S3 da Catalogo dati tramite Spark.

Python

```
# Example: Read an Iceberg table on S3 as a DataFrame from the Glue Data Catalog

dataFrame = spark.read.format("iceberg").load("glue_catalog.databaseName.tableName")
```

Scala

```
// Example: Read an Iceberg table on S3 as a DataFrame from the Glue Data Catalog

val dataFrame =
 spark.read.format("iceberg").load("glue_catalog.databaseName.tableName")
```

Esempio: lettura e scrittura della tabella Iceberg con il controllo delle autorizzazioni di Lake Formation

Questo esempio legge da e scrive su una tabella Iceberg con il controllo delle autorizzazioni di Lake Formation.

> **Note**
>
> Questo esempio funziona solo in AWS Glue 4.0. In AWS Glue 5.0 e versioni successive, segui le istruzioni riportate inthe section called "Lake Formation per FGAC".

1. Crea una tabella Iceberg e registrala in Lake Formation:

   a. Per abilitare il controllo delle autorizzazioni di Lake Formation, devi prima registrare il percorso della tabella Amazon S3 su Lake Formation. Per ulteriori informazioni, consulta la pagina Registrazione di una posizione Amazon S3. Puoi registrarlo dalla console di Lake Formation o utilizzando la AWS CLI:

   ```
   aws lakeformation register-resource --resource-arn arn:aws:s3:::<s3-bucket>/<s3-folder> --use-service-linked-role --region <REGION>
   ```

   Una volta registrata una posizione Amazon S3, qualsiasi tabella AWS Glue che punta alla posizione (o a una delle sue sedi secondarie) restituirà il valore del `IsRegisteredWithLakeFormation` parametro come true nella chiamata. `GetTable`

   b. Crea una tabella Iceberg che punti al percorso registrato tramite Spark SQL:

   > **Note**
   >
   > Di seguito vengono mostrati gli esempi Python.

   ```
   dataFrame.createOrReplaceTempView("tmp_<your_table_name>")

   query = f"""
   CREATE TABLE glue_catalog.<your_database_name>.<your_table_name>
   USING iceberg
   AS SELECT * FROM tmp_<your_table_name>
   """
   spark.sql(query)
   ```

   Puoi anche creare la tabella manualmente tramite AWS Glue `CreateTable` API. Per ulteriori informazioni, consulta Creazione di tabelle Apache Iceberg.

> **Note**
>
> Attualmente, l'API `UpdateTable` non supporta il formato di tabella Iceberg come input per l'operazione.

2. Concedi a Lake Formation l'autorizzazione per il ruolo IAM del processo. Puoi concedere le autorizzazioni dalla console di Lake Formation o utilizzando la AWS CLI. Per ulteriori informazioni, consulta: -table-permissions.html https://docs.aws.amazon.com/lake-formation/ latest/dg/granting
3. Leggi una tabella Iceberg registrata con Lake Formation. Il codice equivale a leggere una tabella Iceberg non registrata. Tieni presente che il tuo ruolo IAM di AWS Glue job deve disporre dell'autorizzazione SELECT affinché la lettura abbia esito positivo.

```
# Example: Read an Iceberg table from the AWS Glue Data Catalog
from awsglue.context import GlueContextfrom pyspark.context import SparkContext

sc = SparkContext()
glueContext = GlueContext(sc)

df = glueContext.create_data_frame.from_catalog(
    database="<your_database_name>",
    table_name="<your_table_name>",
    additional_options=additional_options
)
```

4. Scrivi su una tabella Iceberg registrata con Lake Formation. Il codice equivale a scrivere su una tabella Iceberg non registrata. Tieni presente che il tuo ruolo IAM di AWS Glue job deve disporre dell'autorizzazione SUPER affinché la scrittura abbia esito positivo.

```
glueContext.write_data_frame.from_catalog(
    frame=dataFrame,
    database="<your_database_name>",
    table_name="<your_table_name>",
    additional_options=additional_options
)
```

Riferimento alla configurazione condivisa

È possibile utilizzare i seguenti valori di `format_options` con ogni tipo di formato.

- `attachFilename`: una stringa nel formato appropriato da utilizzare come nome di colonna. Se si fornisce questa opzione, il nome del file di origine del record verrà aggiunto al record. Il valore del parametro verrà utilizzato come nome della colonna.
- `attachTimestamp`: una stringa nel formato appropriato da utilizzare come nome di colonna. Se si fornisce questa opzione, l'ora di modifica del file di origine del record verrà aggiunta al record. Il valore del parametro verrà utilizzato come nome della colonna.

## AWS Supporto di Glue Data Catalog per i job SQL di Spark

Il AWS Glue Data Catalog è un catalogo compatibile con i metastore di Apache Hive. Puoi configurare gli endpoint di sviluppo e i processi AWS Glue affinché usino catalogo dati come metastore Apache Hive esterno. Quindi, puoi eseguire le query Apache Spark SQL direttamente sulle tabelle archiviate nel catalogo dati. I frame dinamici AWS Glue si integrano con catalogo dati per impostazione predefinita. Tuttavia con questa caratteristica i processi Spark SQL possono iniziare a usare il catalogo dati come metastore Hive esterno.

Questa caratteristica richiede l'accesso di rete all' endpoint API AWS Glue. Per processi AWS Glue con connessioni situate in sottoreti private, è necessario configurare un endpoint VPC o un gateway NAT, per fornire l'accesso alla rete. Per informazioni sulla configurazione degli endpoint VPC, consulta Impostazione dell'accesso di rete agli archivi di dati. Per creare un gateway NAT, consulta Gateway NAT nella Guida per l'utente di Amazon VPC.

È possibile configurare gli endpoint di sviluppo e i processi AWS Glue aggiungendo l'argomento `"--enable-glue-datacatalog": ""` agli argomenti degli endpoint di sviluppo e del processo rispettivamente. Questo argomento imposta determinate configurazioni in Spark che gli consentono di accedere al catalogo dati come metastore Hive esterno. Inoltre abilita il supporto Hive nell'oggetto `SparkSession` creato nell'endpoint di sviluppo o nel processo AWS Glue.

Per abilitare l'accesso al Data Catalog, seleziona la casella di controllo Usa AWS Glue Data Catalog come metastore Hive nel gruppo di opzioni del catalogo nella pagina Aggiungi lavoro o Aggiungi endpoint sulla console. Tieni presente che il ruolo IAM utilizzato per il processo o per l'endpoint di sviluppo deve disporre delle autorizzazioni `glue:CreateDatabase`. Viene creato un database chiamato "`default`" nel catalogo dati, nel caso non fosse già presente.

Osserviamo un esempio per utilizzare questa caratteristica nei processi Spark SQL. L'esempio seguente presuppone che hai sottoposto a crawling il set di dati dei legislatori degli Stati Uniti disponibile in `s3://awsglue-datasets/examples/us-legislators`.

Per serialize/deserialize i dati delle tabelle definite nel AWS Glue Data Catalog, Spark SQL necessita della SerDe classe Hive per il formato definito nel AWS Glue Data Catalog nel classpath dello spark job.

SerDes perché alcuni formati comuni sono distribuiti da. AWS Glue Di seguito sono elencati i relativi collegamenti Amazon S3:

- JSON
- XML
- Grok

Aggiungi JSON SerDe come JAR aggiuntivo all'endpoint di sviluppo. Per i lavori, puoi aggiungere l' SerDe utilizzo dell'`--extra-jars`argomento nel campo degli argomenti. Per ulteriori informazioni, consulta Utilizzo dei parametri del lavoro nei lavori AWS Glue.

Ecco un esempio di JSON di input per creare un endpoint di sviluppo con il catalogo dati abilitato per Spark SQL.

```
{
    "EndpointName": "Name",
    "RoleArn": "role_ARN",
    "PublicKey": "public_key_contents",
    "NumberOfNodes": 2,
    "Arguments": {
      "--enable-glue-datacatalog": ""
    },
    "ExtraJarsS3Path": "s3://crawler-public/json/serde/json-serde.jar"
}
```

Ora esegui una query sulle tabelle create dal set di dati dei legislatori degli Stati Uniti utilizzando Spark SQL.

```
>>> spark.sql("use legislators")
DataFrame[]
>>> spark.sql("show tables").show()
+-----------+-----------------+-----------+
|   database|        tableName|isTemporary|
+-----------+-----------------+-----------+
```

```
|legislators|        areas_json|      false|
|legislators|    countries_json|      false|
|legislators|       events_json|      false|
|legislators|  memberships_json|      false|
|legislators|organizations_json|      false|
|legislators|      persons_json|      false|
+-----------+------------------+-----------+
>>> spark.sql("describe memberships_json").show()
+--------------------+---------+-----------------+
|            col_name|data_type|          comment|
+--------------------+---------+-----------------+
|             area_id|   string|from deserializer|
|     on_behalf_of_id|   string|from deserializer|
|     organization_id|   string|from deserializer|
|                role|   string|from deserializer|
|           person_id|   string|from deserializer|
|legislative_perio...|   string|from deserializer|
|          start_date|   string|from deserializer|
|            end_date|   string|from deserializer|
+--------------------+---------+-----------------+
```

Se la SerDe classe per il formato non è disponibile nel classpath del lavoro, verrà visualizzato un errore simile al seguente.

```
>>> spark.sql("describe memberships_json").show()

Caused by: MetaException(message:java.lang.ClassNotFoundException Class
 org.openx.data.jsonserde.JsonSerDe not found)
    at
 org.apache.hadoop.hive.metastore.MetaStoreUtils.getDeserializer(MetaStoreUtils.java:399)
    at
 org.apache.hadoop.hive.ql.metadata.Table.getDeserializerFromMetaStore(Table.java:276)
    ... 64 more
```

Per visualizzare solo gli `organization_id` distinti della tabella `memberships`, esegui la seguente query SQL.

```
>>> spark.sql("select distinct organization_id from memberships_json").show()
+--------------------+
|     organization_id|
+--------------------+
|d56acebe-8fdc-47b...|
|8fa6c3d2-71dc-478...|
```

```
+--------------------+
```

Se è necessaria la stessa operazione per i frame dinamici, esegui la query seguente.

```
>>> memberships = glueContext.create_dynamic_frame.from_catalog(database="legislators",
 table_name="memberships_json")
>>> memberships.toDF().createOrReplaceTempView("memberships")
>>> spark.sql("select distinct organization_id from memberships").show()
+--------------------+
|     organization_id|
+--------------------+
|d56acebe-8fdc-47b...|
|8fa6c3d2-71dc-478...|
+--------------------+
```

Sebbene DynamicFrames siano ottimizzati per le operazioni ETL, consentire a Spark SQL di accedere direttamente al Data Catalog fornisce un modo conciso per eseguire istruzioni SQL complesse o trasferire applicazioni esistenti.

## Utilizzo di segnalibri di processo

AWS Glue per Spark monitora i dati già elaborati attraverso i segnalibri dei processi. Per un riepilogo della funzionalità dei segnalibri di processo e di ciò che supportano, consulta la pagina the section called "Monitoraggio dei dati elaborati mediante segnalibri di processo". Quando si programma un AWS Glue lavoro con segnalibri, si ha accesso a una flessibilità non disponibile nei lavori visivi.

- Durante la lettura da JDBC, è possibile specificare le colonne da utilizzare come chiavi di segnalibro nello script. AWS Glue
- È possibile scegliere quale `transformation_ctx` applicare a ciascuna chiamata al metodo.

Chiamare sempre *`job.init`* all'inizio dello script e *`job.commit`* alla fine dello script con parametri configurati in modo appropriato. Queste due funzioni inizializzano il servizio di segnalibri e aggiornano la modifica dello stato al servizio. I segnalibri non funzioneranno senza che vengano richiamati.

### Specifica delle chiavi dei segnalibri

Per i flussi di lavoro JDBC, il segnalibro tiene traccia delle righe lette dal processo confrontando i valori dei campi chiave con un valore aggiunto ai segnalibri. Questa operazione non è necessaria o applicabile per i flussi di processo Amazon S3. Quando scrivi uno AWS Glue script senza l'editor visivo, puoi specificare quale colonna tenere traccia con i segnalibri. È possibile specificare anche più

colonne. Sono consentite lacune nella sequenza di valori quando si specificano chiavi di segnalibro definite dall'utente.

> **Warning**
>
> Se vengono utilizzate le chiavi di segnalibro definite dall'utente, devono essere fornite rigorosamente e monotonicamente in ordine crescente o decrescente. Quando si selezionano campi aggiuntivi per una chiave composta, i campi relativi a concetti come "versioni secondarie" o "numeri di revisione" non soddisfano questi criteri, poiché i loro valori vengono riutilizzati in tutto il set di dati.

È possibile specificare `jobBookmarkKeys` e `jobBookmarkKeysSortOrder` nei seguenti modi:

- `create_dynamic_frame.from_catalog`: utilizza `additional_options`.
- `create_dynamic_frame.from_options`: utilizza `connection_options`.

Contesto di trasformazione

Molti dei metodi di frame AWS Glue PySpark dinamici includono un parametro opzionale denominato`transformation_ctx`, che è un identificatore univoco per l'istanza dell'operatore ETL. Il parametro `transformation_ctx` viene utilizzato per identificare le informazioni sullo stato all'interno di un segnalibro di processo per un determinato operatore. Nello specifico, AWS Glue utilizza `transformation_ctx` per indicizzare la chiave per lo stato del segnalibro.

> **Warning**
>
> Il parametro `transformation_ctx` serve come chiave per cercare nello stato del segnalibro una fonte specifica nello script. Affinché il segnalibro funzioni correttamente, è necessario mantenere sempre la fonte e il parametro `transformation_ctx` associato coerenti. Modificare la proprietà di origine o rinominare il parametro `transformation_ctx` potrebbe rendere il segnalibro precedente non valido e il filtro basato sulla marca temporale potrebbe non produrre il risultato corretto.

Per fare in modo che i segnalibri del processo funzionino correttamente, abilita il parametro del segnalibro del processo e imposta il parametro `transformation_ctx`. Se non passi il parametro `transformation_ctx`, i segnalibri del processo non sono abilitati per un frame dinamico oppure

nel metodo viene utilizzata una tabella. Ad esempio, in presenza di un processo ETL che legge e unisce due origini Amazon S3, puoi scegliere di passare il parametro `transformation_ctx` solo ai metodi per cui vuoi abilitare i segnalibri. Reimpostando il segnalibro per un processo, si reimpostano tutte le trasformazioni associate al processo, indipendentemente dal `transformation_ctx` utilizzato.

Per ulteriori informazioni sulla classe `DynamicFrameReader`, consulta DynamicFrameReader classe. Per ulteriori informazioni sulle PySpark estensioni, vedere. AWS Riferimento PySpark alle estensioni Glue

Esempi

Example

Di seguito è riportato un esempio di script generato per un'origine dati Amazon S3. Le parti dello script necessarie per l'utilizzo dei segnalibri di processo sono visualizzate in corsivo. Per ulteriori informazioni su questi elementi, consulta le API GlueContext classe e le API DynamicFrameWriter classe.

```
# Sample Script
import sys
from awsglue.transforms import *
from awsglue.utils import getResolvedOptions
from pyspark.context import SparkContext
from awsglue.context import GlueContext
from awsglue.job import Job

args = getResolvedOptions(sys.argv, ['JOB_NAME'])
sc = SparkContext()
glueContext = GlueContext(sc)
spark = glueContext.spark_session
job = Job(glueContext)
job.init(args['JOB_NAME'], args)

datasource0 = glueContext.create_dynamic_frame.from_catalog(
    database = "database",
    table_name = "relatedqueries_csv",
    transformation_ctx = "datasource0"
)

applymapping1 = ApplyMapping.apply(
    frame = datasource0,
```

```
    mappings = [("col0", "string", "name", "string"), ("col1", "string", "number",
 "string")],
    transformation_ctx = "applymapping1"
)

datasink2 = glueContext.write_dynamic_frame.from_options(
    frame = applymapping1,
    connection_type = "s3",
    connection_options = {"path": "s3://input_path"},
    format = "json",
    transformation_ctx = "datasink2"
)


job.commit()
```

Example

Di seguito è riportato un esempio di script generato per un'origine JDBC. La tabella di origine è una tabella dipendente con la colonna empno come chiave primaria. Sebbene per impostazione predefinita il processo utilizzi una chiave primaria sequenziale come chiave di segnalibro se non viene specificata alcuna chiave di segnalibro, poiché empno non è necessariamente sequenziale (potrebbero esserci delle lacune nei valori), non si qualifica come chiave segnalibro predefinita. Di conseguenza, lo script designa esplicitamente empno come chiave di segnalibro. Quella parte del codice è mostrata in corsivo.

```
import sys
from awsglue.transforms import *
from awsglue.utils import getResolvedOptions
from pyspark.context import SparkContext
from awsglue.context import GlueContext
from awsglue.job import Job

args = getResolvedOptions(sys.argv, ['JOB_NAME'])

sc = SparkContext()
glueContext = GlueContext(sc)
spark = glueContext.spark_session
job = Job(glueContext)
job.init(args['JOB_NAME'], args)

datasource0 = glueContext.create_dynamic_frame.from_catalog(
```

```
    database = "hr",
    table_name = "emp",
    transformation_ctx = "datasource0",
    additional_options = {"jobBookmarkKeys":["empno"],"jobBookmarkKeysSortOrder":"asc"}
)

applymapping1 = ApplyMapping.apply(
    frame = datasource0,
    mappings = [("ename", "string", "ename", "string"), ("hrly_rate", "decimal(38,0)",
 "hrly_rate", "decimal(38,0)"), ("comm", "decimal(7,2)", "comm", "decimal(7,2)"),
 ("hiredate", "timestamp", "hiredate", "timestamp"), ("empno", "decimal(5,0)", "empno",
 "decimal(5,0)"), ("mgr", "decimal(5,0)", "mgr", "decimal(5,0)"), ("photo", "string",
 "photo", "string"), ("job", "string", "job", "string"), ("deptno", "decimal(3,0)",
 "deptno", "decimal(3,0)"), ("ssn", "decimal(9,0)", "ssn", "decimal(9,0)"), ("sal",
 "decimal(7,2)", "sal", "decimal(7,2)")],
    transformation_ctx = "applymapping1"
)

datasink2 = glueContext.write_dynamic_frame.from_options(
    frame = applymapping1,
    connection_type = "s3",
    connection_options = {"path": "s3://hr/employees"},
    format = "csv",
    transformation_ctx = "datasink2"
)

job.commit()
```

## Utilizzo del rilevamento di dati sensibili all'esterno di AWS Glue Studio

AWS Glue Studio consente di rilevare dati sensibili, tuttavia è possibile utilizzare la funzionalità Sensitive Data Detection anche al di fuori di AWS Glue Studio.

Per un elenco completo dei tipi di dati sensibili gestiti, consulta la pagina Managed Sensitive Data Types.

Rilevamento del rilevamento di dati sensibili utilizzando i tipi di AWS Managed PII

AWS Glue ne fornisce due APIs in un job AWS Glue ETL. Questi sono `detect()` e `classifyColumns()`:

```
  detect(frame: DynamicFrame,
```

```
      entityTypesToDetect: Seq[String],
      outputColumnName: String = "DetectedEntities",
      detectionSensitivity: String = "LOW"): DynamicFrame

 detect(frame: DynamicFrame,
      detectionParameters: JsonOptions,
      outputColumnName: String = "DetectedEntities",
      detectionSensitivity: String = "LOW"): DynamicFrame

  classifyColumns(frame: DynamicFrame,
      entityTypesToDetect: Seq[String],
      sampleFraction: Double = 0.1,
      thresholdFraction: Double = 0.1,
      detectionSensitivity: String = "LOW")
```

È possibile utilizzare l'`detect()`API per identificare i tipi di informazioni AWS personali gestite e i tipi di entità personalizzati. Una nuova colonna viene creata automaticamente con il risultato del rilevamento. L'API `classifyColumns()` restituisce una mappa in cui le chiavi sono i nomi delle colonne e i valori sono un elenco di tipi di entità rilevati. `SampleFraction` indica la frazione dei dati da campionare durante la scansione di ricerca delle entità PII, mentre `ThresholdFraction` indica la frazione dei dati che devono essere soddisfatti per identificare una colonna come dati PII.

Rilevamento a livello di riga

Nell'esempio, il job esegue le seguenti azioni utilizzando `detect()` e `classifyColumns()` APIs:

- legge i dati da un Amazon S3 bucket e li trasforma in un DynamicFrame
- rilevamento di istanze di "e-mail" e "carta di credito" in dynamicFrame
- restituzione di un dynamicFrame con valori originali più una colonna che include il risultato del rilevamento per ogni riga
- scrivere il DynamicFrame restituito in un altro percorso Amazon S3

```
 import com.amazonaws.services.glue.GlueContext
 import com.amazonaws.services.glue.MappingSpec
 import com.amazonaws.services.glue.errors.CallSite
 import com.amazonaws.services.glue.util.GlueArgParser
 import com.amazonaws.services.glue.util.Job
 import com.amazonaws.services.glue.util.JsonOptions
```

```
 import org.apache.spark.SparkContext
 import scala.collection.JavaConverters._
 import com.amazonaws.services.glue.ml.EntityDetector

 object GlueApp {
   def main(sysArgs: Array[String]) {
     val spark: SparkContext = new SparkContext()
     val glueContext: GlueContext = new GlueContext(spark)
     val args = GlueArgParser.getResolvedOptions(sysArgs, Seq("JOB_NAME").toArray)
     Job.init(args("JOB_NAME"), glueContext, args.asJava)
     val frame=
glueContext.getSourceWithFormat(formatOptions=JsonOptions("""{"quoteChar": "\"",
"withHeader": true, "separator": ","}"""), connectionType="s3", format="csv",
options=JsonOptions("""{"paths": ["s3://pathToSource"], "recurse": true}"""),
transformationContext="AmazonS3_node1650160158526").getDynamicFrame()

     val frameWithDetectedPII = EntityDetector.detect(frame, Seq("EMAIL",
"CREDIT_CARD"))

     glueContext.getSinkWithFormat(connectionType="s3",
options=JsonOptions("""{"path": "s3://pathToOutput/", "partitionKeys": []}"""),
transformationContext="someCtx",
format="json").writeDynamicFrame(frameWithDetectedPII)

     Job.commit()
   }
 }
```

Rilevamento a livello di riga con operazioni granulari

Nell'esempio, il job esegue le seguenti azioni utilizzando: `detect()` APIs

- leggere i dati da un bucket di Amazon S3 e trasformarli in un dynamicFrame
- rilevamento dei tipi di dati sensibili per "USA_PTIN", " BANK_ACCOUNT", "USA_SSN", "USA_PASSPORT_NUMBER" e "PHONE_NUMBER" nel dynamicFrame
- restituzione di un dynamicFrame con valori mascherati modificati più una colonna che include il risultato del rilevamento per ogni riga
- scrittura del dynamicFrame restituito in un altro percorso di Amazon S3

A differenza dell'API `detect()` di cui sopra, questa utilizza operazioni granulari per rilevare i tipi di entità. Per ulteriori informazioni, consulta Parametri di rilevamento per l'utilizzo di `detect()`.

```
import com.amazonaws.services.glue.GlueContext
import com.amazonaws.services.glue.MappingSpec
import com.amazonaws.services.glue.errors.CallSite
import com.amazonaws.services.glue.util.GlueArgParser
import com.amazonaws.services.glue.util.Job
import com.amazonaws.services.glue.util.JsonOptions
import org.apache.spark.SparkContext
import scala.collection.JavaConverters._
import com.amazonaws.services.glue.ml.EntityDetector

object GlueApp {
  def main(sysArgs: Array[String]) {
    val spark: SparkContext = new SparkContext()
    val glueContext: GlueContext = new GlueContext(spark)
    val args = GlueArgParser.getResolvedOptions(sysArgs, Seq("JOB_NAME").toArray)
    Job.init(args("JOB_NAME"), glueContext, args.asJava)
    val frame =
 glueContext.getSourceWithFormat(formatOptions=JsonOptions("""{"quoteChar": "\"",
 "withHeader": true, "separator": ","}"""), connectionType="s3", format="csv",
 options=JsonOptions("""{"paths": ["s3://pathToSource"], "recurse": true}"""),
 transformationContext="AmazonS3_node_source").getDynamicFrame()

    val detectionParameters = JsonOptions(
      """
        {
          "USA_DRIVING_LICENSE": [{
            "action": "PARTIAL_REDACT",
            "sourceColumns": ["Driving License"],
            "actionOptions": {
              "matchPattern": "[0-9]",
              "redactChar": "*"
            }
          }],
          "BANK_ACCOUNT": [{
            "action": "DETECT",
            "sourceColumns": ["*"]
          }],
          "USA_SSN": [{
            "action": "SHA256_HASH",
            "sourceColumns": ["SSN"]
```

```
          }],
          "IP_ADDRESS": [{
            "action": "REDACT",
            "sourceColumns": ["IP Address"],
            "actionOptions": {"redactText": "*****"}
          }],
          "PHONE_NUMBER": [{
            "action": "PARTIAL_REDACT",
            "sourceColumns": ["Phone Number"],
            "actionOptions": {
              "numLeftCharsToExclude": 1,
              "numRightCharsToExclude": 0,
              "redactChar": "*"
            }
          }]
        }
      """
    )

    val frameWithDetectedPII = EntityDetector.detect(frame, detectionParameters,
 "DetectedEntities", "HIGH")

    glueContext.getSinkWithFormat(connectionType="s3", options=JsonOptions("""{"path":
 "s3://pathToOutput/", "partitionKeys": []}"""),
 transformationContext="AmazonS3_node_target",
 format="json").writeDynamicFrame(frameWithDetectedPII)

    Job.commit()
  }
}
```

Rilevamento a livello di colonna

Nell'esempio, il job esegue le seguenti azioni utilizzando `classifyColumns()`APIs:

- leggere i dati da un bucket di Amazon S3 e trasformarli in un dynamicFrame
- rilevamento di istanze di “e-mail” e “carta di credito” in dynamicFrame
- imposta i parametri per campionare il 100% della colonna, contrassegna un'entità come rilevata se si trova nel 10% delle celle e ha una sensibilità "LOW"

- restituisce una mappa in cui le chiavi sono i nomi delle colonne e i valori sono l'elenco dei tipi di entità rilevati
- scrittura del dynamicFrame restituito in un altro percorso di Amazon S3

```
import com.amazonaws.services.glue.GlueContext
import com.amazonaws.services.glue.MappingSpec
import com.amazonaws.services.glue.errors.CallSite
import com.amazonaws.services.glue.util.GlueArgParser
import com.amazonaws.services.glue.util.Job
import com.amazonaws.services.glue.util.JsonOptions
import org.apache.spark.SparkContext
import scala.collection.JavaConverters._
import com.amazonaws.services.glue.DynamicFrame
import com.amazonaws.services.glue.ml.EntityDetector

object GlueApp {
  def main(sysArgs: Array[String]) {
    val spark: SparkContext = new SparkContext()
    val glueContext: GlueContext = new GlueContext(spark)
    val args = GlueArgParser.getResolvedOptions(sysArgs, Seq("JOB_NAME").toArray)
    Job.init(args("JOB_NAME"), glueContext, args.asJava)
    val frame =
 glueContext.getSourceWithFormat(formatOptions=JsonOptions("""{"quoteChar":
 "\"", "withHeader": true, "separator": ",", "optimizePerformance": false}"""),
 connectionType="s3", format="csv", options=JsonOptions("""{"paths": ["s3://
pathToSource"], "recurse": true}"""), transformationContext="frame").getDynamicFrame()

    import glueContext.sparkSession.implicits._

    val detectedDataFrame = EntityDetector.classifyColumns(
        frame,
        entityTypesToDetect = Seq("CREDIT_CARD", "PHONE_NUMBER"),
        sampleFraction = 1.0,
        thresholdFraction = 0.1,
        detectionSensitivity = "LOW"
    )
    val detectedDF = (detectedDataFrame).toSeq.toDF("columnName", "entityTypes")
    val DetectSensitiveData_node = DynamicFrame(detectedDF, glueContext)

    glueContext.getSinkWithFormat(connectionType="s3", options=JsonOptions("""{"path":
 "s3://pathToOutput", "partitionKeys": []}"""), transformationContext="someCtx",
 format="json").writeDynamicFrame(DetectSensitiveData_node)
```

```
    Job.commit()
  }
}
```

Rilevamento del rilevamento di dati sensibili mediante tipi di AWS CustomEntityType PII

È possibile definire entità personalizzate tramite AWS Studio. Tuttavia, per utilizzare questa funzionalità fuori da AWS Studio, è necessario prima definire i tipi di entità personalizzati e quindi aggiungere i tipi di entità personalizzati definiti all'elenco di`entityTypesToDetect`.

Se hai tipi di dati sensibili specifici nei tuoi dati (come "ID dipendente"), puoi creare entità personalizzate chiamando l'API `CreateCustomEntityType()`. L'esempio seguente definisce il tipo di entità personalizzato 'EMPLOYEE_ID' per l'API `CreateCustomEntityType()` con i parametri della richiesta:

```
{
    "name": "EMPLOYEE_ID",
    "regexString": "\d{4}-\d{3}",
    "contextWords": ["employee"]
}
```

Quindi, modifica il lavoro per utilizzare il nuovo tipo di dati sensibili personalizzato aggiungendo il tipo di entità personalizzato (EMPLOYEE_ID) all'API `EntityDetector()`:

```
import com.amazonaws.services.glue.GlueContext
import com.amazonaws.services.glue.MappingSpec
import com.amazonaws.services.glue.errors.CallSite
import com.amazonaws.services.glue.util.GlueArgParser
import com.amazonaws.services.glue.util.Job
import com.amazonaws.services.glue.util.JsonOptions
import org.apache.spark.SparkContext
import scala.collection.JavaConverters._
import com.amazonaws.services.glue.ml.EntityDetector

object GlueApp {
  def main(sysArgs: Array[String]) {
```

```
      val spark: SparkContext = new SparkContext()
      val glueContext: GlueContext = new GlueContext(spark)
      val args = GlueArgParser.getResolvedOptions(sysArgs, Seq("JOB_NAME").toArray)
      Job.init(args("JOB_NAME"), glueContext, args.asJava)
      val frame=
 glueContext.getSourceWithFormat(formatOptions=JsonOptions("""{"quoteChar": "\"",
 "withHeader": true, "separator": ","}"""), connectionType="s3", format="csv",
 options=JsonOptions("""{"paths": ["s3://pathToSource"], "recurse": true}"""),
 transformationContext="AmazonS3_node1650160158526").getDynamicFrame()

      val frameWithDetectedPII = EntityDetector.detect(frame, Seq("EMAIL",
 "CREDIT_CARD", "EMPLOYEE_ID"))

      glueContext.getSinkWithFormat(connectionType="s3",
 options=JsonOptions("""{"path": "s3://pathToOutput/", "partitionKeys": []}"""),
 transformationContext="someCtx",
 format="json").writeDynamicFrame(frameWithDetectedPII)

      Job.commit()
    }
  }
```

> **Note**
>
> Se un tipo di dati sensibili personalizzato è definito con lo stesso nome di un tipo di entità gestita esistente, il tipo di dati sensibili personalizzato avrà la precedenza e sovrascriverà la logica del tipo di entità gestita.

Parametri di rilevamento per l'utilizzo di **`detect()`**

Questo metodo viene utilizzato per rilevare le entità in a DynamicFrame. Ne restituisce una nuova DataFrame con valori originali e una colonna aggiuntiva con outputColumnName metadati di rilevamento PII. Il mascheramento personalizzato può essere eseguito dopo che questo DynamicFrame è stato restituito all'interno dello AWS Glue script, oppure è possibile utilizzare l'API detect () con azioni granulari.

```
detect(frame: DynamicFrame,
       entityTypesToDetect: Seq[String],
       outputColumnName: String = "DetectedEntities",
```

```
        detectionSensitivity: String = "LOW"): DynamicFrame
```

Parametri:

- frame — (type:`DynamicFrame`) L'input DynamicFrame contenente i dati da elaborare.
- entityTypesToRileva — (tipo:`[Seq[String]`) Elenco dei tipi di entità da rilevare. Possono essere tipi di entità gestiti o personalizzati.
- outputColumnName— (type:`String`, default: "DetectedEntities«) Il nome della colonna in cui verranno memorizzate le entità rilevate. Se non viene fornito, il nome di colonna predefinito è "DetectedEntities».
- detectionSensivity — (tipo: `String`, opzioni: "BASSA" oppure "ELEVATA", impostazione predefinita: "BASSA") specifica la distinzione del processo di rilevamento. Le opzioni valide sono "BASSA" oppure "ELEVATA". Se non viene fornita, la distinzione predefinita è impostata su "LOW".

Impostazioni di `outputColumnName`:

Il nome della colonna in cui verranno memorizzate le entità rilevate. Se non viene fornito, il nome di colonna predefinito è "DetectedEntities». Per ogni riga della colonna di output, la colonna supplementare include una mappa del nome della colonna ai metadati dell'entità rilevata, con le seguenti coppie chiave-valore:

- entityType: il tipo di entità rilevato.
- start — la posizione iniziale dell'entità rilevata nei dati originali.
- end — la posizione finale dell'entità rilevata nei dati originali.
- actionUsed — L'azione eseguita sull'entità rilevata (ad esempio, «DETECT», «REDACT», «PARTIAL_REDACT», "_HASH»). SHA256

Esempio:

```
{
   "DetectedEntities":{
      "SSN Col":[
         {
            "entityType":"USA_SSN",
            "actionUsed":"DETECT",
            "start":4,
```

```
            "end":15
         }
      ],
      "Random Data col":[
         {
            "entityType":"BANK_ACCOUNT",
            "actionUsed":"PARTIAL_REDACT",
            "start":4,
            "end":13
         },
         {
            "entityType":"IP_ADDRESS",
            "actionUsed":"REDACT",
            "start":4,
            "end":13
         }
      ]
   }
}
```

Parametri di rilevamento per **`detect()`** con operazioni granulari

Questo metodo viene utilizzato per rilevare le entità in un utilizzando parametri specifici. DynamicFrame Ne restituisce una nuova DataFrame con valori originali sostituiti con dati sensibili mascherati e una colonna aggiuntiva con metadati `outputColumnName` di rilevamento PII.

```
detect(frame: DynamicFrame,
       detectionParameters: JsonOptions,
       outputColumnName: String = "DetectedEntities",
       detectionSensitivity: String = "LOW"): DynamicFrame
```

Parametri:

- frame — (type:`DynamicFrame`): L'input DynamicFrame contenente i dati da elaborare.
- detectionParameters — (tipo: `JsonOptions`): opzioni JSON che specificano i parametri per il processo di rilevamento.
- outputColumnName— (type:`String`, default: "DetectedEntities«): Il nome della colonna in cui verranno memorizzate le entità rilevate. Se non viene fornito, il nome di colonna predefinito è "DetectedEntities».

- detectionSensivity — (tipo: `String`, opzioni: "BASSA" oppure "ELEVATA", impostazione predefinita: "BASSA"): specifica la distinzione del processo di rilevamento. Le opzioni valide sono "BASSA" oppure "ELEVATA". Se non viene fornita, la distinzione predefinita è impostata su "LOW".

Impostazioni di `detectionParameters`

Se non è inclusa alcuna impostazione, verranno utilizzati i valori predefiniti.

- action — (type:`String`, options: «DETECT», «REDACT», «PARTIAL_REDACT», "SHA256_HASH») Specifica l'azione da eseguire sull'entità. Obbligatorio. Le operazioni che eseguono il mascheramento (ad eccezione di "DETECT") possono eseguire solo un'operazione per colonna. Si tratta di una misura preventiva per mascherare le entità unite.
- sourceColumns — (tipo: `List[String]`, impostazione predefinita: ["*"]) elenco dei nomi delle colonne di origine su cui eseguire il rilevamento dell'entità. L'impostazione predefinita è ["*"], se non presente. Viene generato `IllegalArgumentException` se viene utilizzato un nome di colonna non valido.
- sourceColumnsToEscludi — (digitare:`List[String]`) Elenco dei nomi delle colonne di origine su cui eseguire il rilevamento dell'entità. Usa `sourceColumns` o `sourceColumnsToExclude`. Viene generato `IllegalArgumentException` se viene utilizzato un nome di colonna non valido.
- actionOptions — opzioni aggiuntive basate sull'operazione specificata:
  - Per «DETECT» e "SHA256_HASH», non sono consentite opzioni.
  - Per "REDACT":
    - redactText — (tipo: `String`, impostazione predefinita: "*****") testo per sostituire l'entità rilevata.
  - Per "PARTIAL_REDACT":
    - redactChar — (tipo: `String`, impostazione predefinita: "*") carattere per sostituire ogni carattere rilevato nell'entità.
    - matchPattern — (tipo: `String`) modello Regex per la redazione parziale. Non può essere combinato con numLeftChars ToExclude o. `numRightCharsToExclude`
    - numLeftCharsToExclude— (tipo:`String, integer`) Numero di caratteri a sinistra da escludere. Non può essere unito a matchPattern, ma può essere utilizzato con `numRightCharsToExclude`.
    - numRightCharsToExclude— (digitare:`String, integer`) Numero di caratteri a destra da escludere. Non può essere unito a matchPattern, ma può essere utilizzato con `numRightCharsToExclude`.

Impostazioni di `outputColumnName`

Vedi outputColumnName le impostazioni

Parametri di rilevamento per **`classifyColumns()`**

Questo metodo viene utilizzato per rilevare le entità in a DynamicFrame. Restituisce una mappa in cui le chiavi sono i nomi delle colonne e i valori sono l'elenco dei tipi di entità rilevati. Il mascheramento personalizzato può essere eseguito dopo che questo è stato restituito all'interno dello script AWS Glue.

```
classifyColumns(frame: DynamicFrame,
                entityTypesToDetect: Seq[String],
                sampleFraction: Double = 0.1,
                thresholdFraction: Double = 0.1,
                detectionSensitivity: String = "LOW")
```

Parametri:

- frame — (tipo:`DynamicFrame`) L'input DynamicFrame contenente i dati da elaborare.
- entityTypesToRileva — (tipo:`Seq[String]`) Elenco dei tipi di entità da rilevare. Possono essere tipi di entità gestiti o personalizzati.
- sampleFraction — (tipo: `Double`, impostazione predefinita: 10%) la frazione dei dati da campionare durante la scansione di entità PII.
- thresholdFraction — (tipo: `Double`, impostazione predefinita: 10%): la frazione dei dati che devono essere soddisfatti per identificare una colonna come dati PII.
- detectionSensivity — (tipo: `String`, opzioni: "BASSA" oppure "ELEVATA", impostazione predefinita: "BASSA") specifica la distinzione del processo di rilevamento. Le opzioni valide sono "BASSA" oppure "ELEVATA". Se non viene fornita, la distinzione predefinita è impostata su "LOW".

Tipi di dati sensibili gestiti

Entità globali

| Tipo di dati | Categoria | Description |
|---|---|---|
| PERSON_NAME | Universal | Il nome della persona. |

| Tipo di dati | Categoria | Description |
|---|---|---|
| EMAIL | Personale | L'indirizzo e-mail. |
| IP_ADDRESS | Computer | L'indirizzo IP |
| MAC_ADDRESS | Personale | L'indirizzo MAC. |

Tipi di dati negli Stati Uniti

| Tipo di dati | Description |
|---|---|
| BANK_ACCOUNT | Il numero di conto bancario. Non è specifico per un paese o una regione, tuttavia vengono rilevati solo i formati di conto statunitensi e canadesi. |
| CREDIT_CARD | Il numero di carta di credito. |
| PHONE_NUMBER | Il numero di telefono. Non è specifico per un paese o una regione, tuttavia, al momento vengono rilevati solo i numeri di telefono statunitensi e canadesi. |
| USA_ATIN | Il numero identificativo del contribuente per l'adozione negli Stati Uniti rilasciato dall'Internal Revenue Service. |
| USA_CPT_CODE | Il codice CPT (specifico per gli Stati Uniti). |
| USA_DEA_NUMBER | Il numero DEA (specifico per gli Stati Uniti). |
| USA_DRIVING_LICENSE | Il numero della patente di guida (specifico per gli Stati Uniti). |
| USA_HCPCS_CODE | Il codice HCPCS (specifico per gli Stati Uniti). |

| Tipo di dati | Description |
| --- | --- |
| USA_HEALTH_INSURANCE_CLAIM_NUMBER | Il numero di attestazione dell'assicurazione sanitaria (specifico per gli Stati Uniti). |
| USA_ITIN | L'ITIN (per persone o entità statunitensi). |
| USA_MEDICARE_BENEFICIARY_IDENTIFIER | Il numero identificativo del beneficiario di Medicare (specifico per gli Stati Uniti). |
| USA_NATIONAL_DRUG_CODE | Il codice NDC (specifico per gli Stati Uniti). |
| USA_NATIONAL_PROVIDER_IDENTIFIER | Il numero identificativo del fornitore nazionale (specifico per gli Stati Uniti). |
| USA_PASSPORT_NUMBER | Il numero del passaporto (per i cittadini statunitensi). |
| USA_PTIN | Il codice di identificazione fiscale US Preparer TIN rilasciato dall'Internal Revenue Service. |
| USA_SSN | Il numero di previdenza sociale (per i cittadini statunitensi). |

Tipi di dati in Argentina

| Tipo di dati | Description |
| --- | --- |
| ARGENTINA_TAX_IDENTIFICATION_NUMBER | Il codice di identificazione fiscale dell'Argentina. Conosciuto anche come CUIT o CUIL. |

Tipi di dati in Australia

| Tipo di dati | Description |
| --- | --- |
| AUSTRALIA_BUSINESS_NUMBER | Australia Business Number (ABN). Un codice identificativo univoco rilasciato dall'Australian |

| Tipo di dati | Description |
| --- | --- |
| | Business Register (ABR) per identificare le aziende a livello amministrativo e pubblico. |
| AUSTRALIA_COMPANY_NUMBER | Australia Company Number (ACN). Un codice identificativo univoco rilasciato dalla Australian Securities and Investments Commission. |
| AUSTRALIA_DRIVING_LICENSE | Il numero della patente di guida per l'Australia. |
| AUSTRALIA_MEDICARE_NUMBER | Il numero Medicare australiano. Il numero identificativo personale rilasciato dalla Australia n Health Insurance Commission. |
| AUSTRALIA_PASSPORT_NUMBER | Il numero del passaporto australiano. |
| AUSTRALIA_TAX_FILE_NUMBER | Il codice fiscale australiano (TFN). Rilasciat o dall'Australian Taxation Office (ATO) ai contribuenti (persone fisiche, società, ecc.) per le finalità fiscali. |

Tipi di dati in Austria

| Tipo di dati | Description |
| --- | --- |
| AUSTRIA_DRIVING_LICENSE | Il numero della patente di guida (specifico per l'Austria). |
| AUSTRIA_PASSPORT_NUMBER | Il numero del passaporto (specifico per l'Austria ). |
| AUSTRIA_SSN | Il numero di previdenza sociale (per i cittadini austriaci). |
| AUSTRIA_TAX_IDENTIFICATION_NUMBER | Il codice di identificazione fiscale (specifico per l'Austria). |

| Tipo di dati | Description |
| --- | --- |
| AUSTRIA_VALUE_ADDED_TAX | L'imposta sul valore aggiunto (specifica per l'Austria). |

Tipi di dati nell'area balcanica

| Tipo di dati | Description |
| --- | --- |
| BOSNIA_UNIQUE_MASTER_CITIZEN_NUMBER | Il codice identificativo unico (JMBG) per i cittadini della Bosnia-Erzegovina. |
| KOSOVO_UNIQUE_MASTER_CITIZEN_NUMBER | Il codice identificativo unico (JMBG) per il Kosovo. |
| MACEDONIA_UNIQUE_MASTER_CITIZEN_NUMBER | Il codice identificativo unico per la Macedonia. |
| MONTENEGRO_UNIQUE_MASTER_CITIZEN_NUMBER | Il codice identificativo unico (JMBG) per il Montenegro. |
| SERBIA_UNIQUE_MASTER_CITIZEN_NUMBER | Il codice identificativo unico (JMBG) per la Serbia. |
| SERBIA_VALUE_ADDED_TAX | L'imposta sul valore aggiunto (specifica per la Serbia). |
| VOJVODINA_UNIQUE_MASTER_CITIZEN_NUMBER | Il codice identificativo unico (JMBG) per la Vojvodina. |

Tipi di dati in Belgio

| Tipo di dati | Description |
| --- | --- |
| BELGIUM_DRIVING_LICENSE | Il numero della patente di guida (specifico per il Belgio). |

| Tipo di dati | Description |
| --- | --- |
| BELGIUM_NATIONAL_IDENTIFICATION_NUMBER | Il numero nazionale belga (BNN). |
| BELGIUM_PASSPORT_NUMBER | Il numero del passaporto (specifico per il Belgio). |
| BELGIUM_TAX_IDENTIFICATION_NUMBER | Il codice di identificazione fiscale (specifico per il Belgio). |
| BELGIUM_VALUE_ADDED_TAX | L'imposta sul valore aggiunto (specifica per il Belgio). |

Tipi di dati in Brasile

| Tipo di dati | Description |
| --- | --- |
| BRAZIL_BANK_ACCOUNT | Il numero di conto bancario (specifico per il Brasile). |
| BRAZIL_NATIONAL_IDENTIFICATION_NUMBER | Il numero identificativo nazionale (specifico per il Brasile). |
| BRAZIL_NATIONAL_REGISTRY_OF_LEGAL_ENTITIES_NUMBER | Il codice di identificazione rilasciato alle società (specifico per il Brasile), noto anche come CNPJ. |
| BRAZIL_NATURAL_PERSON_REGISTRY_NUMBER | Numero di registro delle persone fisiche, noto anche come CPF. |

Tipi di dati in Bulgaria

| Tipo di dati | Description |
| --- | --- |
| BULGARIA_DRIVING_LICENSE | Il numero della patente di guida (specifico per la Bulgaria). |

| Tipo di dati | Description |
| --- | --- |
| BULGARIA_UNIFORM_CIVIL_NUMBER | Il codice di identificazione unificato (EGN) che funge da numero di identificazione nazionale. |
| BULGARIA_VALUE_ADDED_TAX | L'imposta sul valore aggiunto (specifica per la Bulgaria). |

Tipi di dati in Canada

| Tipo di dati | Description |
| --- | --- |
| CANADA_DRIVING_LICENSE | Il numero della patente di guida (specifico per il Canada). |
| CANADA_GOVERNMENT_IDENTIFICATION_CARD_NUMBER | Il numero identificativo nazionale (specifico per il Canada). |
| CANADA_PASSPORT_NUMBER | Il numero del passaporto (specifico per il Canada). |
| CANADA_PERMANENT_RESIDENCE_NUMBER | Il numero di residenza permanente (numero di PR Card). |
| CANADA_PERSONAL_HEALTH_NUMBER | Il codice identificativo univoco per l'assistenza sanitaria (numero PHN). |
| CANADA_SOCIAL_INSURANCE_NUMBER | Il numero di previdenza sociale (SIN) in Canada. |

Tipi di dati in Cile

| Tipo di dati | Description |
| --- | --- |
| CHILE_DRIVING_LICENSE | Il numero della patente di guida (specifico per il Cile). |

| Tipo di dati | Description |
| --- | --- |
| CHILE_NATIONAL_IDENTIFICATION_NUMBER | Il numero identificativo nazionale per il Cile, noto anche come RUT o RUN. |

Tipi di dati in Cina, Hong Kong, Macao e Taiwan

| Tipo di dati | Description |
| --- | --- |
| CHINA_IDENTIFICATION | L'identificatore cinese. |
| CHINA_LICENSE_PLATE_NUMBER | Il numero della patente di guida (specifico per la Cina). |
| CHINA_MAINLAND_TRAVEL_PERMIT_ID_HONG_KONG_MACAU | Il permesso di viaggio sulla terraferma per i residenti di Hong Kong e Macao. |
| CHINA_MAINLAND_TRAVEL_PERMIT_ID_TAIWAN | Il permesso di viaggio sulla terraferma per i residenti di Taiwan rilasciato dal governo della Repubblica Popolare Cinese (RPC). |
| CHINA_PASSPORT_NUMBER | Il numero del passaporto (specifico per la Cina). |
| CHINA_PHONE_NUMBER | Il numero di telefono (specifico per la Cina). |
| HONG_KONG_IDENTITY_CARD | Il documento di identità ufficiale rilasciato dal Dipartimento dell'Immigrazione di Hong Kong. |
| MACAU_RESIDENT_IDENTITY_CARD | La Macau Resident Identity Card o BIR è una carta d'identità ufficiale emessa dall'Identification Services Bureau di Macao. |
| TAIWAN_NATIONAL_IDENTIFICATION_NUMBER | Il numero identificativo nazionale (specifico per Taiwan). |
| TAIWAN_PASSPORT_NUMBER | Il numero del passaporto (specifico per Taiwan). |

Tipi di dati in Colombia

| Tipo di dati | Description |
| --- | --- |
| COLOMBIA_PERSONAL_IDENTIFICATION_NUMBER | Il codice identificativo univoco assegnato ai colombiani alla nascita. |
| COLOMBIA_TAX_IDENTIFICATION_NUMBER | Il codice di identificazione fiscale (specifico per la Colombia). |

Tipi di dati in Croazia

| Tipo di dati | Description |
| --- | --- |
| CROATIA_DRIVING_LICENSE | Il numero della patente di guida (specifico per la Croazia). |
| CROATIA_IDENTITY_NUMBER | Il numero identificativo nazionale (specifico per la Croazia). |
| CROATIA_PASSPORT_NUMBER | Il numero del passaporto (specifico per la Croazia). |
| CROATIA_PERSONAL_IDENTIFICATION_NUMBER | Il codice di identificazione personale (OIB). |

Tipi di dati a Cipro

| Tipo di dati | Description |
| --- | --- |
| CYPRUS_DRIVING_LICENSE | Il numero della patente di guida (specifico per Cipro). |
| CYPRUS_NATIONAL_IDENTIFICATION_NUMBER | La carta d'identità cipriota. |
| CYPRUS_PASSPORT_NUMBER | Il numero del passaporto (specifico per Cipro). |

| Tipo di dati | Description |
| --- | --- |
| CYPRUS_TAX_IDENTIFICATION_NUMBER | Il codice di identificazione fiscale (specifico per Cipro). |
| CYPRUS_VALUE_ADDED_TAX | L'imposta sul valore aggiunto (specifica per Cipro). |

Tipi di dati in Repubblica Ceca

| Tipo di dati | Description |
| --- | --- |
| CZECHIA_DRIVING_LICENSE | Numero della patente di guida (specifico per la Repubblica Ceca). |
| CZECHIA_PERSONAL_IDENTIFICATION_NUMBER | Il codice di identificazione personale (specifico per la Repubblica Ceca). |
| CZECHIA_VALUE_ADDED_TAX | L'imposta sul valore aggiunto (specifica per la Repubblica Ceca). |

Tipi di dati in Danimarca

| Tipo di dati | Description |
| --- | --- |
| DENMARK_DRIVING_LICENSE | Il numero della patente di guida (specifico per la Danimarca). |
| DENMARK_PERSONAL_IDENTIFICATION_NUMBER | Il codice di identificazione personale (specifico per la Danimarca). |
| DENMARK_TAX_IDENTIFICATION_NUMBER | Il codice di identificazione fiscale (specifico per la Danimarca). |
| DENMARK_VALUE_ADDED_TAX | L'imposta sul valore aggiunto (specifica per la Danimarca). |

Tipi di dati in Estonia

| Tipo di dati | Description |
| --- | --- |
| ESTONIA_DRIVING_LICENSE | Il numero della patente di guida (specifico per l'Estonia). |
| ESTONIA_PASSPORT_NUMBER | Il numero del passaporto (specifico per l'Estonia). |
| ESTONIA_PERSONAL_IDENTIFICATION_CODE | Il codice di identificazione personale (specifico per l'Estonia). |
| ESTONIA_VALUE_ADDED_TAX | L'imposta sul valore aggiunto (specifica per l'Estonia). |

Tipi di dati in Finlandia

| Tipo di dati | Description |
| --- | --- |
| FINLAND_DRIVING_LICENSE | Il numero della patente di guida (specifico per la Finlandia). |
| FINLAND_HEALTH_INSURANCE_NUMBER | Il numero dell'assicurazione sanitaria (specifico per la Finlandia). |
| FINLAND_NATIONAL_IDENTIFICATION_NUMBER | Il numero identificativo nazionale (specifico per la Finlandia). |
| FINLAND_PASSPORT_NUMBER | Il numero del passaporto (specifico per la Finlandia). |
| FINLAND_VALUE_ADDED_TAX | L'imposta sul valore aggiunto (specifica per la Finlandia). |

Tipi di dati in Francia

| Tipo di dati | Description |
| --- | --- |
| FRANCE_BANK_ACCOUNT | Il numero di conto bancario (specifico per la Francia). |
| FRANCE_DRIVING_LICENSE | Il numero della patente di guida (specifico per la Francia). |
| FRANCE_HEALTH_INSURANCE_NUMBER | Il numero dell'assicurazione sanitaria in Francia. |
| FRANCE_INSEE_CODE | Il codice di previdenza sociale, SSN o NIR in Francia. |
| FRANCE_NATIONAL_IDENTIFICATION_NUMBER | Il numero identificativo nazionale (CNI) per la Francia. |
| FRANCE_PASSPORT_NUMBER | Il numero del passaporto (specifico per la Francia). |
| FRANCE_TAX_IDENTIFICATION_NUMBER | Il codice di identificazione fiscale (specifico per la Francia). |
| FRANCE_VALUE_ADDED_TAX | L'imposta sul valore aggiunto (specifica per la Francia). |

Tipi di dati in Germania

| Tipo di dati | Description |
| --- | --- |
| GERMANY_BANK_ACCOUNT | Il numero di conto bancario (specifico per la Germania). |
| GERMANY_DRIVING_LICENSE | Il numero della patente di guida (specifico per la Germania). |
| GERMANY_PASSPORT_NUMBER | Il numero del passaporto (specifico per la Germania). |

| Tipo di dati | Description |
| --- | --- |
| GERMANY_PERSONAL_IDENTIFICATION_NUMBER | Il codice di identificazione personale (specifico per la Germania). |
| GERMANY_TAX_IDENTIFICATION_NUMBER | Il codice di identificazione fiscale (specifico per la Germania). |
| GERMANY_VALUE_ADDED_TAX | L'imposta sul valore aggiunto (specifica per la Germania). |

Tipi di dati in Grecia

| Tipo di dati | Description |
| --- | --- |
| GREECE_DRIVING_LICENSE | Il numero della patente di guida (specifico per la Grecia). |
| GREECE_PASSPORT_NUMBER | Il numero del passaporto (specifico per la Grecia). |
| GREECE_SSN | Il numero di previdenza sociale (per i cittadini greci). |
| GREECE_TAX_IDENTIFICATION_NUMBER | Il codice di identificazione fiscale (specifico per la Grecia). |
| GREECE_VALUE_ADDED_TAX | L'imposta sul valore aggiunto (specifica per la Grecia). |

Tipi di dati in Ungheria

| Tipo di dati | Description |
| --- | --- |
| HUNGARY_DRIVING_LICENSE | Il numero della patente di guida (specifico per l'Ungheria). |

| Tipo di dati | Description |
| --- | --- |
| HUNGARY_PASSPORT_NUMBER | Il numero del passaporto (specifico per l'Ungheria). |
| HUNGARY_SSN | Il numero di previdenza sociale (per i cittadini ungheresi). |
| HUNGARY_TAX_IDENTIFICATION_NUMBER | Il codice di identificazione fiscale (specifico per l'Ungheria). |
| HUNGARY_VALUE_ADDED_TAX | L'imposta sul valore aggiunto (specifica per l'Ungheria). |

Tipi di dati in Islanda

| Tipo di dati | Description |
| --- | --- |
| ICELAND_NATIONAL_IDENTIFICATION_NUMBER | Il numero identificativo nazionale (specifico per l'Islanda). |
| ICELAND_PASSPORT_NUMBER | Il numero del passaporto (specifico per l'Islanda ). |
| ICELAND_VALUE_ADDED_TAX | L'imposta sul valore aggiunto (specifica per l'Islanda). |

Tipi di dati in India

| Tipo di dati | Description |
| --- | --- |
| INDIA_AADHAAR_NUMBER | Il codice di identificazione Aadhaar rilasciato dalla Unique Identification Authority of India. |
| INDIA_PERMANENT_ACCOUNT_NUMBER | il Permanent Account Number (PAN) indiano. |

Tipi di dati in Indonesia

| Tipo di dati | Description |
| --- | --- |
| INDONESIA_IDENTITY_CARD_NUMBER | Il numero identificativo nazionale (specifico per l'Indonesia). |

Tipi di dati in Irlanda

| Tipo di dati | Description |
| --- | --- |
| IRELAND_DRIVING_LICENSE | Il numero della patente di guida (specifico per l'Irlanda). |
| IRELAND_PASSPORT_NUMBER | Il numero del passaporto (specifico per l'Irlanda ). |
| IRELAND_PERSONAL_PUBLIC_SER VICE_NUMBER | Il numero di servizio pubblico personale (PPS) irlandese. |
| IRELAND_TAX_IDENTIFICATION_NUMBER | Il codice di identificazione fiscale (specifico per l'Irlanda). |
| IRELAND_VALUE_ADDED_TAX | L'imposta sul valore aggiunto (specifica per l'Irlanda). |

Tipi di dati in Israele

| Tipo di dati | Description |
| --- | --- |
| ISRAEL_IDENTIFICATION_NUMBER | Il numero identificativo nazionale (specifico per Israele). |

Tipi di dati in Italia

| Tipo di dati | Description |
| --- | --- |
| ITALY_BANK_ACCOUNT | Il numero di conto bancario (specifico per l'Italia). |
| ITALY_DRIVING_LICENSE | Il numero della patente di guida (specifico per l'Italia). |
| ITALY_FISCAL_CODE | Il codice identificativo, noto anche come codice fiscale italiano. |
| ITALY_PASSPORT_NUMBER | Il numero del passaporto (specifico per l'Italia). |
| ITALY_VALUE_ADDED_TAX | L'imposta sul valore aggiunto (specifica per l'Italia). |

Tipi di dati in Giappone

| Tipo di dati | Description |
| --- | --- |
| JAPAN_BANK_ACCOUNT | Il numero di conto bancario in Giappone. |
| JAPAN_DRIVING_LICENSE | Il numero di patente di guida per il Giappone. |
| JAPAN_MY_NUMBER | L'identificativo univoco per i cittadini o le società giapponesi utilizzato per l'amministrazione fiscale, l'amministrazione della sicurezza sociale e la risposta alle catastrofi |
| JAPAN_PASSPORT_NUMBER | Il numero del passaporto giapponese. |

Tipi di dati in Corea

| Tipo di dati | Description |
| --- | --- |
| KOREA_PASSPORT_NUMBER | Il numero del passaporto (specifico per la Corea). |

| Tipo di dati | Description |
| --- | --- |
| KOREA_RESIDENCE_REGISTRATION_NUMBER_FOR_CITIZENS | Il numero di registrazione di residenza coreano per i cittadini. |
| KOREA_RESIDENCE_REGISTRATION_NUMBER_FOR_FOREIGNERS | Il numero di registrazione di residenza coreano per gli stranieri. |

Tipi di dati in Lettonia

| Tipo di dati | Description |
| --- | --- |
| LATVIA_DRIVING_LICENSE | Il numero della patente di guida (specifico per la Lettonia). |
| LATVIA_PASSPORT_NUMBER | Il numero del passaporto (specifico per la Lettonia). |
| LATVIA_PERSONAL_IDENTIFICATION_NUMBER | Il codice di identificazione personale (specifico per la Lettonia). |
| LATVIA_VALUE_ADDED_TAX | L'imposta sul valore aggiunto (specifica per la Lettonia). |

Tipi di dati nel Liechtenstein

| Tipo di dati | Description |
| --- | --- |
| LIECHTENSTEIN_NATIONAL_IDENTIFICATION_NUMBER | Il numero identificativo nazionale (specifico per il Liechtenstein). |
| LIECHTENSTEIN_PASSPORT_NUMBER | Il numero del passaporto (specifico per il Liechtenstein). |
| LIECHTENSTEIN_TAX_IDENTIFICATION_NUMBER | Il codice di identificazione fiscale (specifico del Liechtenstein). |

Tipi di dati in Lituania

| Tipo di dati | Description |
| --- | --- |
| LITHUANIA_DRIVING_LICENSE | Il numero della patente di guida (specifico per la Lituania). |
| LITHUANIA_PERSONAL_IDENTIFICATION_NUMBER | Il codice di identificazione personale (specifico per la Lituania). |
| LITHUANIA_TAX_IDENTIFICATION_NUMBER | Il codice di identificazione fiscale (specifico per la Lituania). |
| LITHUANIA_VALUE_ADDED_TAX | L'imposta sul valore aggiunto (specifica per la Lituania). |

Tipi di dati in Lussemburgo

| Tipo di dati | Description |
| --- | --- |
| LUXEMBOURG_DRIVING_LICENSE | Il numero della patente di guida (specifico per il Lussemburgo). |
| LUXEMBOURG_NATIONAL_INDIVIDUAL_NUMBER | Il numero identificativo nazionale (specifico per il Lussemburgo). |
| LUXEMBOURG_PASSPORT_NUMBER | Il numero del passaporto (specifico per il Lussemburgo). |
| LUXEMBOURG_TAX_IDENTIFICATION_NUMBER | Il codice di identificazione fiscale (specifico per il Lussemburgo). |
| LUXEMBOURG_VALUE_ADDED_TAX | L'imposta sul valore aggiunto (specifica per il Lussemburgo). |

Tipi di dati in Malesia

| Tipo di dati | Description |
| --- | --- |
| MALAYSIA_MYKAD_NUMBER | Il numero identificativo nazionale (specifico per la Malesia). |
| MALAYSIA_PASSPORT_NUMBER | Il numero del passaporto (specifico per la Malesia). |

Tipi di dati a Malta

| Tipo di dati | Description |
| --- | --- |
| MALTA_DRIVING_LICENSE | Il numero della patente di guida (specifico per Malta). |
| MALTA_NATIONAL_IDENTIFICATION_NUMBER | Il numero identificativo nazionale (specifico per Malta). |
| MALTA_TAX_IDENTIFICATION_NUMBER | Il codice di identificazione fiscale (specifico per Malta). |
| MALTA_VALUE_ADDED_TAX | L'imposta sul valore aggiunto (specifica per Malta). |

Tipi di dati in Messico

| Tipo di dati | Description |
| --- | --- |
| MEXICO_CLABE_NUMBER | Codice bancario Mexico CLABE (Clave Bancaria Estandarizada). |
| MEXICO_DRIVING_LICENSE | Numero della patente di guida (specifico per il Messico). |
| MEXICO_PASSPORT_NUMBER | Il numero del passaporto (specifico per il Messico). |

| Tipo di dati | Description |
| --- | --- |
| MEXICO_TAX_IDENTIFICATION_NUMBER | Il codice di identificazione fiscale (specifico per il Messico). |
| MEXICO_UNIQUE_POPULATION_REGISTRY_CODE | La Clave Única de Registro de Población (CURP) o codice di identità del Messico. |

Tipi di dati nei Paesi Bassi

| Tipo di dati | Description |
| --- | --- |
| NETHERLANDS_CITIZEN_SERVICE_NUMBER | Il codice identificativo olandese (BSN, burgerservicenummer). |
| NETHERLANDS_DRIVING_LICENSE | Il numero della patente di guida (specifico per i Paesi Bassi). |
| NETHERLANDS_PASSPORT_NUMBER | Il numero del passaporto (specifico per i Paesi Bassi). |
| NETHERLANDS_TAX_IDENTIFICATION_NUMBER | Il codice di identificazione fiscale (specifico per i Paesi Bassi). |
| NETHERLANDS_VALUE_ADDED_TAX | L'imposta sul valore aggiunto (specifica per i Paesi Bassi). |
| NETHERLANDS_BANK_ACCOUNT | Il numero di conto bancario (specifico per i Paesi Bassi). |

Tipi di dati in Nuova Zelanda

| Tipo di dati | Description |
| --- | --- |
| NEW_ZEALAND_DRIVING_LICENSE | Il numero della patente di guida (specifico per la Nuova Zelanda). |

| Tipo di dati | Description |
| --- | --- |
| NEW_ZEALAND_NATIONAL_HEALTH_INDEX_NUMBER | Il numero del sistema sanitario nazionale della Nuova Zelanda. |
| NEW_ZEALAND_TAX_IDENTIFICATION_NUMBER | Il codice di identificazione fiscale, noto anche come codice fiscale interno (specifico per la Nuova Zelanda). |

Tipi di dati in Norvegia

| Tipo di dati | Description |
| --- | --- |
| NORWAY_BIRTH_NUMBER | Il numero di identità nazionale norvegese. |
| NORWAY_DRIVING_LICENSE | Il numero della patente di guida (specifico per la Norvegia). |
| NORWAY_HEALTH_INSURANCE_NUMBER | Il numero dell'assicurazione sanitaria norvegese. |
| NORWAY_NATIONAL_IDENTIFICATION_NUMBER | Il numero identificativo nazionale (specifico per la Norvegia). |
| NORWAY_VALUE_ADDED_TAX | L'imposta sul valore aggiunto (specifica per la Norvegia). |

Tipi di dati nelle Filippine

| Tipo di dati | Description |
| --- | --- |
| PHILIPPINES_DRIVING_LICENSE | Il numero della patente di guida (specifico per le Filippine). |
| PHILIPPINES_PASSPORT_NUMBER | Il numero del passaporto (specifico per le Filippine). |

Tipi di dati in Polonia

| Tipo di dati | Description |
| --- | --- |
| POLAND_DRIVING_LICENSE | Il numero della patente di guida (specifico per la Polonia). |
| POLAND_IDENTIFICATION_NUMBER | Il codice identificativo della Polonia. |
| POLAND_PASSPORT_NUMBER | Il numero del passaporto (specifico per la Polonia). |
| POLAND_REGON_NUMBER | Il codice di identificazione REGON, noto anche come numero di identificazione statistica. |
| POLAND_SSN | Il numero di previdenza sociale (per i cittadini polacchi). |
| POLAND_TAX_IDENTIFICATION_NUMBER | Il codice di identificazione fiscale (specifico per la Polonia). |
| POLAND_VALUE_ADDED_TAX | L'imposta sul valore aggiunto (specifica per la Polonia). |

Tipi di dati in Portogallo

| Tipo di dati | Description |
| --- | --- |
| PORTUGAL_DRIVING_LICENSE | Il numero della patente di guida (specifico per il Portogallo). |
| PORTUGAL_NATIONAL_IDENTIFIC ATION_NUMBER | Il numero identificativo nazionale (specifico per il Portogallo). |
| PORTUGAL_PASSPORT_NUMBER | Il numero del passaporto (specifico per il Portogallo). |
| PORTUGAL_TAX_IDENTIFICATION_NUMBER | Il codice di identificazione fiscale (specifico per il Portogallo). |

| Tipo di dati | Description |
| --- | --- |
| PORTUGAL_VALUE_ADDED_TAX | L'imposta sul valore aggiunto (specifica per il Portogallo). |

Tipi di dati in Romania

| Tipo di dati | Description |
| --- | --- |
| ROMANIA_DRIVING_LICENSE | Il numero della patente di guida (specifico per la Romania). |
| ROMANIA_NUMERICAL_PERSONAL_CODE | Il codice di identificazione personale (specifico per la Romania). |
| ROMANIA_PASSPORT_NUMBER | Il numero del passaporto (specifico per la Romania). |
| ROMANIA_VALUE_ADDED_TAX | L'imposta sul valore aggiunto (specifica per la Romania). |

Tipi di dati a Singapore

| Tipo di dati | Description |
| --- | --- |
| SINGAPORE_DRIVING_LICENSE | Il numero della patente di guida (specifico per Singapore). |
| SINGAPORE_NATIONAL_REGISTRY _IDENTIFICATION_NUMBER | La carta d'identità nazionale di Singapore. |
| SINGAPORE_PASSPORT_NUMBER | Il numero del passaporto (specifico per Singapore). |
| SINGAPORE_UNIQUE_ENTITY_NUMBER | Il numero di entità univoco (UEN) per Singapore. |

Tipi di dati in Slovacchia

| Tipo di dati | Description |
| --- | --- |
| SLOVAKIA_DRIVING_LICENSE | Il numero della patente di guida (specifico per la Slovacchia). |
| SLOVAKIA_NATIONAL_IDENTIFICATION_NUMBER | Il numero identificativo nazionale (specifico per la Slovacchia). |
| SLOVAKIA_PASSPORT_NUMBER | Il numero del passaporto (specifico per la Slovacchia). |
| SLOVAKIA_VALUE_ADDED_TAX | L'imposta sul valore aggiunto (specifica per la Slovacchia). |

Tipi di dati in Slovenia

| Tipo di dati | Description |
| --- | --- |
| SLOVENIA_DRIVING_LICENSE | Il numero della patente di guida (specifico per la Slovenia). |
| SLOVENIA_PASSPORT_NUMBER | Il numero del passaporto (specifico per la Slovenia). |
| SLOVENIA_TAX_IDENTIFICATION_NUMBER | Il codice di identificazione fiscale (specifico per la Slovenia). |
| SLOVENIA_UNIQUE_MASTER_CITIZEN_NUMBER | Il codice identificativo unico (JMBG) per i cittadini sloveni. |
| SLOVENIA_VALUE_ADDED_TAX | L'imposta sul valore aggiunto (specifica per la Slovenia). |

Tipi di dati in Sudafrica

| Tipo di dati | Description |
| --- | --- |
| SOUTH_AFRICA_PERSONAL_IDENTIFICATION_NUMBER | Il codice di identificazione personale (specifico per il Sud Africa). |

Tipi di dati in Spagna

| Tipo di dati | Description |
| --- | --- |
| SPAIN_BANK_ACCOUNT | Il numero di conto bancario (specifico per la Spagna). |
| SPAIN_DNI | La carta d'identità nazionale (Documento Nacional de Identidad) spagnola. |
| SPAIN_DRIVING_LICENSE | Il numero della patente di guida (specifico per la Spagna). |
| SPAIN_NIE | Il numero identificativo degli stranieri (specifico per la Spagna), noto anche come NIE. |
| SPAIN_NIF | Il codice di identificazione fiscale (specifico per la Spagna), noto anche come NIF. |
| SPAIN_PASSPORT_NUMBER | Il numero del passaporto (specifico per la Spagna). |
| SPAIN_SSN | Il numero di previdenza sociale (per i cittadini spagnoli). |
| SPAIN_VALUE_ADDED_TAX | L'imposta sul valore aggiunto (specifica per la Spagna). |

Tipi di dati nello Sri Lanka

| Tipo di dati | Description |
| --- | --- |
| SRI_LANKA_NATIONAL_IDENTIFICATION_NUMBER | Il numero identificativo nazionale (specifico per lo Sri Lanka). |

Tipi di dati in Svezia

| Tipo di dati | Description |
| --- | --- |
| SWEDEN_DRIVING_LICENSE | Il numero della patente di guida (specifico per la Svezia). |
| SWEDEN_PASSPORT_NUMBER | Il numero del passaporto (specifico per la Svezia). |
| SWEDEN_PERSONAL_IDENTIFICATION_NUMBER | Il numero identificativo nazionale (specifico per la Svezia). |
| SWEDEN_TAX_IDENTIFICATION_NUMBER | Il codice di identificazione fiscale per la Svezia (personnummer). |
| SWEDEN_VALUE_ADDED_TAX | L'imposta sul valore aggiunto (specifica per la Svezia). |

Tipi di dati in Svizzera

| Tipo di dati | Description |
| --- | --- |
| SWITZERLAND_AHV | Il numero di previdenza sociale per i cittadini svizzeri (AHV). |
| SWITZERLAND_HEALTH_INSURANCE_NUMBER | Il numero dell'assicurazione sanitaria svizzera. |
| SWITZERLAND_PASSPORT_NUMBER | Il numero del passaporto (specifico per la Svizzera). |

| Tipo di dati | Description |
| --- | --- |
| SWITZERLAND_VALUE_ADDED_TAX | L'imposta sul valore aggiunto (specifica per la Svizzera). |

Tipi di dati in Thailandia

| Tipo di dati | Description |
| --- | --- |
| THAILAND_PASSPORT_NUMBER | Il numero del passaporto (specifico per la Thailandia). |
| THAILAND_PERSONAL_IDENTIFICATION_NUMBER | Il codice di identificazione personale (specifico per la Thailandia). |

Tipi di dati in Turchia

| Tipo di dati | Description |
| --- | --- |
| TURKEY_NATIONAL_IDENTIFICATION_NUMBER | Il numero identificativo nazionale (specifico per la Turchia). |
| TURKEY_PASSPORT_NUMBER | Il numero del passaporto (specifico per la Turchia). |
| TURKEY_VALUE_ADDED_TAX | L'imposta sul valore aggiunto (specifica per la Turchia). |

Tipi di dati in Ucraina

| Tipo di dati | Description |
| --- | --- |
| UKRAINE_INDIVIDUAL_IDENTIFICATION_NUMBER | Il codice identificativo univoco (specifico per l'Ucraina). |

| Tipo di dati | Description |
| --- | --- |
| UKRAINE_PASSPORT_NUMBER_DOMESTIC | Il numero del passaporto nazionale (specifico per l'Ucraina). |
| UKRAINE_PASSPORT_NUMBER_INTERNATIONAL | Il numero del passaporto internazionale (specifico per l'Ucraina). |

Tipi di dati negli Emirati Arabi Uniti (EAU)

| Tipo di dati | Description |
| --- | --- |
| UNITED_ARAB_EMIRATES_PERSONAL_NUMBER | Il codice di identificazione personale (specifico per gli Emirati Arabi Uniti). |

Tipi di dati nel Regno Unito

| Tipo di dati | Description |
| --- | --- |
| UK_BANK_ACCOUNT | Il numero di conto bancario del Regno Unito (UK). |
| UK_BANK_SORT_CODE | Il codice di ordinamento bancario del Regno Unito (UK). I codici di ordinamento sono codici bancari utilizzati per indirizzare i trasferimenti di denaro tra le banche nei rispettivi paesi tramite le rispettive organizzazioni di liquidazione. |
| UK_DRIVING_LICENSE | Il numero della patente di guida per il Regno Unito di Gran Bretagna e Irlanda del Nord (specifico per il Regno Unito) |
| UK_ELECTORAL_ROLL_NUMBER | Il numero di iscrizione alle liste elettorali (ERN) è il numero identificativo rilasciato a una persona per la registrazione delle elezioni nel Regno Unito. Il formato di questo numero |

| Tipo di dati | Description |
| --- | --- |
| | è specificato dagli standard governativi del Regno Unito del Gabinetto del Regno Unito. |
| UK_NATIONAL_HEALTH_SERVICE_NUMBER | Il numero del National Health Service (NHS) è il numero unico assegnato a un utente registrato di servizi sanitari pubblici nel Regno Unito. |
| UK_NATIONAL_INSURANCE_NUMBER | Il numero di previdenza nazionale (NINO) è un numero utilizzato nel Regno Unito (Regno Unito) per identificare una persona per il programma assicurativo nazionale o il sistema di sicurezza sociale. Talvolta è denominata NO NI o NINO. |
| UK_PASSPORT_NUMBER | Il numero di passaporto del Regno Unito (UK). |
| UK_UNIQUE_TAXPAYER_REFERENC E_NUMBER | Il numero di riferimento unico del contribue nte (UTR) del Regno Unito (UK). Un identific atore utilizzato dal governo del Regno Unito per gestire il sistema fiscale. |
| UK_VALUE_ADDED_TAX | L'IVA è un'imposta sui consumi a carico del consumatore finale. L'IVA viene pagata per ogni transazione nel processo di produzione e distribuzione. Per il Regno Unito, il numero di partita IVA è rilasciato dall'ufficio IVA della regione in cui è stabilita l'azienda. |
| UK_PHONE_NUMBER | Il numero di telefono del Regno Unito (UK). |

Tipi di dati in Venezuela

| Tipo di dati | Description |
| --- | --- |
| VENEZUELA_DRIVING_LICENSE | Il numero della patente di guida (specifico per il Venezuela). |

| Tipo di dati | Description |
| --- | --- |
| VENEZUELA_NATIONAL_IDENTIFICATION_NUMBER | Il numero identificativo nazionale (specifico per il Venezuela). |
| VENEZUELA_VALUE_ADDED_TAX | L'imposta sul valore aggiunto (specifica per il Venezuela). |

Utilizzo del rilevamento dei dati sensibili granulari

> **Note**
>
> Le azioni granulari sono disponibili solo in AWS Glue 3.0 e 4.0. Ciò include l'esperienza AWS Glue Studio. Inoltre, le modifiche persistenti del log di audit non sono disponibili nella versione 2.0.
> Per tutti i lavori visivi AWS Glue Studio 3.0 e 4.0 verrà creato uno script che utilizza automaticamente azioni granulari. APIs

La trasformazione Detect Sensitive Data offre la possibilità di rilevare, mascherare o rimuovere le entità definite o predefinite da AWS Glue. Le azioni granulari consentono inoltre di applicare un'azione specifica per entità. I vantaggi aggiuntivi includono:

- Prestazioni migliorate in quanto le azioni vengono applicate non appena vengono rilevati i dati.
- Possibilità di includere o escludere colonne specifiche.
- La possibilità di utilizzare il mascheramento parziale. Questo consente di mascherare parzialmente le entità di dati sensibili rilevate, anziché mascherare l'intera stringa. Sono supportati sia i parametri semplici con offset che regex.

Di seguito sono riportati frammenti di codice relativi al rilevamento di dati sensibili APIs e alle azioni dettagliate utilizzate nei job di esempio a cui si fa riferimento nella sezione successiva.

Rilevare l'API: le azioni granulari utilizzano il nuovo parametro `detectionParameters`:

```
def detect(
    frame: DynamicFrame,
    detectionParameters: JsonOptions,
```

```
    outputColumnName: String = "DetectedEntities",
    detectionSensitivity: String = "LOW"
): DynamicFrame = {}
```

Utilizzo del rilevamento di dati sensibili con azioni granulari APIs

Il rilevamento dei dati sensibili APIs tramite detect analizza i dati forniti, determina se le righe o le colonne sono tipi di entità di dati sensibili ed eseguirà le azioni specificate dall'utente per ogni tipo di entità.

Utilizzo di API di rilevamento con azioni granulari

Utilizza l'API rilevamento e specifica e `outputColumnName` e `detectionParameters`.

```
   object GlueApp {
     def main(sysArgs: Array[String]) {

       val spark: SparkContext = new SparkContext()
       val glueContext: GlueContext = new GlueContext(spark)

       // @params: [JOB_NAME]
       val args = GlueArgParser.getResolvedOptions(sysArgs, Seq("JOB_NAME").toArray)
       Job.init(args("JOB_NAME"), glueContext, args.asJava)

       // Script generated for node S3 bucket. Creates DataFrame from data stored in
S3.
       val S3bucket_node1 =
glueContext.getSourceWithFormat(formatOptions=JsonOptions("""{"quoteChar":
"\"", "withHeader": true, "separator": ",", "optimizePerformance": false}"""),
connectionType="s3", format="csv", options=JsonOptions("""{"paths":
["s3://189657479688-ddevansh-pii-test-bucket/tiny_pii.csv"], "recurse": true}"""),
transformationContext="S3bucket_node1").getDynamicFrame()

       // Script generated for node Detect Sensitive Data. Will run detect API for the
DataFrame
       // detectionParameter contains information on which EntityType are being
detected
       // and what actions are being applied to them when detected.
       val DetectSensitiveData_node2 = EntityDetector.detect(
           frame = S3bucket_node1,
           detectionParameters = JsonOptions(
```

```
            """
               {
                   "PHONE_NUMBER": [
                       {
                           "action": "PARTIAL_REDACT",
                           "actionOptions": {
                               "numLeftCharsToExclude": "3",
                               "numRightCharsToExclude": "4",
                               "redactChar": "#"
                           },
                           "sourceColumnsToExclude": [ "Passport No", "DL NO#" ]
                       }
                   ],
                   "USA_PASSPORT_NUMBER": [
                       {
                           "action": "SHA256_HASH",
                           "sourceColumns": [ "Passport No" ]
                       }
                   ],
                   "USA_DRIVING_LICENSE": [
                       {
                           "action": "REDACT",
                           "actionOptions": {
                               "redactText": "USA_DL"
                           },
                           "sourceColumns": [ "DL NO#" ]
                       }
                   ]

               }
           """
           ),
           outputColumnName = "DetectedEntities"
       )

       // Script generated for node S3 bucket. Store Results of detect to S3 location
       val S3bucket_node3 = glueContext.getSinkWithFormat(connectionType="s3",
options=JsonOptions("""{"path": "s3://amzn-s3-demo-bucket/test-output/",
"partitionKeys": []}"""), transformationContext="S3bucket_node3",
format="json").writeDynamicFrame(DetectSensitiveData_node2)

       Job.commit()
     }
```

Lo script precedente creerà un DataFrame file da una posizione in Amazon S3 e quindi eseguirà l'`detect`API. Poiché l'`detect`API richiede che il campo `detectionParameters` (una mappa del nome dell'entità su un elenco di tutte le impostazioni di azione da utilizzare per quell'entità) sia rappresentato dall'`JsonOptions`oggetto di AWS Glue, ci consentirà anche di estendere la funzionalità dell'API.

Per ogni azione specificata per entità, inserisci un elenco di tutti i nomi di colonna a cui applicare la entity/action combinazione. Questo consente di personalizzare le entità da rilevare per ogni colonna del set di dati e ignorare le entità che non si trovano in una colonna specifica. Questo consente inoltre di aumentare le prestazioni dei processi, evitando di eseguire chiamate di rilevamento non necessarie a tali entità, e consente di eseguire azioni uniche per ogni combinazione di colonne ed entità.

Esaminando più da vicino `detectionParameters`, ci sono tre tipi di entità nel processo di esempio. Questi sono `Phone Number`, `USA_PASSPORT_NUMBER` e `USA_DRIVING_LICENSE`. Per ognuno di questi tipi di entità AWS Glue eseguirà diverse azioni che sono `PARTIAL_REDACTSHA256_HASH`,`REDACT`, e`DETECT`. and/or `sourceColumnsToExclude`Se rilevato, deve essere applicato anche `sourceColumns` a ciascuno dei tipi di entità.

> **Note**
>
> È possibile utilizzare una sola edit-in-place azione (`PARTIAL_REDACTSHA256_HASH`, o`REDACT`) per colonna, ma l'`DETECT`azione può essere utilizzata con una qualsiasi di queste azioni.

Il campo `detectionParameters` ha il seguente layout:

```
ENTITY_NAME -> List[Actions]
{
 "ENTITY_NAME": [{
  Action, // required
  ColumnSpecs,
  ActionOptionsMap
    }],
    "ENTITY_NAME2": [{
  ...
    }]
```

```
    }
```

I tipi di `actions` e `actionOptions` sono elencati di seguito:

```
DETECT
{
    # Required
    "action": "DETECT",
    # Optional, depending on action chosen
    "actionOptions": {
        // There are no actionOptions for DETECT
    },
    # 1 of below required, both can also used
    "sourceColumns": [
        "COL_1", "COL_2", ..., "COL_N"
    ],
    "sourceColumnsToExclude": [
        "COL_5"
    ]
}

SHA256_HASH
{
    # Required
    "action": "SHA256_HASH",
    # Required or optional, depending on action chosen
    "actionOptions": {
        // There are no actionOptions for SHA256_HASH
    },

    # 1 of below required, both can also used
    "sourceColumns": [
        "COL_1", "COL_2", ..., "COL_N"
    ],
    "sourceColumnsToExclude": [
        "COL_5"
    ]
}

REDACT
{
    # Required
```

```
    "action": "REDACT",
    # Required or optional, depending on action chosen
    "actionOptions": {
        // The text that is being replaced
        "redactText": "USA_DL"
    },

    # 1 of below required, both can also used
    "sourceColumns": [
        "COL_1", "COL_2", ..., "COL_N"
    ],
    "sourceColumnsToExclude": [
        "COL_5"
    ]
}

PARTIAL_REDACT
{
    # Required
    "action": "PARTIAL_REDACT",
    # Required or optional, depending on action chosen
    "actionOptions": {
        // number of characters to not redact from the left side
        "numLeftCharsToExclude": "3",
        // number of characters to not redact from the right side
        "numRightCharsToExclude": "4",
        // the partial redact will be made with this redacted character
        "redactChar": "#",
        // regex pattern for partial redaction
        "matchPattern": "[0-9]"
    },

    # 1 of below required, both can also used
    "sourceColumns": [
        "COL_1", "COL_2", ..., "COL_N"
    ],
    "sourceColumnsToExclude": [
        "COL_5"
    ]
}
```

Una volta eseguito lo script, i risultati vengono inviati alla posizione Amazon S3 specificata. Puoi visualizzare i dati in Amazon S3 ma con i tipi di entità selezionati che vengono sensibilizzati in base all'azione selezionata. Nel caso, avremmo una riga simile a questa:


```
{
    "Name": "Colby Schuster",
    "Address": "39041 Antonietta Vista, South Rodgerside, Nebraska 24151",
    "Car Owned": "Fiat",
    "Email": "Kitty46@gmail.com",
    "Company": "O'Reilly Group",
    "Job Title": "Dynamic Functionality Facilitator",
    "ITIN": "991-22-2906",
    "Username": "Cassandre.Kub43",
    "SSN": "914-22-2906",
    "DOB": "2020-08-27",
    "Phone Number": "1-2#######1718",
    "Bank Account No": "69741187",
    "Credit Card Number": "6441-6289-6867-2162-2711",
    "Passport No": "94f311e93a623c72ccb6fc46cf5f5b0265ccb42c517498a0f27fd4c43b47111e",
    "DL NO#": "USA_DL"
}
```


Nello script precedente, `Phone Number` è stato parzialmente redatto con #. `Passport No`È stato modificato in un SHA256 hash. `DL NO#` è stato rilevato come numero di patente di guida USA ed è stato redatto in "USA_DL" proprio come indicato in `detectionParameters`.

> **Note**
>
> L'API classifyColumns non è disponibile per l'uso con azioni dettagliate a causa della natura dell'API. L'API esegue il campionamento delle colonne (regolabile dall'utente ma con valori predefiniti) per eseguire il rilevamento più rapidamente. Per questo motivo, le azioni dettagliate richiedono l'iterazione su ogni valore.

### Log di audit persistente

Una nuova funzionalità introdotta con azioni granulari (ma disponibile anche quando si utilizza la versione normale APIs) è la presenza di un registro di controllo persistente. Attualmente, l'esecuzione dell'API di rilevamento aggiunge una colonna aggiuntiva (impostazione predefinita

su `DetectedEntities` ma personalizzabile tramite `outputColumnName`) con i metadati di rilevamento PII. Questo ora ha una chiave di metadati "actionUsed", che è una tra `DETECT`, `PARTIAL_REDACT`, `SHA256_HASH` e `REDACT`.

```
"DetectedEntities": {
    "Credit Card Number": [
        {
            "entityType": "CREDIT_CARD",
            "actionUsed": "DETECT",
            "start": 0,
            "end": 19
        }
    ],
    "Phone Number": [
        {
            "entityType": "PHONE_NUMBER",
            "actionUsed": "REDACT",
            "start": 0,
            "end": 14
        }
    ]
}
```

Anche i clienti che utilizzano azioni APIs non granulari, ad esempio, `detect(entityTypesToDetect, outputColumnName)` vedranno questo registro di controllo persistente nel dataframe risultante.

I clienti che utilizzano APIs azioni granulari vedranno tutte le azioni, indipendentemente dal fatto che siano state oscurate o meno. Esempio:

```
+--------------------+---------------
+-------------------------------------------------------------------------------------------------
+
| Credit Card Number  |  Phone Number  |
                                              DetectedEntities
                                                                  |
+--------------------+---------------
+-------------------------------------------------------------------------------------------------
+
| 622126741306XXXX    | +12#####7890   | {"Credit Card Number":
[{"entityType":"CREDIT_CARD","actionUsed":"PARTIAL_REDACT","start":0,"end":16}],"Phone
```

```
 Number":
[{"entityType":"PHONE_NUMBER","actionUsed":"PARTIAL_REDACT","start":0,"end":12}]}} |
| 6221 2674 1306 XXXX | +12#######7890 | {"Credit Card Number":
[{"entityType":"CREDIT_CARD","actionUsed":"PARTIAL_REDACT","start":0,"end":19}],"Phone
 Number":
[{"entityType":"PHONE_NUMBER","actionUsed":"PARTIAL_REDACT","start":0,"end":14}]}} |
| 6221-2674-1306-XXXX | 22#######7890  | {"Credit Card Number":
[{"entityType":"CREDIT_CARD","actionUsed":"PARTIAL_REDACT","start":0,"end":19}],"Phone
 Number":
[{"entityType":"PHONE_NUMBER","actionUsed":"PARTIAL_REDACT","start":0,"end":14}]}} |
+--------------------+---------------
+-------------------------------------------------------------------------------------------------------
+
```

Se non desideri visualizzare la DetectedEntitiescolonna, puoi semplicemente trascinare la colonna aggiuntiva in uno script personalizzato.

## API Visual Job di AWS Glue

AWS Glue fornisce un'API che permette ai clienti di creare processi di integrazione dei dati utilizzando l'API AWS Glue di un oggetto JSON che rappresenta un DAG. I clienti possono quindi utilizzare l'editor visivo in AWS Glue Studio per lavorare con questi processi.

Per ulteriori informazioni sui tipi di dati di Visual Job API, consulta API Visual Job.

Argomenti

- Progettazione di API e CRUD APIs
- Nozioni di base
- Limitazioni Visual job

### Progettazione di API e CRUD APIs

La CreateJob e UpdateJob APIs ora supporta un parametro opzionale aggiuntivo, codeGenConfiguration Nodes. Fornire una struttura json non vuota per questo campo comporterà la registrazione del gruppo di disponibilità del database in AWS Glue Studio per il processo creato e per il codice associato generato. Un valore nullo o una stringa vuota per questo campo durante la creazione di processi verrà ignorato.

Gli aggiornamenti al campo codeGenConfiguration Nodes verranno eseguiti tramite l' UpdateJob AWS GlueAPI in modo simile a CreateJob. L'intero campo deve essere specificato nel UpdateJob punto in cui il DAG è stato modificato come desiderato. Un valore nullo fornito verrà ignorato e non verrà eseguito alcun aggiornamento del DAG. Una struttura o una stringa vuota farà sì che i codeGenConfiguration nodi vengano impostati come vuoti e qualsiasi DAG precedente venga rimosso. L' GetJob API restituirà un DAG, se ne esiste uno. L' DeleteJob API eliminerà anche qualsiasi DAG associato.

Nozioni di base

Per creare un lavoro, usa l' CreateJob azione. L'input della `CreateJob` richiesta avrà un campo aggiuntivo 'codeGenConfigurationNodes' in cui è possibile specificare l'oggetto DAG in JSON.

Cose da tenere a mente:

- Il campo 'codeGenConfigurationNodes' è una mappa da nodeID a nodo.
- Ciascun nodo inizia con una chiave che ne identifica il tipo.
- È possibile specificare una sola chiave, poiché un nodo può essere di un solo tipo.
- Il campo di input contiene i nodi padre del nodo corrente.

Di seguito è riportata una rappresentazione JSON di un input. CreateJob

```
{
  "node-1": {
    "S3CatalogSource": {
      "Table": "csvFormattedTable",
      "PartitionPredicate": "",
      "Name": "S3 bucket",
      "AdditionalOptions": {},
      "Database": "myDatabase"
    }
  },
  "node-3": {
    "S3DirectTarget": {
      "Inputs": ["node-2"],
      "PartitionKeys": [],
      "Compression": "none",
      "Format": "json",
      "SchemaChangePolicy": { "EnableUpdateCatalog": false },
      "Path": "",
```

```
      "Name": "S3 bucket"
    }
  },
  "node-2": {
    "ApplyMapping": {
      "Inputs": ["node-1"],
      "Name": "ApplyMapping",
      "Mapping": [
        {
          "FromType": "long",
          "ToType": "long",
          "Dropped": false,
          "ToKey": "myheader1",
          "FromPath": ["myheader1"]
        },
        {
          "FromType": "long",
          "ToType": "long",
          "Dropped": false,
          "ToKey": "myheader2",
          "FromPath": ["myheader2"]
        },
        {
          "FromType": "long",
          "ToType": "long",
          "Dropped": false,
          "ToKey": "myheader3",
          "FromPath": ["myheader3"]
        }
      ]
    }
  }
}
```

Aggiornamento e acquisizione di processi

Poiché UpdateJobavrà anche un campo 'codeGenConfigurationNodes', il formato di input sarà lo stesso. Vedi UpdateJobAzione.

L'GetJobazione restituirà anche un campo codeGenConfiguration 'Nodes' nello stesso formato. Vedi GetJobAzione.

Limitazioni Visual job

Poiché il parametro 'codeGenConfigurationNodes' è stato aggiunto a quello esistente APIs, eventuali limitazioni APIs verranno ereditate. Inoltre, i codeGenConfiguration nodi e alcuni nodi avranno dimensioni limitate. Per ulteriori informazioni, consulta Struttura processo.

# Script di programmazione Ray

> **Important**
>
> AWS Glue for Ray non è più aperto a nuovi clienti. I clienti esistenti possono continuare a utilizzare il servizio normalmente. Per ulteriori informazioni, consulta AWS Glue Ray End of Support.

AWS Glue semplifica la scrittura e l'esecuzione di script Ray. Questa sezione descrive le funzionalità Ray supportate disponibili in AWS Glue for Ray. Gli script Ray vengono programmati in Python.

Lo script personalizzato deve essere compatibile con la versione di Ray definita dal campo `Runtime` nella definizione del processo. Per ulteriori informazioni sui `Runtime` nell'API Processi, consulta la pagina the section called "Jobs". Per ulteriori informazioni su ciascun ambiente di runtime, consulta la pagina the section called "Ambienti di runtime Ray supportati".

Argomenti

- Tutorial: scrivere uno script ETL AWS Glue per Ray
- Utilizzo di Ray Core e Ray Data in AWS Glue for Ray
- Fornitura di file e librerie Python ai processi Ray
- Connessione ai dati nei processi Ray

## Tutorial: scrivere uno script ETL AWS Glue per Ray

> **Important**
>
> AWS Glue for Ray non è più aperto a nuovi clienti. I clienti esistenti possono continuare a utilizzare il servizio normalmente. Per ulteriori informazioni, consulta AWS Glue Ray End of Support.

Ray ti dà la possibilità di scrivere e scalare attività distribuite in modo nativo in Python. AWS Glue for Ray offre ambienti Ray senza server a cui è possibile accedere sia dai job che dalle sessioni interattive (le sessioni interattive di Ray sono disponibili in anteprima). Il AWS Glue job system offre un modo coerente per gestire ed eseguire le attività, in base a una pianificazione, da un trigger o dalla console. AWS Glue

La combinazione di questi AWS Glue strumenti crea una potente toolchain che puoi usare per i carichi di lavoro di estrazione, trasformazione e caricamento (ETL), un caso d'uso comune per. AWS Glue Questo tutorial ti illustrerà le basi per creare questa soluzione.

Supportiamo anche l'utilizzo di Spark AWS Glue per i tuoi carichi di lavoro ETL. Per un tutorial sulla scrittura di uno script AWS Glue per Spark, consulta. the section called "Tutorial: scrittura di uno script Spark" Per ulteriori informazioni sui motori disponibili, consulta la pagina the section called "AWS Glue per Spark e AWS Glue per Ray". Ray è in grado di affrontare vari tipi di attività nell'ambito dell'analisi, del machine learning (ML) e dello sviluppo di applicazioni.

In questo tutorial, estrarrai, trasformerai e caricherai un set di dati CSV ospitato in Amazon Simple Storage Service (Amazon S3). Inizierai con il set di dati dei dati di record di viaggio della New York City Taxi and Limousine Commission (TLC), archiviato in un bucket Amazon S3 pubblico. Per ulteriori informazioni su questo set di dati, consulta il Registry of Open Data su AWS.

Trasformerai i tuoi dati con le trasformazioni predefinite disponibili nella libreria Ray Data. Ray Data è una libreria per la preparazione di set di dati progettata da Ray e inclusa di default negli ambienti AWS Glue Ray. Per ulteriori informazioni sulle librerie incluse in modo predefinito, consulta la pagina the section called "Moduli disponibili con i processi Ray". Potrai quindi scrivere i dati trasformati in un bucket Amazon S3 da te controllato.

Prerequisiti: per questo tutorial, è necessario un AWS account con accesso ad AWS Glue Amazon S3.

## Passaggio 1: creazione di un bucket in Amazon S3 per contenere i dati di output

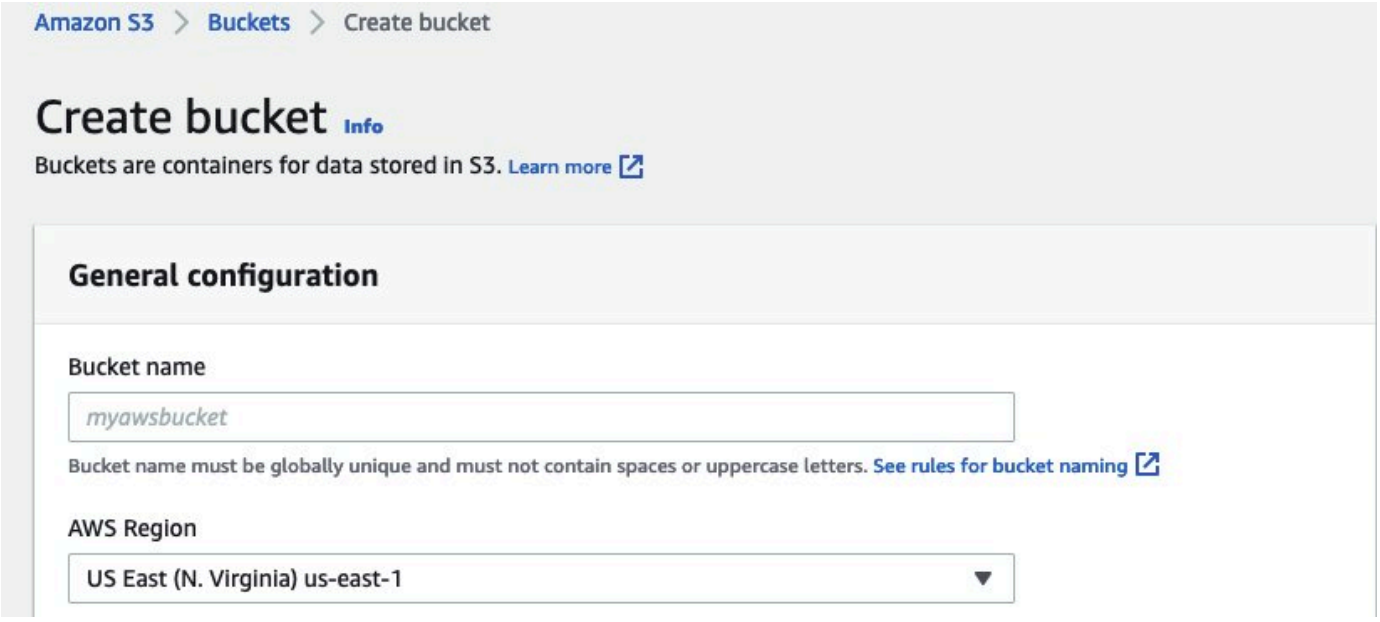
Avrai bisogno di un bucket Amazon S3 da te controllato che funga da sink per i dati creati in questo tutorial. È possibile creare questo bucket con la procedura seguente.

> **Note**
>
> Se desideri scrivere i tuoi dati in un bucket esistente sotto il tuo controllo, puoi saltare questo passaggio. Prendi nota del *yourBucketName* nome del bucket esistente, da utilizzare nei passaggi successivi.

Creazione di un bucket per l'output del processo Ray

- Crea un bucket seguendo i passaggi descritti in Creating a bucket nella Guida per l'utente di Amazon S3.
  - Quando scegli il nome del bucket, prendi nota di *yourBucketName* ciò a cui farai riferimento nei passaggi successivi.
  - Per altre configurazioni, le impostazioni suggerite fornite nella console Amazon S3 dovrebbero funzionare correttamente in questo tutorial.

  Ad esempio, la finestra di dialogo per la creazione del bucket potrebbe avere questo aspetto nella console Amazon S3.




## Passaggio 2: creazione di un ruolo e una policy IAM per il processo Ray

Il tuo lavoro richiederà un ruolo AWS Identity and Access Management (IAM) con quanto segue:

- Autorizzazioni concesse dalla policy gestita da `AWSGlueServiceRole`. Queste sono le autorizzazioni di base necessarie per eseguire un AWS Glue lavoro.
- Autorizzazioni `Read` a livello di accesso per la risorsa `nyc-tlc/*` Amazon S3.
- Autorizzazioni `Write` a livello di accesso per la risorsa *`yourBucketName`*`/*` Amazon S3.
- Una relazione di fiducia che consente al principale `glue.amazonaws.com` di assumere il ruolo.

È possibile creare questo ruolo con la procedura seguente.

Per creare un ruolo IAM per il tuo lavoro su AWS Glue for Ray

> **Note**
>
> È possibile creare un ruolo IAM seguendo molte procedure diverse. Per ulteriori informazioni o opzioni su come effettuare il provisioning delle risorse IAM, consulta la documentazione di AWS Identity and Access Management.

1. Crea una policy che definisca le autorizzazioni Amazon S3 precedentemente delineate seguendo i passaggi descritti in Creating IAM policies (console) with the visual editor nella Guida per l'utente di IAM.
   - Quando selezioni un servizio, scegli Amazon S3.
   - Quando selezioni le autorizzazioni per la tua policy, collega i seguenti set di operazioni per le seguenti risorse (menzionate in precedenza):
     - Autorizzazioni con livello di accesso di lettura per la risorsa `nyc-tlc/*` Amazon S3.
     - Autorizzazioni con livello di accesso di scrittura per la risorsa *`yourBucketName`*`/*` di Amazon S3.
   - Quando selezioni il nome della policy, prendi nota di *`YourPolicyName`* ciò a cui farai riferimento in un passaggio successivo.
2. Crea un ruolo AWS Glue per il tuo lavoro su Ray seguendo i passaggi descritti nella sezione Creazione di un ruolo per un AWS servizio (console) nella Guida per l'utente IAM.
   - Quando selezioni un'entità AWS di servizio affidabile, scegli`Glue`. Questo creerà automaticamente la relazione di attendibilità necessaria per il processo.
   - Quando selezioni le policy per la policy delle autorizzazioni, collega le seguenti policy:
     - `AWSGlueServiceRole`

- *YourPolicyName*
- Quando selezionate il nome del ruolo, prendete nota di *YourRoleName* ciò a cui farete riferimento nei passaggi successivi.

## Passaggio 3: crea ed esegui un job AWS Glue for Ray

In questo passaggio, si crea un AWS Glue lavoro utilizzando il Console di gestione AWS, si fornisce uno script di esempio e si esegue il lavoro. Quando crei un processo, nella console viene creato uno spazio in cui archiviare, configurare e modificare lo script Ray. Per informazioni su come creare i processi, consulta the section called "Accesso alla console ".

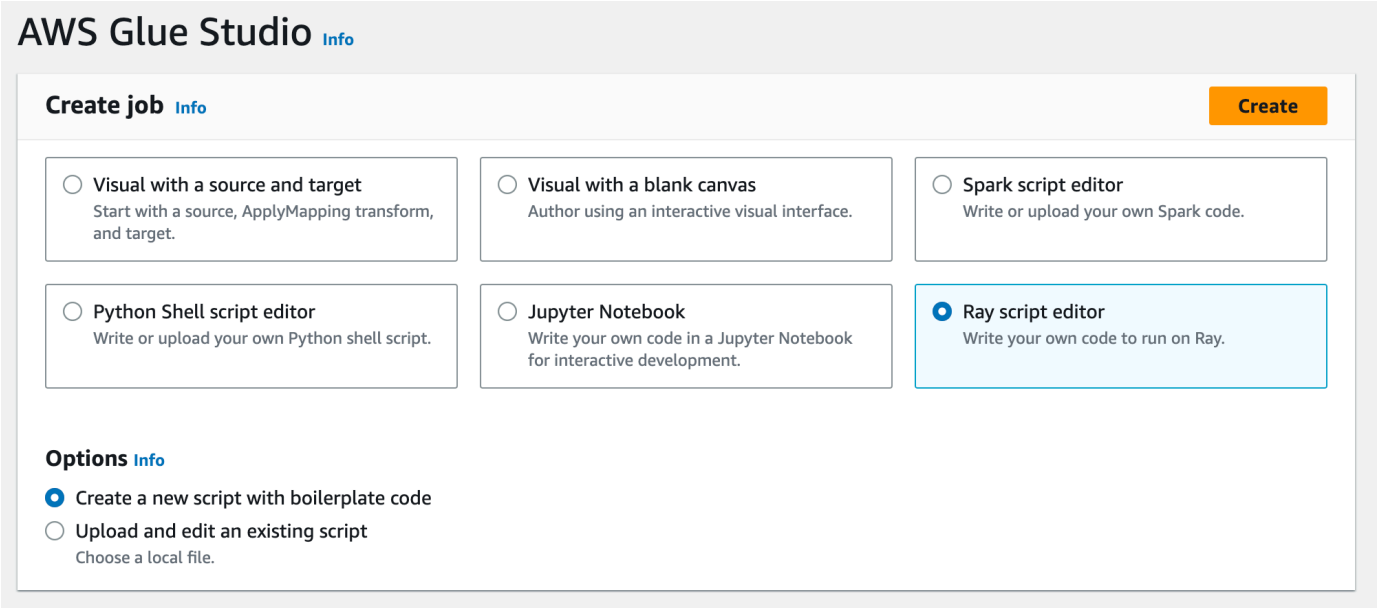
In questo tutorial, affrontiamo il seguente scenario ETL: vorresti leggere i record di gennaio 2022 dal set di dati New York City TLC Trip Record, aggiungere una nuova colonna (`tip_rate`) al set di dati combinando i dati nelle colonne esistenti, quindi rimuovere un numero di colonne che non sono rilevanti per la tua analisi attuale e quindi desideri scrivere i risultati*yourBucketName*. Il seguente script Ray esegue questi passaggi:

```
import ray
import pandas
from ray import data

ray.init('auto')

ds = ray.data.read_csv("s3://nyc-tlc/opendata_repo/opendata_webconvert/yellow/
yellow_tripdata_2022-01.csv")

# Add the given new column to the dataset and show the sample record after adding a new
 column
ds = ds.add_column( "tip_rate", lambda df: df["tip_amount"] / df["total_amount"])

# Dropping few columns from the underlying Dataset
ds = ds.drop_columns(["payment_type", "fare_amount", "extra", "tolls_amount",
 "improvement_surcharge"])

ds.write_parquet("s3://yourBucketName/ray/tutorial/output/")
```

Per creare ed eseguire un job for Ray AWS Glue

1. In Console di gestione AWS, vai alla pagina di AWS Glue destinazione.
2. Nel riquadro di navigazione laterale, scegli Processi ETL.

3. In Crea processo, scegli Ray script editor, quindi scegli Crea, come nella figura seguente.




4. Incolla il testo completo dello script nel riquadro Script e sostituisci l'eventuale testo presente.
5. Vai ai dettagli di Job e imposta la proprietà IAM Role su*YourRoleName*.
6. Seleziona Salva, quindi scegli Esegui.

## Passaggio 4: ispezione dell'output

Dopo aver eseguito il AWS Glue job, dovresti verificare che l'output corrisponda alle aspettative di questo scenario. È possibile farlo con la seguente procedura.

Verifica della corretta esecuzione del processo Ray

1. Nella pagina dei dettagli del processo, vai a Esecuzioni.
2. Dopo alcuni minuti, dovresti vedere un'esecuzione con lo Stato di esecuzione impostato su Operazione riuscita.
3. Accedi alla console Amazon S3 all'indirizzo https://console.aws.amazon.com/s3/e ispeziona. *yourBucketName* Dovresti visualizzare i file scritti nel tuo bucket di output.
4. Leggi i file Parquet e verificane il contenuto. Puoi farlo utilizzando gli strumenti esistenti. Se non disponi di un processo per la convalida dei file Parquet, puoi farlo nella AWS Glue console con una sessione AWS Glue interattiva, usando Spark o Ray (in anteprima).

   In una sessione interattiva, hai accesso alle librerie Ray Data, Spark o pandas, fornite per impostazione predefinita (in base al motore scelto). Per verificare i contenuti del tuo file, è

possibile utilizzare i metodi di ispezione comuni disponibili in tali librerie, ad esempio `count`, `schema` e `show`. Per ulteriori informazioni sulle sessioni interattive nella console, consulta Uso dei notebook con Studio e. AWS Glue AWS Glue

Poiché hai confermato che i file sono stati scritti nel bucket, puoi affermare con relativa certezza che se l'output presenta problemi, non sono correlati alla configurazione IAM. Configura la sessione con *`yourRoleName`* per avere accesso ai file pertinenti.

Se non vedi i risultati previsti, esamina i contenuti per la risoluzione dei problemi in questa guida per identificare e correggere l'origine dell'errore. Puoi trovare i contenuti relativi alla risoluzione dei problemi nel capitolo Risoluzione dei problemi relativi a AWS Glue. Per errori specifici relativi ai processi Ray, consulta la sezione the section called "Risoluzione degli errori relativi ai processi Ray" nel capitolo sulla risoluzione dei problemi.

## Fasi successive

Ora hai visto ed eseguito un processo ETL utilizzando AWS Glue for Ray dall'inizio alla fine. Puoi utilizzare le seguenti risorse per capire quali strumenti offre Ray AWS Glue per trasformare e interpretare i tuoi dati su larga scala.

- Per ulteriori informazioni sul modello di attività di Ray, consulta la pagina the section called "Utilizzo di Ray Core e Ray Data in AWS Glue for Ray". Per una maggiore esperienza nell'uso delle attività di Ray, segui gli esempi nella documentazione di Ray Core. Consulta la pagina Ray Core: Ray Tutorials and Examples (2.4.0) nella documentazione di Ray.
- Per indicazioni sulle librerie di gestione dei dati disponibili in AWS Glue for Ray, consultathe section called "Connessione ai dati". Per ulteriori esperienze con Ray Data per trasformare e scrivere set di dati, segui gli esempi nella documentazione di Ray Data. Consulta la sezione Ray Data: Examples (2.4.0).
- Per ulteriori informazioni sulla configurazione AWS Glue per i lavori Ray, vederethe section called "Utilizzo dei processi Ray".
- Per ulteriori informazioni sulla scrittura di script AWS Glue per Ray, continua a leggere la documentazione in questa sezione.

# Utilizzo di Ray Core e Ray Data in AWS Glue for Ray

> **Important**
>
> AWS Glue for Ray non è più aperto a nuovi clienti. I clienti esistenti possono continuare a utilizzare il servizio normalmente. Per ulteriori informazioni, consulta AWS Glue Ray End of Support.

Ray è un framework per dimensionare gli script Python distribuendo il processo su un cluster. Ray fornisce librerie per ottimizzare determinate attività e puoi usarlo come soluzione a molti tipi di problemi. Nel AWS Glue, ci concentriamo sull'utilizzo di Ray per trasformare set di dati di grandi dimensioni. AWS Glue offre supporto per Ray Data e parti di Ray Core per facilitare questo compito.

## Cos'è Ray Core?

Il primo passaggio per creare un'applicazione distribuita consiste nell'identificare e definire i processi che possono essere eseguiti in simultanea. Ray Core contiene le parti di Ray che si utilizzano per definire le attività che possono essere eseguite contemporaneamente. Ray fornisce informazioni di riferimento e di avvio rapido che è possibile utilizzare per apprendere gli strumenti forniti. Per ulteriori informazioni, consulta le pagine What is Ray Core? e Ray Core Quick Start. Per ulteriori informazioni sulla definizione efficace delle attività simultanee in Ray, consulta la pagina Tips for first-time users.

> **Attività e attori di Ray**
>
> Nella documentazione AWS Glue di Ray, potremmo fare riferimento a compiti e attori, che sono concetti fondamentali di Ray.
> Ray utilizza le funzioni e le classi Python come elementi costitutivi di un sistema di calcolo distribuito. Analogamente a quanto accade con le funzioni e le variabili di Python, che diventano "metodi" e "attributi" quando vengono utilizzate in una classe, le funzioni diventano "attività" e le classi diventano "attori" quando vengono utilizzate in Ray per inviare codice ai worker. È possibile identificare le funzioni e le classi che potrebbero essere utilizzate da Ray tramite l'annotazione `@ray.remote`.
> Le attività e gli attori sono configurabili, hanno un ciclo di vita e occupano risorse di elaborazione per tutto il ciclo di vita. Il codice che genera errori può essere ricondotto a un'attività o a un attore quando si individua la causa principale dei problemi. Pertanto, questi termini potrebbero apparire quando impari a configurare, monitorare o eseguire il debug AWS Glue per i lavori Ray.

Per imparare a utilizzare in modo efficace le attività e gli attori per creare un'applicazione distribuita, consulta la pagina Key Concepts nella documentazione di Ray.

## Ray Core sostituisce AWS Glue Ray

AWS Glue gli ambienti for Ray gestiscono la formazione e la scalabilità dei cluster, nonché la raccolta e la visualizzazione dei log. Poiché gestiamo questi problemi, limitiamo di conseguenza l'accesso e il supporto a quelli APIs in Ray Core che verrebbero utilizzati per risolvere questi problemi in un cluster open source.

Nell'ambiente di runtime gestito `Ray2.4`, non supportiamo:

- CLI Ray Core
- CLI Ray State
- Metodi di utilizzo del parametro `ray.util.metrics` di Prometheus:
  - Contatore
  - Gauge
  - Istogramma
- Altri strumenti di debug:
  - ray.util.pdb.set_trace
  - ray.util.inspect_serializability
  - ray.timeline

## Cos'è Ray Data?

Quando ti connetti a origini e destinazioni dati, gestisci set di dati e avvii trasformazioni comuni, Ray Data è una metodologia semplice per utilizzare Ray per risolvere i problemi di trasformazione dei set di dati Ray. Per ulteriori informazioni sull'utilizzo di Ray Data, consulta la pagina Ray Datasets: Distributed Data Preprocessing.

Puoi utilizzare Ray Data o altri strumenti per accedere ai tuoi dati. Per ulteriori informazioni sull'accesso ai dati in Ray, consulta la pagina the section called “Connessione ai dati”.

### Ray Data sostituisce AWS Glue Ray

Ray Data è supportato e fornito per impostazione predefinita nell'ambiente di runtime gestito `Ray2.4`. Per ulteriori informazioni sui moduli disponibili, consulta the section called "Moduli disponibili con i processi Ray".

## Fornitura di file e librerie Python ai processi Ray

> **Important**
>
> AWS Glue for Ray non è più aperto a nuovi clienti. I clienti esistenti possono continuare a utilizzare il servizio normalmente. Per ulteriori informazioni, consulta AWS Glue Ray End of Support.

Questa sezione fornisce le informazioni necessarie per utilizzare le librerie Python con i lavori AWS Glue Ray. In tutti i processi Ray è possibile utilizzare alcune librerie comuni incluse per impostazione predefinita. Puoi anche fornire le tue librerie Python al tuo processo Ray.

### Moduli disponibili con i processi Ray

È possibile eseguire flussi di lavoro di integrazione dei dati in un processo Ray con i seguenti pacchetti. Per impostazione predefinita, questi pacchetti sono disponibili nei processi Ray.

AWS Glue version 4.0

Nella AWS Glue versione 4.0, l'ambiente Ray (`Ray2.4`runtime) fornisce i seguenti pacchetti:

- boto3 == 1.26.133
- ray == 2.4.0
- pyarrow == 11.0.0
- pandas == 1.5.3
- numpy == 1.24.3
- fsspec == 2023.4.0

Questo elenco include tutti i pacchetti che verrebbero installati con `ray[data] == 2.4.0`. Ray Data è supportato immediatamente.

## Fornitura di file al processo Ray

Puoi fornire file al tuo processo Ray con il parametro `--working-dir`. Fornisci a questo parametro un percorso per un file .zip ospitato su Amazon S3. All'interno del file .zip, i file devono essere contenuti in un'unica directory di primo livello. Nessun altro file deve trovarsi al livello superiore.

I file verranno distribuiti su ogni nodo Ray prima dell'inizio dell'esecuzione dello script. Considera come ciò potrebbe influire sullo spazio su disco disponibile per ogni nodo Ray. Lo spazio disponibile su disco è determinato dal WorkerType set nella configurazione del processo. Se desideri fornire i tuoi dati del processo su larga scala, questo meccanismo non è la soluzione giusta. Per ulteriori informazioni sulla fornitura di dati al processo, consulta la pagina the section called "Connessione ai dati".

I tuoi file saranno accessibili come se la directory fosse stata fornita a Ray tramite il parametro `working_dir`. Ad esempio, per leggere un file denominato `sample.txt` nella directory di primo livello del file .zip, puoi chiamare:

```
@ray.remote
def do_work():
    f = open("sample.txt", "r")
    print(f.read())
```

Per ulteriori informazioni su `working_dir`, consulta la documentazione di Ray. Questa funzionalità si comporta in modo simile alle funzionalità native di Ray.

## Moduli Python aggiuntivi per i processi Ray

### Moduli aggiuntivi di PyPI

I processi Ray utilizzano Python Package Installer (pip3) per installare moduli aggiuntivi da utilizzare con uno script Ray. Puoi utilizzare il parametro `--pip-install` con un elenco di moduli Python separati da virgole per aggiungere un nuovo modulo o modificare la versione di un modulo esistente.

Ad esempio, per aggiornare o aggiungere un nuovo modulo `scikit-learn`, utilizza la seguente coppia chiave-valore:

```
"--pip-install", "scikit-learn==0.21.3"
```

Se disponi di moduli o patch personalizzati, puoi distribuire le tue librerie da Amazon S3 con il parametro `--s3-py-modules`. Prima di caricare la tua distribuzione, potrebbe essere necessario

riconfezionarla e ricompilarla. Segui le linee guida riportate nella sezione the section called "Inclusione del codice Python nei processi Ray".

Distribuzioni personalizzate da Amazon S3

Le distribuzioni personalizzate devono rispettare le linee guida di Ray sulla creazione di pacchetti per le dipendenze. Le istruzioni per creare queste distribuzioni sono riportate nella sezione successiva. Per ulteriori informazioni su come Ray imposta le dipendenze, consulta Dipendenze dell'ambiente nella documentazione di Ray.

Per includere un distribuibile personalizzato dopo averne valutato il contenuto, caricalo in un bucket accessibile dal ruolo IAM del processo. Nella configurazione dei parametri, specifica il percorso Amazon S3 a un archivio zip Python. Se fornisci più distribuibili, separali con una virgola. Esempio:

```
"--s3-py-modules", "s3://s3bucket/pythonPackage.zip"
```

Limitazioni

I processi Ray non supportano la compilazione di codice nativo nell'ambiente del processo. Questo può essere un limite se le tue dipendenze da Python dipendono in modo transitorio dal codice nativo compilato. I Ray job possono eseguire i file binari forniti, ma devono essere compilati per Linux su ARM64 Linux. Ciò significa che potresti essere in grado di utilizzare il contenuto dei wheel `aarch64manylinux`. Puoi fornire le tue dipendenze native in un formato compilato reimpacchettando una ruota secondo gli standard Ray. In genere, ciò significa rimuovere le cartelle `dist-info` in modo che vi sia una sola cartella alla radice dell'archivio.

Non è possibile aggiornare la versione di `ray` o `ray[data]` utilizzando questo parametro. Per utilizzare una nuova versione di Ray, dovrai modificare il campo di runtime del tuo processo dopo che avremo rilasciato il supporto. Per ulteriori informazioni sulle versioni supportate di Ray, consulta la pagina the section called "AWS Glue versioni".

## Inclusione del codice Python nei processi Ray

La Python Software Foundation offre comportamenti standardizzati per la creazione di pacchetti di file Python da utilizzare in diversi runtime. Ray introduce delle limitazioni agli standard di confezionamento di cui dovresti essere a conoscenza. AWS Glue non specifica standard di confezionamento oltre a quelli specificati da Ray. Le seguenti istruzioni forniscono una guida standard sulla creazione di pacchetti Python semplici.

Crea un pacchetto dei file in un archivio `.zip`. Alla root dell'archivio dovrebbe essere presente una directory. Non dovrebbero esserci altri file al livello root dell'archivio, altrimenti si verificherà un

comportamento imprevisto. La directory root è il pacchetto e il suo nome viene utilizzato per fare riferimento al codice Python durante l'importazione.

Se fornisci una distribuzione in questo formato a un processo Ray con `--s3-py-modules`, sarai in grado di importare il codice Python dal tuo pacchetto nello script Ray.

Il tuo pacchetto può fornire un singolo modulo Python con alcuni file Python, oppure puoi confezionare insieme diversi moduli. Quando riconfezioni le dipendenze, come le librerie da PyPI, controlla i file nascosti e le directory di metadati all'interno di quei pacchetti.

> **Warning**
>
> Alcuni comportamenti del sistema operativo rendono difficile seguire correttamente queste istruzioni di confezionamento.
>
> - OSX può aggiungere file nascosti, ad esempio `__MACOSX`, al file zip di livello superiore.
> - Windows può aggiungere automaticamente i file a una cartella all'interno del file zip, creando involontariamente una cartella annidata.

Le seguenti procedure presuppongono che tu stia interagendo con i tuoi file in Amazon Linux 2 o in un sistema operativo simile che fornisce una distribuzione delle utility `zip` e `zipinfo` di Info-ZIP. Ti consigliamo di utilizzare questi strumenti per prevenire comportamenti imprevisti.

Confezionamento di file Python da utilizzare in Ray

1. Crea una directory temporanea con il nome del tuo pacchetto, quindi verifica che la directory di lavoro sia quella padre. A questo scopo, puoi eseguire il comando seguente:

   ```
   cd parent_directory
   mkdir temp_dir
   ```

2. Copia i file nella directory temporanea, quindi verifica la struttura della directory. Il contenuto di questa directory sarà accessibile direttamente come modulo Python. A questo scopo, puoi eseguire il comando seguente:

   ```
   ls -AR temp_dir
   # my_file_1.py
   # my_file_2.py
   ```

3. Comprimi la cartella temporanea utilizzando zip. A questo scopo, puoi eseguire il comando seguente:

```
zip -r zip_file.zip temp_dir
```

4. Verifica che il file sia correttamente confezionato. Ora il *zip_file*.zip dovrebbe essere disponibile nella tua directory di lavoro. È possibile verificarla con il seguente comando:

```
zipinfo -1 zip_file.zip
# temp_dir/
# temp_dir/my_file_1.py
# temp_dir/my_file_2.py
```

Riconfezionamento di un pacchetto Python da utilizzare in Ray.

1. Crea una directory temporanea con il nome del tuo pacchetto, quindi verifica che la directory di lavoro sia quella padre. A questo scopo, puoi eseguire il comando seguente:

```
cd parent_directory
mkdir temp_dir
```

2. Decomprimi il pacchetto e copia il contenuto nella directory temporanea. Rimuovi i file relativi allo standard di confezionamento precedente, lasciando solo il contenuto del modulo. Conferma che la struttura del file sia corretta con il seguente comando:

```
ls -AR temp_dir
# my_module
# my_module/__init__.py
# my_module/my_file_1.py
# my_module/my_submodule/__init__.py
# my_module/my_submodule/my_file_2.py
# my_module/my_submodule/my_file_3.py
```

3. Comprimi la cartella temporanea utilizzando zip. A questo scopo, puoi eseguire il comando seguente:

```
zip -r zip_file.zip temp_dir
```

4. Verifica che il file sia correttamente confezionato. Ora il *zip_file*.zip dovrebbe essere disponibile nella tua directory di lavoro. È possibile verificarla con il seguente comando:

```
zipinfo -1 zip_file.zip
# temp_dir/my_module/
# temp_dir/my_module/__init__.py
# temp_dir/my_module/my_file_1.py
# temp_dir/my_module/my_submodule/
# temp_dir/my_module/my_submodule/__init__.py
# temp_dir/my_module/my_submodule/my_file_2.py
# temp_dir/my_module/my_submodule/my_file_3.py
```

## Connessione ai dati nei processi Ray

> **Important**
>
> AWS Glue for Ray non è più aperto a nuovi clienti. I clienti esistenti possono continuare a utilizzare il servizio normalmente. Per ulteriori informazioni, consulta AWS Glue Ray End of Support.

AWS Glue I lavori Ray possono utilizzare un'ampia gamma di pacchetti Python progettati per integrare rapidamente i dati. Forniamo un set minimo di dipendenze per non appesantire l'ambiente. Per ulteriori informazioni sui componenti inclusi in modo predefinito, consulta la pagina the section called "Moduli disponibili con i processi Ray".

> **Note**
>
> AWS Glue extract, transform, and load (ETL) fornisce l' DynamicFrame astrazione per semplificare i flussi di lavoro ETL in cui risolvi le differenze di schema tra le righe del set di dati. AWS Glue ETL offre funzionalità aggiuntive: segnalibri di lavoro e raggruppamento di file di input. Al momento non forniamo funzionalità corrispondenti nei processi Ray.
> AWS Glue for Spark fornisce supporto diretto per la connessione a determinati formati di dati, fonti e sink. In Ray, l'SDK AWS per pandas e le attuali librerie di terze parti soddisfano sostanzialmente questa esigenza. Dovrai consultare tali librerie per capire quali funzionalità sono disponibili.

AWS Glue l'integrazione di for Ray con Amazon VPC non è attualmente disponibile. Le risorse in Amazon VPC non saranno accessibili senza un percorso pubblico. Per ulteriori informazioni sull'utilizzo AWS Glue con Amazon VPC, consulta. the section called "Configurazione degli endpoint VPC (AWS PrivateLink) per AWS Glue "

## Librerie comuni per lavorare con i dati in Ray

Ray Data: Ray Data fornisce metodi per gestire formati di dati, origini e sink comuni. Per ulteriori informazioni sui formati e le origini supportati in Ray Data, consulta la sezione Input/Output nella documentazione di Ray Data. Ray Data è una libreria prescrittiva anziché generica per la gestione di set di dati.

Ray fornisce alcune indicazioni sui casi d'uso in cui Ray Data potrebbe essere la soluzione migliore per il processo. Per ulteriori informazioni, consulta la sezione Casi d'uso di Ray nella documentazione di Ray.

AWS SDK for pandas (awswrangler) — AWS SDK for pandas è un AWS prodotto che offre soluzioni pulite e testate per la lettura e la scrittura da servizi quando le trasformazioni gestiscono i dati con pandas. AWS DataFrames Per ulteriori informazioni sui formati e le fonti supportati nell'SDK per pandas, consulta l'API Reference nella documentazione dell' AWS SDK per pandas. AWS

Per esempi su come leggere e scrivere dati con l' AWS SDK per panda, consulta Quick Start nella documentazione dell'SDK per pandas. AWS L' AWS SDK per panda non fornisce trasformazioni per i tuoi dati. Fornisce supporto solo per la lettura e la scrittura dalle origini.

Modin: Modin è una libreria Python che implementa le comuni operazioni pandas in modo distribuibile. Per ulteriori informazioni su Modin, consulta la documentazione di Modin. Modin non fornisce supporto per la lettura e la scrittura dalle origini. Fornisce implementazioni distribuite di trasformazioni comuni. Modin è supportato dall'SDK per panda. AWS

Quando esegui Modin e l' AWS SDK per panda insieme in un ambiente Ray, puoi eseguire attività ETL comuni con risultati performanti. Per ulteriori informazioni sull'utilizzo di Modin con l'SDK per pandas, consulta At scale nella documentazione AWS SDK for pandas. AWS

Altri framework: per ulteriori informazioni sui framework supportati da Ray, consulta la sezione L'ecosistema Ray nella documentazione di Ray. Non forniamo supporto per altri framework in for Ray. AWS Glue

## Connessione ai dati tramite Catalogo dati

La gestione dei dati tramite il Data Catalog in combinazione con Ray jobs è supportata dall' AWS SDK per panda. Per ulteriori informazioni, consulta Glue Catalog sul sito Web AWS SDK for pandas.

# Utilizzo di questo servizio con un AWS SDK

AWS I kit di sviluppo software (SDK) sono disponibili per molti linguaggi di programmazione più diffusi. Ogni SDK fornisce un'API, esempi di codice e documentazione che facilitano agli sviluppatori la creazione di applicazioni nel loro linguaggio preferito.

| Documentazione sugli SDK | Esempi di codice |
| --- | --- |
| AWS SDK per C++ | AWS SDK per C++ esempi di codice |
| AWS CLI | AWS CLI esempi di codice |
| AWS SDK per Go | AWS SDK per Go esempi di codice |
| AWS SDK per Java | AWS SDK per Java esempi di codice |
| AWS SDK per JavaScript | AWS SDK per JavaScript esempi di codice |
| AWS SDK per Kotlin | AWS SDK per Kotlin esempi di codice |
| AWS SDK per .NET | AWS SDK per .NET esempi di codice |
| AWS SDK per PHP | AWS SDK per PHP esempi di codice |
| AWS Strumenti per PowerShell | AWS Strumenti per PowerShell esempi di codice |
| AWS SDK per Python (Boto3) | AWS SDK per Python (Boto3) esempi di codice |
| AWS SDK per Ruby | AWS SDK per Ruby esempi di codice |
| AWS SDK per Rust | AWS SDK per Rust esempi di codice |
| AWS SDK per SAP ABAP | AWS SDK per SAP ABAP esempi di codice |
| AWS SDK per Swift | AWS SDK per Swift esempi di codice |

Per esempi specifici del servizio, consulta AWS Glue Esempi di codice API con AWS SDK.

Esempio di disponibilità

Non riesci a trovare quello che ti serve? Richiedi un esempio di codice utilizzando il link Fornisci un feedback nella parte inferiore di questa pagina.

# AWS Glue API

Questa sezione descrive i tipi di dati e le primitive utilizzati da AWS Glue SDKs and Tools. Esistono tre modi generali per interagire a AWS Glue livello di codice al di fuori di Console di gestione AWS, ognuno con la propria documentazione:

- Le librerie di linguaggi SDK consentono di accedere a risorse AWS provenienti da linguaggi di programmazione comuni. Per ulteriori informazioni, consulta Strumenti per costruire in AWS.
- AWS CLI consente di accedere alle AWS risorse dalla riga di comando. Per ulteriori informazioni, consulta Riferimento ai comandi della AWS CLI.
- CloudFormation consente di definire un insieme di AWS risorse da distribuire insieme in modo coerente. Puoi trovare maggiori informazioni su CloudFormation: riferimento al tipo di AWS Glue risorsa.

Questa sezione documenta le primitive condivise indipendentemente da queste SDKs e dagli strumenti. Gli strumenti utilizzano il AWS Glue Web API Reference per comunicare con. AWS

Indice


- Sicurezza APIs in AWS Glue
  - Tipi di dati
  - DataCatalogEncryptionSettings struttura
  - EncryptionAtRest struttura
  - ConnectionPasswordEncryption struttura
  - EncryptionConfiguration struttura
  - Struttura S3Encryption
  - CloudWatchEncryption struttura
  - JobBookmarksEncryption struttura
  - SecurityConfiguration struttura
  - GluePolicy struttura
  - DataQualityEncryption struttura
  - Operazioni
  - GetDataCatalogEncryptionSettings azione (Python: get_data_catalog_encryption_settings)
  - PutDataCatalogEncryptionSettings azione (Python: put_data_catalog_encryption_settings)

- PutResourcePolicy azione (Python: put_resource_policy)
- GetResourcePolicy azione (Python: get_resource_policy)
- DeleteResourcePolicy azione (Python: delete_resource_policy)
- CreateSecurityConfiguration azione (Python: create_security_configuration)
- DeleteSecurityConfiguration azione (Python: delete_security_configuration)
- GetSecurityConfiguration azione (Python: get_security_configuration)
- GetSecurityConfigurations azione (Python: get_security_configurations)
- GetResourcePolicies azione (Python: get_resource_policies)
- API Catalog objects
  - API Cataloghi
    - Tipi di dati
    - Struttura del catalogo
    - CatalogInput struttura
    - TargetRedshiftCatalog struttura
    - CatalogProperties struttura
    - CatalogPropertiesOutput struttura
    - DataLakeAccessProperties struttura
    - IcebergOptimizationProperties struttura
    - DataLakeAccessPropertiesOutput struttura
    - IcebergOptimizationPropertiesOutput struttura
    - FederatedCatalog struttura
    - Operazioni
    - CreateCatalog azione (Python: create_catalog)
    - UpdateCatalog azione (Python: update_catalog)
    - DeleteCatalog azione (Python: delete_catalog)
    - GetCatalog azione (Python: get_catalog)
    - GetCatalogs azione (Python: get_catalogs)
  - API database
    - Tipi di dati
    - Struttura dei database

- DatabaseInput struttura
- PrincipalPermissions struttura
- DataLakePrincipal struttura
- DatabaseIdentifier struttura
- FederatedDatabase struttura
- Operazioni
- CreateDatabase azione (Python: create_database)
- UpdateDatabase azione (Python: update_database)
- DeleteDatabase azione (Python: delete_database)
- GetDatabase azione (Python: get_database)
- GetDatabases azione (Python: get_databases)

- API Table
  - Tipi di dati
  - Struttura della tabella
  - TableInput struttura
  - FederatedTable struttura
  - Struttura delle colonne
  - StorageDescriptor struttura
  - SchemaReference struttura
  - SerDeInfo struttura
  - Struttura dell'ordine
  - SkewedInfo struttura
  - TableVersion struttura
  - TableError struttura
  - TableVersionError struttura
  - SortCriterion struttura
  - TableIdentifier struttura
  - KeySchemaElement struttura
  - PartitionIndex struttura
  - PartitionIndexDescriptor struttura

- BackfillError struttura
- IcebergInput struttura
- OpenTableFormatInput struttura
- ViewDefinition struttura
- ViewDefinitionInput struttura
- ViewRepresentation struttura
- ViewRepresentationInput struttura
- UpdateOpenTableFormatInput struttura
- UpdateIcebergInput struttura
- CreateIcebergTableInput struttura
- UpdateIcebergTableInput struttura
- IcebergSortOrder struttura
- IcebergSortField struttura
- IcebergPartitionSpec struttura
- IcebergPartitionField struttura
- IcebergSchema struttura
- IcebergStructField struttura
- IcebergTableUpdate struttura
- AuditContext struttura
- Operazioni
- CreateTable azione (Python: create_table)
- UpdateTable azione (Python: update_table)
- DeleteTable azione (Python: delete_table)
- BatchDeleteTable azione (Python: batch_delete_table)
- GetTable azione (Python: get_table)
- GetTables azione (Python: get_tables)
- GetTableVersion azione (Python: get_table_version)
- GetTableVersions azione (Python: get_table_versions)
- DeleteTableVersion azione (Python: delete_table_version)
- BatchDeleteTableVersion azione (Python: batch_delete_table_version)

- SearchTables azione (Python: search_tables)
- GetPartitionIndexes azione (Python: get_partition_indexes)
- CreatePartitionIndex azione (Python: create_partition_index)
- DeletePartitionIndex azione (Python: delete_partition_index)
- GetColumnStatisticsForTable azione (Python: get_column_statistics_for_table)
- UpdateColumnStatisticsForTable azione (Python: update_column_statistics_for_table)
- DeleteColumnStatisticsForTable azione (Python: delete_column_statistics_for_table)

- API della partizione
  - Tipi di dati
  - Struttura della partizione
  - PartitionInput struttura
  - PartitionSpecWithSharedStorageDescriptor struttura
  - PartitionListComposingSpec struttura
  - PartitionSpecProxy struttura
  - PartitionValueList struttura
  - Struttura del segmento
  - PartitionError struttura
  - BatchUpdatePartitionFailureEntry struttura
  - BatchUpdatePartitionRequestEntry struttura
  - StorageDescriptor struttura
  - SchemaReference struttura
  - SerDeInfo struttura
  - SkewedInfo struttura
  - Operazioni
  - CreatePartition azione (Python: create_partition)
  - BatchCreatePartition azione (Python: batch_create_partition)
  - UpdatePartition azione (Python: update_partition)
  - DeletePartition azione (Python: delete_partition)
  - BatchDeletePartition azione (Python: batch_delete_partition)
  - GetPartition azione (Python: get_partition)

- GetPartitions azione (Python: get_partitions)
- BatchGetPartition azione (Python: batch_get_partition)
- BatchUpdatePartition azione (Python: batch_update_partition)
- GetColumnStatisticsForPartition azione (Python: get_column_statistics_for_partition)
- UpdateColumnStatisticsForPartition azione (Python: update_column_statistics_for_partition)
- DeleteColumnStatisticsForPartition azione (Python: delete_column_statistics_for_partition)
- API Connessioni
  - API di connessione
    - Tipi di dati
    - Struttura di connessione
    - ConnectionInput struttura
    - TestConnectionInput struttura
    - PhysicalConnectionRequirements struttura
    - GetConnectionsFilter struttura
    - AuthenticationConfiguration struttura
    - AuthenticationConfigurationInput struttura
    - OAuth2Struttura delle proprietà
    - OAuth2PropertiesInput struttura
    - OAuth2ClientApplication struttura
    - AuthorizationCodeProperties struttura
    - BasicAuthenticationCredentials struttura
    - OAuth2struttura delle credenziali
    - Operazioni
    - CreateConnection azione (Python: create_connection)
    - DeleteConnection azione (Python: delete_connection)
    - GetConnection azione (Python: get_connection)
    - GetConnections azione (Python: get_connections)
    - UpdateConnection azione (Python: update_connection)
    - TestConnection azione (Python: test_connection)
    - BatchDeleteConnection azione (Python: batch_delete_connection)

- API Tipi di connessione
  - Gestione della connessione APIs
  - DescribeConnectionType azione (Python: describe_connection_type)
  - ListConnectionTypes azione (Python: list_connection_types)
  - ConnectionTypeBrief struttura
  - ConnectionTypeVariant struttura
  - datatypes
  - Struttura di validazione
  - AuthConfiguration struttura
  - Struttura delle funzionalità
  - Struttura della proprietà
  - AllowedValue struttura
  - ComputeEnvironmentConfiguration struttura
- Metadati di connessione e API di anteprima
  - Tipi di dati
  - Struttura dell'entità
  - Struttura del campo
  - Operazioni
  - ListEntities azione (Python: list_entities)
  - DescribeEntity azione (Python: describe_entity)
  - GetEntityRecords azione (Python: get_entity_records)
- API della funzione definita dall'utente
  - Tipi di dati
  - UserDefinedFunction struttura
  - UserDefinedFunctionInput struttura
  - Operazioni
  - CreateUserDefinedFunction azione (Python: create_user_defined_function)
  - UpdateUserDefinedFunction azione (Python: update_user_defined_function)
  - DeleteUserDefinedFunction azione (Python: delete_user_defined_function)
  - GetUserDefinedFunction azione (Python: get_user_defined_function)

        - GetUserDefinedFunctions azione (Python: get_user_defined_functions)
    - Importazione di un Athena catalogo in AWS Glue
        - Tipi di dati
        - CatalogImportStatus struttura
        - Operazioni
        - ImportCatalogToGlue azione (Python: import_catalog_to_glue)
        - GetCatalogImportStatus azione (Python: get_catalog_import_status)
- API dell'ottimizzatore di tabelle
    - Tipi di dati
    - TableOptimizer struttura
    - TableOptimizerConfiguration struttura
    - TableOptimizerVpcConfiguration struttura
    - CompactionConfiguration struttura
    - IcebergCompactionConfiguration struttura
    - TableOptimizerRun struttura
    - BatchGetTableOptimizerEntry struttura
    - BatchTableOptimizer struttura
    - BatchGetTableOptimizerError struttura
    - RetentionConfiguration struttura
    - IcebergRetentionConfiguration struttura
    - OrphanFileDeletionConfiguration struttura
    - IcebergOrphanFileDeletionConfiguration struttura
    - CompactionMetrics struttura
    - RetentionMetrics struttura
    - OrphanFileDeletionMetrics struttura
    - IcebergCompactionMetrics struttura
    - IcebergRetentionMetrics struttura
    - IcebergOrphanFileDeletionMetrics struttura
    - RunMetrics struttura
    - Operazioni

- GetTableOptimizer azione (Python: get_table_optimizer)
- BatchGetTableOptimizer azione (Python: batch_get_table_optimizer)
- ListTableOptimizerRuns azione (Python: list_table_optimizer_runs)
- CreateTableOptimizer azione (Python: create_table_optimizer)
- DeleteTableOptimizer azione (Python: delete_table_optimizer)
- UpdateTableOptimizer azione (Python: update_table_optimizer)

- API crawler e classificatori
  - API classificatore
    - Tipi di dati
    - Struttura classificatore
    - GrokClassifier struttura
    - XMLClassifier struttura
    - JsonClassifier struttura
    - CsvClassifier struttura
    - CreateGrokClassifierRequest struttura
    - UpdateGrokClassifierRequest struttura
    - Crea la struttura della richiesta XMLClassifier
    - Struttura della XMLClassifier richiesta di aggiornamento
    - CreateJsonClassifierRequest struttura
    - UpdateJsonClassifierRequest struttura
    - CreateCsvClassifierRequest struttura
    - UpdateCsvClassifierRequest struttura
    - Operazioni
    - CreateClassifier azione (Python: create_classifier)
    - DeleteClassifier azione (Python: delete_classifier)
    - GetClassifier azione (Python: get_classifier)
    - GetClassifiers azione (Python: get_classifiers)
    - UpdateClassifier azione (Python: update_classifier)
  - API crawler
    - Tipi di dati

- Struttura dei crawler
- Struttura della pianificazione
- CrawlerTargets struttura
- Struttura S3Target
- Struttura S3 DeltaCatalogTarget
- Struttura S3 DeltaDirectTarget
- JdbcTarget struttura
- struttura Mongo DBTarget
- Struttura a dinamo DBTarget
- DeltaTarget struttura
- IcebergTarget struttura
- HudiTarget struttura
- CatalogTarget struttura
- CrawlerMetrics struttura
- CrawlerHistory struttura
- CrawlsFilter struttura
- SchemaChangePolicy struttura
- LastCrawlInfo struttura
- RecrawlPolicy struttura
- LineageConfiguration struttura
- LakeFormationConfiguration struttura
- Operazioni
- CreateCrawler azione (Python: create_crawler)
- DeleteCrawler azione (Python: delete_crawler)
- GetCrawler azione (Python: get_crawler)
- GetCrawlers azione (Python: get_crawlers)
- GetCrawlerMetrics azione (Python: get_crawler_metrics)
- UpdateCrawler azione (Python: update_crawler)
- StartCrawler azione (Python: start_crawler)
- StopCrawler azione (Python: stop_crawler)

- BatchGetCrawlers azione (Python: batch_get_crawlers)
- ListCrawlers azione (Python: list_crawlers)
- ListCrawls azione (Python: list_crawls)

- API delle statistiche delle colonne
  - Tipi di dati
  - ColumnStatisticsTaskRun struttura
  - ColumnStatisticsTaskSettings struttura
  - ExecutionAttempt struttura
  - Operazioni
  - StartColumnStatisticsTaskRun azione (Python: start_column_statistics_task_run)
  - GetColumnStatisticsTaskRun azione (Python: get_column_statistics_task_run)
  - GetColumnStatisticsTaskRuns azione (Python: get_column_statistics_task_runs)
  - ListColumnStatisticsTaskRuns azione (Python: list_column_statistics_task_runs)
  - StopColumnStatisticsTaskRun azione (Python: stop_column_statistics_task_run)
  - CreateColumnStatisticsTaskSettings azione (Python: create_column_statistics_task_settings)
  - UpdateColumnStatisticsTaskSettings azione (Python: update_column_statistics_task_settings)
  - GetColumnStatisticsTaskSettings azione (Python: get_column_statistics_task_settings)
  - DeleteColumnStatisticsTaskSettings azione (Python: delete_column_statistics_task_settings)
  - StartColumnStatisticsTaskRunSchedule azione (Python: start_column_statistics_task_run_schedule)
  - StopColumnStatisticsTaskRunSchedule azione (Python: stop_column_statistics_task_run_schedule)
  - Eccezioni
  - ColumnStatisticsTaskRunningException struttura
  - ColumnStatisticsTaskNotRunningException struttura
  - ColumnStatisticsTaskStoppingException struttura
  - ColumnStatisticsTaskAutoConcurrencyLimitException struttura
  - InvalidCatalogSettingException struttura



- API del pianificatore del crawler

    - Tipi di dati
    - Struttura della pianificazione
    - Operazioni
    - UpdateCrawlerSchedule azione (Python: update_crawler_schedule)
    - StartCrawlerSchedule azione (Python: start_crawler_schedule)
    - StopCrawlerSchedule azione (Python: stop_crawler_schedule)
- API script ETL auto-generanti
    - Tipi di dati
    - CodeGenNode struttura
    - CodeGenNodeArg struttura
    - CodeGenEdge struttura
    - Struttura della posizione
    - CatalogEntry struttura
    - MappingEntry struttura
    - Operazioni
    - CreateScript azione (Python: create_script)
    - GetDataflowGraph azione (Python: get_dataflow_graph)
    - GetMapping azione (Python: get_mapping)
    - GetPlan azione (Python: get_plan)
- API processo visuale
    - Tipi di dati
    - CodeGenConfigurationNode struttura
    - JDBCConnectorStruttura delle opzioni
    - StreamingDataPreviewOptions struttura
    - AthenaConnectorSource struttura
    - JDBCConnectorStruttura del codice sorgente
    - SparkConnectorSource struttura
    - CatalogSource struttura
    - Struttura My SQL Catalog Source
    - Struttura di Postgree Source SQLCatalog

- Struttura SQLCatalog Oracle Source
- SQLServerCatalogSource Struttura Microsoft
- CatalogKinesisSource struttura
- DirectKinesisSource struttura
- KinesisStreamingSourceOptions struttura
- CatalogKafkaSource struttura
- DirectKafkaSource struttura
- KafkaStreamingSourceOptions struttura
- RedshiftSource struttura
- AmazonRedshiftSource struttura
- AmazonRedshiftNodeData struttura
- AmazonRedshiftAdvancedOption struttura
- Struttura Option
- struttura S3 CatalogSource
- Struttura S3 SourceAdditionalOptions
- Struttura S3 CsvSource
- JDBCSource Struttura diretta
- Struttura S3 DirectSourceAdditionalOptions
- Struttura S3 JsonSource
- Struttura S3 ParquetSource
- Struttura S3 DeltaSource
- Struttura S3 CatalogDeltaSource
- CatalogDeltaSource struttura
- Struttura S3 HudiSource
- Struttura S3 CatalogHudiSource
- Struttura S3 ExcelSource
- CatalogHudiSource struttura
- Struttura Dynamo DBCatalog Source
- RelationalCatalogSource struttura
- JDBCConnectorStruttura dell'obiettivo

- SparkConnectorTarget struttura
- BasicCatalogTarget struttura
- La mia struttura SQLCatalog Target
- Struttura di Postgree Target SQLCatalog
- Struttura di Oracle Target SQLCatalog
- SQLServerCatalogTarget Struttura Microsoft
- RedshiftTarget struttura
- AmazonRedshiftTarget struttura
- UpsertRedshiftTargetOptions struttura
- struttura S3 CatalogTarget
- Struttura S3 GlueParquetTarget
- CatalogSchemaChangePolicy struttura
- struttura S3 DirectTarget
- Struttura S3 HudiCatalogTarget
- Struttura S3 HudiDirectTarget
- Struttura S3 DeltaCatalogTarget
- Struttura S3 DeltaDirectTarget
- Struttura S3 HyperDirectTarget
- Struttura S3 IcebergDirectTarget
- DirectSchemaChangePolicy struttura
- ApplyMapping struttura
- Struttura mappatura
- SelectFields struttura
- DropFields struttura
- RenameField struttura
- Struttura Spigot
- Struttura join
- JoinColumn struttura
- SplitFields struttura
- SelectFromCollection struttura

- FillMissingValues struttura
- Struttura filtro
- FilterExpression struttura
- FilterValue struttura
- CustomCode struttura
- Struttura SparkSQL
- SqlAlias struttura
- DropNullFields struttura
- NullCheckBoxList struttura
- NullValueField struttura
- Struttura Datatype
- Struttura Merge
- Struttura unione
- PIIDetection struttura
- Struttura aggregata
- DropDuplicates struttura
- GovernedCatalogTarget struttura
- GovernedCatalogSource struttura
- AggregateOperation struttura
- GlueSchema struttura
- GlueStudioSchemaColumn struttura
- GlueStudioColumn struttura
- DynamicTransform struttura
- TransformConfigParameter struttura
- EvaluateDataQuality struttura
- DQResultsPublishingOptions struttura
- DQStopJobOnFailureOptions struttura
- EvaluateDataQualityMultiFrame struttura
- Struttura Recipe
- RecipeReference struttura

- SnowflakeNodeData struttura
- SnowflakeSource struttura
- SnowflakeTarget struttura
- ConnectorDataSource struttura
- ConnectorDataTarget struttura
- RecipeStep struttura
- RecipeAction struttura
- ConditionExpression struttura
- Struttura S3 CatalogIcebergSource
- CatalogIcebergSource struttura
- Struttura S3 IcebergCatalogTarget
- DBELTConnectorStruttura Dynamo Source
- DDBELTConnectionStruttura delle opzioni
- DDBELTCatalogAdditionalOptions struttura
- Struttura Route
- GroupFilters struttura
- AutoDataQuality struttura

- API dei processi
  - Jobs
    - Tipi di dati
    - Struttura del processo
    - ExecutionProperty struttura
    - NotificationProperty struttura
    - JobCommand struttura
    - ConnectionsList struttura
    - JobUpdate struttura
    - SourceControlDetails struttura
    - Operazioni
    - CreateJob azione (Python: create_job)
    - UpdateJob azione (Python: update_job)

- GetJob azione (Python: get_job)
- GetJobs azione (Python: get_jobs)
- DeleteJob azione (Python: delete_job)
- ListJobs azione (Python: list_jobs)
- BatchGetJobs azione (Python: batch_get_jobs)

- Esecuzioni di processi
  - Tipi di dati
  - JobRun struttura
  - Struttura Predecessor
  - JobBookmarkEntry struttura
  - BatchStopJobRunSuccessfulSubmission struttura
  - BatchStopJobRunError struttura
  - NotificationProperty struttura
  - Operazioni
  - StartJobRun azione (Python: start_job_run)
  - BatchStopJobRun azione (Python: batch_stop_job_run)
  - GetJobRun azione (Python: get_job_run)
  - GetJobRuns azione (Python: get_job_runs)
  - GetJobBookmark azione (Python: get_job_bookmark)
  - GetJobBookmarks azione (Python: get_job_bookmarks)
  - ResetJobBookmark azione (Python: reset_job_bookmark)
- Triggers
  - Tipi di dati
  - Struttura trigger
  - TriggerUpdate struttura
  - Struttura predicato
  - Struttura condizione
  - Struttura operazione
  - EventBatchingCondition struttura
  - Operazioni

- CreateTrigger azione (Python: create_trigger)
- StartTrigger azione (Python: start_trigger)
- GetTrigger azione (Python: get_trigger)
- GetTriggers azione (Python: get_triggers)
- UpdateTrigger azione (Python: update_trigger)
- StopTrigger azione (Python: stop_trigger)
- DeleteTrigger azione (Python: delete_trigger)
- ListTriggers azione (Python: list_triggers)
- BatchGetTriggers azione (Python: batch_get_triggers)

- Integrazione APIs in AWS Glue
  - Tipi di dati
  - Struttura di integrazione
  - IntegrationConfig struttura
  - IntegrationPartition struttura
  - IntegrationError struttura
  - IntegrationFilter struttura
  - InboundIntegration struttura
  - SourceProcessingProperties struttura
  - TargetProcessingProperties struttura
  - SourceTableConfig struttura
  - TargetTableConfig struttura
  - Operazioni
  - CreateIntegration azione (Python: create_integration)
  - ModifyIntegration azione (Python: modify_integration)
  - DescribeIntegrations azione (Python: describe_integrations)
  - DeleteIntegration azione (Python: delete_integration)
  - DescribeInboundIntegrations azione (Python: describe_inbound_integrazioni)
  - CreateIntegrationTableProperties azione (Python: create_integration_table_properties)
  - UpdateIntegrationTableProperties azione (Python: update_integration_table_properties)
  - GetIntegrationTableProperties azione (Python: get_integration_table_properties)

- DeleteIntegrationTableProperties azione (Python: delete_integration_table_properties)
- CreateIntegrationResourceProperty azione (Python: create_integration_resource_property)
- UpdateIntegrationResourceProperty azione (Python: update_integration_resource_property)
- GetIntegrationResourceProperty azione (Python: get_integration_resource_property)
- UntagResource azione (Python: untag_resource)
- ListTagsForResource azione (Python: list_tags_for_resource)
- Eccezioni
- ResourceNotFoundException struttura
- InternalServerException struttura
- IntegrationAlreadyExistsFault struttura
- IntegrationConflictOperationFault struttura
- IntegrationQuotaExceededFault struttura
- KMSKeyNotAccessibleFault struttura
- IntegrationNotFoundFault struttura
- TargetResourceNotFound struttura
- InvalidIntegrationStateFault struttura
- API Sessioni interattive
  - Tipi di dati
  - Struttura sessione
  - SessionCommand struttura
  - Struttura istruzione
  - StatementOutput struttura
  - StatementOutputData struttura
  - ConnectionsList struttura
  - Operazioni
  - CreateSession azione (Python: create_session)
  - StopSession azione (Python: stop_session)
  - DeleteSession azione (Python: delete_session)
  - GetSession azione (Python: get_session)
  - ListSessions azione (Python: list_sessions)

- RunStatement azione (Python: run_statement)
- CancelStatement azione (Python: cancel_statement)
- GetStatement azione (Python: get_statement)
- ListStatements azione (Python: list_statements)
- GetGlueIdentityCenterConfiguration azione (Python: get_glue_identity_center_configuration)
- UpdateGlueIdentityCenterConfiguration azione (Python: update_glue_identity_center_configuration)
- CreateGlueIdentityCenterConfiguration azione (Python: create_glue_identity_center_configuration)
- DeleteGlueIdentityCenterConfiguration azione (Python: delete_glue_identity_center_configuration)

- API endpoint di sviluppo
  - Tipi di dati
  - DevEndpoint struttura
  - DevEndpointCustomLibraries struttura
  - Operazioni
  - CreateDevEndpoint azione (Python: create_dev_endpoint)
  - UpdateDevEndpoint azione (Python: update_dev_endpoint)
  - DeleteDevEndpoint azione (Python: delete_dev_endpoint)
  - GetDevEndpoint azione (Python: get_dev_endpoint)
  - GetDevEndpoints azione (Python: get_dev_endpoints)
  - BatchGetDevEndpoints azione (Python: batch_get_dev_endpoints)
  - ListDevEndpoints azione (Python: list_dev_endpoints)
- Registro degli schemi
  - Tipi di dati
  - RegistryId struttura
  - RegistryListItem struttura
  - MetadataInfo struttura
  - OtherMetadataValueListItem struttura
  - SchemaListItem struttura
  - SchemaVersionListItem struttura

- MetadataKeyValuePair struttura
- SchemaVersionErrorItem struttura
- ErrorDetails struttura
- SchemaVersionNumber struttura
- SchemaId struttura
- Operazioni
- CreateRegistry azione (Python: create_registry)
- CreateSchema azione (Python: create_schema)
- GetSchema azione (Python: get_schema)
- ListSchemaVersions azione (Python: list_schema_versions)
- GetSchemaVersion azione (Python: get_schema_version)
- GetSchemaVersionsDiff azione (Python: get_schema_versions_diff)
- ListRegistries azione (Python: list_registries)
- ListSchemas azione (Python: list_schemas)
- RegisterSchemaVersion azione (Python: register_schema_version)
- UpdateSchema azione (Python: update_schema)
- CheckSchemaVersionValidity azione (Python: check_schema_version_idity)
- UpdateRegistry azione (Python: update_registry)
- GetSchemaByDefinition azione (Python: get_schema_by_definition)
- GetRegistry azione (Python: get_registry)
- PutSchemaVersionMetadata azione (Python: put_schema_version_metadata)
- QuerySchemaVersionMetadata azione (Python: query_schema_version_metadata)
- RemoveSchemaVersionMetadata azione (Python: remove_schema_version_metadata)
- DeleteRegistry azione (Python: delete_registry)
- DeleteSchema azione (Python: delete_schema)
- DeleteSchemaVersions azione (Python: delete_schema_versions)

- Flussi di lavoro
  - Tipi di dati
  - JobNodeDetails struttura
  - CrawlerNodeDetails struttura

- TriggerNodeDetails struttura
- Struttura crawl
- Struttura nodo
- Struttura edge
- Struttura flusso di lavoro
- WorkflowGraph struttura
- WorkflowRun struttura
- WorkflowRunStatistics struttura
- StartingEventBatchCondition struttura
- Struttura blueprint
- BlueprintDetails struttura
- LastActiveDefinition struttura
- BlueprintRun struttura
- Operazioni
- CreateWorkflow azione (Python: create_workflow)
- UpdateWorkflow azione (Python: update_workflow)
- DeleteWorkflow azione (Python: delete_workflow)
- GetWorkflow azione (Python: get_workflow)
- ListWorkflows azione (Python: list_workflows)
- BatchGetWorkflows azione (Python: batch_get_workflows)
- GetWorkflowRun azione (Python: get_workflow_run)
- GetWorkflowRuns azione (Python: get_workflow_runs)
- GetWorkflowRunProperties azione (Python: get_workflow_run_properties)
- PutWorkflowRunProperties azione (Python: put_workflow_run_properties)
- CreateBlueprint azione (Python: create_blueprint)
- UpdateBlueprint azione (Python: update_blueprint)
- DeleteBlueprint azione (Python: delete_blueprint)
- ListBlueprints azione (Python: list_blueprints)
- BatchGetBlueprints azione (Python: batch_get_blueprints)
- StartBlueprintRun azione (Python: start_blueprint_run)

- GetBlueprintRun azione (Python: get_blueprint_run)
- GetBlueprintRuns azione (Python: get_blueprint_runs)
- StartWorkflowRun azione (Python: start_workflow_run)
- StopWorkflowRun azione (Python: stop_workflow_run)
- ResumeWorkflowRun azione (Python: resume_workflow_run)

- Profili di utilizzo
  - Tipi di dati
  - ProfileConfiguration struttura
  - ConfigurationObject struttura
  - UsageProfileDefinition struttura
  - Operazioni
  - CreateUsageProfile azione (Python: create_usage_profile)
  - GetUsageProfile azione (Python: get_usage_profile)
  - UpdateUsageProfile azione (Python: update_usage_profile)
  - DeleteUsageProfile azione (Python: delete_usage_profile)
  - ListUsageProfiles azione (Python: list_usage_profiles)

- API machine learning
  - Tipi di dati
  - TransformParameters struttura
  - EvaluationMetrics struttura
  - MLTransform struttura
  - FindMatchesParameters struttura
  - FindMatchesMetrics struttura
  - ConfusionMatrix struttura
  - GlueTable struttura
  - TaskRun struttura
  - TransformFilterCriteria struttura
  - TransformSortCriteria struttura
  - TaskRunFilterCriteria struttura
  - TaskRunSortCriteria struttura

- TaskRunProperties struttura
- FindMatchesTaskRunProperties struttura
- ImportLabelsTaskRunProperties struttura
- ExportLabelsTaskRunProperties struttura
- LabelingSetGenerationTaskRunProperties struttura
- SchemaColumn struttura
- TransformEncryption struttura
- MLUserDataEncryption struttura
- ColumnImportance struttura
- Operazioni
- Crea MLTransform azione (Python: create_ml_transform)
- MLTransform Azione di aggiornamento (Python: update_ml_transform)
- MLTransform Azione di cancellazione (Python: delete_ml_transform)
- Ottieni MLTransform un'azione (Python: get_ml_transform)
- Ottieni MLTransforms un'azione (Python: get_ml_transforms)
- MLTransforms Azione elenco (Python: list_ml_transforms)
- Inizia MLEvaluation TaskRun l'azione (Python: start_ml_evaluation_task_run)
- Inizia MLLabeling SetGenerationTaskRun l'azione (Python: start_ml_labeling_set_generation_task_run)
- Azione Get MLTask Run (Python: get_ml_task_run)
- Azione Get MLTask Runs (Python: get_ml_task_runs)
- Annulla azione MLTask Esegui (Python: cancel_ml_task_run)
- StartExportLabelsTaskRun azione (Python: start_export_labels_task_run)
- StartImportLabelsTaskRun azione (Python: start_import_labels_task_run)

- API di qualità dei dati
  - Tipi di dati
  - DataSource struttura
  - DataQualityRulesetListDetails struttura
  - DataQualityTargetTable struttura
  - DataQualityRulesetEvaluationRunDescription struttura

- DataQualityRulesetEvaluationRunFilter struttura
- DataQualityEvaluationRunAdditionalRunOptions struttura
- DataQualityRuleRecommendationRunDescription struttura
- DataQualityRuleRecommendationRunFilter struttura
- DataQualityResult struttura
- DataQualityAnalyzerResult struttura
- DataQualityObservation struttura
- MetricBasedObservation struttura
- DataQualityMetricValues struttura
- DataQualityRuleResult struttura
- DataQualityResultDescription struttura
- DataQualityResultFilterCriteria struttura
- DataQualityRulesetFilterCriteria struttura
- DataQualityAggregatedMetrics struttura
- StatisticAnnotation struttura
- TimestampedInclusionAnnotation struttura
- AnnotationError struttura
- DatapointInclusionAnnotation struttura
- StatisticSummaryList lista
- StatisticSummary struttura
- RunIdentifier struttura
- StatisticModelResult struttura
- DataQualityGlueTable struttura
- Operazioni
- StartDataQualityRulesetEvaluationRun azione (Python: start_data_quality_ruleset_evaluation_run)
- CancelDataQualityRulesetEvaluationRun azione (Python: cancel_data_quality_ruleset_evaluation_run)
- GetDataQualityRulesetEvaluationRun azione (Python: get_data_quality_ruleset_evaluation_run)
- ListDataQualityRulesetEvaluationRuns azione (Python: list_data_quality_ruleset_evaluation_runs)

- StartDataQualityRuleRecommendationRun azione (Python: start_data_quality_rule_recommendation_run)
- CancelDataQualityRuleRecommendationRun azione (Python: cancel_data_quality_rule_recommendation_run)
- GetDataQualityRuleRecommendationRun azione (Python: get_data_quality_rule_recommendation_run)
- ListDataQualityRuleRecommendationRuns azione (Python: list_data_quality_rule_recommendation_runs)
- GetDataQualityResult azione (Python: get_data_quality_result)
- BatchGetDataQualityResult azione (Python: batch_get_data_quality_result)
- ListDataQualityResults azione (Python: list_data_quality_results)
- CreateDataQualityRuleset azione (Python: create_data_quality_ruleset)
- DeleteDataQualityRuleset azione (Python: delete_data_quality_ruleset)
- GetDataQualityRuleset azione (Python: get_data_quality_ruleset)
- ListDataQualityRulesets azione (Python: list_data_quality_rulesets)
- UpdateDataQualityRuleset azione (Python: update_data_quality_ruleset)
- ListDataQualityStatistics azione (Python: list_data_quality_statistics)
- TimestampFilter struttura
- CreateDataQualityRulesetRequest struttura
- GetDataQualityRulesetResponse struttura
- GetDataQualityResultResponse struttura
- StartDataQualityRuleRecommendationRunRequest struttura
- GetDataQualityRuleRecommendationRunResponse struttura
- BatchPutDataQualityStatisticAnnotation azione (Python: batch_put_data_quality_statistic_annotation)
- GetDataQualityModel azione (Python: get_data_quality_model)
- GetDataQualityModelResult azione (Python: get_data_quality_model_result)
- ListDataQualityStatisticAnnotations azione (Python: list_data_quality_statistic_annotations)
- PutDataQualityProfileAnnotation azione (Python: put_data_quality_profile_annotation)

- API di rilevamento dati sensibili
  - Tipi di dati

- CustomEntityType struttura
- Operazioni
- CreateCustomEntityType azione (Python: create_custom_entity_type)
- DeleteCustomEntityType azione (Python: delete_custom_entity_type)
- GetCustomEntityType azione (Python: get_custom_entity_type)
- BatchGetCustomEntityTypes azione (Python: batch_get_custom_entity_types)
- ListCustomEntityTypes azione (Python: list_custom_entity_types)

- Taggare APIs AWS Glue
  - Tipi di dati
  - Struttura tag
  - Operazioni
  - TagResource azione (Python: tag_resource)
  - UntagResource azione (Python: untag_resource)
  - GetTags azione (Python: get_tags)

- Tipi di dati comuni
  - Struttura tag
  - DecimalNumber struttura
  - ErrorDetail struttura
  - PropertyPredicate struttura
  - ResourceUri struttura
  - ColumnStatistics struttura
  - ColumnStatisticsError struttura
  - ColumnError struttura
  - ColumnStatisticsData struttura
  - BooleanColumnStatisticsData struttura
  - DateColumnStatisticsData struttura
  - DecimalColumnStatisticsData struttura
  - DoubleColumnStatisticsData struttura
  - LongColumnStatisticsData struttura
  - StringColumnStatisticsData struttura

- BinaryColumnStatisticsData struttura
- Modelli di stringa
- Eccezioni
  - AccessDeniedException struttura
  - AlreadyExistsException struttura
  - ConcurrentModificationException struttura
  - ConcurrentRunsExceededException struttura
  - CrawlerNotRunningException struttura
  - CrawlerRunningException struttura
  - CrawlerStoppingException struttura
  - EntityNotFoundException struttura
  - FederationSourceException struttura
  - FederationSourceRetryableException struttura
  - GlueEncryptionException struttura
  - IdempotentParameterMismatchException struttura
  - IllegalWorkflowStateException struttura
  - InternalServiceException struttura
  - InvalidExecutionEngineException struttura
  - InvalidInputException struttura
  - InvalidStateException struttura
  - InvalidTaskStatusTransitionException struttura
  - JobDefinitionErrorException struttura
  - JobRunInTerminalStateException struttura
  - JobRunInvalidStateTransitionException struttura
  - JobRunNotInTerminalStateException struttura
  - LateRunnerException struttura
  - NoScheduleException struttura
  - OperationTimeoutException struttura
  - ResourceNotReadyException struttura
  - ResourceNumberLimitExceededException struttura

- SchedulerNotRunningException struttura
- SchedulerRunningException struttura
- SchedulerTransitioningException struttura
- UnrecognizedRunnerException struttura
- ValidationException struttura
- VersionMismatchException struttura

# Sicurezza APIs in AWS Glue

L'API di sicurezza descrive i tipi di dati di sicurezza e l'API relativa alla sicurezza in AWS Glue.

## Tipi di dati

- DataCatalogEncryptionSettings struttura
- EncryptionAtRest struttura
- ConnectionPasswordEncryption struttura
- EncryptionConfiguration struttura
- Struttura S3Encryption
- CloudWatchEncryption struttura
- JobBookmarksEncryption struttura
- SecurityConfiguration struttura
- GluePolicy struttura
- DataQualityEncryption struttura

## DataCatalogEncryptionSettings struttura

Contiene le informazioni di configurazione per mantenere la sicurezza del catalogo dati.

Campi

- `EncryptionAtRest`: un oggetto EncryptionAtRest.

  Speciifica la encryption-at-rest configurazione per il Data Catalog.
- `ConnectionPasswordEncryption`: un oggetto ConnectionPasswordEncryption.

Quando è abilitata la protezione della password di connessione, il catalogo dati utilizza una chiave fornita dal cliente per crittografare la password come parte di `CreateConnection` o `UpdateConnection` e memorizzarla nel campo `ENCRYPTED_PASSWORD` nelle proprietà di connessione. È possibile abilitare la crittografia del catalogo o solo la crittografia delle password.

## EncryptionAtRest struttura

Speciifica la encryption-at-rest configurazione per il Data Catalog.

Campi

- `CatalogEncryptionMode` - obbligatorio: stringa UTF-8 (valori validi: `DISABLED` | `SSE-KMS="SSEKMS"` | `SSE-KMS-WITH-SERVICE-ROLE="SSEKMSWITHSERVICEROLE"`).

  La encryption-at-rest modalità per crittografare i dati del Data Catalog.
- `SseAwsKmsKeyId`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID della AWS KMS chiave da utilizzare per la crittografia a riposo.
- `CatalogEncryptionServiceRole`: stringa UTF-8, corrispondente a Custom string pattern #53.

  Il ruolo che AWS Glue assume di crittografare e decrittografare gli oggetti del Data Catalog per conto del chiamante.

## ConnectionPasswordEncryption struttura

La struttura di dati utilizzata dal catalogo dati per crittografare la password come parte di `CreateConnection` o `UpdateConnection` e memorizzarla nel campo `ENCRYPTED_PASSWORD` nelle proprietà di connessione. È possibile abilitare la crittografia del catalogo o solo la crittografia delle password.

Quando arriva una `CreationConnection` richiesta contenente una password, il Data Catalog crittografa innanzitutto la password utilizzando la AWS KMS chiave dell'utente. Successivamente crittografa l'intero oggetto di connessione di nuovo se anche la crittografia del catalogo è abilitata.

Questa crittografia richiede l'impostazione delle autorizzazioni AWS KMS chiave per abilitare o limitare l'accesso alla chiave della password in base ai requisiti di sicurezza. Ad esempio, è possibile

che si voglia concedere solo agli amministratori l'autorizzazione di decrittografare la chiave della password.

Campi

- `ReturnConnectionPasswordEncrypted`: obbligatorio: booleano.

  Quando il flag `ReturnConnectionPasswordEncrypted` è impostato su "true", le password rimangono crittografate nelle risposte di `GetConnection` e `GetConnections`. Questa crittografia è effettiva indipendentemente dalla crittografia del catalogo.
- `AwsKmsKeyId`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Una AWS KMS chiave utilizzata per crittografare la password di connessione.

  Se la protezione tramite password di connessione è abilitata, il chiamante `CreateConnection` e `UpdateConnection` necessita almeno dell'`kms:Encrypt`autorizzazione sulla AWS KMS chiave specificata per crittografare le password prima di archiviarle nel Data Catalog.

  È possibile impostare l'autorizzazione di decrittografia per consentire o limitare l'accesso alla chiave della password in base ai requisiti di sicurezza.

## EncryptionConfiguration struttura

Specifica una configurazione di crittografia.

Campi

- `S3Encryption`: una matrice di oggetti S3Encryption.

  La configurazione di crittografia per i dati Amazon Simple Storage Service (Amazon S3).
- `CloudWatchEncryption`: un oggetto CloudWatchEncryption.

  La configurazione di crittografia per Amazon CloudWatch.
- `JobBookmarksEncryption`: un oggetto JobBookmarksEncryption.

  Configurazione di crittografia per i segnalibri dei processi.
- `DataQualityEncryption`: un oggetto DataQualityEncryption.

  La configurazione di crittografia per gli asset AWS Glue Data Quality.

## Struttura S3Encryption

Specifica come devono essere crittografati i dati Amazon Simple Storage Service (Amazon S3).

Campi

- `S3EncryptionMode`: stringa UTF-8 (valori validi: `DISABLED` | `SSE-KMS="SSEKMS"` | `SSE-S3="SSES3"`).

  Modalità di crittografia da usare per i dati Amazon S3.
- `KmsKeyArn`: stringa UTF-8, corrispondente a Custom string pattern #42.

  ARN (Amazon Resource Name) della chiave KMS da usare per crittografare i dati.

## CloudWatchEncryption struttura

Speciifica in che modo CloudWatch i dati di Amazon devono essere crittografati.

Campi

- `CloudWatchEncryptionMode`: stringa UTF-8 (valori validi: `DISABLED` | `SSE-KMS="SSEKMS"`).

  La modalità di crittografia da utilizzare per CloudWatch i dati.
- `KmsKeyArn`: stringa UTF-8, corrispondente a Custom string pattern #42.

  ARN (Amazon Resource Name) della chiave KMS da usare per crittografare i dati.

## JobBookmarksEncryption struttura

Specifica come devono essere crittografati i dati dei segnalibri dei processi.

Campi

- `JobBookmarksEncryptionMode`: stringa UTF-8 (valori validi: `DISABLED` | `CSE-KMS="CSEKMS"`).

  Modalità di crittografia da usare per i dati dei segnalibri dei processi.
- `KmsKeyArn`: stringa UTF-8, corrispondente a Custom string pattern #42.

  ARN (Amazon Resource Name) della chiave KMS da usare per crittografare i dati.

## SecurityConfiguration struttura

Specifica una configurazione di sicurezza.

Campi

- `Name`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome della configurazione di sicurezza.

- `CreatedTimeStamp`: timestamp.

  Data e ora in cui questa configurazione di sicurezza è stata creata.

- `EncryptionConfiguration`: un oggetto EncryptionConfiguration.

  Configurazione di crittografia associata a questa configurazione di sicurezza.

## GluePolicy struttura

Una struttura per la restituzione di una policy delle risorse.

Campi

- `PolicyInJson`: stringa UTF-8, almeno 2 byte di lunghezza.

  Contiene il documento di policy richiesto, in formato JSON.

- `PolicyHash`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Contiene il valore hash associato a questa policy.

- `CreateTime`: timestamp.

  La data e l'ora di creazione della policy.

- `UpdateTime`: timestamp.

  La data e l'ora dell'ultimo aggiornamento della policy.

## DataQualityEncryption struttura

Specifica in che modo devono essere crittografate le risorse di Data Quality nell'account.

Campi

- `DataQualityEncryptionMode`: stringa UTF-8 (valori validi: `DISABLED` | `SSE-KMS="SSEKMS"`).

  La modalità di crittografia da utilizzare per crittografare le risorse di Data Quality. Tali risorse includono set di regole, risultati, statistiche, modelli di rilevamento delle anomalie e osservazioni di qualità dei dati.

  I valori validi sono `SSEKMS`, per la crittografia tramite una chiave KMS gestita dal cliente, oppure `DISABLED`.
- `KmsKeyArn`: stringa UTF-8, corrispondente a Custom string pattern #42.

  ARN (Amazon Resource Name) della chiave KMS da usare per crittografare i dati.

## Operazioni

- GetDataCatalogEncryptionSettings azione (Python: get_data_catalog_encryption_settings)
- PutDataCatalogEncryptionSettings azione (Python: put_data_catalog_encryption_settings)
- PutResourcePolicy azione (Python: put_resource_policy)
- GetResourcePolicy azione (Python: get_resource_policy)
- DeleteResourcePolicy azione (Python: delete_resource_policy)
- CreateSecurityConfiguration azione (Python: create_security_configuration)
- DeleteSecurityConfiguration azione (Python: delete_security_configuration)
- GetSecurityConfiguration azione (Python: get_security_configuration)
- GetSecurityConfigurations azione (Python: get_security_configurations)
- GetResourcePolicies azione (Python: get_resource_policies)

## GetDataCatalogEncryptionSettings azione (Python: get_data_catalog_encryption_settings)

Recupera la configurazione di sicurezza per un catalogo specificato.

Richiesta

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  ID del catalogo dati per cui recuperare la configurazione di sicurezza. Se non ne viene fornito nessuno, per impostazione predefinita viene utilizzato l'ID dell'account. AWS

Risposta

- `DataCatalogEncryptionSettings`: un oggetto DataCatalogEncryptionSettings.

  Configurazione di sicurezza richiesta.

Errori

- `InternalServiceException`
- `InvalidInputException`
- `OperationTimeoutException`

## PutDataCatalogEncryptionSettings azione (Python: put_data_catalog_encryption_settings)

Imposta la configurazione di sicurezza per un catalogo specificato. Dopo aver impostato la configurazione, la crittografia specificata viene applicata a ogni scrittura successiva nel catalogo.

Richiesta

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  ID del catalogo dati per cui impostare la configurazione di sicurezza. Se non ne viene fornito nessuno, per impostazione predefinita viene utilizzato l'ID dell'account. AWS
- `DataCatalogEncryptionSettings` - obbligatorio: un oggetto DataCatalogEncryptionSettings.

  Configurazione di sicurezza da impostare.

Risposta

- Nessun parametro di risposta.

Errori

- `InternalServiceException`
- `InvalidInputException`
- `OperationTimeoutException`

## PutResourcePolicy azione (Python: put_resource_policy)

Imposta la policy per la risorsa del catalogo dati per il controllo accessi.

Richiesta

- `PolicyInJson` - obbligatorio: stringa UTF-8, almeno 2 byte di lunghezza.

  Contiene il documento di policy da impostare, in formato JSON.
- `ResourceArn`: stringa UTF-8, non inferiore a 1 o superiore a 10240 byte di lunghezza, corrispondente a Custom string pattern #50.

  Non usare. Solo per uso interno.
- `PolicyHashCondition`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il valore hash restituito quando la policy precedente è stata impostata utilizzando `PutResourcePolicy`. Il suo scopo è prevenire modifiche simultanee di una policy. Non utilizzare questo parametro se non è stata impostata alcuna policy precedente.
- `PolicyExistsCondition`: stringa UTF-8 (valori validi: `MUST_EXIST` | `NOT_EXIST` | `NONE`).

  Il valore `MUST_EXIST` viene utilizzato per aggiornare una policy. Il valore `NOT_EXIST` viene utilizzato per creare una nuova policy. Se viene utilizzato il valore `NONE` o un valore null, la chiamata non dipende dalla presenza di una policy.
- `EnableHybrid`: stringa UTF-8 (valori validi: `TRUE` | `FALSE`).

  Se `'TRUE'`, vuol dire che stai usando entrambi i metodi per concedere l'accesso tra account alle risorse del catalogo dati:

- Aggiornando direttamente la policy delle risorse con `PutResourePolicy`
- Utilizzando il comando Concessione di autorizzazioni sulla Console di gestione AWS.

Se è già stata utilizzata la Console di gestione per concedere l'accesso tra account, deve essere impostato su `'TRUE'`, altrimenti la chiamata non riesce. Il valore di default è 'FALSE'.

Risposta

- `PolicyHash`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Un hash della policy appena impostata. Questo deve essere incluso in una chiamata successiva che sovrascrive o aggiorna questa policy.

Errori

- `EntityNotFoundException`
- `InternalServiceException`
- `OperationTimeoutException`
- `InvalidInputException`
- `ConditionCheckFailureException`

## GetResourcePolicy azione (Python: get_resource_policy)

Recupera una policy di risorse specificata.

Richiesta

- `ResourceArn`: stringa UTF-8, non inferiore a 1 o superiore a 10240 byte di lunghezza, corrispondente a Custom string pattern #50.

  L'ARN della AWS Glue risorsa per cui recuperare la politica delle risorse. Se non viene fornito, viene restituita la policy delle risorse del catalogo dati. Utilizza `GetResourcePolicies` per visualizzare tutte le policy delle risorse esistenti. Per ulteriori informazioni, vedere AWS Glue Specificing Resource. ARNs

Risposta

- `PolicyInJson`: stringa UTF-8, almeno 2 byte di lunghezza.

  Contiene il documento di policy richiesto, in formato JSON.

- `PolicyHash`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Contiene il valore hash associato a questa policy.

- `CreateTime`: timestamp.

  La data e l'ora di creazione della policy.

- `UpdateTime`: timestamp.

  La data e l'ora dell'ultimo aggiornamento della policy.

Errori

- `EntityNotFoundException`
- `InternalServiceException`
- `OperationTimeoutException`
- `InvalidInputException`

## DeleteResourcePolicy azione (Python: delete_resource_policy)

Elimina una policy specificata.

Richiesta

- `PolicyHashCondition`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il valore hash restituito quando è stata impostata questa policy.

- `ResourceArn`: stringa UTF-8, non inferiore a 1 o superiore a 10240 byte di lunghezza, corrispondente a Custom string pattern #50.

  L'ARN della AWS Glue risorsa per la politica delle risorse da eliminare.

Risposta

- Nessun parametro di risposta.

Errori

- `EntityNotFoundException`
- `InternalServiceException`
- `OperationTimeoutException`
- `InvalidInputException`
- `ConditionCheckFailureException`

## CreateSecurityConfiguration azione (Python: create_security_configuration)

Crea una nuova configurazione di sicurezza. Una configurazione della sicurezza è un set di proprietà di sicurezza che AWS Glue può usare. Puoi usare una configurazione della sicurezza per crittografare i dati inattivi. Per informazioni sull'utilizzo delle configurazioni di sicurezza in AWS Glue, consulta Encrypting Data Written by Crawlers, Jobs and Development Endpoints.

Richiesta

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome per la nuova configurazione di sicurezza.

- `EncryptionConfiguration` - obbligatorio: oggetto EncryptionConfiguration.

  Configurazione di crittografia per la nuova configurazione di sicurezza.

Risposta

- `Name`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome assegnato alla nuova configurazione di sicurezza.

- `CreatedTimestamp`: timestamp.

Data e ora in cui la nuova configurazione di sicurezza è stata creata.

Errori

- `AlreadyExistsException`
- `InvalidInputException`
- `InternalServiceException`
- `OperationTimeoutException`
- `ResourceNumberLimitExceededException`

## DeleteSecurityConfiguration azione (Python: delete_security_configuration)

Elimina una configurazione di sicurezza specificata.

Richiesta

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome della configurazione di sicurezza da eliminare.

Risposta

- Nessun parametro di risposta.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `InternalServiceException`
- `OperationTimeoutException`

## GetSecurityConfiguration azione (Python: get_security_configuration)

Recupera una configurazione di sicurezza specificata.

Richiesta

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome della configurazione di sicurezza da recuperare.

Risposta

- `SecurityConfiguration`: un oggetto SecurityConfiguration.

  Configurazione di sicurezza richiesta.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `InternalServiceException`
- `OperationTimeoutException`

## GetSecurityConfigurations azione (Python: get_security_configurations)

Recupera un elenco di tutte le configurazioni di sicurezza.

Richiesta

- `MaxResults`: numero (intero), non inferiore a 1 o superiore a 1.000.

  Numero massimo di risultati da restituire.
- `NextToken`: stringa UTF-8.

  Un token di continuazione, se si tratta di una chiamata di continuazione.

Risposta

- `SecurityConfigurations`: una matrice di oggetti SecurityConfiguration.

  Elenco di configurazioni di sicurezza.

- `NextToken`: stringa UTF-8.

  Token di continuazione, se devono essere restituite più configurazioni di sicurezza.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `InternalServiceException`
- `OperationTimeoutException`

## GetResourcePolicies azione (Python: get_resource_policies)

Recupera le politiche relative alle risorse impostate sulle singole risorse durante le concessioni di autorizzazioni tra account. AWS Resource Access Manager Recupera anche la policy per la risorsa del catalogo dati.

Se hai abilitato la crittografia dei metadati nelle impostazioni di Data Catalog e non disponi dell'autorizzazione sulla AWS KMS chiave, l'operazione non può restituire la politica delle risorse del Data Catalog.

Richiesta

- `NextToken`: stringa UTF-8.

  Token di continuazione, se si tratta di una richiesta di continuazione.
- `MaxResults`: numero (intero), non inferiore a 1 o superiore a 1.000.

  La dimensione massima di un elenco da restituire.

Risposta

- `GetResourcePoliciesResponseList`: una matrice di oggetti GluePolicy.

  Elenco delle singole policy delle risorse e delle policy delle risorse a livello di account.
- `NextToken`: stringa UTF-8.

  Token di continuazione, se l'elenco restituito non contiene l'ultima policy delle risorse disponibile.

Errori

- `InternalServiceException`
- `OperationTimeoutException`
- `InvalidInputException`
- `GlueEncryptionException`

# API Catalog objects

L'API Catalog objects descrive i tipi di dati e l'API relativi all'utilizzo dei cataloghi in AWS Glue.

Argomenti

- API Cataloghi
- API database
- API Table
- API della partizione
- API Connessioni
- API della funzione definita dall'utente
- Importazione di un Athena catalogo in AWS Glue

## API Cataloghi

L'API Catalogs descrive come creare, eliminare, localizzare, aggiornare ed elencare i cataloghi. APIs

### Tipi di dati

- Struttura del catalogo
- CatalogInput struttura
- TargetRedshiftCatalog struttura
- CatalogProperties struttura
- CatalogPropertiesOutput struttura
- DataLakeAccessProperties struttura
- IcebergOptimizationProperties struttura
- DataLakeAccessPropertiesOutput struttura

- IcebergOptimizationPropertiesOutput struttura
- FederatedCatalog struttura

## Struttura del catalogo

L'oggetto catalog rappresenta un raggruppamento logico di database nel AWS Glue Data Catalog o in una fonte federata. Ora puoi creare un catalogo federato Redshift o un catalogo contenente collegamenti a risorse ai database Redshift in un altro account o regione.

Campi

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo. Per concedere l'accesso al catalogo predefinito, questo campo non deve essere fornito.
- `Name`: obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 64 byte di lunghezza, corrispondente a Custom string pattern #25.

  Il nome del catalogo. Non può essere uguale all'ID account.
- `ResourceArn`: stringa UTF-8.

  Il nome della risorsa Amazon (ARN) assegnato alla risorsa del catalogo.
- `Description`: stringa di descrizione, non superiore a 2048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  Stringa di descrizione, non superiore a 2.048 byte di lunghezza, corrispondente al modello di stringa su più righe dell'indirizzo URI. Una descrizione del catalogo.
- `Parameters`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa chiave, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Ogni valore è una stringa UTF-8, lunga non più di 512000 byte.

  Un array della mappa di coppie chiave-valore che definiscono parametri e proprietà del catalogo.
- `CreateTime`: timestamp.

  L'ora in cui è stato creato il catalogo.

- `UpdateTime`: timestamp.

  L'ora dell'ultimo aggiornamento del catalogo.

- `TargetRedshiftCatalog`: un oggetto TargetRedshiftCatalog.

  Un oggetto `TargetRedshiftCatalog` che descrive un catalogo di destinazione per il collegamento del database.

- `FederatedCatalog`: un oggetto FederatedCatalog.

  Un `FederatedCatalog` oggetto che punta a un'entità esterna al AWS Glue Data Catalog.

- `CatalogProperties`: un oggetto CatalogPropertiesOutput.

  Un oggetto `CatalogProperties` che specifica le proprietà di accesso al data lake e altre proprietà personalizzate.

- `CreateTableDefaultPermissions`: una matrice di oggetti PrincipalPermissions.

  Un array di oggetti `PrincipalPermissions`. Crea un set di autorizzazioni predefinite sulle tabelle per i principali. Usato da AWS Lake Formation. Non utilizzato nel normale corso delle AWS Glue operazioni.

- `CreateDatabaseDefaultPermissions`: una matrice di oggetti PrincipalPermissions.

  Un array di oggetti `PrincipalPermissions`. Crea un set di autorizzazioni predefinite sui database per i principali. Usato da AWS Lake Formation. Non utilizzato nel normale corso delle AWS Glue operazioni.

- `AllowFullTableExternalDataAccess`: stringa UTF-8 (valori validi: `True` | `False`).

  Consente ai motori di terze parti di accedere ai dati in Amazon S3 località registrate con Lake Formation.

## CatalogInput struttura

Una struttura che descrive le proprietà del catalogo.

Campi

- `Description`: stringa di descrizione, non superiore a 2048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

Stringa di descrizione, non superiore a 2.048 byte di lunghezza, corrispondente al modello di stringa su più righe dell'indirizzo URI. Una descrizione del catalogo.

- `FederatedCatalog`: un oggetto FederatedCatalog.

  Oggetto `FederatedCatalog`. Una `FederatedCatalog` struttura che fa riferimento a un'entità esterna al AWS Glue Data Catalog, ad esempio un database Redshift.

- `Parameters`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa chiave, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Ogni valore è una stringa UTF-8, lunga non più di 512000 byte.

  Un array della mappa di coppie chiave-valore che definiscono i parametri e le proprietà del catalogo.

- `TargetRedshiftCatalog`: un oggetto TargetRedshiftCatalog.

  Un oggetto `TargetRedshiftCatalog` che descrive un catalogo di destinazione per il collegamento delle risorse.

- `CatalogProperties`: un oggetto CatalogProperties.

  Un oggetto `CatalogProperties` che specifica le proprietà di accesso al data lake e altre proprietà personalizzate.

- `CreateTableDefaultPermissions`: una matrice di oggetti PrincipalPermissions.

  Un array di oggetti `PrincipalPermissions`. Crea un set di autorizzazioni predefinite sulle tabelle per i principali. Usato da AWS Lake Formation. In genere dovrebbe essere impostato in modo esplicito come elenco vuoto.

- `CreateDatabaseDefaultPermissions`: una matrice di oggetti PrincipalPermissions.

  Un array di oggetti `PrincipalPermissions`. Crea un set di autorizzazioni predefinite sui database per i principali. Usato da AWS Lake Formation. In genere dovrebbe essere impostato in modo esplicito come elenco vuoto.

- `AllowFullTableExternalDataAccess`: stringa UTF-8 (valori validi: `True` | `False`).

  Consente ai motori di terze parti di accedere ai dati in Amazon S3 località registrate con Lake Formation.

## TargetRedshiftCatalog struttura

Una struttura che descrive un catalogo di destinazione per il collegamento delle risorse.

Campi

- `CatalogArn` - Obbligatorio: stringa UTF-8.

  Il nome della risorsa Amazon (ARN) della risorsa del catalogo.

## CatalogProperties struttura

Una struttura che specifica le proprietà di accesso al data lake e altre proprietà personalizzate.

Campi

- `DataLakeAccessProperties`: un oggetto DataLakeAccessProperties.

  Un `DataLakeAccessProperties` oggetto che specifica le proprietà per configurare l'accesso al data lake per la risorsa del catalogo nel AWS Glue Data Catalog.
- `IcebergOptimizationProperties`: un oggetto IcebergOptimizationProperties.

  Una struttura che specifica le proprietà di ottimizzazione della tabella Iceberg per il catalogo. Ciò include la configurazione per le operazioni di compattazione, conservazione ed eliminazione dei file orfani che possono essere applicate alle tabelle Iceberg in questo catalogo.
- `CustomProperties`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa chiave, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Ogni valore è una stringa UTF-8, lunga non più di 512000 byte.

  Proprietà chiave-valore aggiuntive per il catalogo, come le ottimizzazioni delle statistiche delle colonne.

## CatalogPropertiesOutput struttura

Attributi di proprietà che includono le proprietà di configurazione per la risorsa del catalogo.

Campi

- `DataLakeAccessProperties`: un oggetto DataLakeAccessPropertiesOutput.

  Un `DataLakeAccessProperties` oggetto con proprietà di input per configurare l'accesso al data lake per la risorsa del catalogo nel AWS Glue Data Catalog.

- `IcebergOptimizationProperties`: un oggetto IcebergOptimizationPropertiesOutput.

  Un oggetto `IcebergOptimizationPropertiesOutput` che specifica le impostazioni di ottimizzazione delle tabelle Iceberg per il catalogo, incluse le configurazioni per le operazioni di compattazione, conservazione ed eliminazione dei file orfani.

- `CustomProperties`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa chiave, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Ogni valore è una stringa UTF-8, lunga non più di 512000 byte.

  Proprietà chiave-valore aggiuntive per il catalogo, come le ottimizzazioni delle statistiche delle colonne.

## DataLakeAccessProperties struttura

Immetti le proprietà per configurare l'accesso al data lake per la risorsa del catalogo nel AWS Glue Data Catalog.

Campi

- `DataLakeAccess`: booleano.

  Attiva o disattiva l'accesso al data lake per le applicazioni Apache Spark che accedono ai database Amazon Redshift nel Data Catalog da qualsiasi motore non Redshift, come Amazon Athena, Amazon EMR o ETL. AWS Glue

- `DataTransferRole`: stringa UTF-8, corrispondente a Custom string pattern #53.

  Un ruolo che verrà assunto AWS Glue per trasferire i dati dello staging bucket durante una query. into/out

- `KmsKey`: stringa UTF-8.

Una chiave di crittografia utilizzata per il bucket di staging che verrà creato insieme al catalogo.

- `CatalogType`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Specifica un tipo di catalogo federato per la risorsa del catalogo nativa. Il tipo attualmente supportato è `aws:redshift`.

## IcebergOptimizationProperties struttura

Una struttura che specifica le proprietà di ottimizzazione delle tabelle Iceberg per il catalogo, incluse le configurazioni per le operazioni di compattazione, conservazione ed eliminazione dei file orfani.

Campi

- `RoleArn`: stringa UTF-8, corrispondente a Custom string pattern #53.

  Il nome della risorsa Amazon (ARN) del ruolo IAM che deve essere assunto per eseguire le operazioni di ottimizzazione delle tabelle Iceberg.

- `Compaction`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa chiave, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Ogni valore è una stringa UTF-8, lunga non più di 512000 byte.

  Una mappa di coppie chiave-valore che specificano i parametri di configurazione per le operazioni di compattazione delle tabelle Iceberg, le quali ottimizzano il layout dei file di dati per migliorare le prestazioni delle query.

- `Retention`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa chiave, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Ogni valore è una stringa UTF-8, lunga non più di 512000 byte.

  Una mappa di coppie chiave-valore che specificano i parametri di configurazione per le operazioni di conservazione delle tabelle Iceberg, le quali gestiscono il ciclo di vita degli snapshot delle tabelle per controllare i costi di archiviazione.

- `OrphanFileDeletion`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa chiave, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Ogni valore è una stringa UTF-8, lunga non più di 512000 byte.

  Una mappa di coppie chiave-valore che specificano i parametri di configurazione per le operazioni di eliminazione dei file orfani di Iceberg, i quali identificano e rimuovono i file a cui non fanno più riferimento i metadati della tabella.

## DataLakeAccessPropertiesOutput struttura

Le proprietà di output della configurazione di accesso al data lake per la risorsa del catalogo nel AWS Glue Data Catalog.

Campi

- `DataLakeAccess`: booleano.

  Attiva o disattiva l'accesso al data lake per le applicazioni Apache Spark che accedono ai database Amazon Redshift nel Catalogo Dati.
- `DataTransferRole`: stringa UTF-8, corrispondente a Custom string pattern #53.

  Un ruolo che verrà assunto AWS Glue per il trasferimento dei dati into/out dello staging bucket durante una query.
- `KmsKey`: stringa UTF-8.

  Una chiave di crittografia utilizzata per il bucket di staging che verrà creato insieme al catalogo.
- `ManagedWorkgroupName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del calcolo gestito di Redshift serverless creato per la risorsa del catalogo.
- `ManagedWorkgroupStatus`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Lo stato del calcolo gestito di Redshift serverless.
- `RedshiftDatabaseName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

Il nome predefinito della risorsa del database Redshift nel calcolo gestito.

- `StatusMessage`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Un messaggio che fornisce informazioni più dettagliate sullo stato del gruppo di lavoro gestito.
- `CatalogType`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Specifica un tipo di catalogo federato per la risorsa del catalogo nativa. Il tipo attualmente supportato è `aws:redshift`.

## IcebergOptimizationPropertiesOutput struttura

Una struttura che contiene le proprietà di output della configurazione di ottimizzazione delle tabelle Iceberg per la risorsa di catalogo nel AWS Glue Data Catalog.

Campi

- `RoleArn`: stringa UTF-8, corrispondente a Custom string pattern #53.

  Il nome della risorsa Amazon (ARN) del ruolo IAM utilizzato per eseguire le operazioni di ottimizzazione delle tabelle Iceberg.
- `Compaction`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa chiave, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Ogni valore è una stringa UTF-8, lunga non più di 512000 byte.

  Una mappa di coppie chiave-valore che specificano i parametri di configurazione per le operazioni di compattazione delle tabelle Iceberg, le quali ottimizzano il layout dei file di dati per migliorare le prestazioni delle query.
- `Retention`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa chiave, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Ogni valore è una stringa UTF-8, lunga non più di 512000 byte.

Una mappa di coppie chiave-valore che specificano i parametri di configurazione per le operazioni di conservazione delle tabelle Iceberg, le quali gestiscono il ciclo di vita degli snapshot delle tabelle per controllare i costi di archiviazione.

- `OrphanFileDeletion`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa chiave, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Ogni valore è una stringa UTF-8, lunga non più di 512000 byte.

  Una mappa di coppie chiave-valore che specificano i parametri di configurazione per le operazioni di eliminazione dei file orfani di Iceberg, i quali identificano e rimuovono i file a cui non fanno più riferimento i metadati della tabella.

- `LastUpdatedTime`: timestamp.

  Il timestamp relativo al momento dell'ultimo aggiornamento delle proprietà di Iceberg.

## FederatedCatalog struttura

Un catalogo che punta a un'entità esterna al AWS Glue Data Catalog.

Campi

- `Identifier`: stringa UTF-8, non inferiore a 1 o superiore a 512 byte di lunghezza, corrispondente a Single-line string pattern.

  Un identificatore univoco per il catalogo federato.

- `ConnectionName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome della connessione a un'origine dati esterna, ad esempio un catalogo federato di Redshift.

- `ConnectionType`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il tipo di connessione utilizzato per accedere al catalogo federato, che specifica il protocollo o il metodo per la connessione all'origine dati esterna.

## Operazioni

- CreateCatalog azione (Python: create_catalog)
- UpdateCatalog azione (Python: update_catalog)
- DeleteCatalog azione (Python: delete_catalog)
- GetCatalog azione (Python: get_catalog)
- GetCatalogs azione (Python: get_catalogs)

## CreateCatalog azione (Python: create_catalog)

Crea un nuovo catalogo nel Data Catalog. AWS Glue

Richiesta

- `Name`: obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 64 byte di lunghezza, corrispondente a Custom string pattern #25.

  Nome del catalogo da creare.
- `CatalogInput` - obbligatorio: un oggetto CatalogInput.

  Un oggetto `CatalogInput` che definisce i metadati per il catalogo.
- `Tags`: una matrice di mappe con coppie chiave-valore, non superiore alle 50 coppie.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Ogni valore è una stringa UTF-8, lunga non più di 256 byte.

  Una matrice di mappe con coppie chiave-valore, non superiore alle 50 coppie. Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza. Ogni valore è una stringa UTF-8, lunga non più di 256 byte. I tag assegnati al catalogo.

Risposta

- Nessun parametro di risposta.

Errori

- `InvalidInputException`

- `AlreadyExistsException`
- `ResourceNumberLimitExceededException`
- `InternalServiceException`
- `OperationTimeoutException`
- `GlueEncryptionException`
- `ConcurrentModificationException`
- `AccessDeniedException`
- `EntityNotFoundException`
- `FederatedResourceAlreadyExistsException`
- `FederationSourceException`

## UpdateCatalog azione (Python: update_catalog)

Aggiorna le proprietà di un catalogo esistente nel Data Catalog. AWS Glue

Richiesta

- `CatalogId` - Obbligatorio:: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern

  L'ID del catalogo.
- `CatalogInput` - obbligatorio: un oggetto CatalogInput.

  Un oggetto `CatalogInput` che specifica le nuove proprietà di un catalogo esistente.

Risposta

- Nessun parametro di risposta.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `InternalServiceException`

- `OperationTimeoutException`
- `GlueEncryptionException`
- `ConcurrentModificationException`
- `AccessDeniedException`
- `FederationSourceException`

## DeleteCatalog azione (Python: delete_catalog)

Rimuove il catalogo specificato dal AWS Glue Data Catalog.

Dopo aver completato questa operazione, non è più possibile accedere ai database, alle tabelle (e a tutte le versioni e partizioni delle tabelle che potrebbero appartenere alle tabelle) e alle funzioni definite dall'utente nel catalogo eliminato. AWS Glue elimina queste risorse «orfane» in modo asincrono in modo tempestivo, a discrezione del servizio.

Per garantire l'eliminazione immediata di tutte le risorse correlate, prima di chiamare l'operazione `DeleteCatalog`, utilizza `DeleteTableVersion` (o `BatchDeleteTableVersion`), `DeletePartition` (o `BatchDeletePartition`), `DeleteTable` (o `BatchDeleteTable`), `DeleteUserDefinedFunction` e `DeleteDatabase` per eliminare eventuali risorse che appartengono al catalogo.

Richiesta

- `CatalogId` - Obbligatorio:: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern

  L'ID del catalogo.

Risposta

- Nessun parametro di risposta.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `InternalServiceException`

- `OperationTimeoutException`
- `GlueEncryptionException`
- `ConcurrentModificationException`
- `AccessDeniedException`
- `FederationSourceException`

## GetCatalog azione (Python: get_catalog)

Il nome del catalogo da recuperare. Deve essere tutto in minuscolo.

Richiesta

- `CatalogId` - Obbligatorio:: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern

  L'ID del catalogo padre in cui si trova il catalogo. Se non ne viene fornito nessuno, per impostazione predefinita viene utilizzato il numero di AWS conto.

Risposta

- `Catalog`: un oggetto Catalogo.

  Oggetto `Catalog`. La definizione del catalogo specificato nel AWS Glue Data Catalog.

Errori

- `InvalidInputException`
- `EntityNotFoundException`
- `InternalServiceException`
- `OperationTimeoutException`
- `GlueEncryptionException`
- `AccessDeniedException`
- `EntityNotFoundException`
- `FederationSourceException`
- `FederationSourceRetryableException`

## GetCatalogs azione (Python: get_catalogs)

Recupera tutti i cataloghi definiti nel Catalogo dati AWS Glue . Per un caso d'uso di un catalogo federato di Redshift, questa operazione restituisce l'elenco dei cataloghi mappati ai database Redshift nel catalogo del namespace Redshift.

Richiesta

- `ParentCatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo padre in cui si trova il catalogo. Se non ne viene fornito nessuno, per impostazione predefinita viene utilizzato il numero di AWS conto.
- `NextToken`: stringa UTF-8.

  Un token di continuazione, se si tratta di una chiamata di continuazione.
- `MaxResults`: numero (intero), non inferiore a 1 o superiore a 1.000.

  Il numero massimo di cataloghi da restituire in una risposta.
- `Recursive`: booleano.

  Se elencare tutti i cataloghi nella gerarchia dei cataloghi, a partire da `ParentCatalogId`. L'impostazione predefinita è `false`. Quando `true`, tutti gli oggetti del catalogo nella gerarchia `ParentCatalogID` vengono enumerati nella risposta.
- `IncludeRoot`: booleano.

  Se elencare il catalogo predefinito nell'account e nella regione nella risposta. L'impostazione predefinita è `false`. Quando `true` e `ParentCatalogId = NULL | AWS Account ID`, tutti i cataloghi e il catalogo predefinito vengono enumerati nella risposta.

  Quando `ParentCatalogId` è diverso da nullo e questo attributo viene passato come `false` o `true`, viene generato un `InvalidInputException`.

Risposta

- `CatalogList` - obbligatorio: una matrice di oggetti Catalogo.

  Un array di oggetti `Catalog`. Un elenco di oggetti `Catalog` dal catalogo padre specificato.
- `NextToken`: stringa UTF-8.

Un token di continuazione per impaginare l'elenco restituito di token, restituiti se il segmento corrente dell'elenco non è l'ultimo.

Errori

- `InvalidInputException`
- `InternalServiceException`
- `OperationTimeoutException`
- `GlueEncryptionException`
- `AccessDeniedException`
- `EntityNotFoundException`
- `FederationSourceException`
- `FederationSourceRetryableException`

## API database

L'API Database descrive i tipi di dati del database e include l'API per creare, eliminare, localizzare, aggiornare ed elencare i database.

### Tipi di dati

- Struttura dei database
- DatabaseInput struttura
- PrincipalPermissions struttura
- DataLakePrincipal struttura
- DatabaseIdentifier struttura
- FederatedDatabase struttura

### Struttura dei database

L'oggetto `Database` rappresenta un raggruppamento logico di tabelle che potrebbero trovarsi in un metastore Hive o in un RDBMS.

Campi

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome del database. Per la compatibilità Hive, questo viene scritto in minuscolo durante la memorizzazione.

- `Description`: stringa di descrizione, non superiore a 2048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  Descrizione del database.

- `LocationUri`: uniform resource identifier (uri), non inferiore a 1 e non superiore a 1024 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  La posizione del database (per esempio, un percorso HDFS).

- `Parameters`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa chiave, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Ogni valore è una stringa UTF-8, lunga non più di 512000 byte.

  Queste coppie chiave-valore definiscono parametri e proprietà del database.

- `CreateTime`: timestamp.

  L'ora in cui è stato creato il database di metadati nel catalogo.

- `CreateTableDefaultPermissions`: una matrice di oggetti PrincipalPermissions.

  Crea un set di autorizzazioni predefinite per la tabella dei principal. Usato da AWS Lake Formation. Non utilizzato nel normale corso delle AWS Glue operazioni.

- `TargetDatabase`: un oggetto DatabaseIdentifier.

  Una struttura `DatabaseIdentifier` che descrive un database di destinazione per il collegamento delle risorse.

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo dati in cui risiede il database.

- `FederatedDatabase`: un oggetto FederatedDatabase.

  Una struttura `FederatedDatabase` che fa riferimento a un'entità esterna al AWS Glue Data Catalog.

## DatabaseInput struttura

Struttura utilizzata per la creazione o per l'aggiornamento di un database.

Campi

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome del database. Per la compatibilità Hive, questo viene scritto in minuscolo durante la memorizzazione.
- `Description`: stringa di descrizione, non superiore a 2048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  Descrizione del database.
- `LocationUri`: uniform resource identifier (uri), non inferiore a 1 e non superiore a 1024 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  La posizione del database (per esempio, un percorso HDFS).
- `Parameters`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa chiave, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Ogni valore è una stringa UTF-8, lunga non più di 512000 byte.

  Queste coppie chiave-valore definiscono parametri e proprietà del database.

  Queste coppie chiave-valore definiscono parametri e proprietà del database.
- `CreateTableDefaultPermissions`: una matrice di oggetti PrincipalPermissions.

  Crea un set di autorizzazioni predefinite per la tabella dei principal. Usato da AWS Lake Formation. Non utilizzato nel normale corso delle AWS Glue operazioni.
- `TargetDatabase`: un oggetto DatabaseIdentifier.

Una struttura `DatabaseIdentifier` che descrive un database di destinazione per il collegamento delle risorse.

- `FederatedDatabase`: un oggetto FederatedDatabase.

  Una struttura `FederatedDatabase` che fa riferimento a un'entità esterna al AWS Glue Data Catalog.

## PrincipalPermissions struttura

Autorizzazioni concesse a un principal.

Campi

- `Principal`: un oggetto DataLakePrincipal.

  Il principal a cui vengono concesse le autorizzazioni.
- `Permissions`: una matrice di stringhe UTF-8.

  Le autorizzazioni concesse al principal.

## DataLakePrincipal struttura

Il AWS Lake Formation preside.

Campi

- `DataLakePrincipalIdentifier`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza.

  Un identificatore per il AWS Lake Formation principale.

## DatabaseIdentifier struttura

Una struttura che descrive un database di destinazione per il collegamento delle risorse.

Campi

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

L'ID del catalogo dati in cui risiede il database.

- `DatabaseName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del database del catalogo.

- `Region`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  La regione della tabella di destinazione.

## FederatedDatabase struttura

Un database che punta a un'entità esterna al AWS Glue Data Catalog.

Campi

- `Identifier`: stringa UTF-8, non inferiore a 1 o superiore a 512 byte di lunghezza, corrispondente a Single-line string pattern.

  Un identificatore univoco per la tabella federata.

- `ConnectionName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome della connessione al metastore esterno.

- `ConnectionType`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il tipo di connessione utilizzato per accedere al database federato, ad esempio JDBC, ODBC o altri protocolli di connessione supportati.

## Operazioni

- CreateDatabase azione (Python: create_database)
- UpdateDatabase azione (Python: update_database)
- DeleteDatabase azione (Python: delete_database)
- GetDatabase azione (Python: get_database)

- GetDatabases azione (Python: get_databases)

## CreateDatabase azione (Python: create_database)

Crea un nuovo database in un catalogo di dati.

Richiesta

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo dati in cui creare il database. Se non ne viene fornito nessuno, per impostazione predefinita viene utilizzato l'ID AWS dell'account.
- `DatabaseInput` - obbligatorio: un oggetto DatabaseInput.

  I metadati per il database.
- `Tags`: una matrice di mappe con coppie chiave-valore, non superiore alle 50 coppie.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Ogni valore è una stringa UTF-8, lunga non più di 256 byte.

  I tag assegnati al database.

Risposta

- Nessun parametro di risposta.

Errori

- `InvalidInputException`
- `AlreadyExistsException`
- `ResourceNumberLimitExceededException`
- `InternalServiceException`
- `OperationTimeoutException`
- `GlueEncryptionException`
- `ConcurrentModificationException`

- `FederatedResourceAlreadyExistsException`
- `FederationSourceException`
- `FederationSourceRetryableException`

## UpdateDatabase azione (Python: update_database)

Aggiorna una definizione di database esistente in un catalogo dati.

Richiesta

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo dati in cui risiede il database dei metadati. Se non ne viene fornito nessuno, per impostazione predefinita viene utilizzato l'ID AWS dell'account.
- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del database da caricare nel catalogo. Per la compatibilità Hive, questo è scritto in caratteri minuscoli.
- `DatabaseInput` - obbligatorio: un oggetto DatabaseInput.

  Un oggetto `DatabaseInput` che specifica la nuova definizione del database di metadati nel catalogo.

Risposta

- Nessun parametro di risposta.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `InternalServiceException`
- `OperationTimeoutException`
- `GlueEncryptionException`

- `ConcurrentModificationException`
- `FederationSourceException`
- `FederationSourceRetryableException`
- `AlreadyExistsException`

## DeleteDatabase azione (Python: delete_database)

Rimuove un database specificato da un catalogo dati.

> **Note**
>
> Dopo aver completato questa operazione, non è più possibile accedere alle tabelle (e a tutte le versioni e partizioni delle tabelle che potrebbero appartenere alle tabelle) e alle funzioni definite dall'utente nel database eliminato. AWS Glue elimina queste risorse «orfane» in modo asincrono in modo tempestivo, a discrezione del servizio.
> Per garantire l'eliminazione immediata di tutte le risorse correlate, prima di chiamare `DeleteDatabase`, utilizza `DeleteTableVersion` o `BatchDeleteTableVersion`, `DeletePartition` o `BatchDeletePartition`, `DeleteUserDefinedFunction` e `DeleteTable` o `BatchDeleteTable` per eliminare eventuali risorse che appartengono al database.

Richiesta

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo dati in cui risiede il database. Se non ne viene fornito nessuno, per impostazione predefinita viene utilizzato l'ID AWS dell'account.
- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del database da eliminare. Per la compatibilità Hive, deve essere interamente in caratteri minuscoli.

Risposta

- Nessun parametro di risposta.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `InternalServiceException`
- `OperationTimeoutException`
- `ConcurrentModificationException`
- `FederationSourceException`
- `FederationSourceRetryableException`

## GetDatabase azione (Python: get_database)

Recupera la definizione di un database specificato.

Richiesta

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo dati in cui risiede il database. Se non ne viene fornito nessuno, per impostazione predefinita viene utilizzato l'ID AWS dell'account.

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome del database da ripristinare. Per la compatibilità Hive, deve essere interamente in caratteri minuscoli.

Risposta

- `Database`: un oggetto Database.

  La definizione del database specificato nel catalogo dati.

Errori

- `InvalidInputException`
- `EntityNotFoundException`
- `InternalServiceException`
- `OperationTimeoutException`
- `GlueEncryptionException`
- `FederationSourceException`
- `FederationSourceRetryableException`

## GetDatabases azione (Python: get_databases)

Recupera tutti i database definiti in un determinato catalogo dati.

Richiesta

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo dati da cui recuperare `Databases`. Se non ne viene fornito nessuno, per impostazione predefinita viene utilizzato l'ID AWS dell'account.
- `NextToken`: stringa UTF-8.

  Un token di continuazione, se si tratta di una chiamata di continuazione.
- `MaxResults`: numero (intero), non inferiore a 1 o superiore a 100.

  Il numero massimo di database da restituire in una risposta.
- `ResourceShareType`: stringa UTF-8 (valori validi: `FOREIGN` | `ALL` | `FEDERATED`).

  Consente di specificare che si desidera elencare i database condivisi con l'account. I valori consentiti sono `FEDERATED`, `FOREIGN` o `ALL`.
  - Se impostato su `FEDERATED`, elencherà i database federati (con riferimento a un'entità esterna) condivisi con l'account.
  - Se impostato su `FOREIGN`, elencherà i database condivisi con l'account.
  - Se impostato su `ALL`, elencherà i database condivisi con l'account, così come i database nell'account locale.

- `AttributesToGet`: una matrice di stringhe UTF-8.

  Specifica i campi del database restituiti dalla chiamata `GetDatabases`. Questo parametro non accetta un elenco vuoto. La richiesta deve includere `NAME`.

Risposta

- `DatabaseList` - obbligatorio: una matrice di oggetti Database.

  Un elenco di oggetti `Database` dal catalogo specificato.
- `NextToken`: stringa UTF-8.

  Un token di continuazione per impaginare l'elenco restituito di token, restituiti se il segmento corrente dell'elenco non è l'ultimo.

Errori

- `InvalidInputException`
- `InternalServiceException`
- `OperationTimeoutException`
- `GlueEncryptionException`
- `EntityNotFoundException`
- `FederationSourceException`
- `FederationSourceRetryableException`

## API Table

L'API Table descrive i tipi di dati e le operazioni associate alle tabelle.

### Tipi di dati

- Struttura della tabella
- TableInput struttura
- FederatedTable struttura
- Struttura delle colonne

- StorageDescriptor struttura
- SchemaReference struttura
- SerDeInfo struttura
- Struttura dell'ordine
- SkewedInfo struttura
- TableVersion struttura
- TableError struttura
- TableVersionError struttura
- SortCriterion struttura
- TableIdentifier struttura
- KeySchemaElement struttura
- PartitionIndex struttura
- PartitionIndexDescriptor struttura
- BackfillError struttura
- IcebergInput struttura
- OpenTableFormatInput struttura
- ViewDefinition struttura
- ViewDefinitionInput struttura
- ViewRepresentation struttura
- ViewRepresentationInput struttura
- UpdateOpenTableFormatInput struttura
- UpdateIcebergInput struttura
- CreateIcebergTableInput struttura
- UpdateIcebergTableInput struttura
- IcebergSortOrder struttura
- IcebergSortField struttura
- IcebergPartitionSpec struttura
- IcebergPartitionField struttura
- IcebergSchema struttura

- IcebergStructField struttura
- IcebergTableUpdate struttura
- AuditContext struttura

## Struttura della tabella

Rappresenta una raccolta di dati correlati organizzati in colonne e righe.

Campi

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome della tabella. Per la compatibilità Hive, deve essere interamente in caratteri minuscoli.
- `DatabaseName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome del database dei metadati in cui risiedono i metadati della tabella. Per la compatibilità Hive, deve essere interamente in caratteri minuscoli.
- `Description`: stringa di descrizione, non superiore a 2048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  Descrizione della tabella.
- `Owner`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il proprietario della tabella.
- `CreateTime`: timestamp.

  Ora della creazione della definizione della tabella nel catalogo dati.
- `UpdateTime`: timestamp.

  L'ultima volta che la tabella è stata aggiornata.
- `LastAccessTime`: timestamp.

  L'ultima volta che la tabella è stata consultata. Questo dato in genere viene fornito dall'HDFS e potrebbe non essere affidabile.

- `LastAnalyzedTime`: timestamp.

  L'ultima volta in cui sono state calcolate le statistiche di colonna per questa tabella.

- `Retention`: numero (intero), non superiore a Nessuno.

  Tempo di conservazione per questa tabella.

- `StorageDescriptor`: un oggetto StorageDescriptor.

  Un descrittore di archiviazione contenente informazioni sull'archiviazione fisica di questa tabella.

- `PartitionKeys`: una matrice di oggetti Colonna.

  Un elenco di colonne in base al quale la tabella è partizionata. Solo i tipi primitivi sono supportati come chiavi di partizione.

  Quando crei una tabella utilizzata da Amazon Athena e non specifichi una `partitionKeys`, è necessario almeno impostare il valore di `partitionKeys` su un elenco vuoto. Ad esempio:

  `"PartitionKeys": []`

- `ViewOriginalText`: stringa UTF-8, non superiore a 409600 byte di lunghezza.

  Incluso per la compatibilità con Apache Hive. Non utilizzato nel normale corso delle AWS Glue operazioni. Se la tabella è una`VIRTUAL_VIEW`, una determinata Athena configurazione è codificata in base64.

- `ViewExpandedText`: stringa UTF-8, non superiore a 409600 byte di lunghezza.

  Incluso per la compatibilità con Apache Hive. Non utilizzato nel normale corso delle operazioni. AWS Glue

- `TableType`: stringa UTF-8, non superiore a 255 byte di lunghezza.

  Il tipo di tabella. AWS Glue creerà tabelle con il `EXTERNAL_TABLE` tipo. Altri servizi, ad esempio Athena, possono creare tabelle con tipi di tabelle aggiuntivi.

  AWS Glue tipi di tabelle correlati:

  EXTERNAL_TABLE

  Attributo compatibile con Hive. Indica una tabella non gestita da Hive.

  GOVERNED

  Usato da AWS Lake Formation. Il AWS Glue Data Catalog capisce`GOVERNED`.

- `Parameters`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa chiave, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Ogni valore è una stringa UTF-8, lunga non più di 512000 byte.

  Queste coppie chiave-valore definiscono proprietà associate a questa tabella.

- `CreatedBy`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Persona o entità che ha creato la tabella.

- `IsRegisteredWithLakeFormation`: booleano.

  Indica se la tabella è stata registrata con AWS Lake Formation.

- `TargetTable`: un oggetto TableIdentifier.

  Una struttura `TableIdentifier` che descrive una tabella di destinazione per il collegamento delle risorse.

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo dati in cui risiede la tabella.

- `VersionId`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID della versione della tabella.

- `FederatedTable`: un oggetto FederatedTable.

  Una struttura `FederatedTable` che fa riferimento a un'entità esterna al AWS Glue Data Catalog.

- `ViewDefinition`: un oggetto ViewDefinition.

  Una struttura che contiene tutte le informazioni che definiscono la vista, compresi il dialetto o i dialetti della vista, e la query.

- `IsMultiDialectView`: booleano.

  Specifica se la vista supporta i dialetti SQL di uno o più motori di query diversi e può quindi essere letta da tali motori.

## TableInput struttura

Una struttura utilizzata per definire una tabella.

Campi

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome della tabella. Per la compatibilità Hive, questo viene scritto in minuscolo durante la memorizzazione.
- `Description`: stringa di descrizione, non superiore a 2048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  Descrizione della tabella.
- `Owner`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il proprietario della tabella. Incluso per la compatibilità con Apache Hive. Non utilizzato nel normale corso delle AWS Glue operazioni.
- `LastAccessTime`: timestamp.

  L'ultima volta che la tabella è stata consultata.
- `LastAnalyzedTime`: timestamp.

  L'ultima volta in cui sono state calcolate le statistiche di colonna per questa tabella.
- `Retention`: numero (intero), non superiore a Nessuno.

  Tempo di conservazione per questa tabella.
- `StorageDescriptor`: un oggetto StorageDescriptor.

  Un descrittore di archiviazione contenente informazioni sull'archiviazione fisica di questa tabella.
- `PartitionKeys`: una matrice di oggetti Colonna.

  Un elenco di colonne in base al quale la tabella è partizionata. Solo i tipi primitivi sono supportati come chiavi di partizione.

  Quando crei una tabella utilizzata da Amazon Athena e non specifichi una `partitionKeys`, è necessario almeno impostare il valore di `partitionKeys` su un elenco vuoto. Ad esempio:

```
"PartitionKeys": []
```

- `ViewOriginalText`: stringa UTF-8, non superiore a 409600 byte di lunghezza.

  Incluso per la compatibilità con Apache Hive. Non utilizzato nel normale corso delle AWS Glue operazioni. Se la tabella è una`VIRTUAL_VIEW`, una determinata Athena configurazione è codificata in base64.

- `ViewExpandedText`: stringa UTF-8, non superiore a 409600 byte di lunghezza.

  Incluso per la compatibilità con Apache Hive. Non utilizzato nel normale corso delle operazioni. AWS Glue

- `TableType`: stringa UTF-8, non superiore a 255 byte di lunghezza.

  Il tipo di tabella. AWS Glue creerà tabelle con il `EXTERNAL_TABLE` tipo. Altri servizi, ad esempio Athena, possono creare tabelle con tipi di tabelle aggiuntivi.

  AWS Glue tipi di tabelle correlati:

  EXTERNAL_TABLE

  Attributo compatibile con Hive. Indica una tabella non gestita da Hive.

  GOVERNED

  Usato da AWS Lake Formation. Il AWS Glue Data Catalog capisce`GOVERNED`.

- `Parameters`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa chiave, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Ogni valore è una stringa UTF-8, lunga non più di 512000 byte.

  Queste coppie chiave-valore definiscono proprietà associate a questa tabella.

- `TargetTable`: un oggetto TableIdentifier.

  Una struttura `TableIdentifier` che descrive una tabella di destinazione per il collegamento delle risorse.

- `ViewDefinition`: un oggetto ViewDefinitionInput.

  Una struttura che contiene tutte le informazioni che definiscono la vista, compresi il dialetto o i dialetti della vista, e la query.

## FederatedTable struttura

Una tabella che punta a un'entità esterna al AWS Glue Data Catalog.

Campi

- `Identifier`: stringa UTF-8, non inferiore a 1 o superiore a 512 byte di lunghezza, corrispondente a Single-line string pattern.

  Un identificatore univoco per la tabella federata.
- `DatabaseIdentifier`: stringa UTF-8, non inferiore a 1 o superiore a 512 byte di lunghezza, corrispondente a Single-line string pattern.

  Un identificatore univoco per la tabella federata.
- `ConnectionName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome della connessione al metastore esterno.
- `ConnectionType`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il tipo di connessione utilizzato per accedere alla tabella federata, che specifica il protocollo o il metodo per la connessione all'origine dati esterna.

## Struttura delle colonne

Una colonna in una `Table`.

Campi

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome della `Column`.
- `Type`: stringa UTF-8, non superiore a 131072 byte di lunghezza, corrispondente a Single-line string pattern.

  Il tipo di dati di `Column`.

- `Comment`: stringa di commento, non superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Commento con testo in formato libero.

- `Parameters`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa chiave, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Ogni valore è una stringa UTF-8, lunga non più di 512000 byte.

  Queste coppie chiave-valore definiscono proprietà associate alla colonna.

## StorageDescriptor struttura

Descrive lo storage fisico dei dati della tabella.

Campi

- `Columns`: una matrice di oggetti Colonna.

  Un elenco delle `Columns` nella tabella.

- `Location`: stringa di posizione, non superiore a 2056 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  La posizione fisica della tabella. Per default utilizza il formato della posizione del warehouse, seguita dalla posizione del database nel warehouse, seguita dal nome della tabella.

- `AdditionalLocations`: una matrice di stringhe UTF-8.

  Un elenco di posizioni che puntano al percorso in cui si trova una tabella Delta.

- `InputFormat`: stringa di formato, non superiore a 128 byte di lunghezza, corrispondente a Single-line string pattern.

  Il formato di input: `SequenceFileInputFormat` (binario) o `TextInputFormat` o un formato personalizzato.

- `OutputFormat`: stringa di formato, non superiore a 128 byte di lunghezza, corrispondente a Single-line string pattern.

Il formato di output: `SequenceFileOutputFormat` (binario) o `IgnoreKeyTextOutputFormat` o un formato personalizzato.

- `Compressed`: booleano.

  `True` se i dati nella tabella sono compressi, in caso contrario `False`.

- `NumberOfBuckets`: numero (intero).

  Deve essere specificato se la tabella contiene una qualsiasi colonna di dimensione.

- `SerdeInfo`: un oggetto SerDeInfo.

  Le informazioni serialization/deserialization (SerDe).

- `BucketColumns`: una matrice di stringhe UTF-8.

  Un elenco di colonne per il raggruppamento del reducer, colonne di clustering, colonne di bucketing nella tabella.

- `SortColumns`: una matrice di oggetti Ordine.

  Un elenco specificando l'ordine di ciascuna bucket nella tabella.

- `Parameters`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa chiave, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Ogni valore è una stringa UTF-8, lunga non più di 512000 byte.

  Le proprietà fornite dall'utente nel modulo chiave-valore.

- `SkewedInfo`: un oggetto SkewedInfo.

  Informazioni sui valori che appaiono di frequente in una colonna (valori disallineati).

- `StoredAsSubDirectories`: booleano.

  `True` se i dati nella tabella sono archiviati nelle sottodirectory, in caso contrario `False`.

- `SchemaReference`: un oggetto SchemaReference.

  Un oggetto che fa riferimento a uno schema memorizzato nel registro degli AWS Glue schemi.

  Quando crei una tabella, puoi passare un elenco vuoto di colonne per lo schema e utilizzare invece un riferimento allo schema.

## SchemaReference struttura

Un oggetto che fa riferimento a uno schema memorizzato nel registro degli AWS Glue schemi.

Campi

- `SchemaId`: un oggetto SchemaId.

  Una struttura che contiene campi di identità dello schema. Deve essere fornito questo o `SchemaVersionId`.
- `SchemaVersionId`: stringa UTF-8, non inferiore a 36 o superiore a 36 byte di lunghezza, corrispondente a Custom string pattern #45.

  L'ID univoco assegnato a una versione dello schema. Deve essere fornito questo o `SchemaId`.
- `SchemaVersionNumber`: numero (intero), non inferiore a 1 o superiore a 100000.

  Il numero di versione dello schema.

## SerDeInfo struttura

Informazioni su un serialization/deserialization programma (SerDe) che funge da estrattore e caricatore.

Campi

- `Name`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome del. SerDe
- `SerializationLibrary`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Di solito la classe che implementa. SerDe Un esempio è `org.apache.hadoop.hive.serde2.columnar.ColumnarSerDe`.
- `Parameters`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa chiave, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Ogni valore è una stringa UTF-8, lunga non più di 512000 byte.

Queste coppie chiave-valore definiscono i parametri di inizializzazione per. SerDe

## Struttura dell'ordine

Specifica l'ordine di una colonna ordinata.

Campi

- `Column` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome della colonna.
- `SortOrder` - obbligatorio: numero (intero), non superiore a 1.

  Indica che la colonna è in ordine crescente (`== 1`) o in ordine decrescente (`==0`).

## SkewedInfo struttura

Specifica i valori disallineati in una tabella. I valori disallineati sono quelli che si verificano con una frequenza molto elevata.

Campi

- `SkewedColumnNames`: una matrice di stringhe UTF-8.

  Un elenco di nomi delle colonne contenenti i valori disallineati.
- `SkewedColumnValues`: una matrice di stringhe UTF-8.

  Un elenco di valori che appaiono così frequentemente da poter essere considerati disallineati.
- `SkewedColumnValueLocationMaps`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8.

  Ogni valore è una stringa UTF-8.

  Una mappatura di valori disallineati per le colonne che li contengono.

## TableVersion struttura

Specifica una versione di una tabella.

Campi

- `Table`: un oggetto Tabella.

  La tabella in questione

- `VersionId`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il valore ID identificativo di questa versione della tabella. `VersionId` è una rappresentazione di stringa di un numero intero. Ogni versione viene incrementata di 1.

## TableError struttura

Un record di errore per le operazioni di tabella.

Campi

- `TableName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome della tabella. Per la compatibilità Hive, deve essere interamente in caratteri minuscoli.

- `ErrorDetail`: un oggetto ErrorDetail.

  I dettagli sull'errore.

## TableVersionError struttura

Un record di errore per le operazioni table-version.

Campi

- `TableName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome della tabella in questione.

- `VersionId`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il valore ID della versione in questione. `VersionID` è una rappresentazione di stringa di un numero intero. Ogni versione viene incrementata di 1.

- `ErrorDetail`: un oggetto ErrorDetail.

  I dettagli sull'errore.

## SortCriterion struttura

Specifica un campo in base al quale ordinare e un ordinamento.

Campi

- `FieldName`: stringa Value, non inferiore a 1 o superiore a 1024 byte di lunghezza.

  Il nome del campo in base al quale eseguire l'ordinamento.

- `Sort`: stringa UTF-8 (valori validi: `ASC="ASCENDING"` | `DESC="DESCENDING"`).

  Un ordinamento crescente o decrescente.

## TableIdentifier struttura

Una struttura che descrive una tabella di destinazione per il collegamento delle risorse.

Campi

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo dati in cui risiede la tabella.

- `DatabaseName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del database di catalogo che contiene la tabella di destinazione.

- `Name`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

Il nome della tabella di destinazione.

- `Region`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  La regione della tabella di destinazione.

## KeySchemaElement struttura

Una coppia di chiavi di partizione costituita da un nome e un tipo.

Campi

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome di una chiave di partizione.
- `Type` - obbligatorio: stringa UTF-8, non superiore a 131072 byte di lunghezza, corrispondente a Single-line string pattern.

  Il tipo di una chiave di partizione.

## PartitionIndex struttura

Una struttura per un indice della partizione.

Campi

- `Keys` - obbligatorio: una matrice di stringhe UTF-8, almeno 1 stringa.

  Le chiavi per l'indice della partizione.
- `IndexName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome dell'indice della partizione.

## PartitionIndexDescriptor struttura

Un descrittore per un indice della partizione in una tabella.

Campi

- `IndexName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome dell'indice della partizione.
- `Keys` - obbligatorio: una matrice di oggetti KeySchemaElement, almeno 1 struttura.

  Un elenco di una o più chiavi, come strutture `KeySchemaElement`, per l'indice della partizione.
- `IndexStatus` - obbligatorio: stringa UTF-8 (valori validi: `CREATING` | `ACTIVE` | `DELETING` | `FAILED`).

  Lo stato dell'indice della partizione.

  I possibili stati sono:
  - CREATING: l'indice è in fase di creazione. Quando un indice è in uno stato CREATING, l'indice o la relativa tabella non possono essere eliminati.
  - ACTIVE: la creazione dell'indice ha esito positivo.
  - FAILED: la creazione dell'indice non riesce.
  - DELETING: l'indice viene eliminato dall'elenco degli indici.
- `BackfillErrors`: una matrice di oggetti BackfillError.

  Un elenco degli errori che possono verificarsi durante la registrazione degli indici delle partizioni per una tabella esistente.

## BackfillError struttura

Un elenco degli errori che possono verificarsi durante la registrazione degli indici delle partizioni per una tabella esistente.

Questi errori forniscono i dettagli sul motivo per cui una registrazione dell'indice non è riuscita e forniscono un numero limitato di partizioni nella risposta, in modo da poter correggere le partizioni in errore e provare a registrare nuovamente l'indice. La serie più comune di errori che possono verificarsi sono classificati come segue:

- EncryptedPartitionError: Le partizioni sono criptate.
- InvalidPartitionTypeDataError: il valore della partizione non corrisponde al tipo di dati per quella colonna di partizione.

- MissingPartitionValueError: le partizioni sono crittografate.
- UnsupportedPartitionCharacterError: I caratteri all'interno del valore della partizione non sono supportati. Ad esempio: U+0000, U+0001, U+0002.
- InternalError: Qualsiasi errore che non appartiene ad altri codici di errore.

Campi

- `Code`: stringa UTF-8 (valori validi: `ENCRYPTED_PARTITION_ERROR` | `INTERNAL_ERROR` | `INVALID_PARTITION_TYPE_DATA_ERROR` | `MISSING_PARTITION_VALUE_ERROR` | `UNSUPPORTED_PARTITION_CHARACTER_ERROR`).

  Il codice di errore per un errore che si è verificato durante la registrazione degli indici delle partizioni per una tabella esistente.
- `Partitions`: una matrice di oggetti PartitionValueList.

  Un elenco di un numero limitato di partizioni nella risposta.

## IcebergInput struttura

Una struttura che definisce una tabella di metadati di Apache Iceberg da creare nel catalogo.

Campi

- `MetadataOperation` - obbligatorio: stringa UTF-8 (valori validi: `CREATE`).

  Un'operazione sui metadati richiesta. Questa opzione può essere impostata solo su `CREATE`.
- `Version`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  La versione della tabella per le tabelle Iceberg. L'impostazione predefinita è 2.
- `CreateIcebergTableInput`: un oggetto CreateIcebergTableInput.

  I parametri di configurazione necessari per creare una nuova tabella Iceberg nel AWS Glue Data Catalog, incluse le proprietà della tabella e le specifiche dei metadati.

## OpenTableFormatInput struttura

Una struttura che rappresenta una tabella in formato aperto.

Campi

- `IcebergInput`: un oggetto IcebergInput.

  Specifica una struttura `IcebergInput` che definisce una tabella di metadati di Apache Iceberg.

## ViewDefinition struttura

Una struttura contenente i dettagli delle rappresentazioni.

Campi

- `IsProtected`: booleano.

  È possibile impostare questo flag come true per ordinare al motore di non inviare le operazioni fornite dall'utente nel piano logico della vista durante la pianificazione delle query. Tuttavia, l'impostazione di questo flag non garantisce che il motore rispetti l'ordine. Consultare la documentazione del motore per comprendere le eventuali garanzie fornite.
- `Definer`: stringa UTF-8, non inferiore a 20 o superiore a 2048 byte di lunghezza, corrispondente a Single-line string pattern.

  Il definitore in una vista in SQL.
- `SubObjects`: una matrice di stringhe UTF-8, non superiore a 10 stringhe.

  Un elenco di tabelle Amazon Resource Names (ARNs).
- `Representations`: una matrice di oggetti ViewRepresentation, non inferiore a 1 o superiore a 1.000 strutture.

  Un elenco di rappresentazioni.

## ViewDefinitionInput struttura

Una struttura contenente i dettagli per la creazione o l'aggiornamento di una AWS Glue vista.

Campi

- `IsProtected`: booleano.

  È possibile impostare questo flag come true per ordinare al motore di non inviare le operazioni fornite dall'utente nel piano logico della vista durante la pianificazione delle query. Tuttavia,

l'impostazione di questo flag non garantisce che il motore rispetti l'ordine. Consultare la documentazione del motore per comprendere le eventuali garanzie fornite.

- `Definer`: stringa UTF-8, non inferiore a 20 o superiore a 2048 byte di lunghezza, corrispondente a Single-line string pattern.

  Il definitore in una vista in SQL.

- `Representations`— Una serie di ViewRepresentationInput oggetti, non meno di 1 o più di 10 strutture.

  Un elenco di strutture contenente il dialetto della vista e la query che la definisce.

- `SubObjects`: una matrice di stringhe UTF-8, non superiore a 10 stringhe.

  Un elenco di tabelle di base ARNs che costituiscono la vista.

## ViewRepresentation struttura

Una struttura contenente il dialetto della vista e la query che la definisce.

Campi

- `Dialect`: stringa UTF-8 (valori validi: `REDSHIFT` | `ATHENA` | `SPARK`).

  Il dialetto del motore di query.

- `DialectVersion`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza.

  La versione del dialetto del motore di query. Ad esempio, 3.0.0.

- `ViewOriginalText`: stringa UTF-8, non superiore a 409600 byte di lunghezza.

  La query `SELECT` fornita dal cliente durante `CREATE VIEW DDL`. Questo codice SQL non viene utilizzato durante la query su una vista (viene invece utilizzato `ViewExpandedText`). `ViewOriginalText` viene usato in casi come `SHOW CREATE VIEW` in cui gli utenti desiderano vedere il comando DDL originale che ha creato la vista.

- `ViewExpandedText`: stringa UTF-8, non superiore a 409600 byte di lunghezza.

  L'SQL esteso per la vista. Questo SQL viene utilizzato dai motori durante l'elaborazione di una query su una vista. I motori possono eseguire operazioni durante la creazione della vista per trasformare `ViewOriginalText` in `ViewExpandedText`. Esempio:

  - Identificatori completi: `SELECT * from table1 -> SELECT * from db1.table1`

- `ValidationConnection`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome della connessione da utilizzare per convalidare la rappresentazione specifica della vista.
- `IsStale`: booleano.

  I dialetti contrassegnati come obsoleti non sono più validi e devono essere aggiornati prima che sia possibile eseguire query nei rispettivi motori di query.

## ViewRepresentationInput struttura

Una struttura contenente i dettagli di una rappresentazione per aggiornare o creare una vista Lake Formation.

Campi

- `Dialect`: stringa UTF-8 (valori validi: `REDSHIFT` | `ATHENA` | `SPARK`).

  Un parametro che specifica il tipo di motore di una rappresentazione specifica.
- `DialectVersion`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza.

  Un parametro che specifica la versione del motore di una rappresentazione specifica.
- `ViewOriginalText`: stringa UTF-8, non superiore a 409600 byte di lunghezza.

  Una stringa che rappresenta la query SQL originale che descrive la vista.
- `ValidationConnection`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome della connessione da utilizzare per convalidare la rappresentazione specifica della vista.
- `ViewExpandedText`: stringa UTF-8, non superiore a 409600 byte di lunghezza.

  Una stringa che rappresenta la query SQL che descrive la vista con risorsa espansa ARNs

## UpdateOpenTableFormatInput struttura

Parametri di input per l'aggiornamento di tabelle aperte in AWS Glue Data Catalog, che fungono da wrapper per operazioni di aggiornamento specifiche di formato come Apache Iceberg.

Campi

- `UpdateIcebergInput`: un oggetto UpdateIcebergInput.

  Parametri di aggiornamento specifici di Apache Iceberg che definiscono le modifiche da apportare alla tabella, tra cui modifiche allo schema, specifiche di partizione e proprietà della tabella.

## UpdateIcebergInput struttura

Parametri di input specifici per l'aggiornamento delle tabelle Apache Iceberg in AWS Glue Data Catalog, contenenti le operazioni di aggiornamento da applicare a una tabella Iceberg esistente.

Campi

- `UpdateIcebergTableInput` - obbligatorio: oggetto UpdateIcebergTableInput.

  Le operazioni di aggiornamento specifiche da applicare alla tabella Iceberg, contenenti un elenco di aggiornamenti che definiscono il nuovo stato della tabella, tra cui schema, partizioni e proprietà.

## CreateIcebergTableInput struttura

I parametri di configurazione necessari per creare una nuova tabella Iceberg nel AWS Glue Data Catalog, incluse le proprietà della tabella e le specifiche dei metadati.

Campi

- `Location` - obbligatorio, stringa di posizione, non superiore a 2056 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  La posizione S3 in cui saranno archiviati i dati della tabella Iceberg.
- `Schema` - obbligatorio: oggetto IcebergSchema.

  La definizione dello schema che specifica la struttura, i tipi di campo e i metadati della tabella Iceberg.
- `PartitionSpec`: un oggetto IcebergPartitionSpec.

  La specifica di partizionamento che definisce come saranno organizzati e partizionati i dati della tabella Iceberg per garantire prestazioni di query ottimali.
- `WriteOrder`: un oggetto IcebergSortOrder.

La specifica di ordinamento che definisce come devono essere ordinati i dati all'interno di ogni partizione per ottimizzare le prestazioni di query.

- `Properties` –

  Coppie chiave-valore di proprietà aggiuntive della tabella e impostazioni di configurazione della tabella Iceberg.

## UpdateIcebergTableInput struttura

Contiene le operazioni di aggiornamento da applicare a una tabella Iceberg esistente in AWS Glue Data Catalog, definendo il nuovo stato dei metadati della tabella.

Campi

- `Updates` - obbligatorio: una matrice di oggetti IcebergTableUpdate.

  L'elenco delle operazioni di aggiornamento della tabella che specificano le modifiche da apportare alla tabella Iceberg, tra cui modifiche dello schema, specifiche di partizione e proprietà della tabella.

## IcebergSortOrder struttura

Definisce la specifica di ordinamento di una tabella Iceberg, che determina come devono essere ordinati i dati all'interno delle partizioni per ottimizzare le prestazioni di query.

Campi

- `OrderId` - obbligatorio: numero (intero).

  L'identificatore univoco per questa specifica di ordinamento all'interno dei metadati della tabella Iceberg.

- `Fields` - obbligatorio: una matrice di oggetti IcebergSortField.

  L'elenco dei campi e delle relative direzioni di ordinamento che definiscono i criteri di ordinamento per i dati della tabella Iceberg.

## IcebergSortField struttura

Definisce un singolo campo all'interno di una specifica di ordinamento Iceberg, compresi il campo di origine, la trasformazione, la direzione di ordinamento e l'ordine dei valori null.

Campi

- `SourceId` - obbligatorio: numero (intero).

  L'identificatore del campo di origine dallo schema della tabella su cui è basato questo campo di ordinamento.
- `Transform` - obbligatorio: stringa UTF-8.

  La funzione di trasformazione applicata al campo di origine prima dell'ordinamento, come identità, bucket o troncamento.
- `Direction` - obbligatorio: stringa UTF-8 (valori validi: `asc="ASC"` | `desc="DESC"`).

  La direzione di ordinamento di questo campo, ascendente o discendente.
- `NullOrder` - obbligatorio: stringa UTF-8 (valori validi: `nulls-first="NULLS_FIRST"` | `nulls-last="NULLS_LAST"`).

  Il comportamento di ordinamento dei valori null in questo campo, che specifica se tali valori devono comparire per primi o per ultimi nell'ordinamento.

## IcebergPartitionSpec struttura

Definisce la specifica di partizionamento di una tabella Iceberg, che determina come saranno organizzati e partizionati i dati della tabella Iceberg per garantire prestazioni di query ottimali.

Campi

- `Fields` - obbligatorio: una matrice di oggetti IcebergPartitionField.

  L'elenco dei campi di partizione che definiscono come devono essere partizionati i dati della tabella, compresi i campi di origine e le relative trasformazioni.
- `SpecId`: numero (intero).

  L'identificatore univoco per questa specifica di partizione all'interno della cronologia dei metadati della tabella Iceberg.

## IcebergPartitionField struttura

Definisce un singolo campo di partizione all'interno di una specifica di partizione Iceberg, compresi il campo di origine, la funzione di trasformazione, il nome della partizione e l'identificatore univoco.

Campi

- `SourceId` - obbligatorio: numero (intero).

  L'identificatore del campo di origine dallo schema della tabella su cui è basato questo campo di partizione.
- `Transform` - obbligatorio: stringa UTF-8.

  La funzione di trasformazione applicata al campo di origine per creare la partizione, come identità, bucket, troncamento, anno, mese, giorno oppure ora.
- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 1024 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del campo di partizione come verrà visualizzato nella struttura della tabella partizionata.
- `FieldId`: numero (intero).

  L'identificatore univoco assegnato a questo campo di partizione all'interno della specifica di partizione della tabella Iceberg.

## IcebergSchema struttura

Definisce la struttura dello schema di una tabella Iceberg, compresi le definizioni dei campi, i tipi di dati e i metadati dello schema.

Campi

- `SchemaId`: numero (intero).

  L'identificatore univoco di questa versione dello schema all'interno della storia di evoluzione dello schema della tabella Iceberg.
- `IdentifierFieldIds`: una matrice di numeri interi a 32 bit firmati.

  L'elenco di identificatori dei campi che identificano in modo univoco i record nella tabella, utilizzati nelle operazioni e nella deduplicazione a livello di riga.

- `Type`: stringa UTF-8 (valori validi: `struct="STRUCT"`).

  Il tipo di root della struttura dello schema, generalmente "struct" per gli schemi delle tabelle Iceberg.

- `Fields` - obbligatorio: una matrice di oggetti IcebergStructField.

  L'elenco delle definizioni dei campi che costituiscono lo schema della tabella, tra cui nomi e tipi di campi e metadati.

## IcebergStructField struttura

Definisce un singolo campo all'interno dello schema di una tabella Iceberg, compresi l'identificatore, il nome, il tipo di dati, la possibilità di ammettere valori null e la documentazione correlati.

Campi

- `Id` - obbligatorio: numero (intero).

  L'identificatore univoco assegnato a questo campo all'interno dello schema della tabella Iceberg, utilizzato per l'evoluzione dello schema e il monitoraggio dei campi.

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 1024 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del campo come visualizzato nello schema della tabella e nelle operazioni di query.

- `Type` -obbligatorio, una struttura vuota chiamata `IcebergDocument`.

  La definizione del tipo di dati di questo campo, che specifica la struttura e il formato dei dati in esso contenuti.

- `Required` - obbligatorio: booleano.

  Indica se questo campo è obbligatorio (non ammette valori null) o facoltativo (ammette valori null) nello schema della tabella.

- `Doc`: stringa di commento, non superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Documentazione o descrizione in formato testuale facoltative che forniscono contesto aggiuntivo sullo scopo e sull'utilizzo di questo campo.

## IcebergTableUpdate struttura

Definisce un set completo di aggiornamenti da applicare a una tabella Iceberg, tra cui modifiche relative a schema, partizione, ordinamento, posizione e proprietà.

Campi

- `Schema` - obbligatorio: oggetto IcebergSchema.

  La definizione dello schema aggiornata per la tabella Iceberg, che specifica eventuali modifiche alla struttura dei campi, ai tipi di dati o ai metadati dello schema.
- `PartitionSpec`: un oggetto IcebergPartitionSpec.

  La specifica di partizionamento aggiornata che definisce il modo in cui devono essere riorganizzati e partizionati i dati della tabella.
- `SortOrder`: un oggetto IcebergSortOrder.

  La specifica di ordinamento aggiornata che definisce come devono essere ordinati i dati all'interno delle partizioni per garantire prestazioni di query ottimali.
- `Location` - obbligatorio, stringa di posizione, non superiore a 2056 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  La posizione S3 aggiornata in cui saranno archiviati i dati della tabella Iceberg.
- `Properties` –

  Le coppie chiave-valore aggiornate di proprietà della tabella e le impostazioni di configurazione della tabella Iceberg.

## AuditContext struttura

Una struttura contenente il contesto Lake Formation di audit.

Campi

- `AdditionalAuditContext`— Stringa UTF-8, lunga non più di 2048 byte.

  Una stringa contenente le informazioni aggiuntive sul contesto di controllo.
- `RequestedColumns`: una matrice di stringhe UTF-8.

  Le colonne richieste per l'audit.

- `AllColumnsRequested`: booleano.

  Richiesta di revisione per tutte le colonne.

## Operazioni

- CreateTable azione (Python: create_table)
- UpdateTable azione (Python: update_table)
- DeleteTable azione (Python: delete_table)
- BatchDeleteTable azione (Python: batch_delete_table)
- GetTable azione (Python: get_table)
- GetTables azione (Python: get_tables)
- GetTableVersion azione (Python: get_table_version)
- GetTableVersions azione (Python: get_table_versions)
- DeleteTableVersion azione (Python: delete_table_version)
- BatchDeleteTableVersion azione (Python: batch_delete_table_version)
- SearchTables azione (Python: search_tables)
- GetPartitionIndexes azione (Python: get_partition_indexes)
- CreatePartitionIndex azione (Python: create_partition_index)
- DeletePartitionIndex azione (Python: delete_partition_index)
- GetColumnStatisticsForTable azione (Python: get_column_statistics_for_table)
- UpdateColumnStatisticsForTable azione (Python: update_column_statistics_for_table)
- DeleteColumnStatisticsForTable azione (Python: delete_column_statistics_for_table)

## CreateTable azione (Python: create_table)

Crea una nuova definizione di tabella nel catalogo dati.

Richiesta

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo dati in cui creare la `Table`. Se non ne viene fornito nessuno, per impostazione predefinita viene utilizzato l'ID AWS dell'account.

- `DatabaseName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il database del catalogo in cui creare la nuova tabella. Per la compatibilità Hive, questo nome è scritto interamente in caratteri minuscoli.

- `Name`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'identificatore univoco per la tabella all'interno del database specificato che verrà creato nel AWS Glue Data Catalog.

- `TableInput`: un oggetto TableInput.

  L'oggetto `TableInput` che definisce la tabella di metadati da creare nel catalogo.

- `PartitionIndexes`: una matrice di oggetti PartitionIndex, non superiore a 3 strutture.

  Un elenco di indici delle partizioni, `PartitionIndex` strutture, da creare nella tabella.

- `TransactionId`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Custom string pattern #44.

  L'ID della transazione.

- `OpenTableFormatInput`: un oggetto OpenTableFormatInput.

  Specifica una struttura `OpenTableFormatInput` durante la creazione di una tabella in formato aperto.

Risposta

- Nessun parametro di risposta.

Errori

- `AlreadyExistsException`
- `InvalidInputException`
- `EntityNotFoundException`
- `ResourceNumberLimitExceededException`
- `InternalServiceException`

- `OperationTimeoutException`
- `GlueEncryptionException`
- `ConcurrentModificationException`
- `ResourceNotReadyException`
- `FederationSourceException`
- `FederationSourceRetryableException`

## UpdateTable azione (Python: update_table)

Aggiorna una tabella di metadati nel catalogo dati.

Richiesta

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo dati in cui risiede la tabella Se non ne viene fornito nessuno, per impostazione predefinita viene utilizzato l'ID AWS dell'account.
- `DatabaseName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del database del catalogo in cui risiede la tabella. Per la compatibilità Hive, questo nome è scritto interamente in caratteri minuscoli.
- `Name`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'identificatore univoco per la tabella all'interno del database specificato che verrà creato nel AWS Glue Data Catalog.
- `TableInput`: un oggetto TableInput.

  Un'oggetto `TableInput` avanzato per la definizione della tabella di metadati nel catalogo.
- `SkipArchive`: booleano.

  Per impostazione predefinita, `UpdateTable` crea sempre una versione archiviata della tabella prima di aggiornarla. Se tuttavia `skipArchive` è impostato su true, `UpdateTable` non crea la versione archiviata.

- `TransactionId`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Custom string pattern #44.

  ID transazione in cui aggiornare il contenuto della tabella.

- `VersionId`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  ID della versione in cui aggiornare il contenuto della tabella.

- `ViewUpdateAction`: stringa UTF-8 (valori validi: `ADD` | `REPLACE` | `ADD_OR_REPLACE` | `DROP`).

  L'operazione da eseguire durante l'aggiornamento della vista.

- `Force`: booleano.

  Un flag che può essere impostato su true per ignorare i requisiti di corrispondenza di descrittore di archiviazione e oggetto secondario.

- `UpdateOpenTableFormatInput`: un oggetto UpdateOpenTableFormatInput.

  Parametri di input per l'aggiornamento di tabelle aperte in AWS Glue Data Catalog, che fungono da wrapper per operazioni di aggiornamento specifiche di formati come Apache Iceberg.

Risposta

- Nessun parametro di risposta.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `InternalServiceException`
- `OperationTimeoutException`
- `ConcurrentModificationException`
- `ResourceNumberLimitExceededException`
- `GlueEncryptionException`
- `ResourceNotReadyException`
- `FederationSourceException`

- `FederationSourceRetryableException`
- `AlreadyExistsException`

## DeleteTable azione (Python: delete_table)

Rimuove una definizione di tabella dal catalogo dati.

> **Note**
>
> Una volta completata questa operazione, non potrai più accedere alle versioni e alle partizioni delle tabelle che appartengono alle tabelle eliminate. AWS Glue elimina tempestivamente queste risorse "orfane" in modo asincrono, a discrezione del servizio.
> Per garantire l'eliminazione immediata di tutte le risorse correlate, prima di chiamare `DeleteTable`, utilizza `DeleteTableVersion` o `BatchDeleteTableVersion` e `DeletePartition` o `BatchDeletePartition` per eliminare eventuali risorse che appartengono alla tabella.

Richiesta

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo dati in cui risiede la tabella Se non ne viene fornito nessuno, per impostazione predefinita viene utilizzato l'ID AWS dell'account.
- `DatabaseName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del database del catalogo in cui risiede la tabella. Per la compatibilità Hive, questo nome è scritto interamente in caratteri minuscoli.
- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome della tabella da eliminare. Per la compatibilità Hive, questo nome è scritto interamente in caratteri minuscoli.
- `TransactionId`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Custom string pattern #44.

ID transazione in cui eliminare il contenuto della tabella.

Risposta

- Nessun parametro di risposta.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `InternalServiceException`
- `OperationTimeoutException`
- `ConcurrentModificationException`
- `ResourceNotReadyException`
- `FederationSourceException`
- `FederationSourceRetryableException`

## BatchDeleteTable azione (Python: batch_delete_table)

Elimina più tabelle contemporaneamente.

> **Note**
>
> Una volta completata questa operazione, non potrai più accedere alle versioni e alle partizioni delle tabelle che appartengono alle tabelle eliminate. AWS Glue elimina tempestivamente queste risorse "orfane" in modo asincrono, a discrezione del servizio.
> Per garantire l'eliminazione immediata di tutte le risorse correlate, prima di chiamare `BatchDeleteTable`, utilizza `DeleteTableVersion` o `BatchDeleteTableVersion` e `DeletePartition` o `BatchDeletePartition` per eliminare eventuali risorse che appartengono alla tabella.

Richiesta

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo dati in cui risiede la tabella Se non ne viene fornito nessuno, per impostazione predefinita viene utilizzato l'ID AWS dell'account.

- `DatabaseName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del database del catalogo in cui risiede la tabella da eliminare. Per la compatibilità Hive, questo nome è scritto interamente in caratteri minuscoli.

- `TablesToDelete` - obbligatorio: una matrice di stringhe UTF-8, non superiore a 100 stringhe.

  Un elenco della tabella da eliminare.

- `TransactionId`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Custom string pattern #44.

  ID transazione in cui eliminare il contenuto della tabella.

Risposta

- `Errors`: una matrice di oggetti TableError.

  Un elenco di errori riscontrati nel tentativo di eliminazione delle tabelle specificate.

Errori

- `InvalidInputException`
- `EntityNotFoundException`
- `InternalServiceException`
- `OperationTimeoutException`
- `GlueEncryptionException`
- `ResourceNotReadyException`

## GetTable azione (Python: get_table)

Consente di recuperare la definizione `Table` in un catalogo dati per una tabella specificata.

Richiesta

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo dati in cui risiede la tabella Se non ne viene fornito nessuno, per impostazione predefinita viene utilizzato l'ID AWS dell'account.

- `DatabaseName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del database nel catalogo in cui risiede la tabella. Per la compatibilità Hive, questo nome è scritto interamente in caratteri minuscoli.

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome della tabella per cui recuperare la definizione. Per la compatibilità Hive, questo nome è scritto interamente in caratteri minuscoli.

- `TransactionId`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Custom string pattern #44.

  ID transazione in cui leggere il contenuto della tabella.

- `QueryAsOfTime`: timestamp.

  Il momento a partire dal quale leggere il contenuto della tabella. Se non è impostato, verrà utilizzato l'orario di esecuzione del commit della transazione più recente. Non può essere specificato insieme a `TransactionId`.

- `AuditContext`: un oggetto AuditContext.

  Una struttura contenente il contesto Lake Formation di controllo.

- `IncludeStatusDetails`: booleano.

  Speciifica se includere i dettagli sullo stato relativi a una richiesta di creazione o aggiornamento di una vista del catalogo AWS Glue dati.

Risposta

- `Table`: un oggetto Tabella.

  L'oggetto `Table` che definisce la tabella specificata.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `InternalServiceException`
- `OperationTimeoutException`
- `GlueEncryptionException`
- `ResourceNotReadyException`
- `FederationSourceException`
- `FederationSourceRetryableException`

## GetTables azione (Python: get_tables)

Consente di recuperare le definizioni di alcune o di tutte le tabelle in un determinato `Database`.

Richiesta

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo dati in cui risiede la tabella. Se non ne viene fornito nessuno, per impostazione predefinita viene utilizzato l'ID AWS dell'account.

- `DatabaseName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il database nel catalogo delle tabelle da elencare. Per la compatibilità Hive, questo nome è scritto interamente in caratteri minuscoli.

- `Expression`: stringa UTF-8, non superiore a 2048 byte di lunghezza, corrispondente a Single-line string pattern.

Un modello di espressione regolare. Se presente, vengono restituite solo le tabelle con i nomi corrispondenti al modello.

- `NextToken`: stringa UTF-8.

  Un token di continuazione, incluso se si tratta di una chiamata di continuazione.

- `MaxResults`: numero (intero), non inferiore a 1 o superiore a 100.

  Il numero massimo di tabelle da restituire in una risposta singola.

- `TransactionId`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Custom string pattern #44.

  ID transazione in cui leggere il contenuto della tabella.

- `QueryAsOfTime`: timestamp.

  Il momento a partire dal quale leggere il contenuto della tabella. Se non è impostato, verrà utilizzato l'orario di esecuzione del commit della transazione più recente. Non può essere specificato insieme a `TransactionId`.

- `AuditContext`: un oggetto AuditContext.

  Una struttura contenente il contesto Lake Formation di controllo.

- `IncludeStatusDetails`: booleano.

  Speciifica se includere i dettagli sullo stato relativi a una richiesta di creazione o aggiornamento di una vista del catalogo AWS Glue dati.

- `AttributesToGet`: una matrice di stringhe UTF-8.

  Specifica i campi della tabella restituiti dalla chiamata `GetTables`. Questo parametro non accetta un elenco vuoto. La richiesta deve includere `NAME`.

  Di seguito sono riportate le combinazioni di valori valide:

  - `NAME`: i nomi di tutte le tabelle nel database.
  - `NAME`, `TABLE_TYPE`: i nomi di tutte le tabelle e i tipi di tabelle.

Risposta

- `TableList`: una matrice di oggetti Tabella.

Un elenco di tutti gli oggetti `Table` richiesti.

- `NextToken`: stringa UTF-8.

  Un token di continuazione, presente se il segmento dell'elenco corrente non è l'ultimo.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `OperationTimeoutException`
- `InternalServiceException`
- `GlueEncryptionException`
- `FederationSourceException`
- `FederationSourceRetryableException`

## GetTableVersion azione (Python: get_table_version)

Consente di recuperare una versione specificata di una tabella.

Richiesta

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo dati in cui risiede la tabella. Se non ne viene fornito nessuno, per impostazione predefinita viene utilizzato l'ID AWS dell'account.

- `DatabaseName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del database del catalogo in cui la tabella risiede. Per la compatibilità Hive, questo nome è scritto interamente in caratteri minuscoli.

- `TableName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome della tabella. Per la compatibilità Hive, questo nome è scritto interamente in caratteri minuscoli.

- `VersionId`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il valore ID della versione della tabella da recuperare. `VersionID` è una rappresentazione di stringa di un numero intero. Ogni versione viene incrementata di 1.

Risposta

- `TableVersion`: un oggetto TableVersion.

  La versione richiesta della tabella.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `InternalServiceException`
- `OperationTimeoutException`
- `GlueEncryptionException`

## GetTableVersions azione (Python: get_table_versions)

Consente di recuperare un elenco di stringhe identificativo delle versioni disponibili di una tabella specificata.

Richiesta

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo dati in cui risiede la tabella. Se non ne viene fornito nessuno, per impostazione predefinita viene utilizzato l'ID AWS dell'account.

- `DatabaseName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del database del catalogo in cui la tabella risiede. Per la compatibilità Hive, questo nome è scritto interamente in caratteri minuscoli.

- `TableName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome della tabella. Per la compatibilità Hive, questo nome è scritto interamente in caratteri minuscoli.
- `NextToken`: stringa UTF-8.

  Un token di continuazione, se non è la prima chiamata.
- `MaxResults`: numero (intero), non inferiore a 1 o superiore a 100.

  Il numero massimo di versioni della tabella da restituire in una risposta.

Risposta

- `TableVersions`: una matrice di oggetti TableVersion.

  Un elenco di stringhe identificativo delle versioni disponibili della tabella specificata.
- `NextToken`: stringa UTF-8.

  Un token di continuazione, se l'elenco delle versioni disponibili non comprende l'ultima.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `InternalServiceException`
- `OperationTimeoutException`
- `GlueEncryptionException`

## DeleteTableVersion azione (Python: delete_table_version)

Elimina una versione specificata di una tabella.

Richiesta

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

L'ID del catalogo dati in cui risiede la tabella. Se non ne viene fornito nessuno, per impostazione predefinita viene utilizzato l'ID AWS dell'account.

- `DatabaseName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del database del catalogo in cui la tabella risiede. Per la compatibilità Hive, questo nome è scritto interamente in caratteri minuscoli.

- `TableName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome della tabella. Per la compatibilità Hive, questo nome è scritto interamente in caratteri minuscoli.

- `VersionId` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il valore ID della versione della tabella da eliminare. `VersionID` è una rappresentazione di stringa di un numero intero. Ogni versione viene incrementata di 1.

Risposta

- Nessun parametro di risposta.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `InternalServiceException`
- `OperationTimeoutException`

## BatchDeleteTableVersion azione (Python: batch_delete_table_version)

Elimina un batch di versioni specificato di una tabella.

Richiesta

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo dati in cui risiede la tabella. Se non ne viene fornito nessuno, per impostazione predefinita viene utilizzato l'ID dell'account. AWS

- `DatabaseName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del database del catalogo in cui la tabella risiede. Per la compatibilità Hive, questo nome è scritto interamente in caratteri minuscoli.

- `TableName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome della tabella. Per la compatibilità Hive, questo nome è scritto interamente in caratteri minuscoli.

- `VersionIds` - obbligatorio: una serie di stringhe UTF-8, non superiore a 100 stringhe.

  Un elenco IDs delle versioni da eliminare. `VersionId` è una rappresentazione di stringa di un numero intero. Ogni versione viene incrementata di 1.

Risposta

- `Errors`: una matrice di oggetti TableVersionError.

  Un elenco di errori riscontrati nel tentativo di eliminazione delle versioni della tabella specificata.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `InternalServiceException`
- `OperationTimeoutException`

## SearchTables azione (Python: search_tables)

Cerca un set di tabelle in base alle proprietà nei metadati della tabella nonché nel database padre. Puoi eseguire ricerche su condizioni di testo o filtro.

Puoi ottenere solo tabelle a cui hai accesso in base alle policy di sicurezza definite in Lake Formation. È necessario almeno un accesso in sola lettura alla tabella affinché venga restituita. Se non disponi dell'accesso a tutte le colonne della tabella, non verranno eseguite ricerche in queste colonne quando l'elenco delle tabelle viene restituito. Se disponi dell'accesso alle colonne ma non ai dati nelle colonne, tali colonne e i metadati associati a tali colonne saranno inclusi nella ricerca.

Richiesta

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Un identificatore univoco, costituito da *`account_id`*.

- `NextToken`: stringa UTF-8.

  Un token di continuazione, incluso se si tratta di una chiamata di continuazione.

- `Filters`: una matrice di oggetti PropertyPredicate.

  Un elenco di coppie chiave-valore e un comparatore utilizzato per filtrare i risultati della ricerca. Restituisce tutte le entità che corrispondono al predicato.

  Il membro `Comparator` della struttura `PropertyPredicate` viene utilizzata solo per i campi ora e può essere omesso per altri tipi di campo. Inoltre, quando si confrontano i valori di stringa, ad esempio quando `Key=Name`, viene utilizzato un algoritmo di corrispondenza parziale. Il campo `Key` (ad esempio, il valore del campo `Name`) è diviso su determinati caratteri di punteggiatura, ad esempio -, :, #, ecc. in token. Quindi ogni token è una corrispondenza esatta rispetto al membro `Value` di `PropertyPredicate`. Ad esempio se `Key=Name` e `Value=link`, le tabelle denominate `customer-link` e `xx-link-yy` vengono restituite, ma`xxlinkyy` non viene restituita.

- `SearchText`: stringa Value, non inferiore a 1 o superiore a 1024 byte di lunghezza.

  Una stringa utilizzata per una ricerca di testo.

  Specifica un valore in filtri tra virgolette basato su una corrispondenza esatta con il valore.

- `SortCriteria`: una matrice di oggetti SortCriterion, non superiore a 1 struttura.

Un elenco di criteri per ordinare i risultati in base a un nome di campo, in ordine crescente o decrescente.

- `MaxResults`: numero (intero), non inferiore a 1 o superiore a 1.000.

  Il numero massimo di tabelle da restituire in una risposta singola.

- `ResourceShareType`: stringa UTF-8 (valori validi: `FOREIGN` | `ALL` | `FEDERATED`).

  Consente di specificare che si desidera eseguire la ricerca nelle tabelle condivise con l'account. I valori consentiti sono `FOREIGN` o `ALL`.

  - Se impostato su `FOREIGN`, cercherà le tabelle condivise con l'account.
  - Se impostato su `ALL`, cercherà le tabelle condivise con l'account, così come le tabelle nell'account locale.

- `IncludeStatusDetails`: booleano.

  Speciifica se includere i dettagli sullo stato relativi a una richiesta di creazione o aggiornamento di una vista del catalogo AWS Glue dati.

Risposta

- `NextToken`: stringa UTF-8.

  Un token di continuazione, presente se il segmento dell'elenco corrente non è l'ultimo.

- `TableList`: una matrice di oggetti Tabella.

  Un elenco di tutti gli oggetti `Table` richiesti. La risposta `SearchTables` restituisce solo le tabelle a cui hai accesso.

Errori

- `InternalServiceException`
- `InvalidInputException`
- `OperationTimeoutException`

## GetPartitionIndexes azione (Python: get_partition_indexes)

Recupera gli indici delle partizioni associati a una tabella.

Richiesta

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo in cui si trova la tabella.

- `DatabaseName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Specifica il nome di un database da cui si desidera recuperare gli indici delle partizioni.

- `TableName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Specifica il nome di una tabella da cui si desidera recuperare gli indici delle partizioni.

- `NextToken`: stringa UTF-8.

  Un token di continuazione, incluso se si tratta di una chiamata di continuazione.

Risposta

- `PartitionIndexDescriptorList`: una matrice di oggetti PartitionIndexDescriptor.

  Un elenco di descrittori di indice.

- `NextToken`: stringa UTF-8.

  Un token di continuazione, presente se il segmento dell'elenco corrente non è l'ultimo.

Errori

- `InternalServiceException`
- `OperationTimeoutException`
- `InvalidInputException`
- `EntityNotFoundException`
- `ConflictException`

## CreatePartitionIndex azione (Python: create_partition_index)

Crea un indice della partizione specificato in una tabella esistente.

Richiesta

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo in cui si trova la tabella.
- `DatabaseName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Specifica il nome di un database in cui si desidera creare un indice della partizione.
- `TableName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Specifica il nome di una tabella in cui si desidera creare un indice della partizione.
- `PartitionIndex` - obbligatorio: un oggetto PartitionIndex.

  Specifica una struttura `PartitionIndex` per creare un indice della partizione in una tabella esistente.

Risposta

- Nessun parametro di risposta.

Errori

- `AlreadyExistsException`
- `InvalidInputException`
- `EntityNotFoundException`
- `ResourceNumberLimitExceededException`
- `InternalServiceException`
- `OperationTimeoutException`
- `GlueEncryptionException`

## DeletePartitionIndex azione (Python: delete_partition_index)

Elimina un indice della partizione specificato da una tabella esistente.

### Richiesta

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo in cui si trova la tabella.
- `DatabaseName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Specifica il nome di un database da cui si desidera eliminare un indice della partizione.
- `TableName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Specifica il nome di una tabella da cui si desidera eliminare un indice della partizione.
- `IndexName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome dell'indice della partizione da eliminare.

### Risposta

- Nessun parametro di risposta.

### Errori

- `InternalServiceException`
- `OperationTimeoutException`
- `InvalidInputException`
- `EntityNotFoundException`
- `ConflictException`
- `GlueEncryptionException`

## GetColumnStatisticsForTable azione (Python: get_column_statistics_for_table)

Recupera le statistiche delle colonne della tabella.

L'autorizzazione Identity and Access Management (IAM) necessaria per questa operazione è `GetTable`.

Richiesta

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo dati in cui si trovano le partizioni. Se non ne viene fornito nessuno, per impostazione predefinita viene utilizzato l'ID dell'account. AWS
- `DatabaseName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del database del catalogo in cui risiedono le partizioni.
- `TableName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome della tabella della partizione.
- `ColumnNames` - obbligatorio: una serie di stringhe UTF-8, non superiore a 100 stringhe.

  Un elenco dei nomi delle colonne.

Risposta

- `ColumnStatisticsList`: una matrice di oggetti ColumnStatistics.

  Elenco di ColumnStatistics.
- `Errors`: una matrice di oggetti ColumnError.

  L'elenco ColumnStatistics di questi non è stato recuperato.

Errori

- `EntityNotFoundException`
- `InvalidInputException`

- `InternalServiceException`
- `OperationTimeoutException`
- `GlueEncryptionException`

## UpdateColumnStatisticsForTable azione (Python: update_column_statistics_for_table)

Crea o aggiorna le statistiche delle tabelle della colonna.

L'autorizzazione Identity and Access Management (IAM) necessaria per questa operazione è `UpdateTable`.

Richiesta

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo dati in cui si trovano le partizioni. Se non ne viene fornito nessuno, per impostazione predefinita viene utilizzato l'ID dell'account. AWS
- `DatabaseName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del database del catalogo in cui risiedono le partizioni.
- `TableName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome della tabella della partizione.
- `ColumnStatisticsList` - obbligatorio: una matrice di oggetti ColumnStatistics, non superiore a 25 strutture.

  Un elenco delle statistiche delle colonne.

Risposta

- `Errors`: una matrice di oggetti ColumnStatisticsError.

  Elenco di ColumnStatisticsErrors.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `InternalServiceException`
- `OperationTimeoutException`
- `GlueEncryptionException`

## DeleteColumnStatisticsForTable azione (Python: delete_column_statistics_for_table)

Recupera le statistiche delle colonne della tabella.

L'autorizzazione Identity and Access Management (IAM) necessaria per questa operazione è `DeleteTable`.

Richiesta

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo dati in cui si trovano le partizioni. Se non ne viene fornito nessuno, per impostazione predefinita viene utilizzato l'ID dell'account. AWS
- `DatabaseName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del database del catalogo in cui risiedono le partizioni.
- `TableName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome della tabella della partizione.
- `ColumnName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome della colonna.

Risposta

- Nessun parametro di risposta.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `InternalServiceException`
- `OperationTimeoutException`
- `GlueEncryptionException`

## API della partizione

L'API Partition descrive i tipi di dati e le operazioni utilizzate per lavorare con le partizioni.

### Tipi di dati

- Struttura della partizione
- PartitionInput struttura
- PartitionSpecWithSharedStorageDescriptor struttura
- PartitionListComposingSpec struttura
- PartitionSpecProxy struttura
- PartitionValueList struttura
- Struttura del segmento
- PartitionError struttura
- BatchUpdatePartitionFailureEntry struttura
- BatchUpdatePartitionRequestEntry struttura
- StorageDescriptor struttura
- SchemaReference struttura
- SerDeInfo struttura
- SkewedInfo struttura

### Struttura della partizione

Rappresenta una porzione dei dati della tabella.

Campi

- `Values`: una matrice di stringhe UTF-8.

  I valori della partizione.

- `DatabaseName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del database del catalogo in cui creare la partizione.

- `TableName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome della tabella di database in cui creare la partizione.

- `CreationTime`: timestamp.

  L'ora in cui è stata creata la partizione.

- `LastAccessTime`: timestamp.

  L'ora in cui è stato effettuato l'ultimo accesso alla partizione.

- `StorageDescriptor`: un oggetto StorageDescriptor.

  Fornisce informazioni sulla posizione fisica in cui è memorizzata la partizione.

- `Parameters`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa chiave, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Ogni valore è una stringa UTF-8, lunga non più di 512000 byte.

  Queste coppie chiave-valore definiscono i parametri per la partizione.

- `LastAnalyzedTime`: timestamp.

  L'ultima volta in cui sono state calcolate le statistiche di colonna per questa partizione.

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo dati in cui si trovano le partizioni.

## PartitionInput struttura

Struttura utilizzata per la creazione e per l'aggiornamento di una partizione.

Campi

- `Values`: una matrice di stringhe UTF-8.

  I valori della partizione. Sebbene questo parametro non sia richiesto dall'SDK, è necessario specificarlo per un input valido.

  I valori delle chiavi per la nuova partizione devono essere passati come una matrice di oggetti String che devono essere sistemati seguendo lo stesso ordine delle chiavi di partizione che appaiono nel prefisso Amazon S3. Altrimenti AWS Glue aggiungerà i valori alle chiavi sbagliate.
- `LastAccessTime`: timestamp.

  L'ora in cui è stato effettuato l'ultimo accesso alla partizione.
- `StorageDescriptor`: un oggetto StorageDescriptor.

  Fornisce informazioni sulla posizione fisica in cui è memorizzata la partizione.
- `Parameters`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa chiave, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Ogni valore è una stringa UTF-8, lunga non più di 512000 byte.

  Queste coppie chiave-valore definiscono i parametri per la partizione.
- `LastAnalyzedTime`: timestamp.

  L'ultima volta in cui sono state calcolate le statistiche di colonna per questa partizione.

## PartitionSpecWithSharedStorageDescriptor struttura

Una specifica per le partizioni che condividono una posizione fisica.

Campi

- `StorageDescriptor`: un oggetto StorageDescriptor.

Le informazioni condivise sullo storage fisico.

- `Partitions`: una matrice di oggetti Partizione.

  Un elenco di partizioni che condividono questa posizione fisica.

## PartitionListComposingSpec struttura

Elenca le partizioni correlate.

Campi

- `Partitions`: una matrice di oggetti Partizione.

  Un elenco di partizioni nella specifica di composizione.

## PartitionSpecProxy struttura

Fornisce un percorso radice per partizioni specifiche.

Campi

- `DatabaseName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il database del catalogo in cui risiedono le partizioni.
- `TableName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome della tabella che contiene le partizioni.
- `RootPath`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il percorso radice del proxy per gestire le partizioni.
- `PartitionSpecWithSharedSD`: un oggetto PartitionSpecWithSharedStorageDescriptor.

  Una specifica delle partizioni che condividono la stessa posizione dello storage fisico.
- `PartitionListComposingSpec`: un oggetto PartitionListComposingSpec.

  Specifica un elenco di partizioni.

## PartitionValueList struttura

Contiene un elenco di valori che definiscono le partizioni.

Campi

- `Values` - obbligatorio: una matrice di stringhe UTF-8.

  L'elenco dei valori.

## Struttura del segmento

Definisce una regione non sovrapposta delle partizioni di una tabella, consentendo l'esecuzione di più richieste in parallelo.

Campi

- `SegmentNumber`: obbligatorio: numero (intero), non superiore a Nessuno.

  Il numero dell'indice a base zero del segmento. Ad esempio, se il numero totale di segmenti è 4, i alori di `SegmentNumber` vanno da 0 a 3.
- `TotalSegments`: obbligatorio: numero (intero), non inferiore a 1 o superiore a 10.

  Il numero totale dei segmenti.

## PartitionError struttura

Contiene informazioni sull'errore di una partizione.

Campi

- `PartitionValues`: una matrice di stringhe UTF-8.

  I valori che definiscono la partizione.
- `ErrorDetail`: un oggetto ErrorDetail.

  Dettagli sull'errore della partizione.

## BatchUpdatePartitionFailureEntry struttura

Contiene informazioni sull'errore di una partizione di aggiornamento in batch.

Campi

- `PartitionValueList`: una matrice di stringhe UTF-8, non superiore a 100.

  Un elenco di valori che definiscono le partizioni.
- `ErrorDetail`: un oggetto ErrorDetail.

  Dettagli sull'errore della partizione di aggiornamento in batch.

## BatchUpdatePartitionRequestEntry struttura

Una struttura che contiene i valori e la struttura utilizzati per aggiornare una partizione.

Campi

- `PartitionValueList`. Obbligatorio: una serie di stringhe UTF-8, non superiore a 100 stringhe.

  Un elenco di valori che definiscono le partizioni.
- `PartitionInput` - obbligatorio: un oggetto PartitionInput.

  Struttura utilizzata per l'aggiornamento di una partizione.

## StorageDescriptor struttura

Descrive lo storage fisico dei dati della tabella.

Campi

- `Columns`: una matrice di oggetti Colonna.

  Un elenco delle `Columns` nella tabella.
- `Location`: stringa di posizione, non superiore a 2056 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  La posizione fisica della tabella. Per default utilizza il formato della posizione del warehouse, seguita dalla posizione del database nel warehouse, seguita dal nome della tabella.

- `AdditionalLocations`: una matrice di stringhe UTF-8.

  Un elenco di posizioni che puntano al percorso in cui si trova una tabella Delta.

- `InputFormat`: stringa di formato, non superiore a 128 byte di lunghezza, corrispondente a Single-line string pattern.

  Il formato di input: `SequenceFileInputFormat` (binario) o `TextInputFormat` o un formato personalizzato.

- `OutputFormat`: stringa di formato, non superiore a 128 byte di lunghezza, corrispondente a Single-line string pattern.

  Il formato di output: `SequenceFileOutputFormat` (binario) o `IgnoreKeyTextOutputFormat` o un formato personalizzato.

- `Compressed`: booleano.

  `True` se i dati nella tabella sono compressi, in caso contrario `False`.

- `NumberOfBuckets`: numero (intero).

  Deve essere specificato se la tabella contiene una qualsiasi colonna di dimensione.

- `SerdeInfo`: un oggetto SerDeInfo.

  Le informazioni serialization/deserialization (SerDe).

- `BucketColumns`: una matrice di stringhe UTF-8.

  Un elenco di colonne per il raggruppamento del reducer, colonne di clustering, colonne di bucketing nella tabella.

- `SortColumns`: una matrice di oggetti Ordine.

  Un elenco specificando l'ordine di ciascuna bucket nella tabella.

- `Parameters`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa chiave, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Ogni valore è una stringa UTF-8, lunga non più di 512000 byte.

  Le proprietà fornite dall'utente nel modulo chiave-valore.

- `SkewedInfo`: un oggetto SkewedInfo.

Informazioni sui valori che appaiono di frequente in una colonna (valori disallineati).

- `StoredAsSubDirectories`: booleano.

  `True` se i dati nella tabella sono archiviati nelle sottodirectory, in caso contrario `False`.

- `SchemaReference`: un oggetto SchemaReference.

  Un oggetto che fa riferimento a uno schema memorizzato nel registro degli AWS Glue schemi.

  Quando crei una tabella, puoi passare un elenco vuoto di colonne per lo schema e utilizzare invece un riferimento allo schema.

## SchemaReference struttura

Un oggetto che fa riferimento a uno schema memorizzato nel registro degli AWS Glue schemi.

Campi

- `SchemaId`: un oggetto SchemaId.

  Una struttura che contiene campi di identità dello schema. Deve essere fornito questo o `SchemaVersionId`.

- `SchemaVersionId`: stringa UTF-8, non inferiore a 36 o superiore a 36 byte di lunghezza, corrispondente a Custom string pattern #45.

  L'ID univoco assegnato a una versione dello schema. Deve essere fornito questo o `SchemaId`.

- `SchemaVersionNumber`: numero (intero), non inferiore a 1 o superiore a 100000.

  Il numero di versione dello schema.

## SerDeInfo struttura

Informazioni su un serialization/deserialization programma (SerDe) che funge da estrattore e caricatore.

Campi

- `Name`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

Nome del. SerDe

- `SerializationLibrary`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Di solito la classe che implementa. SerDe Un esempio è `org.apache.hadoop.hive.serde2.columnar.ColumnarSerDe`.
- `Parameters`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa chiave, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Ogni valore è una stringa UTF-8, lunga non più di 512000 byte.

  Queste coppie chiave-valore definiscono i parametri di inizializzazione per. SerDe

## SkewedInfo struttura

Specifica i valori disallineati in una tabella. I valori disallineati sono quelli che si verificano con una frequenza molto elevata.

Campi

- `SkewedColumnNames`: una matrice di stringhe UTF-8.

  Un elenco di nomi delle colonne contenenti i valori disallineati.
- `SkewedColumnValues`: una matrice di stringhe UTF-8.

  Un elenco di valori che appaiono così frequentemente da poter essere considerati disallineati.
- `SkewedColumnValueLocationMaps`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8.

  Ogni valore è una stringa UTF-8.

  Una mappatura di valori disallineati per le colonne che li contengono.

## Operazioni

- CreatePartition azione (Python: create_partition)

- BatchCreatePartition azione (Python: batch_create_partition)
- UpdatePartition azione (Python: update_partition)
- DeletePartition azione (Python: delete_partition)
- BatchDeletePartition azione (Python: batch_delete_partition)
- GetPartition azione (Python: get_partition)
- GetPartitions azione (Python: get_partitions)
- BatchGetPartition azione (Python: batch_get_partition)
- BatchUpdatePartition azione (Python: batch_update_partition)
- GetColumnStatisticsForPartition azione (Python: get_column_statistics_for_partition)
- UpdateColumnStatisticsForPartition azione (Python: update_column_statistics_for_partition)
- DeleteColumnStatisticsForPartition azione (Python: delete_column_statistics_for_partition)

## CreatePartition azione (Python: create_partition)

Crea una nuova partizione.

Richiesta

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID dell' AWS account del catalogo in cui deve essere creata la partizione.
- `DatabaseName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del database dei metadata in cui deve essere creata la partizione.
- `TableName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome della tabella dei metadata in cui deve essere creata la partizione.
- `PartitionInput` - obbligatorio: un oggetto PartitionInput.

  Una struttura `PartitionInput` che definisce la partizione da creare.

Risposta

- Nessun parametro di risposta.

Errori

- `InvalidInputException`
- `AlreadyExistsException`
- `ResourceNumberLimitExceededException`
- `InternalServiceException`
- `EntityNotFoundException`
- `OperationTimeoutException`
- `GlueEncryptionException`

## BatchCreatePartition azione (Python: batch_create_partition)

Crea una o più partizioni in un'operazione in batch.

Richiesta

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo in cui deve essere creata la partizione. Attualmente, questo dovrebbe essere l'ID dell'account. AWS
- `DatabaseName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del database dei metadata in cui deve essere creata la partizione.
- `TableName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome della tabella dei metadata in cui deve essere creata la partizione.
- `PartitionInputList`: obbligatorio: una matrice di oggetti PartitionInput, non superiore a 100 strutture.

  Un elenco di strutture `PartitionInput` che definiscono le partizioni da creare.

Risposta

- `Errors`: una matrice di oggetti PartitionError.

  Errori rilevati durante il tentativo di creare le partizioni richieste.

Errori

- `InvalidInputException`
- `AlreadyExistsException`
- `ResourceNumberLimitExceededException`
- `InternalServiceException`
- `EntityNotFoundException`
- `OperationTimeoutException`
- `GlueEncryptionException`

## UpdatePartition azione (Python: update_partition)

Aggiorna una partizione.

Richiesta

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo dati in cui si trova la partizione da aggiornare. Se non ne viene fornito nessuno, per impostazione predefinita viene utilizzato l'ID AWS dell'account.
- `DatabaseName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del database del catalogo in cui risiede la tabella.
- `TableName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome della tabella in cui si trova la partizione da aggiornare.
- `PartitionValueList`. Obbligatorio: una serie di stringhe UTF-8, non superiore a 100 stringhe.

Elenco dei valori della chiave della partizione che definiscono la partizione da aggiornare.

- `PartitionInput` - obbligatorio: un oggetto PartitionInput.

  Il nuovo oggetto della partizione a cui aggiornare la partizione.

  La proprietà `Values` non può essere modificata. Se si desidera modificare i valori della chiave per una partizione, è necessario eliminare e creare nuovamente la partizione.

Risposta

- Nessun parametro di risposta.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `InternalServiceException`
- `OperationTimeoutException`
- `GlueEncryptionException`

## DeletePartition azione (Python: delete_partition)

Elimina una partizione specificata.

Richiesta

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo dati in cui si trova la partizione da eliminare. Se non ne viene fornito nessuno, per impostazione predefinita viene utilizzato l'ID AWS dell'account.

- `DatabaseName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del database del catalogo in cui risiede la tabella.

- `TableName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome della tabella che contiene la partizione da eliminare.
- `PartitionValues`. Obbligatorio: una serie di stringhe UTF-8.

  I valori che definiscono la partizione.

Risposta

- Nessun parametro di risposta.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `InternalServiceException`
- `OperationTimeoutException`

## BatchDeletePartition azione (Python: batch_delete_partition)

Cancella una o più partizioni in un'operazione in batch.

Richiesta

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo dati in cui si trova la partizione da eliminare. Se non ne viene fornito nessuno, per impostazione predefinita viene utilizzato l'ID dell' AWS account.
- `DatabaseName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del database del catalogo in cui risiede la tabella.
- `TableName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome della tabella che contiene le partizioni da eliminare.

- `PartitionsToDelete`. Obbligatorio: una serie di oggetti PartitionValueList, non superiore a 25 strutture.

  Un elenco di strutture `PartitionInput` che definiscono le partizioni da eliminare.

Risposta

- `Errors`: una matrice di oggetti PartitionError.

  Errori rilevati durante il tentativo di cancellare le partizioni richieste.

Errori

- `InvalidInputException`
- `EntityNotFoundException`
- `InternalServiceException`
- `OperationTimeoutException`

## GetPartition azione (Python: get_partition)

Consente di recuperare informazioni su una partizione specificata.

Richiesta

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo dati in cui si trova la partizione. Se non ne viene fornito nessuno, per impostazione predefinita viene utilizzato l'ID AWS dell'account.
- `DatabaseName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del database del catalogo in cui risiede la partizione.
- `TableName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome della tabella della partizione.
- `PartitionValues`. Obbligatorio: una serie di stringhe UTF-8.

I valori che definiscono la partizione.

Risposta

- `Partition`: un oggetto Partizione.

  Le informazioni richieste, sotto forma di un oggetto della `Partition`.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `InternalServiceException`
- `OperationTimeoutException`
- `GlueEncryptionException`
- `FederationSourceException`
- `FederationSourceRetryableException`

## GetPartitions azione (Python: get_partitions)

Recupera informazioni sulle partizioni in una tabella.

Richiesta

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo dati in cui si trovano le partizioni. Se non ne viene fornito nessuno, per impostazione predefinita viene utilizzato l'ID AWS dell'account.
- `DatabaseName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del database del catalogo in cui risiedono le partizioni.
- `TableName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

Il nome della tabella della partizione.

- `Expression`: stringa predicato, non superiore a 2048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  Espressione che filtra le partizioni da restituire.

  L'espressione usa la sintassi SQL simile alla clausola di filtro SQL `WHERE`. Il parser di istruzioni SQL JSQLParseranalizza l'espressione.

  Operatori: di seguito sono elencati gli operatori che puoi usare nella chiamata API `Expression`:

  **=**

  Verifica se i valori dei due operandi sono uguali. In caso affermativo, la condizione diventa true.

  Esempio: presupponiamo che "variable a" contenga 10 e "variable b" contenga 20.

  (a = b) non è true.

  **< >**

  Verifica se i valori dei due operandi sono uguali. In caso negativo, la condizione diventa true.

  Esempio: (a < > b) è true.

  **>**

  Verifica se il valore dell'operando di sinistra è superiore al valore dell'operando di destra. In caso affermativo, la condizione diventa true.

  Esempio: (a > b) non è true.

  **<**

  Verifica se il valore dell'operando di sinistra è inferiore al valore dell'operando di destra. In caso affermativo, la condizione diventa true.

  Esempio: (a < b) è true.

  **>=**

  Verifica se il valore dell'operando di sinistra è superiore o uguale al valore dell'operando di destra. In caso affermativo, la condizione diventa true.

Esempio: (a >= b) non è true.

**<=**

Verifica se il valore dell'operando di sinistra è inferiore o uguale al valore dell'operando di destra. In caso affermativo, la condizione diventa true.

Esempio: (a <= b) è true.

AND, OR, IN, BETWEEN, LIKE NOT, IS NULL

Operatori logici.

Tipi di chiave di partizione supportati: di seguito sono indicate le chiavi di partizione supportate.

- `string`
- `date`
- `timestamp`
- `int`
- `bigint`
- `long`
- `tinyint`
- `smallint`
- `decimal`

Se viene riscontrato un tipo non valido, viene generata un'eccezione.

L'elenco seguente mostra gli operatori validi per ogni tipo. Quando definisci un crawler, il tipo `partitionKey` viene creato come `STRING`, perché sia compatibile con le partizioni del catalogo.

Chiamata API di esempio:

Example

La tabella `twitter_partition` contiene tre partizioni:

```
year = 2015
        year = 2016
        year = 2017
```

Example

Get Partition `year` equivale a 2015

```
aws glue get-partitions --database-name dbname --table-name twitter_partition
        --expression "year*=*'2015'"
```

Example

Get partition `year` tra 2016 e 2018 (esclusi)

```
aws glue get-partitions --database-name dbname --table-name twitter_partition
        --expression "year>'2016' AND year<'2018'"
```

Example

Get partition `year` tra 2015 e 2018 (inclusi). Le chiamate API seguenti si equivalgono tra loro

```
aws glue get-partitions --database-name dbname --table-name twitter_partition
        --expression "year>='2015' AND year<='2018'"

        aws glue get-partitions --database-name dbname --table-name
 twitter_partition
        --expression "year BETWEEN 2015 AND 2018"

        aws glue get-partitions --database-name dbname --table-name
 twitter_partition
        --expression "year IN (2015,2016,2017,2018)"
```

Example

Un filtro di partizione con caratteri jolly, in cui l'output di chiamata seguente è l'anno di partizione = 2017. Un'espressione regolare non è supportata in `LIKE`.

```
aws glue get-partitions --database-name dbname --table-name twitter_partition
        --expression "year LIKE '%7'"
```

- `NextToken`: stringa UTF-8.

  Un token di continuazione, se non è la prima chiamata per recuperare le partizioni.

- `Segment`: un oggetto Segment.

  Il segmento delle partizioni della tabella per analizzare questa richiesta.

- `MaxResults`: numero (intero), non inferiore a 1 o superiore a 1.000.

  Il numero massimo di partizioni da restituire in una risposta singola.

- `ExcludeColumnSchema`: booleano.

  Se vero, specifica di non restituire lo schema della colonna di partizione. Utile quando sei interessato solo ad altri attributi di partizione, come i valori o la posizione delle partizioni. Questo approccio evita il problema di una risposta ampia non restituendo dati duplicati.

- `TransactionId`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Custom string pattern #44.

  ID transazione in cui leggere il contenuto della partizione.

- `QueryAsOfTime`: timestamp.

  Il momento a partire dal quale leggere il contenuto della partizione. Se non è impostato, verrà utilizzato l'orario di esecuzione del commit della transazione più recente. Non può essere specificato insieme a `TransactionId`.

Risposta

- `Partitions`: una matrice di oggetti Partizione.

  Un elenco di partizioni richieste.

- `NextToken`: stringa UTF-8.

  Un token di continuazione, se l'elenco restituito di partizioni non include l'ultima.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `OperationTimeoutException`
- `InternalServiceException`
- `GlueEncryptionException`

- `InvalidStateException`
- `ResourceNotReadyException`
- `FederationSourceException`
- `FederationSourceRetryableException`

## BatchGetPartition azione (Python: batch_get_partition)

Recupera le partizioni in una richiesta di batch.

Richiesta

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo dati in cui si trovano le partizioni. Se non ne viene fornito nessuno, per impostazione predefinita viene utilizzato l'ID dell' AWS account.
- `DatabaseName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del database del catalogo in cui risiedono le partizioni.
- `TableName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome della tabella della partizione.
- `PartitionsToGet`: obbligatorio: una matrice di oggetti PartitionValueList, non superiore a 1000 strutture.

  Un elenco di valori della partizione che identificano le partizioni da recuperare.

Risposta

- `Partitions`: una matrice di oggetti Partizione.

  Un elenco di tutte le partizioni richieste.
- `UnprocessedKeys`: una matrice di oggetti PartitionValueList, non superiore a 1000 strutture.

  Un elenco dei valori della partizione nella richiesta per la quale le partizioni non sono state restituite.

Errori

- `InvalidInputException`
- `EntityNotFoundException`
- `OperationTimeoutException`
- `InternalServiceException`
- `GlueEncryptionException`
- `InvalidStateException`
- `FederationSourceException`
- `FederationSourceRetryableException`

## BatchUpdatePartition azione (Python: batch_update_partition)

Aggiorna una o più partizioni in un'operazione in batch.

Richiesta

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo in cui deve essere aggiornata la partizione. Attualmente, questo dovrebbe essere l'ID dell'account. AWS
- `DatabaseName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del database di metadati in cui deve essere aggiornata la partizione.
- `TableName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome della tabella di metadati in cui deve essere aggiornata la partizione.
- `Entries`: obbligatorio: una matrice di oggetti BatchUpdatePartitionRequestEntry, non meno di 1 o più di 100 strutture.

  Un elenco fino a 100 oggetti `BatchUpdatePartitionRequestEntry` da aggiornare.

Risposta

- `Errors`: una matrice di oggetti BatchUpdatePartitionFailureEntry.

  Errori rilevati durante il tentativo di aggiornamento delle partizioni richieste. Elenco di oggetti `BatchUpdatePartitionFailureEntry`.

Errori

- `InvalidInputException`
- `EntityNotFoundException`
- `OperationTimeoutException`
- `InternalServiceException`
- `GlueEncryptionException`

## GetColumnStatisticsForPartition azione (Python: get_column_statistics_for_partition)

Recupera le statistiche della partizione delle colonne.

L'autorizzazione Identity and Access Management (IAM) necessaria per questa operazione è `GetPartition`.

Richiesta

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo dati in cui si trovano le partizioni. Se non ne viene fornito nessuno, per impostazione predefinita viene utilizzato l'ID dell'account. AWS
- `DatabaseName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del database del catalogo in cui risiedono le partizioni.
- `TableName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome della tabella della partizione.
- `PartitionValues`. Obbligatorio: una serie di stringhe UTF-8.

Un elenco di valori della partizione che identificano la partizione.

- `ColumnNames`. Obbligatorio: una serie di stringhe UTF-8, non superiore a 100 stringhe.

  Un elenco dei nomi delle colonne.

Risposta

- `ColumnStatisticsList`: una matrice di oggetti ColumnStatistics.

  L'elenco di ColumnStatistics tali dati non è stato recuperato.
- `Errors`: una matrice di oggetti ColumnError.

  Errore durante il recupero dei dati delle statistiche delle colonne.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `InternalServiceException`
- `OperationTimeoutException`
- `GlueEncryptionException`

## UpdateColumnStatisticsForPartition azione (Python: update_column_statistics_for_partition)

Crea o aggiorna le statistiche delle partizioni delle colonne.

L'autorizzazione Identity and Access Management (IAM) necessaria per questa operazione è `UpdatePartition`.

Richiesta

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo dati in cui si trovano le partizioni. Se non ne viene fornito nessuno, per impostazione predefinita viene utilizzato l'ID dell'account. AWS

- `DatabaseName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del database del catalogo in cui risiedono le partizioni.
- `TableName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome della tabella della partizione.
- `PartitionValues`. Obbligatorio: una serie di stringhe UTF-8.

  Un elenco di valori della partizione che identificano la partizione.
- `ColumnStatisticsList`. Obbligatorio: una serie di oggetti ColumnStatistics, non superiore a 25 strutture.

  Un elenco delle statistiche delle colonne.

Risposta

- `Errors`: una matrice di oggetti ColumnStatisticsError.

  Errore durante l'aggiornamento dei dati delle statistiche delle colonne.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `InternalServiceException`
- `OperationTimeoutException`
- `GlueEncryptionException`

## DeleteColumnStatisticsForPartition azione (Python: delete_column_statistics_for_partition)

Elimina le statistiche della colonna della partizione di una colonna.

L'autorizzazione Identity and Access Management (IAM) necessaria per questa operazione è `DeletePartition`.

Richiesta

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo dati in cui si trovano le partizioni. Se non ne viene fornito nessuno, per impostazione predefinita viene utilizzato l'ID dell'account. AWS
- `DatabaseName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del database del catalogo in cui risiedono le partizioni.
- `TableName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome della tabella della partizione.
- `PartitionValues`. Obbligatorio: una serie di stringhe UTF-8.

  Un elenco di valori della partizione che identificano la partizione.
- `ColumnName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome della colonna.

Risposta

- Nessun parametro di risposta.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `InternalServiceException`
- `OperationTimeoutException`
- `GlueEncryptionException`

# API Connessioni

L'API Connections descrive i tipi di dati e l'API relativi all'utilizzo delle connessioni in AWS Glue.

Argomenti


- API di connessione
- API Tipi di connessione
- Metadati di connessione e API di anteprima


## API di connessione

L'API Connection descrive i tipi di dati di AWS Glue connessione e l'API per creare, eliminare, aggiornare ed elencare le connessioni.

Tipi di dati


- Struttura di connessione
- ConnectionInput struttura
- TestConnectionInput struttura
- PhysicalConnectionRequirements struttura
- GetConnectionsFilter struttura
- AuthenticationConfiguration struttura
- AuthenticationConfigurationInput struttura
- OAuth2Struttura delle proprietà
- OAuth2PropertiesInput struttura
- OAuth2ClientApplication struttura
- AuthorizationCodeProperties struttura
- BasicAuthenticationCredentials struttura
- OAuth2struttura delle credenziali


### Struttura di connessione

Definisce una connessione a un'origine dati.

Campi

- `Name`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome della definizione di connessione.

- `Description`: stringa di descrizione, non superiore a 2048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  La descrizione della connessione.

- `ConnectionType`— Stringa UTF-8 (valori validi: `JDBC SFTP | MONGODB | KAFKA | NETWORK | MARKETPLACE | | CUSTOM | SALESFORCE | VIEW_VALIDATION_REDSHIFT | VIEW_VALIDATION_ATHENA | GOOGLEADS | GOOGLESHEETS | GOOGLEANALYTICS4 | SERVICENOW | MARKETO | SAPODATA | ZENDESK | | JIRACLOUD | NETSUITEERP | HUBSPOT | FACEBOOKADS | INSTAGRAMADS | ZOHOCRM | SALESFORCEPARDOT | SALESFORCEMARKETINGCLOUD | ADOBEANALYTICS | SLACK | | LINKEDIN | MIXPANEL | ASANA | STRIPE | SMARTSHEET | DATADOG WOOCOMMERCE INTERCOM SNAPCHATADS| PAYPAL | QUICKBOOKS | FACEBOOKPAGEINSIGHTS FRESHDESK | TWILIO | DOCUSIGNMONITOR | FRESHSALES | ZOOM | GOOGLESEARCHCONSOLE | SALESFORCECOMMERCECLOUD | SAPCONCUR | DYNATRACE MICROSOFTDYNAMIC365FINANCEANDOPS | MICROSOFTTEAMS | BLACKBAUDRAISEREDGENXT | MAILCHIMP | GITLAB | PENDO | PRODUCTBOARD | CIRCLECI | PIPEDIVE SENDGRID | AZURECOSMOS | AZURESQL | BIGQUERY | BLACKBAUD | CLOUDERAHIVE | CLOUDERAIMPALA | CLOUDWATCH | CLOUDWATCHMETRICS CMDB | DATALAKEGEN2 | DB2 | DB2AS400 | DOCUMENTDB | DOMO DYNAMODB GOOGLECLOUDSTORAGE| HBASE | KUSTOMER | MICROSOFTDYNAMICS365CRM | MONDAY | MYSQL | OKTA | OPENSEARCH | ORACLE | PIPEDRIVE | | POSTGRESQL | SAPHANA | SQLSERVER | SYNAPSE | TERADATA | TERADATANOS | TIMESTREAM | TPCDS |VERTICA`).

  Il tipo di connessione. Attualmente, SFTP non è supportato.

- `MatchCriteria`: una matrice di stringhe UTF-8, non superiore a 10 stringhe.

  Un elenco di criteri che possono essere utilizzati nella selezione di questa connessione.

- `ConnectionProperties`: una matrice di mappe con coppie chiave-valore, non superiore alle 100 coppie.

  Ogni chiave è una stringa UTF-8 (valori validi: `HOST | PORT | USERNAME="USER_NAME" | PASSWORD | ENCRYPTED_PASSWORD | JDBC_DRIVER_JAR_URI | JDBC_DRIVER_CLASS_NAME`

| JDBC_ENGINE | JDBC_ENGINE_VERSION | CONFIG_FILES | INSTANCE_ID
| | JDBC_CONNECTION_URL | JDBC_ENFORCE_SSL | CUSTOM_JDBC_CERT |
SKIP_CUSTOM_JDBC_CERT_VALIDATION | CUSTOM_JDBC_CERT_STRING |
CONNECTION_URL | KAFKA_BOOTSTRAP_SERVERS | KAFKA_SSL_ENABLED
| KAFKA_CUSTOM_CERT | KAFKA_SKIP_CUSTOM_CERT_VALIDATION |
KAFKA_CLIENT_KEYSTORE | KAFKA_CLIENT_KEYSTORE_PASSWORD |
KAFKA_CLIENT_KEY_PASSWORD | ENCRYPTED_KAFKA_CLIENT_KEYSTORE_PASSWORD
| ENCRYPTED_KAFKA_CLIENT_KEY_PASSWORD | | KAFKA_SASL_MECHANISM
| KAFKA_SASL_PLAIN_USERNAME | KAFKA_SASL_PLAIN_PASSWORD |
ENCRYPTED_KAFKA_SASL_PLAIN_PASSWORD | KAFKA_SASL_SCRAM_USERNAME
| KAFKA_SASL_SCRAM_PASSWORD | KAFKA_SASL_SCRAM_SECRETS_ARN |
ENCRYPTED_KAFKA_SASL_SCRAM_PASSWORD KAFKA_SASL_GSSAPI_KEYTAB
KAFKA_SASL_GSSAPI_KRB5_CONF | KAFKA_SASL_GSSAPI_SERVICE |
KAFKA_SASL_GSSAPI_PRINCIPAL | SECRET_ID | CONNECTOR_URL | CONNECTOR_TYPE
| | CONNECTOR_CLASS_NAME | ENDPOINT | ENDPOINT_TYPE | ROLE_ARN | REGION
WORKGROUP_NAME CLUSTER_IDENTIFIER |DATABASE).

Ogni valore è una stringa Value, non inferiore a 1 o superiore a 1.024 byte di lunghezza.

Queste coppie chiave-valore definiscono i parametri per la connessione quando si utilizza lo schema di connessione versione 1:

- HOST- L'URI dell'host: il nome di dominio completo (FQDN) o l' IPv4 indirizzo dell'host del database.
- PORT: il numero della porta, compreso tra 1024 e 65535, della porta su cui l'host del database ascolta le connessioni del database.
- USER_NAME: il nome sotto il quale accedere al database. La stringa del valore per USER_NAME è USERNAME.
- PASSWORD: una password, se usata, per il nome utente.
- ENCRYPTED_PASSWORD: quando si abilita la protezione della password di connessione impostando ConnectionPasswordEncryption nelle impostazioni di crittografia del catalogo dati, questo campo memorizza la password crittografata.
- JDBC_DRIVER_JAR_URI: il percorso Amazon Simple Storage Service (Amazon S3) del file JAR che contiene il driver JDBC da utilizzare.
- JDBC_DRIVER_CLASS_NAME: il nome classe del driver JDBC da utilizzare.
- JDBC_ENGINE: il nome del motore JDBC da utilizzare.

- JDBC_ENGINE_VERSION: la versione del motore JDBC da utilizzare.
- CONFIG_FILES: (Riservato per uso futuro).
- INSTANCE_ID: l'ID istanza da utilizzare.
- JDBC_CONNECTION_URL: l'URL per la connessione a un'origine dati JDBC.
- JDBC_ENFORCE_SSL: una stringa booleana (true, false) che fa distinzione tra maiuscole e minuscole che specifica se il protocollo Secure Sockets Layer (SSL) con nome host corrispondente viene applicato per la connessione JDBC al client. Il valore predefinito è false.
- CUSTOM_JDBC_CERT- Una posizione Amazon S3 che specifica il certificato principale del cliente. AWS Glue utilizza questo certificato radice per convalidare il certificato del cliente durante la connessione al database dei clienti. AWS Glue gestisce solo certificati X.509. Il certificato fornito deve essere codificato DER e fornito in formato PEM con codifica Base64.
- SKIP_CUSTOM_JDBC_CERT_VALIDATION- Per impostazione predefinita, questo è. `false` AWS Glue convalida l'algoritmo Signature e Subject Public Key Algorithm per il certificato del cliente. Gli unici algoritmi consentiti per l'algoritmo Signature sono SHA256with RSA, RSA o SHA384with RSA. SHA512with Per l'algoritmo della chiave pubblica oggetto, la lunghezza della chiave deve essere almeno 2048. Puoi impostare il valore di questa proprietà su `true` per ignorare la convalida di AWS Glue del certificato del cliente.
- CUSTOM_JDBC_CERT_STRING- Una stringa di certificato JDBC personalizzata che viene utilizzata per la corrispondenza tra domini o nomi distinti per prevenire un attacco. man-in-the-middle Nel database Oracle, viene utilizzato come SSL_SERVER_CERT_DN; in Microsoft SQL Server, viene utilizzato come `hostNameInCertificate`.
- CONNECTION_URL: l'URL per la connessione a un'origine dati generale (non JDBC).
- SECRET_ID: l'ID segreto utilizzato per il Secret Manager delle credenziali.
- CONNECTOR_URL: l'URL del connettore per una connessione MARKETPLACE o CUSTOM.
- CONNECTOR_TYPE: il tipo di connettore per una connessione MARKETPLACE o CUSTOM.
- CONNECTOR_CLASS_NAME: il nome di classe del connettore per una connessione MARKETPLACE o CUSTOM.
- KAFKA_BOOTSTRAP_SERVERS: un elenco separato da virgole di coppie host e porte che sono gli indirizzi dei broker Apache Kafka in un cluster Kafka a cui un client Kafka si connetterà e si avvierà.
- KAFKA_SSL_ENABLED: indica se abilitare o disabilitare SSL su una connessione Apache Kafka. Il valore di default è "true".

- KAFKA_CUSTOM_CERT: l'URL Amazon S3 per il file di certificazione CA privato (formato .pem). L'impostazione predefinita è una stringa vuota.
- KAFKA_SKIP_CUSTOM_CERT_VALIDATION- Se saltare o meno la convalida del file di certificato CA. AWS Glue esegue la convalida per tre algoritmi: SHA256with RSA, RSA e RSA. SHA384with SHA512with Il valore predefinito è "false".
- KAFKA_CLIENT_KEYSTORE: la posizione Amazon S3 del file keystore del client per l'autenticazione lato client Kafka (facoltativo).
- KAFKA_CLIENT_KEYSTORE_PASSWORD: la password per accedere al keystore fornito (facoltativo).
- KAFKA_CLIENT_KEY_PASSWORD: un keystore può essere costituito da più chiavi, quindi questa è la password per accedere alla chiave client da utilizzare con la chiave lato server Kafka (facoltativo).
- ENCRYPTED_KAFKA_CLIENT_KEYSTORE_PASSWORD- La versione crittografata della password del keystore del client Kafka (se l'utente ha selezionato l'impostazione di crittografia delle password). AWS Glue
- ENCRYPTED_KAFKA_CLIENT_KEY_PASSWORD- La versione crittografata della password della chiave del client Kafka (se l'utente ha selezionato l'impostazione di crittografia delle password). AWS Glue
- KAFKA_SASL_MECHANISM: "SCRAM-SHA-512", "GSSAPI", "AWS_MSK_IAM" o "PLAIN". Queste sono i due meccanismi SASL supportati.
- KAFKA_SASL_PLAIN_USERNAME: un nome utente in testo normale utilizzato per l'autenticazione con il meccanismo “PLAIN”.
- KAFKA_SASL_PLAIN_PASSWORD: una password in testo normale utilizzata per l'autenticazione con il meccanismo “PLAIN”.
- ENCRYPTED_KAFKA_SASL_PLAIN_PASSWORD- La versione crittografata della password Kafka SASL PLAIN (se l'utente ha selezionato l'impostazione di crittografia delle password). AWS Glue
- KAFKA_SASL_SCRAM_USERNAME: un nome utente in testo semplice utilizzato per autenticarsi con il meccanismo "SCRAM-SHA-512".
- KAFKA_SASL_SCRAM_PASSWORD: una password in testo semplice utilizzata per autenticarsi con il meccanismo "SCRAM-SHA-512".
- ENCRYPTED_KAFKA_SASL_SCRAM_PASSWORD- La versione crittografata della password Kafka SASL SCRAM (se l'utente ha selezionato l'impostazione di crittografia delle password). AWS Glue

- KAFKA_SASL_SCRAM_SECRETS_ARN- Il nome di risorsa Amazon di un segreto in AWS Secrets Manager.
- KAFKA_SASL_GSSAPI_KEYTAB: la posizione S3 di un file keytab Kerberos. Un keytab memorizza le chiavi a lungo termine per uno o più principali. Per ulteriori informazioni, consulta la Documentazione di MIT Kerberos: keytab.
- KAFKA_SASL_GSSAPI_KRB5_CONF: la posizione S3 di un file krb5.conf Kerberos. Un krb5.conf memorizza le informazioni di configurazione Kerberos, ad esempio la posizione del server KDC. Per ulteriori informazioni, consulta la Documentazione di MIT Kerberos: krb5.conf.
- KAFKA_SASL_GSSAPI_SERVICE: il nome del servizio Kerberos, come impostato con sasl.kerberos.service.name nella Configurazione Kafka.
- KAFKA_SASL_GSSAPI_PRINCIPAL- Il nome del principale Kerberos utilizzato da. AWS Glue Per ulteriori informazioni, consulta la Documentazione di Kafka: configurazione dei broker Kafka.
- ROLE_ARN: il ruolo da utilizzare per l'esecuzione delle query.
- REGION- La AWS regione in cui verranno eseguite le interrogazioni.
- WORKGROUP_NAME: il nome di un gruppo di lavoro Amazon Redshift serverless o Amazon Athena in cui verranno eseguite le query.
- CLUSTER_IDENTIFIER: l'identificatore del cluster Amazon Redshift in cui verranno eseguite le query.
- DATABASE: il database Amazon Redshift a cui stai effettuando la connessione.

- SparkProperties: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 2.048 byte di lunghezza.

  Proprietà di connessione specifiche dell'ambiente di calcolo Spark.
- AthenaProperties: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 2.048 byte di lunghezza.

  Proprietà di connessione specifiche dell'ambiente di calcolo Athena.
- PythonProperties: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 2.048 byte di lunghezza.

Proprietà di connessione specifiche dell'ambiente di calcolo Python.

- `PhysicalConnectionRequirements`: un oggetto PhysicalConnectionRequirements.

  I requisiti di connessione fisici, come un cloud privato virtuale (VPC) e un `SecurityGroup`, necessari per effettuare questa connessione.

- `CreationTime`: timestamp.

  Il timestamp dell'ora in cui è stata creata questa definizione della connessione.

- `LastUpdatedTime`: timestamp.

  Il timestamp dell'ultima volta in cui è stata aggiornata la definizione della connessione.

- `LastUpdatedBy`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'utente, gruppo o ruolo che ha aggiornato per ultimo questa definizione di connessione.

- `Status`: stringa UTF-8 (valori validi: `READY` | `IN_PROGRESS` | `FAILED`).

  Lo stato della connessione. Può essere `READY`, `IN_PROGRESS` o `FAILED`.

- `StatusReason`: stringa UTF-8, non inferiore a 1 o superiore a 16.384 byte di lunghezza.

  Il motivo dello stato della connessione.

- `LastConnectionValidationTime`: timestamp.

  Un timestamp dell'ora in cui questa connessione è stata convalidata l'ultima volta.

- `AuthenticationConfiguration`: un oggetto AuthenticationConfiguration.

  Le proprietà di autenticazione della connessione.

- `ConnectionSchemaVersion`: numero (intero), non inferiore a 1 o superiore a 2.

  La versione dello schema di connessione per questa connessione. La versione 2 supporta proprietà per ambienti di calcolo specifici.

- `CompatibleComputeEnvironments`: una matrice di stringhe UTF-8.

  Un elenco di ambienti di calcolo compatibili con la connessione.

ConnectionInput struttura

Una struttura utilizzata per specificare una connessione da creare o aggiornare.

Campi

- Name - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome della connessione.

- Description: stringa di descrizione, non superiore a 2048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  La descrizione della connessione.

- ConnectionType— Obbligatoria: stringa UTF-8 (valori validi: JDBC SFTP MONGODB | KAFKA | | NETWORK | MARKETPLACE | CUSTOM | SALESFORCE | VIEW_VALIDATION_REDSHIFT | VIEW_VALIDATION_ATHENA | | GOOGLEADS | GOOGLESHEETS | GOOGLEANALYTICS4 | SERVICENOW | MARKETO | SAPODATA | | ZENDESK | JIRACLOUD | NETSUITEERP | HUBSPOT | FACEBOOKADS | INSTAGRAMADS | | ZOHOCRM | SALESFORCEPARDOT | SALESFORCEMARKETINGCLOUD | ADOBEANALYTICS | SLACK | LINKEDIN | | MIXPANEL | ASANA | STRIPE | SMARTSHEET DATADOG WOOCOMMERCE| INTERCOM SNAPCHATADS | PAYPAL | QUICKBOOKS | FACEBOOKPAGEINSIGHTS | FRESHDESK | TWILIO DOCUSIGNMONITOR | FRESHSALES | ZOOM | GOOGLESEARCHCONSOLE | SALESFORCECOMMERCECLOUD | SAPCONCUR DYNATRACE | MICROSOFTDYNAMIC365FINANCEANDOPS | MICROSOFTTEAMS | BLACKBAUDRAISEREDGENXT | MAILCHIMP | GITLAB | PENDO PRODUCTBOARD | CIRCLECI | PIPEDIVE | SENDGRID | AZURECOSMOS AZURESQL | BIGQUERY | BLACKBAUD | CLOUDERAHIVE | CLOUDERAIMPALA | CLOUDWATCH CLOUDWATCHMETRICS | CMDB | DATALAKEGEN2 | DB2 | DB2AS400 DOCUMENTDB DOMO| DYNAMODB | GOOGLECLOUDSTORAGE | HBASE | KUSTOMER | MICROSOFTDYNAMICS365CRM | MONDAY | MYSQL | OKTA | OPENSEARCH | ORACLE | PIPEDRIVE | POSTGRESQL | SAPHANA | SQLSERVER | SYNAPSE | TERADATA | TERADATANOS | TIMESTREAM | TPCDS |VERTICA).

  Il tipo di connessione. Attualmente, sono supportati questi tipi:

  - JDBC: designa una connessione a un database tramite Java Database Connectivity (JDBC).

    JDBCLe connessioni utilizzano quanto segue ConnectionParameters.

    - Obbligatorio: tutti (HOST, PORT, JDBC_ENGINE) o JDBC_CONNECTION_URL.
    - Obbligatorio: tutti (USERNAME, PASSWORD) o SECRET_ID.

  - Facoltativo: JDBC_ENFORCE_SSL, CUSTOM_JDBC_CERT, CUSTOM_JDBC_CERT_STRING, SKIP_CUSTOM_JDBC_CERT_VALIDATION. Questi parametri vengono utilizzati per configurare SSL con JDBC.
- KAFKA: indica una connessione a una piattaforma di streaming Apache Kafka.

  KAFKALe connessioni utilizzano quanto segue ConnectionParameters.

  - Obbligatorio: KAFKA_BOOTSTRAP_SERVERS.
  - Facoltativo: KAFKA_SSL_ENABLED, KAFKA_CUSTOM_CERT, KAFKA_SKIP_CUSTOM_CERT_VALIDATION. Questi parametri vengono utilizzati per configurare SSL con KAFKA.
  - Facoltativo: KAFKA_CLIENT_KEYSTORE, KAFKA_CLIENT_KEYSTORE_PASSWORD, KAFKA_CLIENT_KEY_PASSWORD, ENCRYPTED_KAFKA_CLIENT_KEYSTORE_PASSWORD, ENCRYPTED_KAFKA_CLIENT_KEY_PASSWORD. Questi parametri vengono utilizzati per impostare la configurazione del client TLS con SSL in KAFKA.
  - Facoltativo: KAFKA_SASL_MECHANISM. Può essere specificato come SCRAM-SHA-512, GSSAPI o AWS_MSK_IAM.
  - Facoltativo: KAFKA_SASL_SCRAM_USERNAME, KAFKA_SASL_SCRAM_PASSWORD, ENCRYPTED_KAFKA_SASL_SCRAM_PASSWORD. Questi parametri vengono utilizzati per configurare l'autenticazione SASL/SCRAM-SHA-512 con KAFKA.
  - Facoltativo: KAFKA_SASL_GSSAPI_KEYTAB, KAFKA_SASL_GSSAPI_KRB5_CONF, KAFKA_SASL_GSSAPI_SERVICE, KAFKA_SASL_GSSAPI_PRINCIPAL. Questi parametri vengono utilizzati per configurare SASL/GSSAPI l'autenticazione conKAFKA.
- MONGODB: designa una connessione a un database di documenti MongoDB.

  MONGODBLe connessioni utilizzano quanto segue ConnectionParameters.

  - Obbligatorio: CONNECTION_URL.
  - Obbligatorio: tutti (USERNAME, PASSWORD) o SECRET_ID.
- VIEW_VALIDATION_REDSHIFT: indica una connessione utilizzata per la convalida della visualizzazione da parte di Amazon Redshift.
- VIEW_VALIDATION_ATHENA: indica una connessione utilizzata per la convalida della visualizzazione da parte di Amazon Athena.
- NETWORK: designa una connessione di rete a un'origine dati all'interno di un ambiente Amazon Virtual Private Cloud (Amazon VPC).

`NETWORK`Le connessioni non richiedono ConnectionParameters. Fornisci invece un PhysicalConnectionRequirements.

- `MARKETPLACE`- Utilizza le impostazioni di configurazione contenute in un connettore acquistato Marketplace AWS per leggere e scrivere su archivi dati che non sono supportati nativamente da AWS Glue.

  `MARKETPLACE`Le connessioni utilizzano quanto segue ConnectionParameters.

  - Obbligatorio: `CONNECTOR_TYPE`, `CONNECTOR_URL`, `CONNECTOR_CLASS_NAME`, `CONNECTION_URL`.
  - Obbligatorio per le connessioni `JDBC` `CONNECTOR_TYPE`: tutti (`USERNAME`, `PASSWORD`) o `SECRET_ID`.
- `CUSTOM`: utilizza le impostazioni di configurazione contenute in un connettore per leggere e scrivere in archivi dati non supportati nativamente da AWS Glue.

  Per ulteriori informazioni sui parametri di connessione necessari per un particolare connettore, consultate la documentazione relativa al connettore in Aggiungere una AWS Glue connessione nella Guida per l' AWS Glue utente.

  `SFTP` non è supportato.

  Per ulteriori informazioni sull'utilizzo delle funzionalità ConnectionProperties opzionali in AWS Glue, consulta le proprietà di AWS Glue connessione.

  Per ulteriori informazioni su come ConnectionProperties vengono utilizzate le funzionalità opzionali per configurare le funzionalità in AWS Glue Studio, consulta Utilizzo di connettori e connessioni.
- `MatchCriteria`: una matrice di stringhe UTF-8, non superiore a 10 stringhe.

  Un elenco di criteri che possono essere utilizzati nella selezione di questa connessione.
- `ConnectionProperties`: obbligatorio: una matrice di mappe di coppie chiave-valore, non superiore a 100 coppie.

  Ogni chiave è una stringa UTF-8 (valori validi: `HOST` | `PORT` | `USERNAME="USER_NAME"` | `PASSWORD` | `ENCRYPTED_PASSWORD` | | `JDBC_DRIVER_JAR_URI` | `JDBC_DRIVER_CLASS_NAME` | `JDBC_ENGINE` | `JDBC_ENGINE_VERSION` | `CONFIG_FILES` | `INSTANCE_ID` | `JDBC_CONNECTION_URL` | `JDBC_ENFORCE_SSL` | `CUSTOM_JDBC_CERT` | `SKIP_CUSTOM_JDBC_CERT_VALIDATION` | `CUSTOM_JDBC_CERT_STRING` | | `CONNECTION_URL` | `KAFKA_BOOTSTRAP_SERVERS` | `KAFKA_SSL_ENABLED`

| KAFKA_CUSTOM_CERT | KAFKA_SKIP_CUSTOM_CERT_VALIDATION | KAFKA_CLIENT_KEYSTORE | KAFKA_CLIENT_KEYSTORE_PASSWORD | KAFKA_CLIENT_KEY_PASSWORD | ENCRYPTED_KAFKA_CLIENT_KEYSTORE_PASSWORD | ENCRYPTED_KAFKA_CLIENT_KEY_PASSWORD | KAFKA_SASL_MECHANISM | | KAFKA_SASL_PLAIN_USERNAME | KAFKA_SASL_PLAIN_PASSWORD | ENCRYPTED_KAFKA_SASL_PLAIN_PASSWORD | KAFKA_SASL_SCRAM_USERNAME | KAFKA_SASL_SCRAM_PASSWORD | KAFKA_SASL_SCRAM_SECRETS_ARN | ENCRYPTED_KAFKA_SASL_SCRAM_PASSWORD KAFKA_SASL_GSSAPI_KEYTAB KAFKA_SASL_GSSAPI_KRB5_CONF | KAFKA_SASL_GSSAPI_SERVICE | KAFKA_SASL_GSSAPI_PRINCIPAL | SECRET_ID | CONNECTOR_URL | CONNECTOR_TYPE | | CONNECTOR_CLASS_NAME | ENDPOINT | ENDPOINT_TYPE | ROLE_ARN | REGION WORKGROUP_NAME CLUSTER_IDENTIFIER |DATABASE).

Ogni valore è una stringa Value, non inferiore a 1 o superiore a 1.024 byte di lunghezza.

Queste coppie chiave-valore definiscono i parametri per la connessione.

- `SparkProperties`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 2.048 byte di lunghezza.

  Proprietà di connessione specifiche dell'ambiente di calcolo Spark.
- `AthenaProperties`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 2.048 byte di lunghezza.

  Proprietà di connessione specifiche dell'ambiente di calcolo Athena.
- `PythonProperties`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 2.048 byte di lunghezza.

  Proprietà di connessione specifiche dell'ambiente di calcolo Python.
- `PhysicalConnectionRequirements`: un oggetto PhysicalConnectionRequirements.

I requisiti di connessione fisici, come un cloud privato virtuale (VPC) e un `SecurityGroup`, necessari per effettuare correttamente questa connessione.

- `AuthenticationConfiguration`: un oggetto AuthenticationConfigurationInput.

  Le proprietà di autenticazione della connessione.

- `ValidateCredentials`: booleano.

  Una spunta per convalidare le credenziali durante la creazione della connessione. Il valore predefinito è true.

- `ValidateForComputeEnvironments`: una matrice di stringhe UTF-8.

  Gli ambienti di calcolo in cui vengono convalidate le proprietà di connessione specificate.

TestConnectionInput struttura

Una struttura utilizzata per specificare il test di una connessione a un servizio.

Campi

- `ConnectionType`— Obbligatoria: stringa UTF-8 (valori validi: `JDBC` `SFTP` `MONGODB` | `KAFKA` | | `NETWORK` | `MARKETPLACE` | `CUSTOM` | `SALESFORCE` | `VIEW_VALIDATION_REDSHIFT` | `VIEW_VALIDATION_ATHENA` | | `GOOGLEADS` | `GOOGLESHEETS` | `GOOGLEANALYTICS4` | `SERVICENOW` | `MARKETO` | `SAPODATA` | | `ZENDESK` | `JIRACLOUD` | `NETSUITEERP` | `HUBSPOT` | `FACEBOOKADS` | `INSTAGRAMADS` | | `ZOHOCRM` | `SALESFORCEPARDOT` | `SALESFORCEMARKETINGCLOUD` | `ADOBEANALYTICS` | `SLACK` | `LINKEDIN` | | `MIXPANEL` | `ASANA` | `STRIPE` | `SMARTSHEET` `DATADOG` `WOOCOMMERCE`| `INTERCOM` `SNAPCHATADS` | `PAYPAL` | `QUICKBOOKS` | `FACEBOOKPAGEINSIGHTS` | `FRESHDESK` | `TWILIO` `DOCUSIGNMONITOR` | `FRESHSALES` | `ZOOM` | `GOOGLESEARCHCONSOLE` | `SALESFORCECOMMERCECLOUD` | `SAPCONCUR` `DYNATRACE` | `MICROSOFTDYNAMIC365FINANCEANDOPS` | `MICROSOFTTEAMS` | `BLACKBAUDRAISEREDGENXT` | `MAILCHIMP` | `GITLAB` | `PENDO` `PRODUCTBOARD` | `CIRCLECI` | `PIPEDIVE` | `SENDGRID` | `AZURECOSMOS` `AZURESQL` | `BIGQUERY` | `BLACKBAUD` | `CLOUDERAHIVE` | `CLOUDERAIMPALA` | `CLOUDWATCH` `CLOUDWATCHMETRICS` | `CMDB` | `DATALAKEGEN2` | `DB2` | `DB2AS400` `DOCUMENTDB` `DOMO`| `DYNAMODB` | `GOOGLECLOUDSTORAGE` | `HBASE` | `KUSTOMER` | `MICROSOFTDYNAMICS365CRM` | `MONDAY` | `MYSQL` | `OKTA` | `OPENSEARCH` | `ORACLE` | `PIPEDRIVE` | `POSTGRESQL` | `SAPHANA` | `SQLSERVER` | `SYNAPSE` | `TERADATA` | `TERADATANOS` | `TIMESTREAM` | `TPCDS` |`VERTICA`).

Il tipo di connessione da testare. Questa operazione è disponibile solo per i tipi di connessione JDBC o SALESFORCE.

- ConnectionProperties: obbligatorio: una matrice di mappe di coppie chiave-valore, non superiore a 100 coppie.

  Ogni chiave è una stringa UTF-8 (valori validi: HOST | PORT | USERNAME="USER_NAME" | PASSWORD | ENCRYPTED_PASSWORD | JDBC_DRIVER_JAR_URI | JDBC_DRIVER_CLASS_NAME | JDBC_ENGINE | JDBC_ENGINE_VERSION | CONFIG_FILES | INSTANCE_ID | | JDBC_CONNECTION_URL | JDBC_ENFORCE_SSL | CUSTOM_JDBC_CERT | SKIP_CUSTOM_JDBC_CERT_VALIDATION | CUSTOM_JDBC_CERT_STRING | CONNECTION_URL | KAFKA_BOOTSTRAP_SERVERS | KAFKA_SSL_ENABLED | KAFKA_CUSTOM_CERT | KAFKA_SKIP_CUSTOM_CERT_VALIDATION | KAFKA_CLIENT_KEYSTORE | KAFKA_CLIENT_KEYSTORE_PASSWORD | KAFKA_CLIENT_KEY_PASSWORD | ENCRYPTED_KAFKA_CLIENT_KEYSTORE_PASSWORD | ENCRYPTED_KAFKA_CLIENT_KEY_PASSWORD | | KAFKA_SASL_MECHANISM | KAFKA_SASL_PLAIN_USERNAME | KAFKA_SASL_PLAIN_PASSWORD | ENCRYPTED_KAFKA_SASL_PLAIN_PASSWORD | KAFKA_SASL_SCRAM_USERNAME | KAFKA_SASL_SCRAM_PASSWORD | KAFKA_SASL_SCRAM_SECRETS_ARN | ENCRYPTED_KAFKA_SASL_SCRAM_PASSWORD KAFKA_SASL_GSSAPI_KEYTAB KAFKA_SASL_GSSAPI_KRB5_CONF | KAFKA_SASL_GSSAPI_SERVICE | KAFKA_SASL_GSSAPI_PRINCIPAL | SECRET_ID | CONNECTOR_URL | CONNECTOR_TYPE | | CONNECTOR_CLASS_NAME | ENDPOINT | ENDPOINT_TYPE | ROLE_ARN | REGION WORKGROUP_NAME CLUSTER_IDENTIFIER |DATABASE).

  Ogni valore è una stringa Value, non inferiore a 1 o superiore a 1.024 byte di lunghezza.

  Le coppie chiave-valore che definiscono i parametri per la connessione.

  Le connessioni JDBC utilizzano le seguenti proprietà di connessione:

  - Obbligatorio: tutti (HOST, PORT, JDBC_ENGINE) o JDBC_CONNECTION_URL.
  - Obbligatorio: tutti (USERNAME, PASSWORD) o SECRET_ID.
  - Facoltativo: JDBC_ENFORCE_SSL, CUSTOM_JDBC_CERT, CUSTOM_JDBC_CERT_STRING, SKIP_CUSTOM_JDBC_CERT_VALIDATION. Questi parametri vengono utilizzati per configurare SSL con JDBC.

Le connessioni SALESFORCE richiedono la configurazione del membro `AuthenticationConfiguration`.

- `AuthenticationConfiguration`: un oggetto AuthenticationConfigurationInput.

  Una struttura contenente la configurazione di autenticazione nella TestConnection richiesta. Richiesto per una connessione a Salesforce tramite OAuth autenticazione.

PhysicalConnectionRequirements struttura

L'app OAuth client in GetConnection risposta.

Campi

- `SubnetId`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID della sottorete utilizzato dalla connessione.
- `SecurityGroupIdList`: una matrice di stringhe UTF-8, non superiore a 50 stringhe.

  L'elenco di ID del gruppo di sicurezza utilizzato dalla connessione.
- `AvailabilityZone`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  La zona di disponibilità della connessione.

GetConnectionsFilter struttura

Filtra le definizioni di connessione restituite dall'operazione API `GetConnections`.

Campi

- `MatchCriteria`: una matrice di stringhe UTF-8, non superiore a 10 stringhe.

  Una stringa di criteri che deve corrispondere ai criteri registrati nella definizione di connessione affinché venga restituita quella definizione di connessione.
- `ConnectionType`— stringa UTF-8 (valori validi: `JDBC` | | `SFTP` | `MONGODB` | `KAFKA` | | `NETWORK` | `MARKETPLACE` | `CUSTOM` | `SALESFORCE` | `VIEW_VALIDATION_REDSHIFT` | | `VIEW_VALIDATION_ATHENA` | `GOOGLEADS` | `GOOGLESHEETS` | `GOOGLEANALYTICS4`

| SERVICENOW | MARKETO | | SAPODATA | ZENDESK | JIRACLOUD | NETSUITEERP | HUBSPOT | | FACEBOOKADS | INSTAGRAMADS | ZOHOCRM | SALESFORCEPARDOT | SALESFORCEMARKETINGCLOUD | | ADOBEANALYTICS | SLACK | LINKEDIN | MIXPANEL | ASANA | | STRIPE | SMARTSHEET | DATADOG | WOOCOMMERCE INTERCOM SNAPCHATADS| PAYPAL | QUICKBOOKS | FACEBOOKPAGEINSIGHTS FRESHDESK | TWILIO | DOCUSIGNMONITOR | FRESHSALES | ZOOM | GOOGLESEARCHCONSOLE | SALESFORCECOMMERCECLOUD | SAPCONCUR | DYNATRACE MICROSOFTDYNAMIC365FINANCEANDOPS | MICROSOFTTEAMS | BLACKBAUDRAISEREDGENXT | MAILCHIMP | GITLAB | PENDO | PRODUCTBOARD | CIRCLECI | PIPEDIVE SENDGRID | AZURECOSMOS | AZURESQL | BIGQUERY | BLACKBAUD | CLOUDERAHIVE | CLOUDERAIMPALA | CLOUDWATCH | CLOUDWATCHMETRICS CMDB | DATALAKEGEN2 | DB2 | DB2AS400 | DOCUMENTDB | DOMO DYNAMODB GOOGLECLOUDSTORAGE| HBASE | KUSTOMER | MICROSOFTDYNAMICS365CRM | MONDAY | MYSQL | OKTA | OPENSEARCH | ORACLE | PIPEDRIVE | | POSTGRESQL | SAPHANA | SQLSERVER | SYNAPSE | TERADATA | TERADATANOS | TIMESTREAM | TPCDS |VERTICA).

  Il tipo di connessioni da restituire. Attualmente, SFTP non è supportato.

- `ConnectionSchemaVersion`: numero (intero), non inferiore a 1 o superiore a 2.

  Indica se la connessione è stata creata con la versione 1 o 2 dello schema.

AuthenticationConfiguration struttura

Una struttura contenente la configurazione dell'autenticazione.

Campi

- `AuthenticationType`: stringa UTF-8 (valori validi: `BASIC` | `OAUTH2` | `CUSTOM` | `IAM`).

  Una struttura contenente la configurazione dell'autenticazione.

- `SecretArn`: stringa UTF-8, corrispondente a Custom string pattern #36.

  L'ARN del gestore segreto per archiviare le credenziali.

- `KmsKeyArn`: stringa UTF-8, corrispondente a Custom string pattern #42.

  Il nome della risorsa Amazon (ARN) della chiave KMS utilizzato per crittografare le informazioni di autenticazione sensibili. Questa chiave viene utilizzata per proteggere le credenziali e altri dati sensibili archiviati nella configurazione di autenticazione.

- `OAuth2Properties`: un oggetto OAuth2Proprietà.

Le proprietà per OAuth2 l'autenticazione.

### AuthenticationConfigurationInput struttura

Una struttura contenente la configurazione di autenticazione nella CreateConnection richiesta.

Campi

- `AuthenticationType`: stringa UTF-8 (valori validi: `BASIC` | `OAUTH2` | `CUSTOM` | `IAM`).

  Una struttura contenente la configurazione di autenticazione nella CreateConnection richiesta.
- `OAuth2Properties`: un oggetto OAuth2PropertiesInput.

  Le proprietà per OAuth2 l'autenticazione nella CreateConnection richiesta.
- `SecretArn`: stringa UTF-8, corrispondente a Custom string pattern #36.

  L'ARN del gestore segreto per memorizzare le credenziali nella richiesta. CreateConnection
- `KmsKeyArn`: stringa UTF-8, corrispondente a Custom string pattern #42.

  L'ARN della chiave KMS da utilizzare per crittografare la connessione. Viene preso solo come input nella richiesta e archiviato in Secrets Manager.
- `BasicAuthenticationCredentials`: un oggetto BasicAuthenticationCredentials.

  Le credenziali utilizzate quando il tipo di autenticazione è di base.
- `CustomAuthenticationCredentials`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 2.048 byte di lunghezza.

  Le credenziali utilizzate quando il tipo di autenticazione è personalizzato.

### OAuth2Struttura delle proprietà

Una struttura contenente proprietà per OAuth2 l'autenticazione.

Campi

- `OAuth2GrantType`: stringa UTF-8 (valori validi: `AUTHORIZATION_CODE` | `CLIENT_CREDENTIALS` | `JWT_BEARER`).

  Il tipo di OAuth2 concessione. Ad esempio, `AUTHORIZATION_CODE`, `JWT_BEARER` o `CLIENT_CREDENTIALS`.

- `OAuth2ClientApplication`: un oggetto OAuth2ClientApplication.

  Il tipo di applicazione client. Ad esempio, AWS_MANAGED o USER_MANAGED.

- `TokenUrl`: stringa UTF-8, non superiore a 256 byte di lunghezza, corrispondente a Custom string pattern #40.

  L'URL del server di autenticazione del provider, per scambiare un codice di autorizzazione per un token di accesso.

- `TokenUrlParametersMap`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Ogni valore è una stringa UTF-8, non inferiore a 1 o superiore a 512 byte di lunghezza.

  Una mappa dei parametri che vengono aggiunti alla richiesta del token `GET`.

OAuth2PropertiesInput struttura

Una struttura contenente le proprietà per OAuth2 la CreateConnection richiesta.

Campi

- `OAuth2GrantType`: stringa UTF-8 (valori validi: `AUTHORIZATION_CODE` | `CLIENT_CREDENTIALS` | `JWT_BEARER`).

  Il tipo di OAuth2 concessione nella CreateConnection richiesta. Ad esempio, `AUTHORIZATION_CODE`, `JWT_BEARER` o `CLIENT_CREDENTIALS`.

- `OAuth2ClientApplication`: un oggetto OAuth2ClientApplication.

  Il tipo di applicazione client nella CreateConnection richiesta. Ad esempio `AWS_MANAGED` o `USER_MANAGED`.

- `TokenUrl`: stringa UTF-8, non superiore a 256 byte di lunghezza, corrispondente a Custom string pattern #40.

  L'URL del server di autenticazione del provider, per scambiare un codice di autorizzazione per un token di accesso.

- `TokenUrlParametersMap`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Ogni valore è una stringa UTF-8, non inferiore a 1 o superiore a 512 byte di lunghezza.

  Una mappa dei parametri che vengono aggiunti alla richiesta del token `GET`.

- `AuthorizationCodeProperties`: un oggetto AuthorizationCodeProperties.

  L'insieme di proprietà richieste per il tipo di OAuth2 `AUTHORIZATION_CODE` concessione.

- `OAuth2Credentials`: un oggetto OAuth2Credenziali.

  Le credenziali utilizzate quando il tipo di autenticazione è OAuth2 l'autenticazione.

OAuth2ClientApplication struttura

L'app OAuth2 client utilizzata per la connessione.

Campi

- `UserManagedClientApplicationClientId`: stringa UTF-8, non superiore a 2048 byte di lunghezza, corrispondente a Custom string pattern #37.

  L'applicazione client ClientID, se lo è ClientAppType . `USER_MANAGED`

- `AWSManagedClientApplicationReference`: stringa UTF-8, non superiore a 2048 byte di lunghezza, corrispondente a Custom string pattern #37.

  Il riferimento all'app client lato SaaS gestita. AWS

AuthorizationCodeProperties struttura

L'insieme di proprietà richieste per il flusso di lavoro relativo al tipo di OAuth2 `AUTHORIZATION_CODE` sovvenzione.

Campi

- `AuthorizationCode`: stringa UTF-8, non inferiore a 1 o superiore a 4.096 byte di lunghezza, corrispondente a Custom string pattern #37.

  Un codice di autorizzazione da utilizzare nella terza fase del flusso di lavoro di concessione `AUTHORIZATION_CODE`. Si tratta di un codice monouso che diventa non valido una volta scambiato con un token di accesso, quindi è accettabile avere tale valore come parametro di richiesta.
- `RedirectUri`: stringa UTF-8, non superiore a 512 byte di lunghezza, corrispondente a Custom string pattern #41.

  L'URI di reindirizzamento a cui l'utente viene reindirizzato dal server di autorizzazione quando emette un codice di autorizzazione. L'URI viene successivamente utilizzato quando il codice di autorizzazione viene scambiato per un token di accesso.

BasicAuthenticationCredentials struttura

Per assegnare credenziali di autenticazione di base quando non si fornisce un valore `SecretArn`.

Campi

- `Username`: stringa UTF-8, non superiore a 512 byte di lunghezza, corrispondente a Custom string pattern #37.

  Il nome utente da connettere all'origine dati.
- `Password`: stringa UTF-8, non superiore a 512 byte di lunghezza, corrispondente a Custom string pattern #33.

  La password per la connessione all'origine dati.

OAuth2struttura delle credenziali

Le credenziali utilizzate quando il tipo di autenticazione è OAuth2 l'autenticazione.

Campi

- `UserManagedClientApplicationClientSecret`: stringa UTF-8, non superiore a 512 byte di lunghezza, corrispondente a Custom string pattern #38.

Il segreto del client dell'applicazione client se è gestita dall'utente.

- `AccessToken`: stringa UTF-8, non superiore a 4.096 byte di lunghezza, corrispondente a Custom string pattern #38.

  Il token di accesso utilizzato quando il tipo di autenticazione è OAuth2.

- `RefreshToken`: stringa UTF-8, non superiore a 4.096 byte di lunghezza, corrispondente a Custom string pattern #38.

  Il token di aggiornamento utilizzato quando il tipo di autenticazione è OAuth2.

- `JwtToken`: stringa UTF-8, non superiore a 8.000 byte di lunghezza, corrispondente a Custom string pattern #39.

  Il JSON Web Token (JWT) utilizzato quando il tipo di autenticazione è. OAuth2

Operazioni

- CreateConnection azione (Python: create_connection)
- DeleteConnection azione (Python: delete_connection)
- GetConnection azione (Python: get_connection)
- GetConnections azione (Python: get_connections)
- UpdateConnection azione (Python: update_connection)
- TestConnection azione (Python: test_connection)
- BatchDeleteConnection azione (Python: batch_delete_connection)

CreateConnection azione (Python: create_connection)

Crea una definizione di connessione nel catalogo dati.

Le connessioni utilizzate per creare risorse federate richiedono l'autorizzazione IAM `glue:PassConnection`.

Richiesta

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

L'ID del catalogo dati in cui creare la connessione. Se non ne viene fornito nessuno, per impostazione predefinita viene utilizzato l'ID AWS dell'account.

- `ConnectionInput` - obbligatorio: un oggetto ConnectionInput.

  Un oggetto `ConnectionInput` che definisce la connessione da creare.

- `Tags`: una matrice di mappe con coppie chiave-valore, non superiore alle 50 coppie.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Ogni valore è una stringa UTF-8, lunga non più di 256 byte.

  I tag assegnati alla connessione.

Risposta

- `CreateConnectionStatus`: stringa UTF-8 (valori validi: `READY` | `IN_PROGRESS` | `FAILED`).

  Lo stato della richiesta di creazione della connessione. La richiesta può richiedere del tempo per determinati tipi di autenticazione, ad esempio quando si crea una OAuth connessione con token exchange tramite VPC.

Errori

- `AlreadyExistsException`
- `InvalidInputException`
- `OperationTimeoutException`
- `ResourceNumberLimitExceededException`
- `GlueEncryptionException`

DeleteConnection azione (Python: delete_connection)

Elimina una connessione dal catalogo dati.

Richiesta

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

L'ID del catalogo dati in cui risiede la connessione. Se non ne viene fornito nessuno, per impostazione predefinita viene utilizzato l'ID AWS dell'account.

- `ConnectionName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome della connessione da eliminare.

Risposta

- Nessun parametro di risposta.

Errori

- `EntityNotFoundException`
- `OperationTimeoutException`

GetConnection azione (Python: get_connection)

Recupera una definizione di connessione dal catalogo dati.

Richiesta

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo dati in cui risiede la connessione. Se non ne viene fornito nessuno, per impostazione predefinita viene utilizzato l'ID AWS dell'account.

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome della definizione di connessione da recuperare.

- `HidePassword`: booleano.

  Consente di recuperare i metadati di connessione senza restituire la password. Ad esempio, la AWS Glue console utilizza questo flag per recuperare la connessione e non visualizza la password. Imposta questo parametro quando il chiamante potrebbe non avere l'autorizzazione per utilizzare la

AWS KMS chiave per decrittografare la password, ma dispone dell'autorizzazione per accedere al resto delle proprietà della connessione.

- `ApplyOverrideForComputeEnvironment`: stringa UTF-8 (valori validi: `SPARK` | `ATHENA` | `PYTHON`).

  Per le connessioni che possono essere utilizzate in più servizi, specifica le proprietà di restituzione per l'ambiente di calcolo specificato.

Risposta

- `Connection`: un oggetto Connessione.

  La definizione di connessione richiesta.

Errori

- `EntityNotFoundException`
- `OperationTimeoutException`
- `InvalidInputException`
- `GlueEncryptionException`

GetConnections azione (Python: get_connections)

Recupera un elenco di definizioni di connessione dal catalogo dati.

Richiesta

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo dati in cui risiedono le connessioni. Se non ne viene fornito nessuno, per impostazione predefinita viene utilizzato l'ID AWS dell'account.

- `Filter`: un oggetto GetConnectionsFilter.

  Un filtro che controlla quali connessioni vengono restituite.

- `HidePassword`: booleano.

Consente di recuperare i metadati di connessione senza restituire la password. Ad esempio, la AWS Glue console utilizza questo flag per recuperare la connessione e non visualizza la password. Imposta questo parametro quando il chiamante potrebbe non avere l'autorizzazione per utilizzare la AWS KMS chiave per decrittografare la password, ma dispone dell'autorizzazione per accedere al resto delle proprietà della connessione.

- `NextToken`: stringa UTF-8.

  Un token di continuazione, se si tratta di una chiamata di continuazione.

- `MaxResults`: numero (intero), non inferiore a 1 o superiore a 1.000.

  Il numero massimo di connessioni da restituire in una risposta.

Risposta

- `ConnectionList`: una matrice di oggetti Connessione.

  Un elenco di definizioni di connessione richieste.

- `NextToken`: stringa UTF-8.

  Un token di continuazione, se l'elenco di connessioni restituite non include l'ultima delle connessioni filtrate.

Errori

- `EntityNotFoundException`
- `OperationTimeoutException`
- `InvalidInputException`
- `GlueEncryptionException`

UpdateConnection azione (Python: update_connection)

Aggiorna una definizione di connessione nel catalogo dati.

Richiesta

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

L'ID del catalogo dati in cui risiede la connessione. Se non ne viene fornito nessuno, per impostazione predefinita viene utilizzato l'ID AWS dell'account.

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome della definizione di connessione da aggiornare.

- `ConnectionInput` - obbligatorio: un oggetto ConnectionInput.

  Un oggetto `ConnectionInput` che ridefinisce la connessione in questione.

Risposta

- Nessun parametro di risposta.

Errori

- `InvalidInputException`
- `EntityNotFoundException`
- `OperationTimeoutException`
- `InvalidInputException`
- `GlueEncryptionException`

TestConnection azione (Python: test_connection)

Verifica una connessione a un servizio per convalidare le credenziali del servizio fornite.

Puoi fornire un nome di connessione esistente o un `TestConnectionInput` per testare un input di connessione inesistente. La fornitura di entrambi nello stesso momento genera un errore.

Se l'operazione riesce, il servizio restituisce una risposta HTTP 200.

Richiesta

- `ConnectionName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

Opzionale. Il nome della connessione da testare. Se viene fornito solo il nome, l'operazione otterrà la connessione e la utilizzerà per il test.

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo in cui si trova la connessione.

- `TestConnectionInput`: un oggetto TestConnectionInput.

  Una struttura utilizzata per specificare il test di una connessione a un servizio.

Risposta

- Nessun parametro di risposta.

Errori

- `InvalidInputException`
- `OperationTimeoutException`
- `ResourceNumberLimitExceededException`
- `GlueEncryptionException`
- `FederationSourceException`
- `AccessDeniedException`
- `EntityNotFoundException`
- `ConflictException`
- `InternalServiceException`

BatchDeleteConnection azione (Python: batch_delete_connection)

Elimina un elenco di definizioni di connessione dal catalogo dati.

Richiesta

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

L'ID del catalogo dati in cui risiedono le connessioni. Se non ne viene fornito nessuno, per impostazione predefinita viene utilizzato l'ID AWS dell'account.

- `ConnectionNameList`: obbligatorio: matrice di stringhe UTF-8, non superiore a 25 stringhe.

  Un elenco di nomi delle connessioni da eliminare.

Risposta

- `Succeeded`: una matrice di stringhe UTF-8.

  Un elenco di nomi delle definizioni di connessione la cui eliminazione è riuscita.

- `Errors`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Ogni valore è un oggetto ErrorDetail.

  Una mappa dei nomi delle connessioni la cui eliminazione non è riuscita per i dettagli di errore.

Errori

- `InternalServiceException`
- `OperationTimeoutException`

## API Tipi di connessione

L'API Connection Type descrive AWS Glue APIs come descrivere i tipi di connessione.

Gestione della connessione APIs

- DescribeConnectionType azione (Python: describe_connection_type)
- ListConnectionTypes azione (Python: list_connection_types)
- ConnectionTypeBrief struttura
- ConnectionTypeVariant struttura

DescribeConnectionType azione (Python: describe_connection_type)

L'API `DescribeConnectionType` fornisce tutti i dettagli delle opzioni supportate per un determinato tipo di connessione in AWS Glue.

Richiesta

- `ConnectionType` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del tipo di connessione da descrivere.

Risposta

- `ConnectionType`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del tipo di connessione.
- `Description`: stringa UTF-8, non superiore a 1.024 byte di lunghezza.

  Una descrizione del tipo di connessione.
- `Capabilities`: un oggetto Funzionalità.

  I tipi di autenticazione supportati, i tipi di interfaccia dati (ambienti di calcolo) e le operazioni sui dati del connettore.
- `ConnectionProperties`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Ogni valore è un oggetto Proprietà.

  Proprietà di connessione comuni in tutti gli ambienti di calcolo.
- `ConnectionOptions`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Ogni valore è un oggetto Proprietà.

  Restituisce proprietà che possono essere impostate durante la creazione di una connessione in `ConnectionInput.ConnectionProperties`. `ConnectionOptions`definisce i parametri che

possono essere impostati in uno script ETL di Spark nella mappatura delle opzioni di connessione passata a un dataframe.

- `AuthenticationConfiguration`: un oggetto AuthConfiguration.

  Il tipo di autenticazione utilizzato per la connessione.

- `ComputeEnvironmentConfigurations`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Ogni valore è un oggetto ComputeEnvironmentConfiguration.

  Gli ambienti di calcolo supportati dalla connessione.

- `PhysicalConnectionRequirements`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Ogni valore è un oggetto Proprietà.

  Requisiti fisici per una connessione, come le specifiche del VPC, della sottorete e del gruppo di sicurezza.

- `AthenaConnectionProperties`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Ogni valore è un oggetto Proprietà.

  Proprietà di connessione specifiche dell'ambiente di calcolo Athena.

- `PythonConnectionProperties`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Ogni valore è un oggetto Proprietà.

  Proprietà di connessione specifiche dell'ambiente di calcolo Python.

- `SparkConnectionProperties`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Ogni valore è un oggetto Proprietà.

  Proprietà di connessione specifiche dell'ambiente di calcolo Spark.

Errori

- `ValidationException`
- `InvalidInputException`
- `InternalServiceException`

ListConnectionTypes azione (Python: list_connection_types)

L'API `ListConnectionTypes` fornisce un meccanismo di individuazione in cui apprendere i tipi di connessione disponibili in AWS Glue. La risposta contiene un elenco di tipi di connessione con dettagli di alto livello su ciò che è supportato per ogni tipo di connessione. I tipi di connessione elencati sono l'insieme di opzioni supportate per il valore `ConnectionType` nell'API `CreateConnection`.

Richiesta

- `MaxResults`: numero (intero), non inferiore a 1 o superiore a 1.000.

  Numero massimo di risultati da restituire.
- `NextToken`: stringa UTF-8, non inferiore a 1 o superiore a 2048 byte di lunghezza, corrispondente a Custom string pattern #11.

  Un token di continuazione, se si tratta di una chiamata di continuazione.

Risposta

- `ConnectionTypes`: una matrice di oggetti ConnectionTypeBrief.

  Un elenco di oggetti `ConnectionTypeBrief` contenente brevi informazioni sui tipi di connessione supportati.
- `NextToken`: stringa UTF-8, non inferiore a 1 o superiore a 2048 byte di lunghezza, corrispondente a Custom string pattern #11.

  Un token di continuazione, se il segmento dell'elenco corrente non è l'ultimo.

Errori

- `InternalServiceException`

ConnectionTypeBrief struttura

Brevi informazioni su un tipo di connessione supportato restituito dall'API `ListConnectionTypes`.

Campi

- `ConnectionType`— stringa UTF-8 (valori validi: `JDBC` `SFTP` `MONGODB` | `KAFKA` | `NETWORK` | `MARKETPLACE` | `CUSTOM` | `SALESFORCE` | `VIEW_VALIDATION_REDSHIFT` | `VIEW_VALIDATION_ATHENA` | `GOOGLEADS` | `GOOGLESHEETS` | `GOOGLEANALYTICS4` | `SERVICENOW` | `MARKETO` | `SAPODATA` | `ZENDESK` | `JIRACLOUD` | `NETSUITEERP` | `HUBSPOT` | `FACEBOOKADS` | `INSTAGRAMADS` `ZOHOCRM` | `SALESFORCEPARDOT` | `SALESFORCEMARKETINGCLOUD` | `ADOBEANALYTICS` | `SLACK` | `LINKEDIN` | `MIXPANEL` | `ASANA` | `STRIPE` | `SMARTSHEET` | `DATADOG` | `WOOCOMMERCE` `INTERCOM` `SNAPCHATADS`| `PAYPAL` | `QUICKBOOKS` | `FACEBOOKPAGEINSIGHTS` `FRESHDESK` | `TWILIO` | `DOCUSIGNMONITOR` | `FRESHSALES` | `ZOOM` | `GOOGLESEARCHCONSOLE` | `SALESFORCECOMMERCECLOUD` | `SAPCONCUR` | `DYNATRACE` `MICROSOFTDYNAMIC365FINANCEANDOPS` | `MICROSOFTTEAMS` | `BLACKBAUDRAISEREDGENXT` | `MAILCHIMP` | `GITLAB` | `PENDO` | `PRODUCTBOARD` | `CIRCLECI` | `PIPEDIVE` `SENDGRID` | `AZURECOSMOS` | `AZURESQL` | `BIGQUERY` | `BLACKBAUD` | `CLOUDERAHIVE` | `CLOUDERAIMPALA` | `CLOUDWATCH` | `CLOUDWATCHMETRICS` `CMDB` | `DATALAKEGEN2` | `DB2` | `DB2AS400` | `DOCUMENTDB` | `DOMO` `DYNAMODB` `GOOGLECLOUDSTORAGE`| `HBASE` | `KUSTOMER` | `MICROSOFTDYNAMICS365CRM` | `MONDAY` | `MYSQL` | `OKTA` | `OPENSEARCH` | `ORACLE` | `PIPEDRIVE` | | `POSTGRESQL` | `SAPHANA` | `SQLSERVER` | `SYNAPSE` | `TERADATA` | `TERADATANOS` | `TIMESTREAM` | `TPCDS` |`VERTICA`).

  Il nome del tipo di connessione.

- `DisplayName`: stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Il nome leggibile dalle persone per il tipo di connessione visualizzato nella console AWS Glue .

- `Vendor`: stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Il nome del fornitore o del provider che ha creato o gestisce questo tipo di connessione.

- `Description`: stringa UTF-8, non superiore a 1.024 byte di lunghezza.

  Una descrizione del tipo di connessione.

- `Categories` – .

  Un elenco di categorie a cui appartiene questo tipo di connessione. Le categorie aiutano gli utenti a filtrare e trovare i tipi di connessione appropriati in base ai loro casi d'uso.

- `Capabilities`: un oggetto Funzionalità.

  I tipi di autenticazione supportati, i tipi di interfaccia dati (ambienti di calcolo) e le operazioni sui dati del connettore.

- `LogoUrl`: stringa UTF-8.

  L'URL del logo associato a un tipo di connessione.

- `ConnectionTypeVariants`: una matrice di oggetti ConnectionTypeVariant.

  Un elenco di varianti disponibili per questo tipo di connessione. Varianti diverse possono fornire configurazioni specializzate per casi d'uso specifici o implementazioni dello stesso tipo di connessione generale.

ConnectionTypeVariant struttura

Rappresenta una variante di un tipo di connessione in AWS Glue Data Catalog. Le varianti del tipo di connessione forniscono configurazioni e comportamenti specifici per diverse implementazioni dello stesso tipo di connessione generale.

Campi

- `ConnectionTypeVariantName`: stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  L'identificatore univoco per la variante del tipo di connessione. Questo nome viene utilizzato internamente per identificare la variante specifica di un tipo di connessione.

- `DisplayName`: stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Il nome leggibile dall'uomo per la variante del tipo di connessione visualizzata nella console. AWS Glue

- `Description`: stringa UTF-8, non superiore a 1.024 byte di lunghezza.

  Una descrizione dettagliata della variante del tipo di connessione, inclusi lo scopo, i casi d'uso e gli eventuali requisiti di configurazione specifici.

- `LogoUrl`: stringa UTF-8.

  L'URL del logo associato a una variante del tipo di connessione.

datatypes

- Struttura di validazione
- AuthConfiguration struttura
- Struttura delle funzionalità
- Struttura della proprietà
- AllowedValue struttura
- ComputeEnvironmentConfiguration struttura

Struttura di validazione

Definisce come viene eseguita una convalida su una proprietà di connessione.

Campi

- `ValidationType` - obbligatorio: stringa UTF-8 (valori validi: `REGEX` | `RANGE`).

  Il tipo di convalida da eseguire, ad esempio `REGEX`.
- `Patterns` – .

  Un elenco di modelli che si applicano alla convalida.
- `Description`: obbligatorio: stringa UTF-8, non inferiore a 1 o non superiore a 1.024 byte di lunghezza.

  Una descrizione della validazione.
- `MaxLength`: numero (intero).

  Lunghezza massima per la proprietà di connessione di una stringa.
- `Maximum`: numero (intero).

  Un valore massimo quando si specifica un tipo di convalida `RANGE`.
- `Minimum`: numero (intero).

  Un valore minimo quando si specifica un tipo di convalida `RANGE`.

## AuthConfiguration struttura

La configurazione di autenticazione per una connessione restituita dall'API `DescribeConnectionType`.

### Campi

- `AuthenticationType` - obbligatorio: un oggetto Proprietà.

  Il tipo di autenticazione da utilizzare per una connessione.

- `SecretArn`: un oggetto Proprietà.

  Il nome della risorsa Amazon (ARN) per Secrets Manager.

- `OAuth2Properties`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Ogni valore è un oggetto Proprietà.

  Una mappa di coppie chiave-valore per le OAuth2 proprietà. Ogni valore è un oggetto `Property`.

- `BasicAuthenticationProperties`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Ogni valore è un oggetto Proprietà.

  Una mappa di coppie chiave-valore per le proprietà. OAuth2 Ogni valore è un oggetto `Property`.

- `CustomAuthenticationProperties`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Ogni valore è un oggetto Proprietà.

  Una mappa di coppie chiave-valore per le proprietà di autenticazione personalizzate. Ogni valore è un oggetto `Property`.

## Struttura delle funzionalità

Specifica i tipi di autenticazione supportati restituiti dall'API `DescribeConnectionType`.

Campi

- `SupportedAuthenticationTypes` - obbligatorio: una matrice di stringhe UTF-8.

  Un elenco di tipi di autenticazione supportati.
- `SupportedDataOperations` - obbligatorio: una matrice di stringhe UTF-8.

  Un elenco delle operazioni dei dati supportate.
- `SupportedComputeEnvironments` - obbligatorio: una matrice di stringhe UTF-8.

  Un elenco di ambienti di calcolo supportati.

Struttura della proprietà

Un oggetto che definisce un tipo di connessione per un ambiente di calcolo.

Campi

- `Name`: obbligatorio: stringa UTF-8, non inferiore a 1 o non superiore a 128 byte di lunghezza.

  Il nome della proprietà.
- `Description`: obbligatorio: stringa UTF-8, non superiore a 1.024 byte di lunghezza.

  Una descrizione della proprietà.
- `Required` - obbligatorio: booleano.

  Indica se la proprietà è obbligatoria.
- `PropertyTypes` - obbligatorio: una matrice di stringhe UTF-8.

  Descrive il tipo di proprietà.
- `AllowedValues`: una matrice di oggetti AllowedValue.

  Un elenco di oggetti `AllowedValue` che rappresentano i valori consentiti per la proprietà.
- `DataOperationScopes`: una matrice di stringhe UTF-8.

  Indica quali operazioni sui dati sono applicabili alla proprietà.

AllowedValue struttura

Un oggetto che rappresenta un valore consentito per una proprietà.

Campi

- `Description`: stringa UTF-8, non superiore a 1.024 byte di lunghezza.

  Una descrizione del valore consentito.
- `Value`: obbligatorio: stringa UTF-8, non inferiore a 1 o non superiore a 128 byte di lunghezza.

  Il valore consentito per la proprietà.

ComputeEnvironmentConfiguration struttura

Un oggetto contenente la configurazione per un ambiente di calcolo (come Spark, Python o Athena) restituito dall'API `DescribeConnectionType`.

Campi

- `Name`: obbligatorio: stringa UTF-8, non inferiore a 1 o non superiore a 128 byte di lunghezza.

  Un nome per la configurazione dell'ambiente di calcolo.
- `Description`: obbligatorio: stringa UTF-8, non superiore a 1.024 byte di lunghezza.

  Una descrizione dell'ambiente di calcolo.
- `ComputeEnvironment` - obbligatorio: stringa UTF-8 (valori validi: `SPARK` | `ATHENA` | `PYTHON`).

  Tipo di ambiente di calcolo.
- `SupportedAuthenticationTypes` - obbligatorio: una matrice di stringhe UTF-8.

  I tipi di autenticazione supportati per l'ambiente di calcolo.
- `ConnectionOptions` - obbligatorio: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Ogni valore è un oggetto Proprietà.

  I parametri utilizzati come opzioni di connessione per l'ambiente di calcolo.

- `ConnectionPropertyNameOverrides` - obbligatorio: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Ogni valore è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Il nome della proprietà di connessione ha la precedenza per l'ambiente di calcolo.
- `ConnectionOptionNameOverrides` - obbligatorio: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Ogni valore è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Il nome dell'opzione di connessione ha la precedenza per l'ambiente di calcolo.
- `ConnectionPropertiesRequiredOverrides`— Richiesto:.

  Le proprietà di connessione richieste come sostituzioni per l'ambiente di calcolo.
- `PhysicalConnectionPropertiesRequired`: booleano.

  Indica se `PhysicalConnectionProperties` sono necessarie per l'ambiente di calcolo.

## Metadati di connessione e API di anteprima

La seguente connessione APIs descrive le operazioni per descrivere i metadati di connessione.

Tipi di dati

- Struttura dell'entità
- Struttura del campo

Struttura dell'entità

Un'entità supportata da un dato `ConnectionType`.

Campi

- `EntityName`: stringa UTF-8.

Il nome dell'entità.

- `Label`: stringa UTF-8.

  Etichetta utilizzata per l'entità.

- `IsParentEntity`: booleano.

  Un valore booleano che aiuta a determinare se ci sono oggetti secondari che possono essere elencati.

- `Description`: stringa UTF-8.

  Una descrizione dell'entità.

- `Category`: stringa UTF-8.

  Il tipo di entità presenti nella risposta. Questo valore dipende dalla connessione di origine. Ad esempio, questo è `SObjects` per Salesforce e `databases`, `schemas` o `tables` per origini come Amazon Redshift.

- `CustomProperties` –

  Una mappa facoltativa di chiavi che può essere restituita per un'entità da un connettore.

Struttura del campo

L'oggetto `Field` contiene informazioni sulle diverse proprietà associate a un campo nel connettore.

Campi

- `FieldName`: stringa UTF-8.

  L'identificatore univoco del campo.

- `Label`: stringa UTF-8.

  Un'etichetta leggibile utilizzata per il campo.

- `Description`: stringa UTF-8.

  Una descrizione del campo.

- `FieldType`: stringa UTF-8 (valori validi: `INT` | `SMALLINT` | `BIGINT` | `FLOAT` | `LONG` | `DATE` | `BOOLEAN` | `MAP` | `ARRAY` | `STRING` | `TIMESTAMP` | `DECIMAL` | `BYTE` | `SHORT` | `DOUBLE` | `STRUCT`).

Il tipo di dati nel campo.

- `IsPrimaryKey`: booleano.

  Indica se questo campo può essere utilizzato come chiave primaria per l'entità specificata.

- `IsNullable`: booleano.

  Indica se questo campo può essere annullabile o meno.

- `IsRetrievable`: booleano.

  Indica se questo campo può essere aggiunto nella clausola Select della query SQL o se è recuperabile o meno.

- `IsFilterable`: booleano.

  Indica se questo campo può essere utilizzato in una clausola di filtro (clausola `WHERE`) di un'istruzione SQL durante l'esecuzione di query sui dati.

- `IsPartitionable`: booleano.

  Indica se un determinato campo può essere utilizzato per partizionare la query eseguita su SaaS.

- `IsCreateable`: booleano.

  Indica se questo campo può essere creato come parte di una scrittura di destinazione.

- `IsUpdateable`: booleano.

  Indica se questo campo può essere aggiornato come parte di una scrittura di destinazione.

- `IsUpsertable`: booleano.

  Indica se questo campo può essere alterato come parte di una scrittura di destinazione.

- `IsDefaultOnCreate`: booleano.

  Indica se questo campo viene compilato automaticamente al momento della creazione dell'oggetto, ad esempio un timestamp Ora di creazione.

- `SupportedValues` – .

  Un elenco di valori supportati per il campo.

- `SupportedFilterOperators`: una matrice di stringhe UTF-8.

  Indica gli operatori di filtro di supporto per questo campo.

- `CustomProperties` –

  Mappa opzionale delle chiavi che possono essere restituite.

Operazioni

- ListEntities azione (Python: list_entities)
- DescribeEntity azione (Python: describe_entity)
- GetEntityRecords azione (Python: get_entity_records)

ListEntities azione (Python: list_entities)

Restituisce le entità disponibili supportate dal tipo di connessione.

Richiesta

- `ConnectionName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Un nome per la connessione che ha bisogno di credenziali per eseguire query su qualsiasi tipo di connessione.
- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo che contiene la connessione. Questo può essere nullo. Per impostazione predefinita, l'ID AWS account è l'ID del catalogo.
- `ParentEntityName`: stringa UTF-8.

  Il nome dell'entità padre per la quale intendi elencare i figli. Questo parametro utilizza un percorso completo dell'entità per elencare le entità figlio.
- `NextToken`: stringa UTF-8, non inferiore a 1 o superiore a 2048 byte di lunghezza, corrispondente a Custom string pattern #11.

  Un token di continuazione, incluso se si tratta di una chiamata di continuazione.
- `DataStoreApiVersion`: stringa UTF-8, non inferiore a 1 o superiore a 256 byte di lunghezza, corrispondente a Custom string pattern #23.

  La versione API del connettore SaaS.

Risposta

- `Entities`: una matrice di oggetti Entità.

  Elenco di oggetti `Entity`.
- `NextToken`: stringa UTF-8, non inferiore a 1 o superiore a 2048 byte di lunghezza, corrispondente a Custom string pattern #11.

  Un token di continuazione, presente se il segmento corrente non è l'ultimo.

Errori

- `EntityNotFoundException`
- `OperationTimeoutException`
- `InvalidInputException`
- `GlueEncryptionException`
- `ValidationException`
- `FederationSourceException`
- `AccessDeniedException`

DescribeEntity azione (Python: describe_entity)

Fornisce dettagli sull'entità utilizzata con il tipo di connessione, con una descrizione del modello di dati per ogni campo dell'entità selezionata.

La risposta include tutti i campi che compongono l'entità.

Richiesta

- `ConnectionName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome della connessione che contiene le credenziali del tipo di connessione.
- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo che contiene la connessione. Questo può essere nullo. Per impostazione predefinita, l'ID AWS account è l'ID del catalogo.

- `EntityName` - Obbligatorio: stringa UTF-8.

  Il nome dell'entità che desideri descrivere in base al tipo di connessione.

- `NextToken`: stringa UTF-8, non inferiore a 1 o superiore a 2048 byte di lunghezza, corrispondente a Custom string pattern #11.

  Un token di continuazione, incluso se si tratta di una chiamata di continuazione.

- `DataStoreApiVersion`: stringa UTF-8, non inferiore a 1 o superiore a 256 byte di lunghezza, corrispondente a Custom string pattern #23.

  La versione dell'API utilizzata per il datastore.

Risposta

- `Fields`: una matrice di oggetti Campo.

  Descrive i campi per tale entità del connettore. Questo è l'elenco degli oggetti `Field`. `Field` è molto simile alla colonna in un database. L'oggetto `Field` contiene informazioni sulle diverse proprietà associate ai campi del connettore.

- `NextToken`: stringa UTF-8, non inferiore a 1 o superiore a 2048 byte di lunghezza, corrispondente a Custom string pattern #11.

  Un token di continuazione, presente se il segmento corrente non è l'ultimo.

Errori

- `EntityNotFoundException`
- `OperationTimeoutException`
- `InvalidInputException`
- `GlueEncryptionException`
- `ValidationException`
- `FederationSourceException`
- `AccessDeniedException`

GetEntityRecords azione (Python: get_entity_records)

Questa API viene utilizzata per interrogare i dati di anteprima da un determinato tipo di connessione o da un catalogo AWS Glue dati nativo basato su Amazon S3.

Restituisce i record sotto forma di un array di blob JSON. Ogni record è formattato utilizzando Jackson JsonNode in base al tipo di campo definito dall'`DescribeEntity`API.

I connettori Spark generano schemi in base alla stessa mappatura dei tipi di dati dell'API `DescribeEntity`. I connettori Spark convertono i dati nei tipi di dati appropriati che corrispondono allo schema quando restituiscono le righe.

Richiesta

- `ConnectionName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome della connessione che contiene le credenziali del tipo di connessione.
- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo che contiene la connessione. Questo può essere nullo. Per impostazione predefinita, l'ID AWS account è l'ID del catalogo.
- `EntityName` - Obbligatorio: stringa UTF-8.

  Nome dell'entità per cui vogliamo eseguire le query sui dati di anteprima relativi al tipo di connessione specificato.
- `NextToken`: stringa UTF-8, non inferiore a 1 o superiore a 2048 byte di lunghezza, corrispondente a Custom string pattern #11.

  Un token di continuazione, incluso se si tratta di una chiamata di continuazione.
- `DataStoreApiVersion`: stringa UTF-8, non inferiore a 1 o superiore a 256 byte di lunghezza, corrispondente a Custom string pattern #23.

  La versione API del connettore SaaS.
- `ConnectionOptions`: una matrice di mappe con coppie chiave-valore, non superiore alle 100 coppie.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 256 byte di lunghezza, corrispondente a Custom string pattern #18.

Ogni valore è una stringa UTF-8, non inferiore a 1 o superiore a 256 byte di lunghezza, corrispondente a Custom string pattern #17.

Opzioni di connettore necessarie per eseguire query sui dati.

- `FilterPredicate`: stringa UTF-8, non inferiore a 1 o superiore a 100.000 byte di lunghezza.

  Un predicato di filtro che puoi applicare nella richiesta di query.

- `Limit`: obbligatorio: numero (long), non inferiore a 1 o superiore a 1.000.

  Limita il numero di record recuperati con la richiesta.

- `SelectedFields`: un array di stringhe UTF-8, non inferiore a 1 o superiore a 1.000 stringhe.

  Elenco di campi da recuperare come parte dei dati di anteprima.

Risposta

- `Records`: un array di strutture.

  Un elenco di tutti gli oggetti richiesti.

- `NextToken`: stringa UTF-8, non inferiore a 1 o superiore a 2048 byte di lunghezza, corrispondente a Custom string pattern #11.

  Un token di continuazione, presente se il segmento corrente non è l'ultimo.

Errori

- `EntityNotFoundException`
- `OperationTimeoutException`
- `InvalidInputException`
- `GlueEncryptionException`
- `ValidationException`
- `FederationSourceException`
- `AccessDeniedException`

# API della funzione definita dall'utente

L'API User-defined Function descrive AWS Glue i tipi di dati e le operazioni utilizzate nell'utilizzo delle funzioni.

## Tipi di dati

- UserDefinedFunction struttura
- UserDefinedFunctionInput struttura

## UserDefinedFunction struttura

Rappresenta l'equivalente di una definizione di funzione Hive definita dall'utente (UDF).

Campi

- `FunctionName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome della funzione.
- `DatabaseName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del database del catalogo che contiene la funzione.
- `ClassName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  La classe Java che contiene il codice della funzione.
- `OwnerName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il proprietario della funzione.
- `OwnerType`: stringa UTF-8 (valori validi: `USER` | `ROLE` | `GROUP`).

  Il tipo di proprietario.
- `CreateTime`: timestamp.

  L'ora in cui è stata creata la funzione.

- `ResourceUris`: una matrice di oggetti ResourceUri, non superiore a 1000 strutture.

  La risorsa URIs per la funzione.
- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo dati in cui si trova la funzione.

## UserDefinedFunctionInput struttura

Una struttura utilizzata per creare o aggiornare una funzione definita dall'utente.

Campi

- `FunctionName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome della funzione.
- `ClassName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  La classe Java che contiene il codice della funzione.
- `OwnerName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il proprietario della funzione.
- `OwnerType`: stringa UTF-8 (valori validi: `USER` | `ROLE` | `GROUP`).

  Il tipo di proprietario.
- `ResourceUris`: una matrice di oggetti ResourceUri, non superiore a 1000 strutture.

  La risorsa URIs per la funzione.

## Operazioni

- CreateUserDefinedFunction azione (Python: create_user_defined_function)
- UpdateUserDefinedFunction azione (Python: update_user_defined_function)
- DeleteUserDefinedFunction azione (Python: delete_user_defined_function)

- GetUserDefinedFunction azione (Python: get_user_defined_function)
- GetUserDefinedFunctions azione (Python: get_user_defined_functions)

## CreateUserDefinedFunction azione (Python: create_user_defined_function)

Crea una nuova definizione di funzione nel catalogo dati.

Richiesta

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo dati in cui creare la funzione. Se non ne viene fornito nessuno, per impostazione predefinita viene utilizzato l'ID dell'account. AWS
- `DatabaseName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del database del catalogo in cui creare la funzione.
- `FunctionInput` - obbligatorio: oggetto UserDefinedFunctionInput.

  Un oggetto `FunctionInput` che definisce la funzione da creare nel catalogo dati.

Risposta

- Nessun parametro di risposta.

Errori

- `AlreadyExistsException`
- `InvalidInputException`
- `InternalServiceException`
- `EntityNotFoundException`
- `OperationTimeoutException`
- `ResourceNumberLimitExceededException`
- `GlueEncryptionException`

## UpdateUserDefinedFunction azione (Python: update_user_defined_function)

Aggiorna una definizione di funzione esistente nel catalogo dati.

### Richiesta

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo dati in cui si trova la funzione da aggiornare. Se non ne viene fornito nessuno, per impostazione predefinita viene utilizzato l'ID dell'account. AWS
- `DatabaseName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del database del catalogo in cui si trova la funzione da aggiornare.
- `FunctionName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome della funzione.
- `FunctionInput` - obbligatorio: oggetto UserDefinedFunctionInput.

  Un oggetto `FunctionInput` che ridefinisce la funzione nel catalogo dati.

### Risposta

- Nessun parametro di risposta.

### Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `InternalServiceException`
- `OperationTimeoutException`
- `GlueEncryptionException`

## DeleteUserDefinedFunction azione (Python: delete_user_defined_function)

Elimina una definizione di funzione esistente dal catalogo dati.

Richiesta

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo dati in cui si trova la funzione da eliminare. Se non ne viene fornito nessuno, per impostazione predefinita viene utilizzato l'ID dell'account AWS .
- `DatabaseName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del database del catalogo in cui si trova la funzione.
- `FunctionName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome della definizione della funzione da eliminare.

Risposta

- Nessun parametro di risposta.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `InternalServiceException`
- `OperationTimeoutException`

## GetUserDefinedFunction azione (Python: get_user_defined_function)

Richiama una definizione di funzione specificata dal catalogo dati.

Richiesta

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo dati in cui si trova la funzione da richiamare. Se non ne viene fornito nessuno, per impostazione predefinita viene utilizzato l'ID dell'account AWS .
- `DatabaseName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del database del catalogo in cui si trova la funzione.
- `FunctionName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome della funzione.

Risposta

- `UserDefinedFunction`: un oggetto UserDefinedFunction.

  La definizione di funzione richiesta.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `InternalServiceException`
- `OperationTimeoutException`
- `GlueEncryptionException`

## GetUserDefinedFunctions azione (Python: get_user_defined_functions)

Richiama definizioni di funzione multiple dal catalogo dati.

Richiesta

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

L'ID del catalogo dati in cui si trovano le funzioni da recuperare. Se non ne viene fornito nessuno, per impostazione predefinita viene utilizzato l'ID dell'account AWS .

- `DatabaseName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del database del catalogo in cui si trovano le funzioni. Se non ne viene fornito nessuno, verranno restituite le funzioni di tutti i database del catalogo.

- `Pattern` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Una stringa di modello nome-funzione facoltativa che filtra le definizioni di funzione restituite.

- `NextToken`: stringa UTF-8.

  Un token di continuazione, se si tratta di una chiamata di continuazione.

- `MaxResults` – Numero (intero), non inferiore a 1 o superiore a 100.

  Il numero massimo di funzioni da restituire in una risposta.

Risposta

- `UserDefinedFunctions`: una matrice di oggetti UserDefinedFunction.

  Un elenco di definizioni di funzione richieste.

- `NextToken`: stringa UTF-8.

  Un token di continuazione, se l'elenco di funzioni restituite non include l'ultima funzione richiesta.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `OperationTimeoutException`
- `InternalServiceException`
- `GlueEncryptionException`

# Importazione di un Athena catalogo in AWS Glue

L'API di migrazione descrive AWS Glue i tipi di dati e le operazioni relative alla migrazione di un catalogo di Athena dati verso. AWS Glue

## Tipi di dati

- CatalogImportStatus struttura

## CatalogImportStatus struttura

Una struttura che contiene informazioni sullo stato della migrazione.

Campi

- `ImportCompleted`: booleano.

  `True` se la migrazione è stata completata, in caso contrario `False`.
- `ImportTime`: timestamp.

  L'ora in cui la migrazione è stata avviata.
- `ImportedBy`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome della persona che ha avviato la migrazione.

## Operazioni

- ImportCatalogToGlue azione (Python: import_catalog_to_glue)
- GetCatalogImportStatus azione (Python: get_catalog_import_status)

## ImportCatalogToGlue azione (Python: import_catalog_to_glue)

Importa un catalogo dati Amazon Athena esistente in. AWS Glue

Richiesta

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

L'ID del catalogo da importare. Attualmente, questo dovrebbe essere l'ID dell' AWS account.

Risposta

- Nessun parametro di risposta.

Errori

- `InternalServiceException`
- `OperationTimeoutException`

## GetCatalogImportStatus azione (Python: get_catalog_import_status)

Recupera lo stato di un'operazione di migrazione.

Richiesta

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo da migrare. Attualmente, questo dovrebbe essere l'ID dell'account. AWS

Risposta

- `ImportStatus`: un oggetto CatalogImportStatus.

  Lo stato della migrazione del catalogo specificata.

Errori

- `InternalServiceException`
- `OperationTimeoutException`

# API dell'ottimizzatore di tabelle

L'API di ottimizzazione delle tabelle descrive l' AWS Glue API per abilitare la compattazione per migliorare le prestazioni di lettura.

## Tipi di dati


- TableOptimizer struttura
- TableOptimizerConfiguration struttura
- TableOptimizerVpcConfiguration struttura
- CompactionConfiguration struttura
- IcebergCompactionConfiguration struttura
- TableOptimizerRun struttura
- BatchGetTableOptimizerEntry struttura
- BatchTableOptimizer struttura
- BatchGetTableOptimizerError struttura
- RetentionConfiguration struttura
- IcebergRetentionConfiguration struttura
- OrphanFileDeletionConfiguration struttura
- IcebergOrphanFileDeletionConfiguration struttura
- CompactionMetrics struttura
- RetentionMetrics struttura
- OrphanFileDeletionMetrics struttura
- IcebergCompactionMetrics struttura
- IcebergRetentionMetrics struttura
- IcebergOrphanFileDeletionMetrics struttura
- RunMetrics struttura


## TableOptimizer struttura

Contiene dettagli su un ottimizzatore associato a una tabella.

Campi

- `type`: stringa UTF-8 (valori validi: `compaction="COMPACTION"` | `retention="RETENTION"` | `orphan_file_deletion="ORPHAN_FILE_DELETION"`).

  Il tipo di ottimizzatore di tabelle. I valori validi sono:

  - `compaction`, per gestire la compattazione con un ottimizzatore di tabelle.
  - `retention`, per gestire la conservazione degli snapshot con un ottimizzatore di tabelle.
  - `orphan_file_deletion`, per gestire l'eliminazione di file orfani con un ottimizzatore di tabelle.
- `configuration`: un oggetto TableOptimizerConfiguration.

  Un oggetto `TableOptimizerConfiguration` specificato durante la creazione o l'aggiornamento di un ottimizzatore di tabelle.
- `lastRun`: un oggetto TableOptimizerRun.

  Un oggetto `TableOptimizerRun` che rappresenta l'ultima esecuzione dell'ottimizzatore di tabelle.
- `configurationSource`: stringa UTF-8 (valori validi: `catalog="CATALOG"` | `table="TABLE"`).

  Specifica l'origine della configurazione dell'ottimizzatore. Questo indica come è stato configurato l'ottimizzatore di tabelle e quale entità o servizio ha avviato la configurazione.

## TableOptimizerConfiguration struttura

Contiene dettagli sulla configurazione di un ottimizzatore di tabelle. Questa configurazione viene passata quando si crea o si aggiorna un ottimizzatore di tabelle.

Campi

- `roleArn`: stringa UTF-8, non inferiore a 20 o superiore a 2048 byte di lunghezza, corrispondente a Single-line string pattern.

  Un ruolo passato dal chiamante che autorizza il servizio ad aggiornare le risorse associate all'ottimizzatore per suo conto.
- `enabled`: booleano.

  Se l'ottimizzazione delle tabelle è abilitata.
- `vpcConfiguration`: un oggetto TableOptimizerVpcConfiguration.

Un oggetto `TableOptimizerVpcConfiguration` che rappresenta la configurazione del VPC per l'ottimizzatore di tabelle.

Questa configurazione è necessaria per eseguire l'ottimizzazione sulle tabelle che si trovano nel VPC del cliente.

- `compactionConfiguration`: un oggetto CompactionConfiguration.

  La configurazione per un ottimizzatore di compattazione. Questa configurazione definisce come verranno compattati i file di dati nella tabella per migliorare le prestazioni delle query e ridurre i costi di archiviazione.

- `retentionConfiguration`: un oggetto RetentionConfiguration.

  La configurazione per un ottimizzatore della conservazione degli snapshot.

- `orphanFileDeletionConfiguration`: un oggetto OrphanFileDeletionConfiguration.

  La configurazione per un ottimizzatore per l'eliminazione di file orfani.

## TableOptimizerVpcConfiguration struttura

Un oggetto che descrive la configurazione del VPC per un ottimizzatore di tabelle.

Questa configurazione è necessaria per eseguire l'ottimizzazione sulle tabelle che si trovano nel VPC del cliente.

Campi

- `glueConnectionName`: stringa UTF-8, almeno 1 byte di lunghezza.

  Il nome della AWS Glue connessione utilizzata per il VPC per l'ottimizzatore di tabelle.

## CompactionConfiguration struttura

La configurazione per un ottimizzatore di compattazione. Questa configurazione definisce come verranno compattati i file di dati nella tabella per migliorare le prestazioni delle query e ridurre i costi di archiviazione.

Campi

- `icebergConfiguration`: un oggetto IcebergCompactionConfiguration.

  La configurazione per un ottimizzatore di compattazione Iceberg.

## IcebergCompactionConfiguration struttura

La configurazione per un ottimizzatore di compattazione Iceberg. Questa configurazione definisce i parametri per l'ottimizzazione del layout dei file di dati nelle tabelle Iceberg.

Campi

- `strategy`: stringa UTF-8 (valori validi: `binpack="BINPACK"` | `sort="SORT"` | `z-order="ZORDER"`).

  La strategia da utilizzare per la compattazione. I valori validi sono:

  - `binpack`: combina file di piccole dimensioni in file più grandi, in genere con dimensioni superiori a 100 MB, applicando eventuali eliminazioni in sospeso. Questa è la strategia di compattazione consigliata per la maggior parte dei casi d'uso.
  - `sort`: organizza i dati in base a colonne specifiche che vengono ordinate gerarchicamente durante la compattazione, migliorando le prestazioni delle query per le operazioni filtrate. Questa strategia è consigliata quando le query vengono spesso filtrate in base a colonne specifiche. Per utilizzare questa strategia, è necessario innanzitutto definire un criterio di ordinamento nelle proprietà della tabella Iceberg utilizzando la proprietà della tabella `sort_order`.
  - `z-order`: ottimizza l'organizzazione dei dati fondendo più attributi in un unico valore scalare che può essere utilizzato per l'ordinamento, consentendo query efficienti su più dimensioni. Questa strategia è consigliata quando è necessario eseguire query sui dati su più dimensioni contemporaneamente. Per utilizzare questa strategia, è necessario innanzitutto definire un criterio di ordinamento nelle proprietà della tabella Iceberg utilizzando la proprietà della tabella `sort_order`.

  In assenza di un input preciso, sarà utilizzato il valore predefinito 'binpack'.
- `minInputFiles`: numero (intero).

  Il numero minimo di file di dati che devono essere presenti in una partizione prima della compattazione comprimerà effettivamente i file. Questo parametro aiuta a controllare quando viene

attivata la compattazione, evitando operazioni di compattazione non necessarie su partizioni con pochi file. In assenza di un input preciso, sarà utilizzato il valore predefinito 100.

- `deleteFileThreshold`: numero (intero).

  Il numero minimo di eliminazioni che devono essere presenti in un file di dati per renderlo idoneo alla compattazione. Questo parametro aiuta a ottimizzare la compattazione concentrandosi sui file che contengono un numero significativo di operazioni di eliminazione, il che può migliorare le prestazioni delle query rimuovendo i record eliminati. In assenza di un input preciso, sarà utilizzato il valore predefinito 1.

## TableOptimizerRun struttura

Contiene i dettagli per l'esecuzione di un ottimizzatore di tabelle.

Campi

- `eventType`: stringa UTF-8 (valori validi: `starting="STARTING"` | `completed="COMPLETED"` | `failed="FAILED"` | `in_progress="IN_PROGRESS"`).

  Un tipo di evento che rappresenta lo stato dell'esecuzione dell'ottimizzatore di tabella.

- `startTimestamp`: timestamp.

  Rappresenta il timestamp dell'epoca in cui è stato avviato il processo di compattazione all'interno di Lake Formation.

- `endTimestamp`: timestamp.

  Rappresenta il timestamp dell'epoca in cui è terminato il processo di compattazione.

- `metrics`: un oggetto RunMetrics.

  Un oggetto `RunMetrics` contenente i parametri per l'esecuzione dell'ottimizzatore.

  Questo membro è obsoleto. Visualizza i singoli membri del parametro per la compattazione, la conservazione e l'eliminazione dei file orfani.

- `error`: stringa UTF-8.

  Un errore che si è verificato durante l'esecuzione dell'ottimizzatore.

- `compactionMetrics`: un oggetto CompactionMetrics.

Un oggetto `CompactionMetrics` contenente i parametri per l'esecuzione dell'ottimizzatore.

- `compactionStrategy`: stringa UTF-8 (valori validi: `binpack="BINPACK"` | `sort="SORT"` | `z-order="ZORDER"`).

  La strategia da utilizzare per l'esecuzione della compattazione. Indica quale algoritmo è stato applicato per determinare il modo in cui i file sono stati selezionati e combinati durante il processo di compattazione. I valori validi sono:

  - `binpack`: combina file di piccole dimensioni in file più grandi, in genere con dimensioni superiori a 100 MB, applicando eventuali eliminazioni in sospeso. Questa è la strategia di compattazione consigliata per la maggior parte dei casi d'uso.
  - `sort`: organizza i dati in base a colonne specifiche che vengono ordinate gerarchicamente durante la compattazione, migliorando le prestazioni delle query per le operazioni filtrate. Questa strategia è consigliata quando le query vengono spesso filtrate in base a colonne specifiche. Per utilizzare questa strategia, è necessario innanzitutto definire un criterio di ordinamento nelle proprietà della tabella Iceberg utilizzando la proprietà della tabella `sort_order`.
  - `z-order`: ottimizza l'organizzazione dei dati fondendo più attributi in un unico valore scalare che può essere utilizzato per l'ordinamento, consentendo query efficienti su più dimensioni. Questa strategia è consigliata quando è necessario eseguire query sui dati su più dimensioni contemporaneamente. Per utilizzare questa strategia, è necessario innanzitutto definire un criterio di ordinamento nelle proprietà della tabella Iceberg utilizzando la proprietà della tabella `sort_order`.

- `retentionMetrics`: un oggetto RetentionMetrics.

  Un oggetto `RetentionMetrics` contenente i parametri per l'esecuzione dell'ottimizzatore.

- `orphanFileDeletionMetrics`: un oggetto OrphanFileDeletionMetrics.

  Un oggetto `OrphanFileDeletionMetrics` contenente i parametri per l'esecuzione dell'ottimizzatore.

## BatchGetTableOptimizerEntry struttura

Rappresenta un ottimizzatore di tabella da recuperare durante l'operazione `BatchGetTableOptimizer`.

Campi

- `catalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo della tabella.

- `databaseName`: stringa UTF-8, almeno 1 byte di lunghezza.

  Il nome del database nel catalogo in cui risiede la tabella.

- `tableName`: stringa UTF-8, almeno 1 byte di lunghezza.

  Nome della tabella.

- `type`: stringa UTF-8 (valori validi: `compaction="COMPACTION"` | `retention="RETENTION"` | `orphan_file_deletion="ORPHAN_FILE_DELETION"`).

  Il tipo di ottimizzatore di tabelle.

## BatchTableOptimizer struttura

Contiene i dettagli per uno degli ottimizzatori di tabella restituiti dall'operazione `BatchGetTableOptimizer`.

Campi

- `catalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo della tabella.

- `databaseName`: stringa UTF-8, almeno 1 byte di lunghezza.

  Il nome del database nel catalogo in cui risiede la tabella.

- `tableName`: stringa UTF-8, almeno 1 byte di lunghezza.

  Nome della tabella.

- `tableOptimizer`: un oggetto TableOptimizer.

  Un oggetto `TableOptimizer` che contiene i dettagli sulla configurazione e l'ultima esecuzione di un ottimizzatore di tabella.

## BatchGetTableOptimizerError struttura

Contiene dettagli su uno degli errori nell'elenco degli errori restituito dall'operazione `BatchGetTableOptimizer`.

Campi

- `error`: un oggetto ErrorDetail.

  Un oggetto `ErrorDetail` contenente i dettagli del codice e del messaggio di errore.
- `catalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo della tabella.
- `databaseName`: stringa UTF-8, almeno 1 byte di lunghezza.

  Il nome del database nel catalogo in cui risiede la tabella.
- `tableName`: stringa UTF-8, almeno 1 byte di lunghezza.

  Nome della tabella.
- `type`: stringa UTF-8 (valori validi: `compaction="COMPACTION"` | `retention="RETENTION"` | `orphan_file_deletion="ORPHAN_FILE_DELETION"`).

  Il tipo di ottimizzatore di tabelle.

## RetentionConfiguration struttura

La configurazione per un ottimizzatore della conservazione degli snapshot.

Campi

- `icebergConfiguration`: un oggetto IcebergRetentionConfiguration.

  La configurazione per un ottimizzatore della conservazione degli snapshot Iceberg.

## IcebergRetentionConfiguration struttura

La configurazione per un ottimizzatore della conservazione degli snapshot Iceberg.

Campi

- `snapshotRetentionPeriodInDays`: numero (intero).

  Il numero di giorni in cui gli snapshot Iceberg devono essere mantenuti. In assenza di un input preciso, verrà utilizzato il campo di configurazione della tabella Iceberg corrispondente o, se non è presente, verrà utilizzato il valore predefinito 5.

- `numberOfSnapshotsToRetain`: numero (intero).

  Il numero di snapshot Iceberg da mantenere entro il periodo di conservazione. In assenza di un input preciso, verrà utilizzato il campo di configurazione della tabella Iceberg corrispondente o, se non è presente, verrà utilizzato il valore predefinito 1.

- `cleanExpiredFiles`: booleano.

  Se impostato su false, gli snapshot vengono eliminati solo dai metadati della tabella e i file di dati e metadati sottostanti non vengono eliminati.

- `runRateInHours`: numero (intero).

  L'intervallo in ore tra le esecuzioni del processo di conservazione. Questo parametro controlla la frequenza con cui verrà eseguito l'ottimizzatore di conservazione per la pulizia degli snapshot scaduti. Il valore deve essere compreso tra 3 e 168 ore (7 giorni). In assenza di un input preciso, sarà utilizzato il valore predefinito 24.

## OrphanFileDeletionConfiguration struttura

La configurazione per un ottimizzatore per l'eliminazione di file orfani.

Campi

- `icebergConfiguration`: un oggetto IcebergOrphanFileDeletionConfiguration.

  La configurazione per un ottimizzatore per l'eliminazione di file orfani Iceberg.

## IcebergOrphanFileDeletionConfiguration struttura

La configurazione per un ottimizzatore per l'eliminazione di file orfani Iceberg.

Campi

- `orphanFileRetentionPeriodInDays`: numero (intero).

  Il numero di giorni in cui i file orfani devono essere mantenuti prima dell'eliminazione dei file. In assenza di un input preciso, sarà utilizzato il valore predefinito 3.

- `location`: stringa UTF-8.

  Specifica una directory in cui cercare i file (il valore predefinito è la posizione della tabella). È possibile scegliere una sottodirectory anziché la posizione della tabella di primo livello.

- `runRateInHours`: numero (intero).

  L'intervallo in ore tra le esecuzioni del processo di eliminazione di file orfani. Questo parametro controlla la frequenza con cui verrà eseguito l'ottimizzatore per l'eliminazione dei file orfani per la pulizia dei file orfani. Il valore deve essere compreso tra 3 e 168 ore (7 giorni). In assenza di un input preciso, sarà utilizzato il valore predefinito 24.

## CompactionMetrics struttura

Una struttura che contiene i parametri di compattazione per l'esecuzione dell'ottimizzatore.

Campi

- `IcebergMetrics`: un oggetto IcebergCompactionMetrics.

  Una struttura che contiene i parametri di compattazione Iceberg per l'esecuzione dell'ottimizzatore.

## RetentionMetrics struttura

Una struttura che contiene i parametri di conservazione per l'esecuzione dell'ottimizzatore.

Campi

- `IcebergMetrics`: un oggetto IcebergRetentionMetrics.

  Una struttura che contiene i parametri di conservazione Iceberg per l'esecuzione dell'ottimizzatore.

## OrphanFileDeletionMetrics struttura

Una struttura che contiene i parametri di eliminazione dei file orfani per l'esecuzione dell'ottimizzatore.

Campi

- `IcebergMetrics`: un oggetto IcebergOrphanFileDeletionMetrics.

  Una struttura che contiene i parametri di eliminazione dei file orfani Iceberg per l'esecuzione dell'ottimizzatore.

## IcebergCompactionMetrics struttura

Parametri di compattazione per Iceberg per l'esecuzione dell'ottimizzatore.

Campi

- `DpuHours`: numero (doppio).

  Il numero di ore DPU utilizzate dal processo.
- `NumberOfDpus`: numero (intero).

  Il numero di DPUs energia consumata dal lavoro, arrotondato al numero intero più vicino.
- `JobDurationInHour`: numero (doppio).

  La durata del processo in ore.

## IcebergRetentionMetrics struttura

Parametri di conservazione degli snapshot per Iceberg per l'esecuzione dell'ottimizzatore.

Campi

- `DpuHours`: numero (doppio).

  Il numero di ore DPU utilizzate dal processo.
- `NumberOfDpus`: numero (intero).

  Il numero di DPUs energia consumata dal lavoro, arrotondato al numero intero più vicino.
- `JobDurationInHour`: numero (doppio).

La durata del processo in ore.

## IcebergOrphanFileDeletionMetrics struttura

Parametri di eliminazione dei file orfani per Iceberg per l'esecuzione dell'ottimizzatore.

Campi

- `DpuHours`: numero (doppio).

  Il numero di ore DPU utilizzate dal processo.
- `NumberOfDpus`: numero (intero).

  Il numero di DPUs energia consumata dal lavoro, arrotondato al numero intero più vicino.
- `JobDurationInHour`: numero (doppio).

  La durata del processo in ore.

## RunMetrics struttura

Parametri per l'esecuzione dell'ottimizzatore.

Questa struttura è obsoleta. Visualizza i singoli membri del parametro per la compattazione, la conservazione e l'eliminazione dei file orfani.

Campi

- `NumberOfBytesCompacted`: stringa UTF-8.

  Il numero di byte rimossi dall'esecuzione del processo di compattazione.
- `NumberOfFilesCompacted`: stringa UTF-8.

  Il numero di file rimossi dall'esecuzione del processo di compattazione.
- `NumberOfDpus`: stringa UTF-8.

  Il numero di DPUs energia consumata dal lavoro, arrotondato al numero intero più vicino.
- `JobDurationInHour`: stringa UTF-8.

  La durata del processo in ore.

## Operazioni

- GetTableOptimizer azione (Python: get_table_optimizer)
- BatchGetTableOptimizer azione (Python: batch_get_table_optimizer)
- ListTableOptimizerRuns azione (Python: list_table_optimizer_runs)
- CreateTableOptimizer azione (Python: create_table_optimizer)
- DeleteTableOptimizer azione (Python: delete_table_optimizer)
- UpdateTableOptimizer azione (Python: update_table_optimizer)

## GetTableOptimizer azione (Python: get_table_optimizer)

Restituisce la configurazione di tutti gli ottimizzatori associati a una tabella specificata.

Richiesta

- `CatalogId` - Obbligatorio:: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern

  L'ID del catalogo della tabella.
- `DatabaseName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del database nel catalogo in cui risiede la tabella.
- `TableName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome della tabella.
- `Type` - obbligatorio: stringa UTF-8 (valori validi: `compaction="COMPACTION"` | `retention="RETENTION"` | `orphan_file_deletion="ORPHAN_FILE_DELETION"`).

  Il tipo di ottimizzatore di tabelle.

Risposta

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

L'ID del catalogo della tabella.

- `DatabaseName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del database nel catalogo in cui risiede la tabella.

- `TableName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome della tabella.

- `TableOptimizer`: un oggetto TableOptimizer.

  L'ottimizzatore associato alla tabella specificata.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `AccessDeniedException`
- `InternalServiceException`
- `ThrottlingException`

## BatchGetTableOptimizer azione (Python: batch_get_table_optimizer)

Restituisce la configurazione per gli ottimizzatori di tabella specificati.

Richiesta

- `Entries` - obbligatorio: una matrice di oggetti BatchGetTableOptimizerEntry.

  Un elenco di oggetti `BatchGetTableOptimizerEntry` che specificano gli ottimizzatori di tabella da recuperare.

Risposta

- `TableOptimizers`: una matrice di oggetti BatchTableOptimizer.

  Elenco di oggetti `BatchTableOptimizer`.

- `Failures`: una matrice di oggetti BatchGetTableOptimizerError.

  Un elenco di errori derivanti dall'operazione.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `AccessDeniedException`
- `InternalServiceException`
- `ThrottlingException`

## ListTableOptimizerRuns azione (Python: list_table_optimizer_runs)

Elenca la cronologia delle esecuzioni dell'ottimizzatore precedenti per una tabella specifica.

Richiesta

- `CatalogId` - Obbligatorio:: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern

  L'ID del catalogo della tabella.
- `DatabaseName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del database nel catalogo in cui risiede la tabella.
- `TableName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome della tabella.
- `Type` - obbligatorio: stringa UTF-8 (valori validi: `compaction="COMPACTION"` | `retention="RETENTION"` | `orphan_file_deletion="ORPHAN_FILE_DELETION"`).

  Il tipo di ottimizzatore di tabelle.
- `MaxResults`: numero (intero).

  Il numero massimo di esecuzioni dell'ottimizzatore da restituire per ogni chiamata.

- `NextToken`: stringa UTF-8.

  Un token di continuazione, se si tratta di una chiamata di continuazione.

Risposta

- `CatalogId`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo della tabella.

- `DatabaseName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del database nel catalogo in cui risiede la tabella.

- `TableName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome della tabella.

- `NextToken`: stringa UTF-8.

  Un token di continuazione per impaginare l'elenco restituito di esecuzioni dell'ottimizzatore, restituite se il segmento corrente dell'elenco non è l'ultimo.

- `TableOptimizerRuns`: una matrice di oggetti TableOptimizerRun.

  Un elenco delle esecuzioni di ottimizzazione associate a una tabella.

Errori

- `EntityNotFoundException`
- `AccessDeniedException`
- `InvalidInputException`
- `ValidationException`
- `InternalServiceException`
- `ThrottlingException`

# CreateTableOptimizer azione (Python: create_table_optimizer)

Crea un nuovo ottimizzatore di tabelle per una funzione specifica.

Richiesta

- `CatalogId` - Obbligatorio:: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern

  L'ID del catalogo della tabella.
- `DatabaseName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del database nel catalogo in cui risiede la tabella.
- `TableName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome della tabella.
- `Type` - obbligatorio: stringa UTF-8 (valori validi: `compaction="COMPACTION"` | `retention="RETENTION"` | `orphan_file_deletion="ORPHAN_FILE_DELETION"`).

  Il tipo di ottimizzatore di tabelle.
- `TableOptimizerConfiguration` - obbligatorio: un oggetto TableOptimizerConfiguration.

  Un oggetto `TableOptimizerConfiguration` che rappresenta la configurazione dell'ottimizzatore di tabelle.

Risposta

- Nessun parametro di risposta.

Errori

- `EntityNotFoundException`
- `ValidationException`
- `InvalidInputException`
- `AccessDeniedException`

- `AlreadyExistsException`
- `InternalServiceException`
- `ThrottlingException`

## DeleteTableOptimizer azione (Python: delete_table_optimizer)

Elimina un ottimizzatore e tutti i metadati associati per una tabella. L'ottimizzazione non verrà più eseguita sulla tabella.

Richiesta

- `CatalogId` - Obbligatorio:: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern

  L'ID del catalogo della tabella.
- `DatabaseName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del database nel catalogo in cui risiede la tabella.
- `TableName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome della tabella.
- `Type` - obbligatorio: stringa UTF-8 (valori validi: `compaction="COMPACTION"` | `retention="RETENTION"` | `orphan_file_deletion="ORPHAN_FILE_DELETION"`).

  Il tipo di ottimizzatore di tabelle.

Risposta

- Nessun parametro di risposta.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `AccessDeniedException`

- `InternalServiceException`
- `ThrottlingException`

## UpdateTableOptimizer azione (Python: update_table_optimizer)

Aggiorna la configurazione per un ottimizzatore di tabelle esistente.

Richiesta

- `CatalogId` - Obbligatorio:: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern

  L'ID del catalogo della tabella.
- `DatabaseName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del database nel catalogo in cui risiede la tabella.
- `TableName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome della tabella.
- `Type` - obbligatorio: stringa UTF-8 (valori validi: `compaction="COMPACTION"` | `retention="RETENTION"` | `orphan_file_deletion="ORPHAN_FILE_DELETION"`).

  Il tipo di ottimizzatore di tabelle.
- `TableOptimizerConfiguration` - obbligatorio: un oggetto TableOptimizerConfiguration.

  Un oggetto `TableOptimizerConfiguration` che rappresenta la configurazione dell'ottimizzatore di tabelle.

Risposta

- Nessun parametro di risposta.

Errori

- `EntityNotFoundException`

- `InvalidInputException`
- `AccessDeniedException`
- `ValidationException`
- `InternalServiceException`
- `ThrottlingException`
- `ConcurrentModificationException`

# API crawler e classificatori

L'API Crawler and classifiers descrive i tipi di dati del AWS Glue crawler e del classificatore e include l'API per la creazione, l'eliminazione, l'aggiornamento e l'elenco di crawler o classificatori.

Argomenti


- API classificatore
- API crawler
- API delle statistiche delle colonne
- API del pianificatore del crawler


## API classificatore

L'API Classifier descrive i tipi di dati del AWS Glue classificatore e include l'API per la creazione, l'eliminazione, l'aggiornamento e l'elenco dei classificatori.

### Tipi di dati


- Struttura classificatore
- GrokClassifier struttura
- XMLClassifier struttura
- JsonClassifier struttura
- CsvClassifier struttura
- CreateGrokClassifierRequest struttura
- UpdateGrokClassifierRequest struttura

- Crea la struttura della richiesta XMLClassifier
- Struttura della XMLClassifier richiesta di aggiornamento
- CreateJsonClassifierRequest struttura
- UpdateJsonClassifierRequest struttura
- CreateCsvClassifierRequest struttura
- UpdateCsvClassifierRequest struttura

## Struttura classificatore

I classificatori vengono attivati durante un'attività di crawling. Un classificatore verifica se un determinato file è in un formato che è in grado di gestire. In questo caso il classificatore crea uno schema nel formato di un oggetto `StructType` che corrisponde a quel formato di dati.

Puoi utilizzare i classificatori standard che AWS Glue fornisce oppure puoi scrivere classificatori personalizzati per classificare al meglio le tue fonti di dati e specificare gli schemi appropriati da utilizzare per esse. Un classificatore può essere di tipo `grok`, `XML`, `JSON` o `CSV` personalizzato come specificato in uno dei campi dell'oggetto `Classifier`.

Campi

- `GrokClassifier`: un oggetto GrokClassifier.

  Un classificatore che utilizza `grok`.
- `XMLClassifier`: un oggetto XMLClassifier.

  Classificatore per contenuto XML.
- `JsonClassifier`: un oggetto JsonClassifier.

  Classificatore per contenuto JSON.
- `CsvClassifier`: un oggetto CsvClassifier.

  Un classificatore per i valori separati da virgole (CSV).

## GrokClassifier struttura

Un classificatore che utilizza i pattern `grok`.

Campi

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del classificatore.

- `Classification`: obbligatorio: stringa UTF-8.

  Identificatore del formato di dati corrisposto dal classificatore, ad esempio log Twitter, JSON, Omniture e così via.

- `CreationTime`: timestamp.

  L'ultima volta in cui è stato registrato il classificatore.

- `LastUpdated`: timestamp.

  L'ultima volta in cui è stato aggiornato il classificatore.

- `Version`: numero (lungo).

  La versione del classificatore.

- `GrokPattern`: obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 2048 byte di lunghezza, corrispondente a A Logstash Grok string pattern.

  Il pattern grok applicato a un datastore da questo classificatore. Per ulteriori informazioni, consulta i pattern integrati in Scrittura di classificatori personalizzati.

- `CustomPatterns`: stringa UTF-8, non superiore a 16000 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  Pattern grok personalizzati opzionali definiti da questo classificatore. Per ulteriori informazioni, consulta i pattern personalizzati in Scrittura di classificatori personalizzati.

## XMLClassifier struttura

Classificatore per contenuto `XML`.

Campi

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

Il nome del classificatore.

- `Classification`. Obbligatorio: stringa UTF-8.

  Identificatore del formato di dati corrisposto dal classificatore.

- `CreationTime`: timestamp.

  L'ultima volta in cui è stato registrato il classificatore.

- `LastUpdated`: timestamp.

  L'ultima volta in cui è stato aggiornato il classificatore.

- `Version`: numero (lungo).

  La versione del classificatore.

- `RowTag`: stringa UTF-8.

  Il tag XML che designa l'elemento contenente ogni record in un documento XML da analizzare. Non è in grado di identificare un elemento con chiusura automatica (chiuso da `/>`). Un elemento riga vuota contenente solo attributi può essere analizzato fintantoché termina con un tag di chiusura (ad esempio, `<row item_a="A" item_b="B"></row>` è corretto, mentre `<row item_a="A" item_b="B" />` non lo è).

## JsonClassifier struttura

Classificatore per contenuto `JSON`.

Campi

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del classificatore.

- `CreationTime`: timestamp.

  L'ultima volta in cui è stato registrato il classificatore.

- `LastUpdated`: timestamp.

  L'ultima volta in cui è stato aggiornato il classificatore.

- `Version`: numero (lungo).

La versione del classificatore.

- `JsonPath`. Obbligatorio: stringa UTF-8.

  Una `JsonPath` stringa che definisce i dati JSON che il classificatore deve classificare. AWS Glue supporta un sottoinsieme di JsonPath, come descritto in Writing Custom Classifiers. JsonPath

## CsvClassifier struttura

Classificatore per contenuto `CSV` personalizzato.

Campi

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del classificatore.

- `CreationTime`: timestamp.

  L'ultima volta in cui è stato registrato il classificatore.

- `LastUpdated`: timestamp.

  L'ultima volta in cui è stato aggiornato il classificatore.

- `Version`: numero (lungo).

  La versione del classificatore.

- `Delimiter`: obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 1 byte di lunghezza, corrispondente a Custom string pattern #26.

  Un simbolo personalizzato per indicare il separatore di ogni voce di colonna nella riga.

- `QuoteSymbol`. Obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 1 byte di lunghezza, corrispondente a Custom string pattern #26.

  Un simbolo personalizzato per indicare la combinazione dei contenuti in un singolo valore di colonna. Deve essere diverso dal delimitatore di colonna.

- `ContainsHeader`: stringa UTF-8 (valori validi: `UNKNOWN` | `PRESENT` | `ABSENT`).

  Indica se il file CSV contiene un'intestazione.

- `Header`: una matrice di stringhe UTF-8.

Un elenco di stringhe che rappresenta i nomi delle colonne.

- `DisableValueTrimming`: booleano.

  Specifica di non tagliare i valori prima di individuare il tipo di valori di colonna. Il valore predefinito è `true`.

- `AllowSingleColumn`: booleano.

  Abilita l'elaborazione dei file che contengono una sola colonna.

- `CustomDatatypeConfigured`: booleano.

  Consente di configurare il tipo di dati personalizzato.

- `CustomDatatypes`: una matrice di stringhe UTF-8.

  Un elenco di tipi di dati personalizzati tra cui "BINARY", "BOOLEAN", "DATE", "DECIMAL", "DOUBLE", "FLOAT", "INT", "LONG", "SHORT", "STRING", "TIMESTAMP".

- `Serde`: stringa UTF-8 (valori validi: `OpenCSVSerDe` | `LazySimpleSerDe` | `None`).

  Imposta il file CSV SerDe per l'elaborazione del classificatore, che verrà applicato nel Data Catalog. I valori validi sono `OpenCSVSerDe`, `LazySimpleSerDe` e `None`. È possibile specificare il valore `None` quando si desidera che il crawler esegua il rilevamento.

## CreateGrokClassifierRequest struttura

Specifica un classificatore `grok` per `CreateClassifier`.

Campi

- `Classification` - Obbligatorio: stringa UTF-8.

  Un identificatore del formato di dati a cui corrisponde il classificatore, ad esempio Twitter, JSON, Omniture logs, Amazon CloudWatch Logs e così via.

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del nuovo classificatore.

- `GrokPattern`. Obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 2048 byte di lunghezza, corrispondente a A Logstash Grok string pattern.

Il pattern grok utilizzato da questo classificatore.

- `CustomPatterns`: stringa UTF-8, non superiore a 16000 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  Pattern grok personalizzati opzionali utilizzati da questo classificatore.

## UpdateGrokClassifierRequest struttura

Specifica un classificatore grok da aggiornare quando viene passato a `UpdateClassifier`.

Campi

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome della `GrokClassifier`.

- `Classification`: stringa UTF-8.

  Un identificatore del formato di dati a cui corrisponde il classificatore, ad esempio Twitter, JSON, Omniture logs, Amazon CloudWatch Logs e così via.

- `GrokPattern`: stringa UTF-8, non inferiore a 1 o superiore a 2048 byte di lunghezza, corrispondente a A Logstash Grok string pattern.

  Il pattern grok utilizzato da questo classificatore.

- `CustomPatterns`: stringa UTF-8, non superiore a 16000 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  Pattern grok personalizzati opzionali utilizzati da questo classificatore.

## Crea la struttura della richiesta XMLClassifier

Specifica un classificatore XML per `CreateClassifier`.

Campi

- `Classification`. Obbligatorio: stringa UTF-8.

  Identificatore del formato di dati corrisposto dal classificatore.

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del classificatore.

- `RowTag`: stringa UTF-8.

  Il tag XML che designa l'elemento contenente ogni record in un documento XML da analizzare. Non è in grado di identificare un elemento con chiusura automatica (chiuso da `/>`). Un elemento riga vuota contenente solo attributi può essere analizzato fintantoché termina con un tag di chiusura (ad esempio, `<row item_a="A" item_b="B"></row>` è corretto, mentre `<row item_a="A" item_b="B" />` non lo è).

## Struttura della XMLClassifier richiesta di aggiornamento

Specifica un classificatore XML da aggiornare.

Campi

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del classificatore.

- `Classification`: stringa UTF-8.

  Identificatore del formato di dati corrisposto dal classificatore.

- `RowTag`: stringa UTF-8.

  Il tag XML che designa l'elemento contenente ogni record in un documento XML da analizzare. Non è in grado di identificare un elemento con chiusura automatica (chiuso da `/>`). Un elemento riga vuota contenente solo attributi può essere analizzato fintantoché termina con un tag di chiusura (ad esempio, `<row item_a="A" item_b="B"></row>` è corretto, mentre `<row item_a="A" item_b="B" />` non lo è).

## CreateJsonClassifierRequest struttura

Specifica un classificatore JSON per `CreateClassifier`.

Campi

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del classificatore.
- `JsonPath`: obbligatorio: stringa UTF-8.

  Una `JsonPath` stringa che definisce i dati JSON che il classificatore deve classificare. AWS Glue supporta un sottoinsieme di JsonPath, come descritto in Writing Custom Classifiers. JsonPath

## UpdateJsonClassifierRequest struttura

Specifica un classificatore JSON da aggiornare.

Campi

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del classificatore.
- `JsonPath`: stringa UTF-8.

  Una `JsonPath` stringa che definisce i dati JSON che il classificatore deve classificare. AWS Glue supporta un sottoinsieme di JsonPath, come descritto in Writing Custom Classifiers. JsonPath

## CreateCsvClassifierRequest struttura

Specifica un classificatore CSV personalizzato per `CreateClassifier`.

Campi

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del classificatore.
- `Delimiter`. Obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 1 byte di lunghezza, corrispondente a Custom string pattern #26.

  Un simbolo personalizzato per indicare il separatore di ogni voce di colonna nella riga.

- `QuoteSymbol`. Obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 1 byte di lunghezza, corrispondente a Custom string pattern #26.

  Un simbolo personalizzato per indicare la combinazione dei contenuti in un singolo valore di colonna. Deve essere diverso dal delimitatore di colonna.

- `ContainsHeader`: stringa UTF-8 (valori validi: `UNKNOWN` | `PRESENT` | `ABSENT`).

  Indica se il file CSV contiene un'intestazione.

- `Header`: una matrice di stringhe UTF-8.

  Un elenco di stringhe che rappresenta i nomi delle colonne.

- `DisableValueTrimming`: booleano.

  Specifica di non tagliare i valori prima di individuare il tipo di valori di colonna. Il valore di default è true.

- `AllowSingleColumn`: booleano.

  Abilita l'elaborazione dei file che contengono una sola colonna.

- `CustomDatatypeConfigured`: booleano.

  Consente di configurare tipi di dati personalizzati.

- `CustomDatatypes`: una matrice di stringhe UTF-8.

  Crea un elenco di tipi di dati personalizzati supportati.

- `Serde`: stringa UTF-8 (valori validi: `OpenCSVSerDe` | `LazySimpleSerDe` | `None`).

  Imposta il file CSV SerDe per l'elaborazione del classificatore, che verrà applicato nel Data Catalog. I valori validi sono `OpenCSVSerDe`, `LazySimpleSerDe` e `None`. È possibile specificare il valore `None` quando si desidera che il crawler esegua il rilevamento.

## UpdateCsvClassifierRequest struttura

Specifica un classificatore CSV personalizzato da aggiornare.

Campi

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

Il nome del classificatore.

- `Delimiter`. Obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 1 byte di lunghezza, corrispondente a Custom string pattern #26.

  Un simbolo personalizzato per indicare il separatore di ogni voce di colonna nella riga.

- `QuoteSymbol`. Obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 1 byte di lunghezza, corrispondente a Custom string pattern #26.

  Un simbolo personalizzato per indicare la combinazione dei contenuti in un singolo valore di colonna. Deve essere diverso dal delimitatore di colonna.

- `ContainsHeader`: stringa UTF-8 (valori validi: `UNKNOWN` | `PRESENT` | `ABSENT`).

  Indica se il file CSV contiene un'intestazione.

- `Header`: una matrice di stringhe UTF-8.

  Un elenco di stringhe che rappresenta i nomi delle colonne.

- `DisableValueTrimming`: booleano.

  Specifica di non tagliare i valori prima di individuare il tipo di valori di colonna. Il valore di default è true.

- `AllowSingleColumn`: booleano.

  Abilita l'elaborazione dei file che contengono una sola colonna.

- `CustomDatatypeConfigured`: booleano.

  Specifica la configurazione di tipi di dati personalizzati.

- `CustomDatatypes`: una matrice di stringhe UTF-8.

  Specifica un elenco di tipi di dati personalizzati supportati.

- `Serde`: stringa UTF-8 (valori validi: `OpenCSVSerDe` | `LazySimpleSerDe` | `None`).

  Imposta il file CSV SerDe per l'elaborazione del classificatore, che verrà applicato nel Data Catalog. I valori validi sono `OpenCSVSerDe`, `LazySimpleSerDe` e `None`. È possibile specificare il valore `None` quando si desidera che il crawler esegua il rilevamento.

## Operazioni

- CreateClassifier azione (Python: create_classifier)
- DeleteClassifier azione (Python: delete_classifier)
- GetClassifier azione (Python: get_classifier)
- GetClassifiers azione (Python: get_classifiers)
- UpdateClassifier azione (Python: update_classifier)

## CreateClassifier azione (Python: create_classifier)

Crea un classificatore nell'account utente. L'operazione può essere un `GrokClassifier`, un `XMLClassifier`, un `JsonClassifier` o un `CsvClassifier` a seconda del campo in cui è presente la richiesta.

### Richiesta

- `GrokClassifier`: un oggetto CreateGrokClassifierRequest.

  Oggetto `GrokClassifier` che specifica il classificatore da creare.
- `XMLClassifier`: un oggetto Crea XMLClassifier richiesta.

  Oggetto `XMLClassifier` che specifica il classificatore da creare.
- `JsonClassifier`: un oggetto CreateJsonClassifierRequest.

  Oggetto `JsonClassifier` che specifica il classificatore da creare.
- `CsvClassifier`: un oggetto CreateCsvClassifierRequest.

  Oggetto `CsvClassifier` che specifica il classificatore da creare.

### Risposta

- Nessun parametro di risposta.

### Errori

- `AlreadyExistsException`
- `InvalidInputException`

- `OperationTimeoutException`

## DeleteClassifier azione (Python: delete_classifier)

Rimuove un classificatore dal catalogo dati.

Richiesta

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome del classificatore da rimuovere.

Risposta

- Nessun parametro di risposta.

Errori

- `EntityNotFoundException`
- `OperationTimeoutException`

## GetClassifier azione (Python: get_classifier)

Recupera un classificatore per nome.

Richiesta

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome del classificatore da recuperare.

Risposta

- `Classifier`: un oggetto Classificatore.

  Il classificatore richiesto.

Errori

- `EntityNotFoundException`
- `OperationTimeoutException`

## GetClassifiers azione (Python: get_classifiers)

Visualizza l'elenco di tutti gli oggetti classificatore nel catalogo dati.

Richiesta

- `MaxResults`: numero (intero), non inferiore a 1 o superiore a 1.000.

  Dimensione dell'elenco da restituire (opzionale).
- `NextToken`: stringa UTF-8.

  Token di continuazione opzionale.

Risposta

- `Classifiers`: una matrice di oggetti Classificatore.

  L'elenco richiesto di tutti gli oggetti classificatore.
- `NextToken`: stringa UTF-8.

  Token di continuazione.

Errori

- `OperationTimeoutException`

## UpdateClassifier azione (Python: update_classifier)

Modifica un classificatore esistente (`GrokClassifier`, `XMLClassifier`, `JsonClassifier` o `CsvClassifier` a seconda del campo in cui è presente).

Richiesta

- `GrokClassifier`: un oggetto UpdateGrokClassifierRequest.

Oggetto `GrokClassifier` con i campi aggiornati.

- `XMLClassifier`: un oggetto XMLClassifierRichiesta di aggiornamento.

  Oggetto `XMLClassifier` con i campi aggiornati.
- `JsonClassifier`: un oggetto UpdateJsonClassifierRequest.

  Oggetto `JsonClassifier` con i campi aggiornati.
- `CsvClassifier`: un oggetto UpdateCsvClassifierRequest.

  Oggetto `CsvClassifier` con i campi aggiornati.

Risposta

- Nessun parametro di risposta.

Errori

- `InvalidInputException`
- `VersionMismatchException`
- `EntityNotFoundException`
- `OperationTimeoutException`

## API crawler

L'API Crawler descrive i tipi di dati dei AWS Glue crawler, oltre all'API per la creazione, l'eliminazione, l'aggiornamento e l'elenco dei crawler.

### Tipi di dati

- Struttura dei crawler
- Struttura della pianificazione
- CrawlerTargets struttura
- Struttura S3Target
- Struttura S3 DeltaCatalogTarget
- Struttura S3 DeltaDirectTarget

- JdbcTarget struttura
- struttura Mongo DBTarget
- Struttura a dinamo DBTarget
- DeltaTarget struttura
- IcebergTarget struttura
- HudiTarget struttura
- CatalogTarget struttura
- CrawlerMetrics struttura
- CrawlerHistory struttura
- CrawlsFilter struttura
- SchemaChangePolicy struttura
- LastCrawlInfo struttura
- RecrawlPolicy struttura
- LineageConfiguration struttura
- LakeFormationConfiguration struttura

## Struttura dei crawler

Specifica un programma crawler che esamina un'origine dati e usa i classificatori per cercare di determinarne lo schema. Se l'esito è positivo, il crawler registra i metadati riguardanti l'origine dati in AWS Glue Data Catalog.

Campi

- `Name`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del crawler.
- `Role`: stringa UTF-8.

  Il nome della risorsa Amazon (ARN) di un ruolo IAM utilizzato per accedere alle risorse del cliente, ad esempio i dati di Amazon Simple Storage Service (Amazon S3).
- `Targets`: un oggetto CrawlerTargets.

  Raccolta di destinazioni da sottoporre al crawling.

- `DatabaseName`: stringa UTF-8.

  Il nome del database di catalogo in cui viene archiviato l'output del crawler.

- `Description`: stringa di descrizione, non superiore a 2048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  Descrizione del crawler.

- `Classifiers`: una matrice di stringhe UTF-8.

  Elenco di stringhe UTF-8 che specificano i classificatori personalizzati associati al crawler.

- `RecrawlPolicy`: un oggetto RecrawlPolicy.

  Una policy che specifica se eseguire nuovamente il crawling dell'intero set di dati o solo delle cartelle aggiunte dall'ultima esecuzione del crawler.

- `SchemaChangePolicy`: un oggetto SchemaChangePolicy.

  La policy che specifica i comportamenti di aggiornamento ed eliminazione per il crawler.

- `LineageConfiguration`: un oggetto LineageConfiguration.

  Una configurazione che specifica se la derivazione dei dati è abilitata per il crawler.

- `State`: stringa UTF-8 (valori validi: `READY` | `RUNNING` | `STOPPING`).

  Indica se il crawler è in esecuzione o se una sessione è in sospeso.

- `TablePrefix`: stringa UTF-8, non superiore a 128 byte di lunghezza.

  Il prefisso aggiunto ai nomi delle tabelle create.

- `Schedule`: un oggetto Schedule.

  Per i crawler pianificati, la pianificazione dell'esecuzione del crawler.

- `CrawlElapsedTime`: numero (lungo).

  Se il crawler è in esecuzione, contiene il tempo totale trascorso dall'inizio dell'ultimo crawling.

- `CreationTime`: timestamp.

  L'ora di creazione del crawler.

- `LastUpdated`: timestamp.

  L'ora dell'ultimo aggiornamento del crawler.

- `LastCrawl`: un oggetto LastCrawlInfo.

  Lo stato dell'ultimo crawling ed eventualmente le informazioni sull'errore, se presente.

- `Version`: numero (lungo).

  La versione del crawler.

- `Configuration`: stringa UTF-8.

  Le informazioni di configurazione del crawler. Questa stringa JSON con versione consente agli utenti di specificare gli aspetti del comportamento di un crawler. Per ulteriori informazioni, consulta la pagina Impostazione delle opzioni di configurazione del crawler.

- `CrawlerSecurityConfiguration`: stringa UTF-8, non superiore a 128 byte di lunghezza.

  Il nome della struttura `SecurityConfiguration` che questo crawler deve utilizzare.

- `LakeFormationConfiguration`: un oggetto LakeFormationConfiguration.

  Speciifica se il crawler deve utilizzare le credenziali per il crawler anziché AWS Lake Formation le credenziali del ruolo IAM.

## Struttura della pianificazione

Oggetto di pianificazione che utilizza una dichiarazione `cron` per pianificare un evento.

Campi

- `ScheduleExpression`: stringa UTF-8.

  Espressione `cron` usata per specificare la pianificazione (consulta Pianificazioni basate sul tempo per processi e crawler. Ad esempio, per eseguire un processo ogni giorno alle 12:15 UTC, devi specificare: `cron(15 12 * * ? *)`.

- `State`: stringa UTF-8 (valori validi: `SCHEDULED` | `NOT_SCHEDULED` | `TRANSITIONING`).

  Lo stato della pianificazione.

## CrawlerTargets struttura

Specifica gli archivi dati da sottoporre al crawling.

Campi

- `S3Targets`: una matrice di oggetti S3Target.

  Specifica le destinazioni di Amazon Simple Storage Service (Amazon S3).
- `JdbcTargets`: una matrice di oggetti JdbcTarget.

  Specifica le destinazioni JDBC.
- `MongoDBTargets`: una matrice di oggetti Mongo DBTarget.

  Specifica destinazioni Amazon DocumentDB o MongoDB.
- `DynamoDBTargets`: una matrice di oggetti Dinamo DBTarget.

  Specifica le destinazioni di Amazon DynamoDB.
- `CatalogTargets`: una matrice di oggetti CatalogTarget.

  Speciifica gli AWS Glue Data Catalog obiettivi.
- `DeltaTargets`: una matrice di oggetti DeltaTarget.

  Specifica le destinazioni dell'archivio dati Delta.
- `IcebergTargets`: una matrice di oggetti IcebergTarget.

  Specifica le destinazioni del datastore Apache Iceberg.
- `HudiTargets`: una matrice di oggetti HudiTarget.

  Specifica le destinazioni del datastore Apache Hudi.

## Struttura S3Target

Specifica un archivio dati in Amazon Simple Storage Service (Amazon S3).

Campi

- `Path`: stringa UTF-8.

  Il percorso della destinazione Amazon S3.
- `Exclusions`: una matrice di stringhe UTF-8.

Elenco di modelli globali utilizzati per l'esclusione dal crawling. Per ulteriori informazioni, consulta la sezione relativa alla catalogazione delle tabelle con un crawler.

- `ConnectionName`: stringa UTF-8, non inferiore a 1 o superiore a 2048 byte di lunghezza.

  Il nome di una connessione che consente a un processo o a un crawler di accedere ai dati in Amazon S3 all'interno di un ambiente Amazon Virtual Private Cloud (Amazon VPC).

- `SampleSize`: numero (intero).

  Imposta il numero di file in ogni cartella foglia da sottoporre al crawling durante il crawling di file di esempio in un set di dati. Se non è impostato, tutti i file vengono sottoposti al crawling. Un valore valido è un numero intero compreso tra 1 e 249.

- `EventQueueArn`: stringa UTF-8.

  Un ARN Amazon SQS valido. Ad esempio, `arn:aws:sqs:region:account:sqs`.

- `DlqEventQueueArn`: stringa UTF-8.

  Un ARN Amazon SQS di messaggi non recapitabili valido. Ad esempio, `arn:aws:sqs:region:account:deadLetterQueue`.

## Struttura S3 DeltaCatalogTarget

Speciifica una destinazione che scrive su un'origine dati Delta Lake nel AWS Glue Data Catalog.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome di destinazione dati.

- `Inputs` - obbligatorio: una matrice di stringhe UTF-8, non inferiore o superiore a 1 stringa.

  I nodi che sono input per la destinazione di dati.

- `PartitionKeys`: una matrice di stringhe UTF-8.

  Specifica il partizionamento nativo utilizzando una sequenza di chiavi.

- `Table` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome della tabella del database in cui scrivere.

- `Database` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

Il nome del database in cui scrivere.

- `AdditionalOptions`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8 corrispondente al Custom string pattern #61.

  Ogni valore è una stringa UTF-8 corrispondente al Custom string pattern #61.

  Specifica le opzioni di connessione aggiuntive per il connettore.
- `SchemaChangePolicy`: un oggetto CatalogSchemaChangePolicy.

  Una policy che specifica i comportamenti di aggiornamento per il crawler.
- `AutoDataQuality`: un oggetto AutoDataQuality.

  Specifica se abilitare automaticamente la valutazione della qualità dei dati per il catalogo S3 Delta di destinazione. Se impostato su `true`, i controlli di qualità dei dati vengono eseguiti automaticamente durante l'operazione di scrittura.
- `OutputSchemas`: una matrice di oggetti GlueSchema.

  Specifica lo schema di dati per il catalogo S3 Delta di destinazione.

## Struttura S3 DeltaDirectTarget

Speciifica una destinazione che scrive su un'origine dati Delta Lake in. Amazon S3

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome di destinazione dati.
- `Inputs` - obbligatorio: una matrice di stringhe UTF-8, non inferiore o superiore a 1 stringa.

  I nodi che sono input per la destinazione di dati.
- `PartitionKeys`: una matrice di stringhe UTF-8.

  Specifica il partizionamento nativo utilizzando una sequenza di chiavi.
- `Path` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il percorso Amazon S3 dell'origine dati Delta Lake su cui scrivere.

- `Compression` - obbligatorio: stringa UTF-8 (valori validi: `uncompressed="UNCOMPRESSED"` | `snappy="SNAPPY"`).

  Specifica il modo in cui i dati sono compressi. In genere questo non è necessario se i dati hanno un'estensione del file standard. I valori possibili sono `"gzip"` e `"bzip"`).

- `NumberTargetPartitions`: stringa UTF-8.

  Specifica il numero di partizioni della destinazione per la distribuzione dei file del set di dati Delta Lake in Amazon S3.

- `Format` - obbligatorio, stringa UTF-8 (valori validi: `json="JSON"` | `csv="CSV"` | `avro="AVRO"` | `orc="ORC"` | `parquet="PARQUET"` | `hudi="HUDI"` | `delta="DELTA"` | `iceberg="ICEBERG"` | `hyper="HYPER"` | `xml="XML"`).

  Specifica il formato di output dei dati per la destinazione.

- `AdditionalOptions`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8 corrispondente al Custom string pattern #61.

  Ogni valore è una stringa UTF-8 corrispondente al Custom string pattern #61.

  Specifica le opzioni di connessione aggiuntive per il connettore.

- `SchemaChangePolicy`: un oggetto DirectSchemaChangePolicy.

  Una policy che specifica i comportamenti di aggiornamento per il crawler.

- `AutoDataQuality`: un oggetto AutoDataQuality.

  Specifica se abilitare automaticamente la valutazione della qualità dei dati per la destinazione diretta S3 Delta. Se impostato su `true`, i controlli di qualità dei dati vengono eseguiti automaticamente durante l'operazione di scrittura.

## JdbcTarget struttura

Specifica un archivio dati JDBC da sottoporre al crawling.

Campi

- `ConnectionName`: stringa UTF-8, non inferiore a 1 o superiore a 2048 byte di lunghezza.

  Il nome della connessione da usare per connettersi alla destinazione JDBC.

- `Path`: stringa UTF-8.

  Il percorso della destinazione JDBC.
- `Exclusions`: una matrice di stringhe UTF-8.

  Elenco di modelli globali utilizzati per l'esclusione dal crawling. Per ulteriori informazioni, consulta la sezione relativa alla catalogazione delle tabelle con un crawler.
- `EnableAdditionalMetadata`: una matrice di stringhe UTF-8.

  Specifica un valore di `RAWTYPES` o `COMMENTS` per abilitare metadati aggiuntivi nelle risposte della tabella. `RAWTYPES` fornisce il tipo di dati a livello nativo. `COMMENTS` fornisce commenti associati a una colonna o a una tabella del database.

  Se non hai bisogno di metadati aggiuntivi, lascia il campo vuoto.

## struttura Mongo DBTarget

Specifica un archivio dati Amazon DocumentDB o MongoDB da sottoporre al crawling.

Campi

- `ConnectionName`: stringa UTF-8, non inferiore a 1 o superiore a 2048 byte di lunghezza.

  Il nome della connessione da usare per connettersi alla destinazione Amazon DocumentDB o MongoDB.
- `Path`: stringa UTF-8.

  Il percorso della destinazione Amazon DocumentDB o MongoDB (database/raccolta).
- `ScanAll`: booleano.

  Indica se eseguire la scansione di tutti i registri o campionare le righe della tabella. La scansione di tutti i registri può richiedere molto tempo quando la tabella non è una tabella di throughput elevato.

  Un valore di `true` significa analizzare tutti i registri, mentre un valore di `false` significa campionare i registri. Se non viene specificato alcun valore, il valore di default è `true`.

## Struttura a dinamo DBTarget

Specifica una tabella Amazon DynamoDB per eseguire il crawling.

Campi

- `Path`: stringa UTF-8.

  Nome della tabella DynamoDB di cui eseguire il crawling.

- `scanAll`: booleano.

  Indica se eseguire la scansione di tutti i registri o campionare le righe della tabella. La scansione di tutti i registri può richiedere molto tempo quando la tabella non è una tabella di throughput elevato.

  Un valore di `true` significa analizzare tutti i registri, mentre un valore di `false` significa campionare i registri. Se non viene specificato alcun valore, il valore di default è `true`.

- `scanRate`: numero (doppio).

  La percentuale di unità di capacità di lettura configurate da utilizzare dal crawler. AWS Glue L'unità di capacità di lettura è un termine definito da DynamoDB ed è un valore numerico che funge da limitatore di velocità per il numero di letture che possono essere eseguite su tale tabella al secondo.

  I valori validi sono null o un valore compreso tra 0,1 e 1,5. Un valore null viene utilizzato quando l'utente non fornisce un valore e il valore predefinito è 0,5 dell'unità di capacità di lettura massima configurata (per le tabelle con provisioning) o 0,25 dell'unità di capacità di lettura massima configurata (per le tabelle che utilizzano la modalità on demand).

## DeltaTarget struttura

Specifica un archivio dati Delta per eseguire la scansione di una o più tabelle Delta.

Campi

- `DeltaTables`: una matrice di stringhe UTF-8.

  Un elenco dei percorsi Amazon S3 alle tabelle Delta.

- `ConnectionName`: stringa UTF-8, non inferiore a 1 o superiore a 2048 byte di lunghezza.

  Il nome della connessione da usare per connettersi alla destinazione della tabella Delta.

- `WriteManifest`: booleano.

  Specifica se scrivere i file manifest sul percorso della tabella Delta.

- `CreateNativeDeltaTable`: booleano.

  Specifica se il crawler creerà tabelle native per consentire l'integrazione con i motori di query che supportano l'interrogazione diretta del log delle transazioni Delta.

## IcebergTarget struttura

Specifica un'origine dati Apache Iceberg in cui sono archiviate le tabelle Iceberg all'interno di Amazon S3.

Campi

- `Paths`: una matrice di stringhe UTF-8.

  Uno o più Amazon S3 percorsi che contengono le cartelle di metadati Iceberg come. `s3://bucket/prefix`
- `ConnectionName`: stringa UTF-8, non inferiore a 1 o superiore a 2048 byte di lunghezza.

  Il nome della connessione da utilizzare per connettersi alla destinazione Iceberg.
- `Exclusions`: una matrice di stringhe UTF-8.

  Elenco di modelli globali utilizzati per l'esclusione dal crawling. Per ulteriori informazioni, consulta la sezione relativa alla catalogazione delle tabelle con un crawler.
- `MaximumTraversalDepth`: numero (intero).

  La profondità massima dei Amazon S3 percorsi che il crawler può attraversare per scoprire la cartella di metadati Iceberg nel percorso. Amazon S3 Viene utilizzata per limitare il tempo di esecuzione del crawler.

## HudiTarget struttura

Specifica un'origine dati Apache Hudi.

Campi

- `Paths`: una matrice di stringhe UTF-8.

Una serie di stringhe di Amazon S3 posizione per Hudi, ognuna delle quali indica la cartella principale in cui risiedono i file di metadati per una tabella Hudi. La cartella Hudi può trovarsi in una cartella figlia della principale.

Il crawler scansionerà tutte le cartelle al di sotto del percorso di una cartella Hudi.

- `ConnectionName`: stringa UTF-8, non inferiore a 1 o superiore a 2048 byte di lunghezza.

  Il nome della connessione da utilizzare per connettersi alla destinazione Hudi. Se i tuoi file Hudi sono archiviati in bucket che richiedono l'autorizzazione VPC, puoi impostarne le proprietà di connessione qui.

- `Exclusions`: una matrice di stringhe UTF-8.

  Elenco di modelli globali utilizzati per l'esclusione dal crawling. Per ulteriori informazioni, consulta la sezione relativa alla catalogazione delle tabelle con un crawler.

- `MaximumTraversalDepth`: numero (intero).

  La profondità massima dei Amazon S3 percorsi che il crawler può attraversare per scoprire la cartella di metadati Hudi nel percorso. Amazon S3 Viene utilizzata per limitare il tempo di esecuzione del crawler.

## CatalogTarget struttura

Speciifica un AWS Glue Data Catalog obiettivo.

Campi

- `DatabaseName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del database da sincronizzare.

- `Tables` - obbligatorio: una matrice di stringhe UTF-8, almeno 1 stringa.

  Elenco di tabelle da sincronizzare.

- `ConnectionName`: stringa UTF-8, non inferiore a 1 o superiore a 2048 byte di lunghezza.

  Il nome della connessione per una tabella di Catalogo dati supportata da Amazon S3 come destinazione del crawling quando si utilizza un tipo di connessione `Catalog` abbinato a un tipo di connessione `NETWORK`.

- `EventQueueArn`: stringa UTF-8.

  Un ARN Amazon SQS valido. Ad esempio, `arn:aws:sqs:region:account:sqs`.
- `DlqEventQueueArn`: stringa UTF-8.

  Un ARN Amazon SQS di messaggi non recapitabili valido. Ad esempio, `arn:aws:sqs:region:account:deadLetterQueue`.

## CrawlerMetrics struttura

I parametri di un determinato crawler.

Campi

- `CrawlerName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del crawler.
- `TimeLeftSeconds`: numero (doppio), non superiore a None (Nessuno).

  Il tempo stimato che rimane per completare un crawling in esecuzione.
- `StillEstimating`: booleano.

  True se il crawler sta ancora valutando il tempo necessario per completare la sessione.
- `LastRuntimeSeconds`: numero (doppio), non superiore a None (Nessuno).

  La durata in secondi della sessione più recente del crawler.
- `MedianRuntimeSeconds`: numero (doppio), non superiore a None (Nessuno).

  La durata media in secondi delle sessioni del crawler.
- `TablesCreated`: numero (intero), non superiore a Nessuno.

  Il numero di tabelle create dal crawler.
- `TablesUpdated`: numero (intero), non superiore a Nessuno.

  Il numero di tabelle aggiornate dal crawler.
- `TablesDeleted`: numero (intero), non superiore a Nessuno.

  Il numero di tabelle eliminate dal crawler.

## CrawlerHistory struttura

Contiene le informazioni per l'esecuzione di un crawler.

Campi

- `CrawlId`: stringa UTF-8.

  Un identificatore UUID per ogni crawling.
- `State`: stringa UTF-8 (valori validi: `RUNNING` | `COMPLETED` | `FAILED` | `STOPPED`).

  Lo stato del crawling.
- `StartTime`: timestamp.

  La data e l'ora in cui è stata avviata l'esecuzione del crawler.
- `EndTime`: timestamp.

  La data e l'ora in cui è terminato il crawling.
- `Summary`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Un riepilogo dell'esecuzione per il crawling in JSON. Contiene le tabelle e le partizioni del catalogo che sono state aggiunte, aggiornate o eliminate.
- `ErrorMessage`: stringa di descrizione, non superiore a 2048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  Se si è verificato un errore, il messaggio di errore è associato al crawling.
- `LogGroup`: stringa UTF-8, non inferiore a 1 o superiore a 512 byte di lunghezza, corrispondente a Log group string pattern.

  Il gruppo di log associato al crawler.
- `LogStream`: stringa UTF-8, non inferiore a 1 o superiore a 512 byte di lunghezza, corrispondente a Log-stream string pattern.

  Il flusso di log associato all'esecuzione del crawler.
- `MessagePrefix`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il prefisso per un CloudWatch messaggio relativo a questo crawl.

- `DPUHour`: numero (doppio), non superiore a None (Nessuno).

  Il numero di unità di elaborazione dati (DPU) utilizzate in ore per il crawling.

## CrawlsFilter struttura

Un elenco di campi, comparatori e valori che puoi utilizzare per filtrare le esecuzioni del crawler per un crawler specificato.

Campi

- `FieldName`: stringa UTF-8 (valori validi: `CRAWL_ID` | `STATE` | `START_TIME` | `END_TIME` | `DPU_HOUR`).

  Una chiave utilizzata per filtrare le esecuzioni del crawler per un crawler specificato. I valori validi per ciascuno dei nomi di campo sono:

  - `CRAWL_ID`: una stringa che rappresenta l'identificatore UUID per un crawling.
  - `STATE`: una stringa che rappresenta lo stato del crawling.
  - `START_TIME` e `END_TIME`: il timestamp epoch in millisecondi.
  - `DPU_HOUR`: il numero di unità di elaborazione dati (DPU) utilizzate in ore per il crawling.
- `FilterOperator`: stringa UTF-8 (valori validi: `GT` | `GE` | `LT` | `LE` | `EQ` | `NE`).

  Un comparatore definito che opera sul valore. Gli operatori disponibili sono:

  - `GT`: maggiore di.
  - `GE`: maggiore o uguale a.
  - `LT`: minore di.
  - `LE`: minore o uguale a.
  - `EQ`: uguale a.
  - `NE`: non uguale a.
- `FieldValue`: stringa UTF-8.

  Il valore fornito per il confronto nel campo del crawling.

## SchemaChangePolicy struttura

Una policy che specifica i comportamenti di aggiornamento ed eliminazione per il crawler.

Campi

- `UpdateBehavior`: stringa UTF-8 (valori validi: `LOG` | `UPDATE_IN_DATABASE`).

  Il comportamento di aggiornamento quando il crawler riscontra una variazione dello schema.
- `DeleteBehavior`: stringa UTF-8 (valori validi: `LOG` | `DELETE_FROM_DATABASE` | `DEPRECATE_IN_DATABASE`).

  Il comportamento di eliminazione quando il crawler riscontra un oggetto eliminato.

## LastCrawlInfo struttura

Informazioni sullo stato e sull'errore relative al crawling più recente.

Campi

- `Status`: stringa UTF-8 (valori validi: `SUCCEEDED` | `CANCELLED` | `FAILED`).

  Stato dell'ultimo crawling.
- `ErrorMessage`: stringa di descrizione, non superiore a 2048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  Le informazioni sull'errore dell'ultimo crawling, se presente.
- `LogGroup`: stringa UTF-8, non inferiore a 1 o superiore a 512 byte di lunghezza, corrispondente a Log group string pattern.

  Il gruppo di log per l'ultimo crawling.
- `LogStream`: stringa UTF-8, non inferiore a 1 o superiore a 512 byte di lunghezza, corrispondente a Log-stream string pattern.

  Il flusso di log per l'ultimo crawling.
- `MessagePrefix`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il prefisso per un messaggio sul crawling.
- `StartTime`: timestamp.

  L'ora di inizio del crawling.

## RecrawlPolicy struttura

Quando si esegue il crawling di un'origine dati Amazon S3 dopo il completamento del primo crawling, specifica se eseguire nuovamente il crawling dell'intero set di dati o solo delle cartelle aggiunte dopo l'ultima esecuzione del crawler. Per ulteriori informazioni, consulta Crawling incrementali in AWS Glue nella guida per sviluppatori.

Campi

- `RecrawlBehavior`: stringa UTF-8 (valori validi: `CRAWL_EVERYTHING` | `CRAWL_NEW_FOLDERS_ONLY` | `CRAWL_EVENT_MODE`).

  Specifica se eseguire nuovamente il crawling dell'intero set di dati o solo delle cartelle aggiunte dall'ultima esecuzione del crawler.

  Un valore di `CRAWL_EVERYTHING` specifica nuovamente il crawling dell'intero set di dati.

  Un valore di `CRAWL_NEW_FOLDERS_ONLY` specifica il crawling solo delle cartelle che sono state aggiunte dopo l'ultima esecuzione del crawler.

  Un valore di `CRAWL_EVENT_MODE` specifica il crawling solo delle modifiche identificate dagli eventi Amazon S3.

## LineageConfiguration struttura

Specifica le impostazioni di configurazione della derivazione dei dati per il crawler.

Campi

- `CrawlerLineageSettings`: stringa UTF-8 (valori validi: `ENABLE` | `DISABLE`).

  Specifica se la derivazione dei dati è abilitata per il crawler. I valori validi sono:

  - ENABLE: abilita la derivazione dei dati per il crawler
  - DISABLE: disabilita la derivazione dei dati per il crawler

## LakeFormationConfiguration struttura

Specifica le impostazioni AWS Lake Formation di configurazione per il crawler.

Campi

- `UseLakeFormationCredentials`: booleano.

  Speciifica se utilizzare le AWS Lake Formation credenziali per il crawler anziché le credenziali del ruolo IAM.
- `AccountId`: stringa UTF-8, non superiore a 12 byte di lunghezza.

  Obbligatorio per i crawling tra più account. Per il crawling degli stessi account dei dati di destinazione, può essere lasciato come null.

## Operazioni

- CreateCrawler azione (Python: create_crawler)
- DeleteCrawler azione (Python: delete_crawler)
- GetCrawler azione (Python: get_crawler)
- GetCrawlers azione (Python: get_crawlers)
- GetCrawlerMetrics azione (Python: get_crawler_metrics)
- UpdateCrawler azione (Python: update_crawler)
- StartCrawler azione (Python: start_crawler)
- StopCrawler azione (Python: stop_crawler)
- BatchGetCrawlers azione (Python: batch_get_crawlers)
- ListCrawlers azione (Python: list_crawlers)
- ListCrawls azione (Python: list_crawls)

## CreateCrawler azione (Python: create_crawler)

Crea un nuovo crawler con destinazioni, ruolo, configurazione specifici e pianificazione opzionale. Deve essere specificata almeno una destinazione di crawling nel campo `s3Targets`, nel campo, `jdbcTargets` o nel campo `DynamoDBTargets`.

Richiesta

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

Il nome del nuovo crawler.

- `Role` - obbligatorio: stringa UTF-8.

  Il ruolo IAM o il nome della risorsa Amazon (ARN) di un ruolo IAM utilizzato dal nuovo crawler per accedere alle risorse dei clienti.

- `DatabaseName`: stringa UTF-8.

  Il AWS Glue database in cui vengono scritti i risultati, ad esempio:. `arn:aws:daylight:us-east-1::database/sometable/*`

- `Description`: stringa di descrizione, non superiore a 2048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  Descrizione del nuovo crawler.

- `Targets` - obbligatorio: un oggetto CrawlerTargets.

  Elenco della raccolta di destinazioni da sottoporre al crawling.

- `Schedule`: stringa UTF-8.

  Espressione `cron` usata per specificare la pianificazione (consulta Pianificazioni basate sul tempo per processi e crawler. Ad esempio, per eseguire un processo ogni giorno alle 12:15 UTC, devi specificare: `cron(15 12 * * ? *)`.

- `Classifiers`: una matrice di stringhe UTF-8.

  Elenco di classificatori personalizzati registrati dall'utente. Per impostazione predefinita, tutti i classificatori integrati sono inclusi in un crawling, ma i classificatori personalizzati sovrascrivono sempre i classificatori predefiniti per una determinata classificazione.

- `TablePrefix`: stringa UTF-8, non superiore a 128 byte di lunghezza.

  Il prefisso di tabella utilizzato per le tabelle di catalogo create.

- `SchemaChangePolicy`: un oggetto SchemaChangePolicy.

  Policy per il comportamento di aggiornamento ed eliminazione del crawler.

- `RecrawlPolicy`: un oggetto RecrawlPolicy.

  Una policy che specifica se eseguire nuovamente il crawling dell'intero set di dati o solo delle cartelle aggiunte dall'ultima esecuzione del crawler.

- `LineageConfiguration`: un oggetto LineageConfiguration.

Specifica le impostazioni di configurazione della derivazione dei dati per il crawler.

- `LakeFormationConfiguration`: un oggetto LakeFormationConfiguration.

  Specifica le impostazioni AWS Lake Formation di configurazione per il crawler.

- `Configuration`: stringa UTF-8.

  Le informazioni di configurazione del crawler. Questa stringa JSON con versione consente agli utenti di specificare gli aspetti del comportamento di un crawler. Per ulteriori informazioni, consulta la pagina Impostazione delle opzioni di configurazione del crawler.

- `CrawlerSecurityConfiguration`: stringa UTF-8, non superiore a 128 byte di lunghezza.

  Il nome della struttura `SecurityConfiguration` che questo crawler deve utilizzare.

- `Tags`: una matrice di mappe con coppie chiave-valore, non superiore alle 50 coppie.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Ogni valore è una stringa UTF-8, lunga non più di 256 byte.

  I tag da usare con questa richiesta crawler. Puoi usare i tag per limitare l'accesso al crawler. Per ulteriori informazioni sui tag in AWS Glue, consulta AWS Tags in AWS Glue nella guida per sviluppatori.

Risposta

- Nessun parametro di risposta.

Errori

- `InvalidInputException`
- `AlreadyExistsException`
- `OperationTimeoutException`
- `ResourceNumberLimitExceededException`

## DeleteCrawler azione (Python: delete_crawler)

Rimuove un crawler specificato da, a meno che lo stato del crawler non lo sia AWS Glue Data Catalog. `RUNNING`

Richiesta

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome del crawler da rimuovere.

Risposta

- Nessun parametro di risposta.

Errori

- `EntityNotFoundException`
- `CrawlerRunningException`
- `SchedulerTransitioningException`
- `OperationTimeoutException`

## GetCrawler azione (Python: get_crawler)

Recupera i metadati per un determinato crawler.

Richiesta

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome del crawler per cui recuperare i metadati.

Risposta

- `Crawler`: un oggetto Crawler.

  I metadati per il crawler specificato.

Errori

- `EntityNotFoundException`
- `OperationTimeoutException`

## GetCrawlers azione (Python: get_crawlers)

Recupera i metadati per tutti i crawler definiti nell'account del cliente.

Richiesta

- `MaxResults`: numero (intero), non inferiore a 1 o superiore a 1.000.

  Il numero di crawler da restituire per ciascuna chiamata.
- `NextToken`: stringa UTF-8.

  Token di continuazione, se si tratta di una richiesta di continuazione.

Risposta

- `Crawlers`: una matrice di oggetti Crawler.

  Elenco di metadati di crawler.
- `NextToken`: stringa UTF-8.

  Token di continuazione, se l'elenco restituito non ha raggiunto la fine delle voci definite in questo account del cliente.

Errori

- `OperationTimeoutException`

## GetCrawlerMetrics azione (Python: get_crawler_metrics)

Recupera i parametri sul crawler specificato.

Richiesta

- `CrawlerNameList`: una matrice di stringhe UTF-8, non superiore a 100.

Elenco di nomi di crawler su cui recuperare i parametri.

- `MaxResults`: numero (intero), non inferiore a 1 o superiore a 1.000.

  La dimensione massima di un elenco da restituire.

- `NextToken`: stringa UTF-8.

  Un token di continuazione, se si tratta di una chiamata di continuazione.

Risposta

- `CrawlerMetricsList`: una matrice di oggetti CrawlerMetrics.

  Elenco di parametri per il crawler specificato.

- `NextToken`: stringa UTF-8.

  Token di continuazione, se l'elenco restituito non contiene l'ultimo parametro disponibile.

Errori

- `OperationTimeoutException`

## UpdateCrawler azione (Python: update_crawler)

Aggiorna un crawler. Se un crawler è in esecuzione, è necessario arrestarlo utilizzando `StopCrawler` prima dell'aggiornamento.

Richiesta

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del nuovo crawler.

- `Role`: stringa UTF-8.

  Il ruolo IAM o il nome della risorsa Amazon (ARN) di un ruolo IAM utilizzato dal nuovo crawler per accedere alle risorse dei clienti.

- `DatabaseName`: stringa UTF-8.

Il AWS Glue database in cui sono archiviati i risultati, ad esempio:. `arn:aws:daylight:us-east-1::database/sometable/*`

- `Description`: stringa UTF-8, non superiore a 2048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  Descrizione del nuovo crawler.

- `Targets`: un oggetto CrawlerTargets.

  Elenco di destinazioni da sottoporre al crawling.

- `Schedule`: stringa UTF-8.

  Espressione `cron` usata per specificare la pianificazione (consulta Pianificazioni basate sul tempo per processi e crawler. Ad esempio, per eseguire un processo ogni giorno alle 12:15 UTC, devi specificare: `cron(15 12 * * ? *)`.

- `Classifiers`: una matrice di stringhe UTF-8.

  Elenco di classificatori personalizzati registrati dall'utente. Per impostazione predefinita, tutti i classificatori integrati sono inclusi in un crawling, ma i classificatori personalizzati sovrascrivono sempre i classificatori predefiniti per una determinata classificazione.

- `TablePrefix`: stringa UTF-8, non superiore a 128 byte di lunghezza.

  Il prefisso di tabella utilizzato per le tabelle di catalogo create.

- `SchemaChangePolicy`: un oggetto SchemaChangePolicy.

  Policy per il comportamento di aggiornamento ed eliminazione del crawler.

- `RecrawlPolicy`: un oggetto RecrawlPolicy.

  Una policy che specifica se eseguire nuovamente il crawling dell'intero set di dati o solo delle cartelle aggiunte dall'ultima esecuzione del crawler.

- `LineageConfiguration`: un oggetto LineageConfiguration.

  Specifica le impostazioni di configurazione della derivazione dei dati per il crawler.

- `LakeFormationConfiguration`: un oggetto LakeFormationConfiguration.

  Specifica le impostazioni AWS Lake Formation di configurazione per il crawler.

- `Configuration`: stringa UTF-8.

Le informazioni di configurazione del crawler. Questa stringa JSON con versione consente agli utenti di specificare gli aspetti del comportamento di un crawler. Per ulteriori informazioni, consulta la pagina Impostazione delle opzioni di configurazione del crawler.

- `CrawlerSecurityConfiguration`: stringa UTF-8, non superiore a 128 byte di lunghezza.

  Il nome della struttura `SecurityConfiguration` che questo crawler deve utilizzare.

Risposta

- Nessun parametro di risposta.

Errori

- `InvalidInputException`
- `VersionMismatchException`
- `EntityNotFoundException`
- `CrawlerRunningException`
- `OperationTimeoutException`

## StartCrawler azione (Python: start_crawler)

Avvia un crawling utilizzando il crawler specificato, indipendentemente dalla pianificazione. Se il crawler è già in esecuzione, restituisce un. CrawlerRunningException

Richiesta

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome del crawler da avviare.

Risposta

- Nessun parametro di risposta.

Errori

- `EntityNotFoundException`
- `CrawlerRunningException`
- `OperationTimeoutException`

## StopCrawler azione (Python: stop_crawler)

Se il crawler specificato è in esecuzione, arresta il crawling.

Richiesta

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome del crawler da arrestare.

Risposta

- Nessun parametro di risposta.

Errori

- `EntityNotFoundException`
- `CrawlerNotRunningException`
- `CrawlerStoppingException`
- `OperationTimeoutException`

## BatchGetCrawlers azione (Python: batch_get_crawlers)

Restituisce un elenco di metadati di risorse per un elenco di nomi di crawler. Dopo aver chiamato l'operazione `ListCrawlers`, puoi chiamare questa operazione per accedere ai dati a cui sono state concesse le autorizzazioni. Questa operazione supporta tutte le autorizzazioni IAM, tra cui le condizioni di autorizzazione che utilizzano i tag.

Richiesta

- `CrawlerNames` - obbligatorio: una serie di stringhe UTF-8, non superiore a 100 stringhe.

L'elenco dei nomi di crawler che potrebbero essere i nomi restituiti dall'operazione `ListCrawlers`.

Risposta

- `Crawlers`: una matrice di oggetti Crawler.

  Un elenco di definizioni di crawler.
- `CrawlersNotFound`: una matrice di stringhe UTF-8, non superiore a 100.

  Un elenco di nomi di crawler non trovati.

Errori

- `InvalidInputException`
- `OperationTimeoutException`

## ListCrawlers azione (Python: list_crawlers)

Recupera i nomi di tutte le risorse del crawler in questo AWS account o delle risorse con il tag specificato. Questa operazione consente di vedere quali risorse sono disponibili nel proprio account e i relativi nomi.

L'operazione accetta il campo facoltativo `Tags` che si può utilizzare come filtro per la risposta in modo che le risorse con tag possano essere recuperate come gruppo. Se si sceglie di utilizzare il filtro dei tag, potranno essere recuperate solo le risorse con tag.

Richiesta

- `MaxResults`: numero (intero), non inferiore a 1 o superiore a 1.000.

  La dimensione massima di un elenco da restituire.
- `NextToken`: stringa UTF-8.

  Token di continuazione, se si tratta di una richiesta di continuazione.
- `Tags`: una matrice di mappe con coppie chiave-valore, non superiore alle 50 coppie.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

Ogni valore è una stringa UTF-8, lunga non più di 256 byte.

Specifica che vengono restituite solo le risorse con tag.

Risposta

- `CrawlerNames`: una matrice di stringhe UTF-8, non superiore a 100.

  I nomi di tutti i crawler nell'account oppure i crawler con i tag specificati.
- `NextToken`: stringa UTF-8.

  Token di continuazione, se l'elenco restituito non contiene l'ultimo parametro disponibile.

Errori

- `OperationTimeoutException`

## ListCrawls azione (Python: list_crawls)

Restituisce tutti i crawling di un determinato crawler. Restituisce solo i crawling che si sono verificati dalla data di avvio della funzione cronologia del crawler e conserva solo fino a 12 mesi di crawling. I crawling più vecchi non verranno restituiti.

È possibile utilizzare questa API per:

- Recuperare tutti i crawling di un determinato crawler.
- Recuperare tutti i crawling di un crawler specificato entro un conteggio limitato.
- Recuperare tutti i crawling di un crawler specificato in un intervallo di tempo specifico.
- Recuperare tutti i crawling di un crawler specificato con uno stato particolare, un ID di crawling o un valore orario della DPU.

Richiesta

- `CrawlerName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del crawler di cui vuoi recuperare le esecuzioni.

- `MaxResults`: numero (intero), non inferiore a 1 o superiore a 1.000.

  Numero massimo di risultati da restituire. Il valore predefinito è 20 e il valore massimo è 100.
- `Filters`: una matrice di oggetti CrawlsFilter.

  Filtra i crawling in base ai criteri specificati in un elenco di oggetti `CrawlsFilter`.
- `NextToken`: stringa UTF-8.

  Un token di continuazione, se si tratta di una chiamata di continuazione.

Risposta

- `Crawls`: una matrice di oggetti CrawlerHistory.

  Un elenco di oggetti `CrawlerHistory` che rappresentano le esecuzioni del crawling che soddisfano i criteri specificati.
- `NextToken`: stringa UTF-8.

  Un token di continuazione per impaginare l'elenco restituito di token, restituiti se il segmento corrente dell'elenco non è l'ultimo.

Errori

- `EntityNotFoundException`
- `OperationTimeoutException`
- `InvalidInputException`

## API delle statistiche delle colonne

L'API Column Statistics descrive come AWS Glue APIs restituire statistiche sulle colonne di una tabella.

### Tipi di dati

- ColumnStatisticsTaskRun struttura
- ColumnStatisticsTaskSettings struttura
- ExecutionAttempt struttura

## ColumnStatisticsTaskRun struttura

L'oggetto che mostra i dettagli dell'esecuzione delle statistiche delle colonne.

Campi

- `CustomerId`: stringa UTF-8, non superiore a 12 byte di lunghezza.

  L'ID AWS dell'account.
- `ColumnStatisticsTaskRunId`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'identificatore per l'esecuzione dell'attività delle statistiche delle colonne specifica.
- `DatabaseName`: stringa UTF-8.

  Il database in cui risiede la tabella.
- `TableName`: stringa UTF-8.

  Il nome della tabella per cui vengono generate le statistiche delle colonne.
- `ColumnNameList`: una matrice di stringhe UTF-8.

  Un elenco dei nomi delle colonne. Se non viene fornito, per impostazione predefinita verranno utilizzati tutti i nomi delle colonne della tabella.
- `CatalogID`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo dati in cui risiede la tabella Se non ne viene fornito nessuno, per impostazione predefinita viene utilizzato l'ID dell' AWS account.
- `Role`: stringa UTF-8.

  Il ruolo IAM che assume il servizio per generare statistiche.
- `SampleSize`: numero (doppio), non superiore a 100.

  La percentuale di righe utilizzate per generare statistiche. Se non viene fornita, per generare statistiche verrà utilizzata l'intera tabella.
- `SecurityConfiguration`: stringa UTF-8, non superiore a 128 byte di lunghezza.

  Nome della configurazione di sicurezza utilizzata per crittografare CloudWatch i log per l'esecuzione dell'attività Column stats.

- `NumberOfWorkers`: numero (intero), almeno 1.

  Il numero di worker utilizzati per generare statistiche delle colonne. Il processo è preconfigurato per scalare automaticamente fino a 25 istanze.

- `WorkerType`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il tipo di worker utilizzati per generare statistiche. Il valore predefinito è `g.1x`.

- `ComputationType`: stringa UTF-8 (valori validi: `FULL` | `INCREMENTAL`).

  Il tipo di calcolo delle statistiche sulle colonne.

- `Status`: stringa UTF-8 (valori validi: `STARTING` | `RUNNING` | `SUCCEEDED` | `FAILED` | `STOPPED`).

  Lo stato dell'esecuzione dell'attività.

- `CreationTime`: timestamp.

  L'ora di creazione di questa attività.

- `LastUpdated`: timestamp.

  Il momento dell'ultima modifica di questa attività.

- `StartTime`: timestamp.

  L'orario di inizio dell'attività.

- `EndTime`: timestamp.

  L'orario di fine dell'attività.

- `ErrorMessage`: stringa di descrizione, non superiore a 2048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  Il messaggio di errore per il processo.

- `DPUSeconds`: numero (doppio), non superiore a None (Nessuno).

  L'utilizzo della DPU calcolato in secondi per tutti i worker con scalabilità automatica.

## ColumnStatisticsTaskSettings struttura

Le impostazioni di un'attività di statistica sulle colonne.

Campi

- `DatabaseName`: stringa UTF-8.

  Nome del database in cui risiede la tabella.

- `TableName`: stringa UTF-8.

  Il nome della tabella per cui generare le statistiche sulle colonne.

- `Schedule`: un oggetto Schedule.

  Una pianificazione per l'esecuzione delle statistiche sulle colonne, specificata in sintassi CRON.

- `ColumnNameList`: una matrice di stringhe UTF-8.

  Un elenco di nomi delle colonne per le quali eseguire le statistiche.

- `CatalogID`: stringa ID catalogo, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo dati in cui risiede il database.

- `Role`: stringa UTF-8.

  Il ruolo utilizzato per l'esecuzione delle statistiche sulle colonne.

- `SampleSize`: numero (doppio), non superiore a 100.

  La percentuale di dati da campionare.

- `SecurityConfiguration`: stringa UTF-8, non superiore a 128 byte di lunghezza.

  Nome della configurazione di sicurezza utilizzata per crittografare i CloudWatch log.

- `ScheduleType`: stringa UTF-8 (valori validi: `CRON` | `AUTO`).

  Il tipo di pianificazione per un'attività di statistica sulle colonne. I valori possibili sono `CRON` o `AUTO`.

- `SettingSource`: stringa UTF-8 (valori validi: `CATALOG` | `TABLE`).

  L'origine dell'impostazione dell'attività di statistica sulle colonne. I valori possibili sono `CATALOG` o `TABLE`.

- `LastExecutionAttempt`: un oggetto ExecutionAttempt.

  L'ultimo `ExecutionAttempt` per l'esecuzione dell'attività di statistica sulle colonne.

## ExecutionAttempt struttura

Un tentativo di esecuzione dell'attività di statistica sulle colonne.

Campi

- `Status`: stringa UTF-8 (valori validi: `FAILED` | `STARTED`).

  Lo stato dell'ultima esecuzione dell'attività di statistica sulle colonne.
- `ColumnStatisticsTaskRunId`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Un ID esecuzione attività per l'esecuzione dell'attività di statistica sulle colonne.
- `ExecutionTimestamp`: timestamp.

  Un timestamp del momento in cui si è verificata l'ultima esecuzione dell'attività di statistica sulle colonne.
- `ErrorMessage`: stringa di descrizione, non superiore a 2048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  Un messaggio di errore associato all'ultima esecuzione dell'attività di statistica sulle colonne.

## Operazioni

- StartColumnStatisticsTaskRun azione (Python: start_column_statistics_task_run)
- GetColumnStatisticsTaskRun azione (Python: get_column_statistics_task_run)
- GetColumnStatisticsTaskRuns azione (Python: get_column_statistics_task_runs)
- ListColumnStatisticsTaskRuns azione (Python: list_column_statistics_task_runs)
- StopColumnStatisticsTaskRun azione (Python: stop_column_statistics_task_run)
- CreateColumnStatisticsTaskSettings azione (Python: create_column_statistics_task_settings)
- UpdateColumnStatisticsTaskSettings azione (Python: update_column_statistics_task_settings)
- GetColumnStatisticsTaskSettings azione (Python: get_column_statistics_task_settings)
- DeleteColumnStatisticsTaskSettings azione (Python: delete_column_statistics_task_settings)
- StartColumnStatisticsTaskRunSchedule azione (Python: start_column_statistics_task_run_schedule)

- StopColumnStatisticsTaskRunSchedule azione (Python: stop_column_statistics_task_run_schedule)

## StartColumnStatisticsTaskRun azione (Python: start_column_statistics_task_run)

Avvia l'esecuzione di un'attività di statistica delle colonne, per una tabella e delle colonne specificate.

Richiesta

- `DatabaseName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome del database in cui risiede la tabella.

- `TableName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome della tabella per generare statistiche.

- `ColumnNameList`: una matrice di stringhe UTF-8.

  Un elenco dei nomi delle colonne per generare statistiche. Se non viene fornito, per impostazione predefinita verranno utilizzati tutti i nomi delle colonne della tabella.

- `Role` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il ruolo IAM che assume il servizio per generare statistiche.

- `SampleSize`: numero (doppio), non superiore a 100.

  La percentuale di righe utilizzate per generare statistiche. Se non viene fornita, per generare statistiche verrà utilizzata l'intera tabella.

- `CatalogID`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo dati in cui risiede la tabella. Se non viene fornito, per impostazione predefinita viene utilizzato l'ID dell'account AWS .

- `SecurityConfiguration`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

Nome della configurazione di sicurezza utilizzata per crittografare i log per l'esecuzione dell'attività Column stats. CloudWatch

Risposta

- `ColumnStatisticsTaskRunId`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'identificatore per l'esecuzione dell'attività delle statistiche delle colonne.

Errori

- `AccessDeniedException`
- `EntityNotFoundException`
- `ColumnStatisticsTaskRunningException`
- `OperationTimeoutException`
- `ResourceNumberLimitExceededException`
- `InvalidInputException`

## GetColumnStatisticsTaskRun azione (Python: get_column_statistics_task_run)

Ottieni l'associato per l'esecuzione di un'attività, con un ID di esecuzione dell'attività. metadata/ information

Richiesta

- `ColumnStatisticsTaskRunId` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'identificatore per l'esecuzione dell'attività delle statistiche delle colonne specifica.

Risposta

- `ColumnStatisticsTaskRun`: un oggetto ColumnStatisticsTaskRun.

Un oggetto `ColumnStatisticsTaskRun` che rappresenta i dettagli dell'esecuzione delle statistiche delle colonne.

Errori

- `EntityNotFoundException`
- `OperationTimeoutException`
- `InvalidInputException`

## GetColumnStatisticsTaskRuns azione (Python: get_column_statistics_task_runs)

Recupera le informazioni su tutte le esecuzioni associate alla tabella specificata.

Richiesta

- `DatabaseName` - obbligatorio: stringa UTF-8.

  Nome del database in cui risiede la tabella.
- `TableName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome della tabella.
- `MaxResults`: numero (intero), non inferiore a 1 o superiore a 1.000.

  La dimensione massima della risposta.
- `NextToken`: stringa UTF-8.

  Un token di continuazione, se si tratta di una chiamata di continuazione.

Risposta

- `ColumnStatisticsTaskRuns`: una matrice di oggetti ColumnStatisticsTaskRun.

  Un elenco delle esecuzioni dell'attività delle statistiche delle colonne.
- `NextToken`: stringa UTF-8.

  Un token di continuazione, se non sono ancora stati restituite tutte le esecuzioni dell'attività.

Errori

- `OperationTimeoutException`

## ListColumnStatisticsTaskRuns azione (Python: list_column_statistics_task_runs)

Elenca tutte le attività eseguite per un determinato account.

Richiesta

- `MaxResults`: numero (intero), non inferiore a 1 o superiore a 1.000.

  La dimensione massima della risposta.
- `NextToken`: stringa UTF-8.

  Un token di continuazione, se si tratta di una chiamata di continuazione.

Risposta

- `ColumnStatisticsTaskRunIds`: una matrice di stringhe UTF-8, non superiore a 100.

  Un elenco di attività di statistica delle colonne eseguite. IDs
- `NextToken`: stringa UTF-8.

  Un token di continuazione, se non tutte le attività eseguite IDs sono state ancora restituite.

Errori

- `OperationTimeoutException`

## StopColumnStatisticsTaskRun azione (Python: stop_column_statistics_task_run)

Interrompe l'esecuzione di un'operazione per la tabella specificata.

Richiesta

- `DatabaseName` - obbligatorio: stringa UTF-8.

  Nome del database in cui risiede la tabella.

- `TableName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome della tabella.

Risposta

- Nessun parametro di risposta.

Errori

- `EntityNotFoundException`
- `ColumnStatisticsTaskNotRunningException`
- `ColumnStatisticsTaskStoppingException`
- `OperationTimeoutException`

## CreateColumnStatisticsTaskSettings azione (Python: create_column_statistics_task_settings)

Crea le impostazioni per un'attività di statistica sulle colonne.

Richiesta

- `DatabaseName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome del database in cui risiede la tabella.
- `TableName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome della tabella per cui generare le statistiche sulle colonne.
- `Role` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il ruolo utilizzato per l'esecuzione delle statistiche sulle colonne.
- `Schedule`: stringa UTF-8.

Una pianificazione per l'esecuzione delle statistiche sulle colonne, specificata in sintassi CRON.

- `ColumnNameList`: una matrice di stringhe UTF-8.

  Un elenco di nomi delle colonne per le quali eseguire le statistiche.

- `SampleSize`: numero (doppio), non superiore a 100.

  La percentuale di dati da campionare.

- `CatalogID`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo dati in cui risiede il database.

- `SecurityConfiguration`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome della configurazione di sicurezza utilizzata per crittografare i log. CloudWatch

- `Tags`: una matrice di mappe con coppie chiave-valore, non superiore alle 50 coppie.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Ogni valore è una stringa UTF-8, lunga non più di 256 byte.

  Una mappa di tag.

Risposta

- Nessun parametro di risposta.

Errori

- `AlreadyExistsException`
- `AccessDeniedException`
- `EntityNotFoundException`
- `InvalidInputException`
- `OperationTimeoutException`
- `ResourceNumberLimitExceededException`
- `ColumnStatisticsTaskRunningException`

## UpdateColumnStatisticsTaskSettings azione (Python: update_column_statistics_task_settings)

Aggiorna le impostazioni per un'attività di statistica sulle colonne.

Richiesta

- `DatabaseName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome del database in cui risiede la tabella.

- `TableName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome della tabella per cui generare le statistiche sulle colonne.

- `Role`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il ruolo utilizzato per l'esecuzione delle statistiche sulle colonne.

- `Schedule`: stringa UTF-8.

  Una pianificazione per l'esecuzione delle statistiche sulle colonne, specificata in sintassi CRON.

- `ColumnNameList`: una matrice di stringhe UTF-8.

  Un elenco di nomi delle colonne per le quali eseguire le statistiche.

- `SampleSize`: numero (doppio), non superiore a 100.

  La percentuale di dati da campionare.

- `CatalogID`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del catalogo dati in cui risiede il database.

- `SecurityConfiguration`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome della configurazione di sicurezza utilizzata per crittografare i log. CloudWatch

Risposta

- Nessun parametro di risposta.

Errori

- `AccessDeniedException`
- `EntityNotFoundException`
- `InvalidInputException`
- `VersionMismatchException`
- `OperationTimeoutException`

## GetColumnStatisticsTaskSettings azione (Python: get_column_statistics_task_settings)

Recupera le impostazioni per un'attività di statistica sulle colonne.

Richiesta

- `DatabaseName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome del database in cui risiede la tabella.
- `TableName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome della tabella per cui recuperare le statistiche sulle colonne.

Risposta

- `ColumnStatisticsTaskSettings`: un oggetto ColumnStatisticsTaskSettings.

  Un oggetto `ColumnStatisticsTaskSettings` che rappresenta le impostazioni per l'attività di statistica sulle colonne.

Errori

- `EntityNotFoundException`

- `InvalidInputException`
- `OperationTimeoutException`

## DeleteColumnStatisticsTaskSettings azione (Python: delete_column_statistics_task_settings)

Elimina le impostazioni per un'attività di statistica sulle colonne.

Richiesta

- `DatabaseName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome del database in cui risiede la tabella.
- `TableName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome della tabella per cui eliminare le statistiche sulle colonne.

Risposta

- Nessun parametro di risposta.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `OperationTimeoutException`

## StartColumnStatisticsTaskRunSchedule azione (Python: start_column_statistics_task_run_schedule)

Avvia la pianificazione di esecuzione di un'attività di statistica sulle colonne.

Richiesta

- `DatabaseName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome del database in cui risiede la tabella.
- `TableName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome della tabella per cui avviare la pianificazione di esecuzione di un'attività di statistica sulle colonne.

Risposta

- Nessun parametro di risposta.

Errori

- `AccessDeniedException`
- `EntityNotFoundException`
- `InvalidInputException`
- `OperationTimeoutException`

## StopColumnStatisticsTaskRunSchedule azione (Python: stop_column_statistics_task_run_schedule)

Interrompe la pianificazione di esecuzione di un'attività di statistica sulle colonne.

Richiesta

- `DatabaseName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome del database in cui risiede la tabella.
- `TableName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

Il nome della tabella per cui interrompere la pianificazione di esecuzione di un'attività di statistica sulle colonne.

Risposta

- Nessun parametro di risposta.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `OperationTimeoutException`

## Eccezioni

- ColumnStatisticsTaskRunningException struttura
- ColumnStatisticsTaskNotRunningException struttura
- ColumnStatisticsTaskStoppingException struttura
- ColumnStatisticsTaskAutoConcurrencyLimitException struttura
- InvalidCatalogSettingException struttura

## ColumnStatisticsTaskRunningException struttura

Un'eccezione generata quando si cerca di avviare un altro processo durante l'esecuzione di un processo di generazione di statistiche delle colonne.

Campi

- `Message`: stringa UTF-8.

  Messaggio che descrive il problema.

## ColumnStatisticsTaskNotRunningException struttura

Un'eccezione generata quando si tenta di interrompere l'esecuzione di un'attività quando non è in esecuzione alcuna attività.

Campi

- `Message`: stringa UTF-8.

  Messaggio che descrive il problema.

## ColumnStatisticsTaskStoppingException struttura

Un'eccezione generata quando si tenta di interrompere l'esecuzione di un'attività.

Campi

- `Message`: stringa UTF-8.

  Messaggio che descrive il problema.

## ColumnStatisticsTaskAutoConcurrencyLimitException struttura

Un'eccezione generata quando è già stato raggiunto il limite di processi automatici di statistica simultanei.

Campi

- `Message`: stringa UTF-8.

  Messaggio che descrive il problema.

## InvalidCatalogSettingException struttura

Un'eccezione generata quando si verifica un problema con le impostazioni del catalogo.

Campi

- `Message`: stringa UTF-8.

  Messaggio che descrive il problema.

# API del pianificatore del crawler

L'API Crawler scheduler descrive i tipi di dati del AWS Glue crawler, oltre all'API per la creazione, l'eliminazione, l'aggiornamento e l'elenco dei crawler.

## Tipi di dati

- Struttura della pianificazione

## Struttura della pianificazione

Oggetto di pianificazione che utilizza una dichiarazione `cron` per pianificare un evento.

Campi

- `ScheduleExpression`: stringa UTF-8.

  Espressione `cron` usata per specificare la pianificazione (consulta Pianificazioni basate sul tempo per processi e crawler. Ad esempio, per eseguire un processo ogni giorno alle 12:15 UTC, devi specificare: `cron(15 12 * * ? *)`.
- `State`: stringa UTF-8 (valori validi: `SCHEDULED` | `NOT_SCHEDULED` | `TRANSITIONING`).

  Lo stato della pianificazione.

## Operazioni

- UpdateCrawlerSchedule azione (Python: update_crawler_schedule)
- StartCrawlerSchedule azione (Python: start_crawler_schedule)
- StopCrawlerSchedule azione (Python: stop_crawler_schedule)

## UpdateCrawlerSchedule azione (Python: update_crawler_schedule)

Aggiorna la pianificazione di un crawler utilizzando un'espressione `cron`.

Richiesta

- `CrawlerName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

Nome del crawler la cui programmazione è da aggiornare.

- `Schedule`: stringa UTF-8.

  Espressione `cron` aggiornata usata per specificare la pianificazione, consulta Pianificazioni basate sul tempo per processi e crawler. Ad esempio, per eseguire un processo ogni giorno alle 12:15 UTC, devi specificare: `cron(15 12 * * ? *)`.

Risposta

- Nessun parametro di risposta.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `VersionMismatchException`
- `SchedulerTransitioningException`
- `OperationTimeoutException`

## StartCrawlerSchedule azione (Python: start_crawler_schedule)

Cambia lo stato della pianificazione del crawler specificato su `SCHEDULED`, a meno che il crawler non sia già in esecuzione o lo stato della pianificazione sia già impostata su `SCHEDULED`.

Richiesta

- `CrawlerName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome del crawler da pianificare.

Risposta

- Nessun parametro di risposta.

Errori

- `EntityNotFoundException`
- `SchedulerRunningException`
- `SchedulerTransitioningException`
- `NoScheduleException`
- `OperationTimeoutException`

### StopCrawlerSchedule azione (Python: stop_crawler_schedule)

Imposta lo stato della pianificazione del crawler specificato su `NOT_SCHEDULED`, ma non arresta il crawler se è già in esecuzione.

Richiesta

- `CrawlerName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome del crawler il cui stato della programmazione è da impostare.

Risposta

- Nessun parametro di risposta.

Errori

- `EntityNotFoundException`
- `SchedulerNotRunningException`
- `SchedulerTransitioningException`
- `OperationTimeoutException`

## API script ETL auto-generanti

L'API di generazione degli script ETL descrive i tipi di dati e l'API per la generazione di script ETL in AWS Glue.

# Tipi di dati


- CodeGenNode struttura
- CodeGenNodeArg struttura
- CodeGenEdge struttura
- Struttura della posizione
- CatalogEntry struttura
- MappingEntry struttura


## CodeGenNode struttura

Rappresenta un nodo in un grafo aciclico orientato (DAG)

Campi

- `Id` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Identifier string pattern.

  Un identificatore del nodo univoco all'interno del grafo del nodo.
- `NodeType`. Obbligatorio: stringa UTF-8.

  Il tipo di nodo.
- `Args`: obbligatorio: una matrice di oggetti CodeGenNodeArg, non superiore a 50 strutture.

  Proprietà del nodo sotto forma di coppie nome-valore.
- `LineNumber`: numero (intero).

  Il numero di riga del nodo.

## CodeGenNodeArg struttura

Un argomento o una proprietà di un nodo.

Campi

- `Name`. Obbligatorio: stringa UTF-8.

Il nome dell'argomento o della proprietà.

- `Value`. Obbligatorio: stringa UTF-8.

  Il valore dell'argomento o della proprietà.

- `Param`: booleano.

  True se il valore viene utilizzato come parametro.

## CodeGenEdge struttura

Rappresenta un edge direzionale in un grafo aciclico orientato (DAG).

Campi

- `Source` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Identifier string pattern.

  L'ID del nodo in cui inizia l'edge.

- `Target` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Identifier string pattern.

  L'ID del nodo in cui termina l'edge.

- `TargetParameter`: stringa UTF-8.

  La destinazione dell'edge.

## Struttura della posizione

La posizione delle risorse.

Campi

- `Jdbc`: una matrice di oggetti CodeGenNodeArg, non superiore a 50 strutture.

  Una posizione JDBC.

- `S3`: una matrice di oggetti CodeGenNodeArg, non superiore a 50 strutture.

  Posizione Amazon Simple Storage Service (Amazon S3).

- `DynamoDB`: una matrice di oggetti CodeGenNodeArg, non superiore a 50 strutture.

  Posizione di una tabella Amazon DynamoDB.

## CatalogEntry struttura

Specifica una definizione di tabella in AWS Glue Data Catalog.

Campi

- `DatabaseName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il database in cui risiedono i metadata della tabella.
- `TableName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome della tabella in questione.

## MappingEntry struttura

Definisce una mappatura.

Campi

- `SourceTable`: stringa UTF-8.

  Il nome della tabella di origine.
- `SourcePath`: stringa UTF-8.

  Il percorso di origine .
- `SourceType`: stringa UTF-8.

  Il tipo di sorgente.
- `TargetTable`: stringa UTF-8.

  La tabella di destinazione.
- `TargetPath`: stringa UTF-8.

Il percorso di destinazione.

- `TargetType`: stringa UTF-8.

  Il tipo di destinazione.

## Operazioni

- CreateScript azione (Python: create_script)
- GetDataflowGraph azione (Python: get_dataflow_graph)
- GetMapping azione (Python: get_mapping)
- GetPlan azione (Python: get_plan)

## CreateScript azione (Python: create_script)

Trasforma un grafo aciclico orientato (DAG) in codice.

Richiesta

- `DagNodes`: una matrice di oggetti CodeGenNode.

  Un elenco dei nodi del DAG.

- `DagEdges`: una matrice di oggetti CodeGenEdge.

  Un elenco dei confini del DAG.

- `Language`: stringa UTF-8 (valori validi: `PYTHON` | `SCALA`).

  Il linguaggio di programmazione del codice derivante dal DAG.

Risposta

- `PythonScript`: stringa UTF-8.

  Lo script in Python generato dal DAG.

- `ScalaCode`: stringa UTF-8.

  Il codice Scala generato dal DAG.

Errori

- `InvalidInputException`
- `InternalServiceException`
- `OperationTimeoutException`

## GetDataflowGraph azione (Python: get_dataflow_graph)

Trasforma uno script in Python in un grafo aciclico orientato (DAG).

Richiesta

- `PythonScript`: stringa UTF-8.

  Lo script in Python da trasformare.

Risposta

- `DagNodes`: una matrice di oggetti CodeGenNode.

  Un elenco dei nodi del DAG risultante.
- `DagEdges`: una matrice di oggetti CodeGenEdge.

  Un elenco dei confini del DAG risultante.

Errori

- `InvalidInputException`
- `InternalServiceException`
- `OperationTimeoutException`

## GetMapping azione (Python: get_mapping)

Crea mappature.

Richiesta

- `Source` - obbligatorio: un oggetto CatalogEntry.

Specifica la tabella di origine.

- `Sinks`: una matrice di oggetti CatalogEntry.

  Un elenco di tabelle di destinazione.

- `Location`: un oggetto Location (Ubicazione).

  Parametri per la mappatura.

Risposta

- `Mapping` - obbligatorio: una matrice di oggetti MappingEntry.

  Un elenco delle mappature per le destinazioni specificate.

Errori

- `InvalidInputException`
- `InternalServiceException`
- `OperationTimeoutException`
- `EntityNotFoundException`

## GetPlan azione (Python: get_plan)

Ottiene il codice per eseguire una mappatura specificata.

Richiesta

- `Mapping` - obbligatorio: una matrice di oggetti MappingEntry.

  L'elenco delle mappature da una tabella di origine per le tabelle di destinazione.

- `Source` - obbligatorio: un oggetto CatalogEntry.

  La tabella di origine.

- `Sinks`: una matrice di oggetti CatalogEntry.

  Le tabelle di destinazione.

- `Location`: un oggetto Location (Ubicazione).

  Parametri per la mappatura.

- `Language`: stringa UTF-8 (valori validi: `PYTHON` | `SCALA`).

  Il linguaggio di programmazione del codice per eseguire la mappatura.

- `AdditionalPlanOptionsMap`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8.

  Ogni valore è una stringa UTF-8.

  Una mappa per contenere parametri facoltativi chiave-valore aggiuntivi.

  Attualmente, vengono supportate queste coppie chiave-valore:

  - `inferSchema`— Speciifica se impostare su `inferSchema` true o false per lo script predefinito generato da un processo. AWS Glue Ad esempio, per impostare `inferSchema` su true, bisogna fornire la seguente coppia di chiave-valore:

    ```
    --additional-plan-options-map '{"inferSchema":"true"}'
    ```

Risposta

- `PythonScript`: stringa UTF-8.

  Uno script in Python per eseguire la mappatura.

- `ScalaCode`: stringa UTF-8.

  Codice Scala per eseguire la mappatura.

Errori

- `InvalidInputException`
- `InternalServiceException`
- `OperationTimeoutException`

# API processo visuale

L'API Visual Job consente di creare processi di integrazione dei dati utilizzando l' AWS Glue API di un oggetto JSON che rappresenta una configurazione visiva di un AWS Glue lavoro.

Viene fornito un elenco `CodeGenConfigurationNodes` di API per la creazione o l'aggiornamento del lavoro per registrare un DAG in AWS Glue Studio per il lavoro creato e generare il codice associato.

## Tipi di dati

- CodeGenConfigurationNode struttura
- JDBCConnectorStruttura delle opzioni
- StreamingDataPreviewOptions struttura
- AthenaConnectorSource struttura
- JDBCConnectorStruttura del codice sorgente
- SparkConnectorSource struttura
- CatalogSource struttura
- Struttura My SQLCatalog Source
- Struttura di Postgree Source SQLCatalog
- Struttura SQLCatalog Oracle Source
- SQLServerCatalogSource Struttura Microsoft
- CatalogKinesisSource struttura
- DirectKinesisSource struttura
- KinesisStreamingSourceOptions struttura
- CatalogKafkaSource struttura
- DirectKafkaSource struttura
- KafkaStreamingSourceOptions struttura
- RedshiftSource struttura
- AmazonRedshiftSource struttura
- AmazonRedshiftNodeData struttura
- AmazonRedshiftAdvancedOption struttura

- Struttura Option
- struttura S3 CatalogSource
- Struttura S3 SourceAdditionalOptions
- Struttura S3 CsvSource
- JDBCSource Struttura diretta
- Struttura S3 DirectSourceAdditionalOptions
- Struttura S3 JsonSource
- Struttura S3 ParquetSource
- Struttura S3 DeltaSource
- Struttura S3 CatalogDeltaSource
- CatalogDeltaSource struttura
- Struttura S3 HudiSource
- Struttura S3 CatalogHudiSource
- Struttura S3 ExcelSource
- CatalogHudiSource struttura
- Struttura Dynamo DBCatalog Source
- RelationalCatalogSource struttura
- JDBCConnectorStruttura dell'obiettivo
- SparkConnectorTarget struttura
- BasicCatalogTarget struttura
- La mia struttura SQLCatalog Target
- Struttura di Postgree Target SQLCatalog
- Struttura di Oracle Target SQLCatalog
- SQLServerCatalogTarget Struttura Microsoft
- RedshiftTarget struttura
- AmazonRedshiftTarget struttura
- UpsertRedshiftTargetOptions struttura
- struttura S3 CatalogTarget
- Struttura S3 GlueParquetTarget

- CatalogSchemaChangePolicy struttura
- struttura S3 DirectTarget
- Struttura S3 HudiCatalogTarget
- Struttura S3 HudiDirectTarget
- Struttura S3 DeltaCatalogTarget
- Struttura S3 DeltaDirectTarget
- Struttura S3 HyperDirectTarget
- Struttura S3 IcebergDirectTarget
- DirectSchemaChangePolicy struttura
- ApplyMapping struttura
- Struttura mappatura
- SelectFields struttura
- DropFields struttura
- RenameField struttura
- Struttura Spigot
- Struttura join
- JoinColumn struttura
- SplitFields struttura
- SelectFromCollection struttura
- FillMissingValues struttura
- Struttura filtro
- FilterExpression struttura
- FilterValue struttura
- CustomCode struttura
- Struttura SparkSQL
- SqlAlias struttura
- DropNullFields struttura
- NullCheckBoxList struttura
- NullValueField struttura

- Struttura Datatype
- Struttura Merge
- Struttura unione
- PIIDetection struttura
- Struttura aggregata
- DropDuplicates struttura
- GovernedCatalogTarget struttura
- GovernedCatalogSource struttura
- AggregateOperation struttura
- GlueSchema struttura
- GlueStudioSchemaColumn struttura
- GlueStudioColumn struttura
- DynamicTransform struttura
- TransformConfigParameter struttura
- EvaluateDataQuality struttura
- DQResultsPublishingOptions struttura
- DQStopJobOnFailureOptions struttura
- EvaluateDataQualityMultiFrame struttura
- Struttura Recipe
- RecipeReference struttura
- SnowflakeNodeData struttura
- SnowflakeSource struttura
- SnowflakeTarget struttura
- ConnectorDataSource struttura
- ConnectorDataTarget struttura
- RecipeStep struttura
- RecipeAction struttura
- ConditionExpression struttura
- Struttura S3 CatalogIcebergSource

- CatalogIcebergSource struttura
- Struttura S3 IcebergCatalogTarget
- DBELTConnectorStruttura Dynamo Source
- DDBELTConnectionStruttura delle opzioni
- DDBELTCatalogAdditionalOptions struttura
- Struttura Route
- GroupFilters struttura
- AutoDataQuality struttura

## CodeGenConfigurationNode struttura

`CodeGenConfigurationNode` enumera tutti i tipi di nodo validi. È possibile compilare una e solo una delle variabili membro.

Campi

- `AthenaConnectorSource`: un oggetto AthenaConnectorSource.

  Specifica un connettore per un'origine dati Amazon Athena.
- `JDBCConnectorSource`: un oggetto JDBCConnectorFonte.

  Specifica un connettore per un'origine dati JDBC.
- `SparkConnectorSource`: un oggetto SparkConnectorSource.

  Specifica un connettore per un'origine dati Apache Spark.
- `CatalogSource`: un oggetto CatalogSource.

  Speciifica un data store nel AWS Glue Data Catalog.
- `RedshiftSource`: un oggetto RedshiftSource.

  Specifica un archivio dati Amazon Redshift.
- `S3CatalogSource`: un oggetto S3 CatalogSource.

  Speciifica un data store Amazon S3 nel Data Catalog AWS Glue .
- `S3CsvSource`: un oggetto S3 CsvSource.

Specifica un archivio dati CSV (valori delimitati da comandi) archiviati in Amazon S3.

- `S3JsonSource`: un oggetto S3 JsonSource.

  Specifica un archivio dati JSON in Amazon S3.

- `S3ParquetSource`: un oggetto S3 ParquetSource.

  Specifica un archivio dati di Apache Parquet archiviato in Amazon S3.

- `RelationalCatalogSource`: un oggetto RelationalCatalogSource.

  Speciifica un data store di catalogo relazionale nel Data Catalog. AWS Glue

- `DynamoDBCatalogSource`: un oggetto Sorgente Dynamo DBCatalog.

  Specifica un data store DynamoDBC Catalog nel Data Catalog. AWS Glue

- `JDBCConnectorTarget`: un oggetto JDBCConnectorObiettivo.

  Specifica una destinazioni di dati che scrive su Amazon S3 nell'archiviazione a colonne di Apache Parquet.

- `SparkConnectorTarget`: un oggetto SparkConnectorTarget.

  Specifica una destinazione che utilizza un connettore Apache Spark.

- `CatalogTarget`: un oggetto BasicCatalogTarget.

  Specifica una destinazione che utilizza una AWS Glue tabella Data Catalog.

- `RedshiftTarget`: un oggetto RedshiftTarget.

  Specifica una destinazione che utilizza Amazon Redshift.

- `S3CatalogTarget`: un oggetto S3 CatalogTarget.

  Speciifica un target di dati che scrive su Amazon S3 utilizzando AWS Glue il Data Catalog.

- `S3GlueParquetTarget`: un oggetto S3 GlueParquetTarget.

  Specifica una destinazioni di dati che scrive su Amazon S3 nell'archiviazione a colonne di Apache Parquet.

- `S3DirectTarget`: un oggetto S3 DirectTarget.

  Specifica una destinazione di dati che scrive su Amazon S3.

- `ApplyMapping`: un oggetto ApplyMapping.

Specifica una trasformazione che mappa le chiavi delle proprietà dei dati nell'origine dei dati alle chiavi delle proprietà dei dati nella destinazione. È possibile rinominare le chiavi, modificare i tipi di dati per le chiavi e scegliere le chiavi da eliminare dal set di dati.

- `SelectFields`: un oggetto SelectFields.

  Specifica una trasformazione che sceglie le chiavi della proprietà dati che si desidera conservare.

- `DropFields`: un oggetto DropFields.

  Specifica una trasformazione che sceglie le chiavi della proprietà dati che si desidera eliminare.

- `RenameField`: un oggetto RenameField.

  Specifica una trasformazione che rinominerà una singola chiave di proprietà dati.

- `Spigot`: un oggetto Spigot.

  Specifica una trasformazione che scrive campioni dei dati in un bucket Amazon S3.

- `Join`: un oggetto Join.

  Specifica una trasformazione che unisce due set di dati in un unico set di dati utilizzando una frase di confronto sulle chiavi di proprietà dei dati specificate. È possibile utilizzare inner, outer, left, right, left semi e left anti join.

- `SplitFields`: un oggetto SplitFields.

  Specifica una trasformazione che divide le chiavi della proprietà dati in due `DynamicFrames`. L'output è una raccolta di `DynamicFrames`: uno con le chiavi di proprietà dei dati selezionate e uno con le chiavi di proprietà dei dati rimanenti.

- `SelectFromCollection`: un oggetto SelectFromCollection.

  Specifica una trasformazione che sceglie un `DynamicFrame` da una raccolta di `DynamicFrames`. L'output è il `DynamicFrame` selezionato

- `FillMissingValues`: un oggetto FillMissingValues.

  Specifica una trasformazione che individua i registri nel set di dati che hanno valori mancanti e aggiunge un nuovo campo con un valore determinato dall'imputazione. Il set di dati di input viene utilizzato per addestrare il modello di machine learning che determina quale dovrebbe essere il valore mancante.

- `Filter`: un oggetto Filtro.

Specifica una trasformazione che divide un set di dati in due, in base a una condizione di filtro.

- `CustomCode`: un oggetto CustomCode.

  Specifica una trasformazione che utilizza il codice personalizzato fornito per eseguire la trasformazione dei dati. L'output è una raccolta di. DynamicFrames

- `SparkSQL`: un oggetto SparkSQL.

  Specifica una trasformazione in cui si inserisce una query SQL utilizzando la sintassi Spark SQL per trasformare i dati. L'output è un singolo `DynamicFrame`.

- `DirectKinesisSource`: un oggetto DirectKinesisSource.

  Specifica un'origine dati Amazon Kinesis diretta.

- `DirectKafkaSource`: un oggetto DirectKafkaSource.

  Specifica un archivio dati Apache Kafka.

- `CatalogKinesisSource`: un oggetto CatalogKinesisSource.

  Specifica un'origine dati Kinesis nel Data Catalog AWS Glue .

- `CatalogKafkaSource`: un oggetto CatalogKafkaSource.

  Specifica un archivio dati Apache Kafka nel catalogo dati.

- `DropNullFields`: un oggetto DropNullFields.

  Specifica una trasformazione che rimuove le colonne dal set di dati se tutti i valori nella colonna sono "null". Per impostazione predefinita, AWS Glue Studio riconosce gli oggetti nulli, ma alcuni valori come stringhe vuote, stringhe «nulle», numeri interi -1 o altri segnaposto come gli zeri, non vengono riconosciuti automaticamente come nulli.

- `Merge`: un oggetto Unione.

  Specifica una trasformazione che unisce `DynamicFrame` a con un `DynamicFrame` di staging basato sulle chiavi primarie specificate per identificare i registri. I registri duplicati (registri con le stesse chiavi primarie) non vengono deduplicati.

- `Union`: un oggetto Union.

  Specifica una trasformazione che combina le righe di due o più set di dati in un unico risultato.

- `PIIDetection`: un oggetto PIIDetection.

Specifica una trasformazione che identifica, rimuove o maschera i dati PII.

- `Aggregate`: un oggetto Aggregazione.

  Specifica una trasformazione che raggruppa le righe in base ai campi scelti e calcola il valore aggregato in base alla funzione specificata.

- `DropDuplicates`: un oggetto DropDuplicates.

  Specifica una trasformazione che rimuove le righe di dati ripetuti da un set di dati.

- `GovernedCatalogTarget`: un oggetto GovernedCatalogTarget.

  Specifica una destinazione di dati che scrive su un catalogo governato.

- `GovernedCatalogSource`: un oggetto GovernedCatalogSource.

  Specifica un'origine dei dati in un catalogo dati governato.

- `MicrosoftSQLServerCatalogSource`: un oggetto Microsoft SQLServer CatalogSource.

  Specifica un'origine dei dati di Microsoft SQL Server nel Catalogo dati di AWS Glue .

- `MySQLCatalogSource`: un oggetto La mia SQLCatalog fonte.

  Specifica un'origine dati MySQL nel Data Catalog. AWS Glue

- `OracleSQLCatalogSource`: un oggetto SQLCatalogFonte Oracle.

  Specifica un'origine dati Oracle nel Data Catalog. AWS Glue

- `PostgreSQLCatalogSource`: un oggetto Fonte Postgre SQLCatalog.

  Specifica un'origine dati PostgressQL nel Data Catalog. AWS Glue

- `MicrosoftSQLServerCatalogTarget`: un oggetto Microsoft SQLServer CatalogTarget.

  Specifica una destinazione che utilizza Microsoft SQL.

- `MySQLCatalogTarget`: un oggetto Il mio SQLCatalog obiettivo.

  Specifica una destinazione che utilizza MySQL.

- `OracleSQLCatalogTarget`: un oggetto Oracle SQLCatalog Target.

  Specifica una destinazione che utilizza Oracle SQL.

- `PostgreSQLCatalogTarget`: un oggetto Postgre Target SQLCatalog.

Specifica una destinazione che utilizza Postgres SQL.

- `Route`: un oggetto Route.

  Specifica un nodo di routing che indirizza i dati verso diversi percorsi di output in base a condizioni di filtraggio definite.

- `DynamicTransform`: un oggetto DynamicTransform.

  Specifica una trasformazione visiva personalizzata creata da un utente.

- `EvaluateDataQuality`: un oggetto EvaluateDataQuality.

  Specifica i criteri di valutazione della qualità dei dati.

- `S3CatalogHudiSource`: un oggetto S3 CatalogHudiSource.

  Specifica un'origine dati Hudi registrata nel Data Catalog. AWS Glue L'origine dati deve essere archiviata in. Amazon S3

- `CatalogHudiSource`: un oggetto CatalogHudiSource.

  Specifica una fonte di dati Hudi registrata nel AWS Glue Data Catalog.

- `S3HudiSource`: un oggetto S3 HudiSource.

  Specifica un'origine dati Hudi memorizzata in. Amazon S3

- `S3HudiCatalogTarget`: un oggetto S3 HudiCatalogTarget.

  Specifica una destinazione che scrive su un'origine dati Hudi nel Data Catalog. AWS Glue

- `S3HudiDirectTarget`: un oggetto S3 HudiDirectTarget.

  Specifica una destinazione che scrive su una fonte di dati Hudi in. Amazon S3

- `S3CatalogDeltaSource`: un oggetto S3 CatalogDeltaSource.

  Specifica un'origine dati Delta Lake registrata nel Data Catalog. AWS Glue L'origine dati deve essere archiviata in Amazon S3.

- `CatalogDeltaSource`: un oggetto CatalogDeltaSource.

  Specifica un'origine dati Delta Lake registrata nel AWS Glue Data Catalog.

- `S3DeltaSource`: un oggetto S3 DeltaSource.

  Specifica un'origine dati Delta Lake memorizzata in. Amazon S3

- `S3DeltaCatalogTarget`: un oggetto S3 DeltaCatalogTarget.

  Speciifica una destinazione che scrive su un'origine dati Delta Lake nel AWS Glue Data Catalog.
- `S3DeltaDirectTarget`: un oggetto S3 DeltaDirectTarget.

  Specifica una destinazione che esegue la scrittura su un'origine dati Delta Lake in. Amazon S3
- `AmazonRedshiftSource`: un oggetto AmazonRedshiftSource.

  Specifica una destinazione che scrive su un'origine dati in Amazon Redshift.
- `AmazonRedshiftTarget`: un oggetto AmazonRedshiftTarget.

  Specifica una destinazione che scrive su una destinazione dati in Amazon Redshift.
- `EvaluateDataQualityMultiFrame`: un oggetto EvaluateDataQualityMultiFrame.

  Specifica i criteri di valutazione della qualità dei dati. Consente più dati di input e restituisce una raccolta di frame dinamici.
- `Recipe`: un oggetto Recipe.

  Specifica un nodo di AWS Glue DataBrew ricetta.
- `SnowflakeSource`: un oggetto SnowflakeSource.

  Specifica un'origine dati Snowflake.
- `SnowflakeTarget`: un oggetto SnowflakeTarget.

  Specifica una destinazione che scrive su un'origine dati Snowflake.
- `ConnectorDataSource`: un oggetto ConnectorDataSource.

  Specifica un'origine generata con opzioni di connessione standard.
- `ConnectorDataTarget`: un oggetto ConnectorDataTarget.

  Specifica un a destinazione generata con opzioni di connessione standard.
- `S3CatalogIcebergSource`: un oggetto S3 CatalogIcebergSource.

  Specifica un'origine dati Apache Iceberg registrata nel Data Catalog. AWS Glue L'origine dati Iceberg deve essere archiviata in. Amazon S3
- `CatalogIcebergSource`: un oggetto CatalogIcebergSource.

  Speciifica un'origine dati Apache Iceberg registrata nel Data Catalog. AWS Glue

- `S3IcebergCatalogTarget`: un oggetto S3 IcebergCatalogTarget.

  Specifica una destinazione del catalogo Apache Iceberg che scrive dati Amazon S3 e registra la tabella nel Data Catalog. AWS Glue
- `S3IcebergDirectTarget`: un oggetto S3 IcebergDirectTarget.

  Definisce i parametri di configurazione per la scrittura dei dati in Amazon S3 come tabella Apache Iceberg.
- `S3ExcelSource`: un oggetto S3 ExcelSource.

  Definisce i parametri di configurazione per la lettura dei file Excel da Amazon S3.
- `S3HyperDirectTarget`: un oggetto S3 HyperDirectTarget.

  Definisce i parametri di configurazione per la scrittura di dati su Amazon S3 utilizzando l' HyperDirect ottimizzazione.
- `DynamoDBELTConnectorSource`: un oggetto Sorgente Dynamo DBELTConnector.

  Specifica un connettore ELT DynamoDB di origine per l'estrazione dei dati dalle tabelle DynamoDB.

## JDBCConnectorStruttura delle opzioni

Opzioni di connessione aggiuntive per il connettore.

Campi

- `FilterPredicate`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Clausola condizione extra per filtrare i dati dall'origine. Ad esempio:

  ```
  BillingCity='Mountain View'
  ```

  Quando si utilizza una query anziché un nome di tabella, è necessario verificare che la query funzioni con il `filterPredicate` specificato.
- `PartitionColumn`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome di una colonna intera utilizzata per il partizionamento. Questa opzione funziona solo quando è inclusa con `lowerBound`, `upperBound` e `numPartitions`. Questa opzione funziona allo stesso modo del lettore Spark SQL JDBC.

- `LowerBound`: numero (long), non superiore a Nessuno.

  Il valore minimo di `partitionColumn` che viene utilizzato per decidere lo stride della partizione.

- `UpperBound`: numero (long), non superiore a Nessuno.

  Il valore massimo di `partitionColumn` che viene utilizzato per decidere lo stride della partizione.

- `NumPartitions`: numero (long), non superiore a Nessuno.

  Il numero di partizioni. Questo valore, insieme a `lowerBound` (incluso) e `upperBound` (escluso), forma stride di partizione per espressioni con le clausole `WHERE` generate che vengono utilizzate per dividere la `partitionColumn`.

- `JobBookmarkKeys`: una matrice di stringhe UTF-8.

  Il nome delle chiavi dei segnalibri di processo su cui eseguire l'ordinamento.

- `JobBookmarkKeysSortOrder`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Specifica il criterio di ordinamento crescente o decrescente.

- `DataTypeMapping`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8 (valori validi:`ARRAY` | `BIGINT` | `BINARY` | `BIT` | `BLOB` | `BOOLEAN` | `CHAR` | `CLOB` | `DATALINK` | `DATE` | `DECIMAL` | `DISTINCT` | `DOUBLE` | `FLOAT` | `INTEGER` | `JAVA_OBJECT` | `LONGNVARCHAR` | `LONGVARBINARY` | `LONGVARCHAR` | `NCHAR` | `NCLOB` | `NULL` | `NUMERIC` | `NVARCHAR` | `OTHER` | `REAL` | `REF` | `REF_CURSOR` | `ROWID` | `SMALLINT` | `SQLXML` | `STRUCT` | `TIME` | `TIME_WITH_TIMEZONE` | `TIMESTAMP` | `TIMESTAMP_WITH_TIMEZONE` | `TINYINT` | `VARBINARY` | `VARCHAR`).

  Ogni valore è una stringa UTF-8 (valori validi: `DATE` | `STRING` | `TIMESTAMP` | `INT` | `FLOAT` | `LONG` | `BIGDECIMAL` | `BYTE` | `SHORT` | `DOUBLE`).

  Mappatura del tipo di dati personalizzata che crea una mappatura da un tipo di dati JDBC a un tipo di dati AWS Glue . Ad esempio, l'opzione `"dataTypeMapping":{"FLOAT":"STRING"}` mappa i campi di dati di tipo JDBC `FLOAT` nel `String` tipo Java chiamando il `ResultSet.getString()` metodo del driver e lo utilizza per creare il AWS Glue record. L'oggetto `ResultSet` viene implementato da ciascun driver, quindi il comportamento è specifico del driver utilizzato. Consulta la documentazione relativa al driver JDBC per capire come il driver esegue le conversioni.

## StreamingDataPreviewOptions struttura

Specifica le opzioni relative all'anteprima dei dati per la visualizzazione di un campione dei dati.

Campi

- `PollingTime`: numero (lungo), almeno 10.

  Il tempo di polling in millisecondi.
- `RecordPollingLimit`: numero (lungo), almeno 1.

  Il limite al numero di registri per cui è stato fatto il polling.

## AthenaConnectorSource struttura

Specifica un connettore per un'origine dati Amazon Athena.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome dell'origine dati.
- `ConnectionName` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome della connessione associata al connettore.
- `ConnectorName` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome di un connettore che consente di accedere all'archivio dati in AWS Glue Studio.
- `ConnectionType` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il tipo di connessione, come marketplace.athena o custom.athena, che designa una connessione a un archivio dati Amazon Athena.
- `ConnectionTable`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome della tabella nell'origine dati.
- `SchemaName` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome del gruppo di log CloudWatch da cui leggere. Ad esempio, `/aws-glue/jobs/output`.
- `OutputSchemas`: una matrice di oggetti GlueSchema.

Specifica lo schema di dati per l'origine Athena personalizzata.

## JDBCConnectorStruttura del codice sorgente

Specifica un connettore per un'origine dati JDBC.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome dell'origine dati.

- `ConnectionName` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome della connessione associata al connettore.

- `ConnectorName` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome di un connettore che consente di accedere all'archivio dati in AWS Glue Studio.

- `ConnectionType` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il tipo di connessione, come marketplace.jdbc o custom.jdbc, che designa una connessione a un archivio dati JDBC.

- `AdditionalOptions`: un oggetto JDBCConnectorOpzioni.

  Opzioni di connessione aggiuntive per il connettore.

- `ConnectionTable`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome della tabella nell'origine dati.

- `Query`: stringa UTF-8, corrispondente a Custom string pattern #62.

  La tabella o la query SQL da cui ottenere i dati. Puoi specificare `ConnectionTable` o `query`, ma non entrambi.

- `OutputSchemas`: una matrice di oggetti GlueSchema.

  Specifica lo schema di dati per l'origine JDBC personalizzata.

## SparkConnectorSource struttura

Specifica un connettore per un'origine dati Apache Spark.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome dell'origine dati.
- `ConnectionName` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome della connessione associata al connettore.
- `ConnectorName` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome di un connettore che consente di accedere all'archivio dati in AWS Glue Studio.
- `ConnectionType` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il tipo di connessione, come marketplace.spark o custom.spark, che designa una connessione a un archivio dati di Apache Spark.
- `AdditionalOptions`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8 corrispondente al Custom string pattern #61.

  Ogni valore è una stringa UTF-8 corrispondente al Custom string pattern #61.

  Opzioni di connessione aggiuntive per il connettore.
- `OutputSchemas`: una matrice di oggetti GlueSchema.

  Specifica lo schema di dati per l'origine Spark personalizzata.

## CatalogSource struttura

Speciifica un data store nel AWS Glue Data Catalog.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome del archivio dati.
- `Database` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

Il nome del database da cui leggere.

- `Table` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome della tabella nel database da cui leggere.

- `PartitionPredicate`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Le partizioni che soddisfano questo predicato vengono eliminate. I file all'interno del periodo di conservazione in queste partizioni non vengono eliminati.

- `OutputSchemas`: una matrice di oggetti GlueSchema.

  Specifica lo schema di dati per l'origine del catalogo.

## Struttura My SQLCatalog Source

Specifica un'origine dati MySQL nel Data Catalog. AWS Glue

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome dell'origine dati.

- `Database` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome del database da cui leggere.

- `Table` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome della tabella nel database da cui leggere.

## Struttura di Postgree Source SQLCatalog

Specifica un'origine dati PostgressQL nel Data Catalog. AWS Glue

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome dell'origine dati.

- `Database` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

Il nome del database da cui leggere.

- `Table` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome della tabella nel database da cui leggere.

## Struttura SQLCatalog Oracle Source

Speciifica un'origine dati Oracle nel AWS Glue Data Catalog.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome dell'origine dati.

- `Database` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome del database da cui leggere.

- `Table` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome della tabella nel database da cui leggere.

## SQLServerCatalogSource Struttura Microsoft

Specifica un'origine dei dati di Microsoft SQL Server nel Catalogo dati di AWS Glue .

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome dell'origine dati.

- `Database` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome del database da cui leggere.

- `Table` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome della tabella nel database da cui leggere.

## CatalogKinesisSource struttura

Specifica un'origine dati Kinesis nel Data Catalog AWS Glue .

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome dell'origine dati.
- `WindowSize`: numero (intero), non superiore a Nessuno.

  La quantità di tempo da dedicare all'elaborazione di ciascun micro batch.
- `DetectSchema`: booleano.

  Se determinare automaticamente o meno lo schema dai dati in entrata.
- `Table` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome della tabella nel database da cui leggere.
- `Database` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome del database da cui leggere.
- `StreamingOptions`: un oggetto KinesisStreamingSourceOptions.

  Opzioni aggiuntive per l'origine dati di streaming Kinesis.
- `DataPreviewOptions`: un oggetto StreamingDataPreviewOptions.

  Opzioni aggiuntive per l'anteprima dei dati.

## DirectKinesisSource struttura

Specifica un'origine dati Amazon Kinesis diretta.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome dell'origine dati.
- `WindowSize`: numero (intero), non superiore a Nessuno.

  La quantità di tempo da dedicare all'elaborazione di ciascun micro batch.

- `DetectSchema`: booleano.

  Se determinare automaticamente o meno lo schema dai dati in entrata.
- `StreamingOptions`: un oggetto KinesisStreamingSourceOptions.

  Opzioni aggiuntive per l'origine dati di streaming Kinesis.
- `DataPreviewOptions`: un oggetto StreamingDataPreviewOptions.

  Opzioni aggiuntive per l'anteprima dei dati.

## KinesisStreamingSourceOptions struttura

Opzioni aggiuntive per l'origine dati di streaming Amazon Kinesis.

Campi

- `EndpointUrl`: stringa UTF-8, corrispondente a Custom string pattern #61.

  L'URL dell'endpoint di Kinesis.
- `StreamName`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome del flusso di dati Kinesis.
- `Classification`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Una classificazione facoltativa.
- `Delimiter`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Specifica il carattere delimitatore.
- `StartingPosition`: stringa UTF-8 (valori validi: `latest="LATEST"` | `trim_horizon="TRIM_HORIZON"` | `earliest="EARLIEST"` | `timestamp="TIMESTAMP"`).

  La posizione di partenza nel flusso dei dati Kinesis da cui leggere i dati. I valori possibili sono `"latest"`, `"trim_horizon"`, `"earliest"` o una stringa di timestamp in formato UTC con il modello `yyyy-mm-ddTHH:MM:SSZ` (dove `Z` rappresenta uno scostamento del fuso orario UTC con un segno +/-. Ad esempio: “2023-04-04T08:00:00-04:00”). Il valore predefinito è `"latest"`.

  Nota: l'utilizzo di un valore che è una stringa di timestamp in formato UTC per «startingPosition» è supportato solo per AWS Glue la versione 4.0 o successiva.
- `MaxFetchTimeInMs`: numero (long), non superiore a Nessuno.

Il tempo massimo impiegato affinché l'esecutore del processo legga i record della batch attuale dal flusso di dati Kinesis, specificato in millisecondi (ms). Entro questo periodo è possibile effettuate più chiamate API `GetRecords`. Il valore di default è `1000`.

- `MaxFetchRecordsPerShard`: numero (long), non superiore a Nessuno.

  Il numero massimo di record da recuperare per shard nel flusso di dati Kinesis per microbatch. Nota: il client può superare questo limite se il processo di streaming ha già letto i record aggiuntivi da Kinesis (nella stessa chiamata get-records). Se `MaxFetchRecordsPerShard` deve essere rigoroso, allora deve essere un multiplo di `MaxRecordPerRead`. Il valore di default è `100000`.
- `MaxRecordPerRead`: numero (long), non superiore a Nessuno.

  Il numero massimo di registri da recuperare nel flusso dei dati Kinesis in ciascuna operazione getRecords. Il valore predefinito è `10000`.
- `AddIdleTimeBetweenReads`: booleano.

  Aggiunge un ritardo tra due operazioni consecutive getRecords. Il valore predefinito è `"False"`. Questa opzione è configurabile solo per AWS Glue versione 2.0 e successive.
- `IdleTimeBetweenReadsInMs`: numero (long), non superiore a Nessuno.

  Il ritardo minimo tra due operazioni consecutive getRecords, specificato in ms. Il valore predefinito è `1000`. Questa opzione è configurabile solo per la versione 2.0 e successive. AWS Glue
- `DescribeShardInterval`: numero (long), non superiore a Nessuno.

  L'intervallo di tempo minimo tra due chiamate ListShards API entro il quale lo script deve prendere in considerazione il resharding. Il valore predefinito è `1s`.
- `NumRetries`: numero (intero), non superiore a Nessuno.

  Il numero massimo di tentativi per le richieste API Kinesis Data Streams. Il valore di default è `3`.
- `RetryIntervalMs`: numero (long), non superiore a Nessuno.

  Il periodo di raffreddamento (specificato in ms) prima di riprovare la chiamata API Kinesis Data Streams. Il valore di default è `1000`.
- `MaxRetryIntervalMs`: numero (long), non superiore a Nessuno.

  Il periodo di raffreddamento (specificato in ms) tra due tentativi di chiamata API Kinesis Data Streams. Il valore predefinito è `10000`.
- `AvoidEmptyBatches`: booleano.

Impedisce la creazione di un processo microbatch vuoto controllando la presenza di dati non letti nel flusso dei dati Kinesis prima che il batch venga avviato. Il valore predefinito è `"False"`.

- `StreamArn`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome della risorsa Amazon (ARN) del flusso di dati Kinesis.

- `RoleArn`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome della risorsa Amazon (ARN) del ruolo da assumere tramite il Servizio di token di sicurezza AWS (AWS STS). Questo ruolo deve disporre delle autorizzazioni per descrivere o leggere le operazioni dei registri per il flusso di dati Kinesis. Quando si accede a un flusso di dati in un altro account, è necessario utilizzare questo parametro. Usato in combinazione con `"awsSTSSessionName"`.

- `RoleSessionName`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Un identificatore della sessione che assume il ruolo tramite AWS STS. Quando si accede a un flusso di dati in un altro account, è necessario utilizzare questo parametro. Usato in combinazione con `"awsSTSRoleARN"`.

- `AddRecordTimestamp`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Quando questa opzione è impostata su "true", l'output dei dati conterrà una colonna aggiuntiva denominata "__src_timestamp" che indica l'ora in cui il record corrispondente è stato ricevuto dal flusso. Il valore predefinito è "false". Questa opzione è supportata nella AWS Glue versione 4.0 o successiva.

- `EmitConsumerLagMetrics`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Quando questa opzione è impostata su «true», per ogni batch emette le metriche relative alla durata compresa tra il record più vecchio ricevuto dallo stream e il momento in AWS Glue cui arriva. CloudWatch Il nome della metrica è «glue.driver.streaming. maxConsumerLagInMs». Il valore predefinito è "false". Questa opzione è supportata in AWS Glue versione 4.0 o successive.

- `StartingTimestamp`: stringa UTF-8.

  Il timestamp del record nel flusso di dati Kinesis da cui iniziare la lettura dei dati. I valori possibili sono una stringa di timestamp in formato UTC del modello `yyyy-mm-ddTHH:MM:SSZ`, dove Z rappresenta uno scostamento del fuso orario UTC con un segno +/- (ad esempio: "2023-04-04T 08:00:00+08:00").

- `FanoutConsumerARN`: stringa UTF-8, corrispondente a Custom string pattern #61.

Il nome della risorsa Amazon (ARN) del consumatore con fan-out avanzato del flusso di dati Kinesis. Quando è specificato, abilita il fan-out avanzato per un throughput dedicato e un consumo di dati a latenza inferiore.

## CatalogKafkaSource struttura

Specifica un archivio dati Apache Kafka nel catalogo dati.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome del archivio dati.

- `WindowSize`: numero (intero), non superiore a Nessuno.

  La quantità di tempo da dedicare all'elaborazione di ciascun micro batch.

- `DetectSchema`: booleano.

  Se determinare automaticamente o meno lo schema dai dati in entrata.

- `Table` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome della tabella nel database da cui leggere.

- `Database` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome del database da cui leggere.

- `StreamingOptions`: un oggetto KafkaStreamingSourceOptions.

  Specifica le opzioni di streaming.

- `DataPreviewOptions`: un oggetto StreamingDataPreviewOptions.

  Specifica le opzioni relative all'anteprima dei dati per la visualizzazione di un campione dei dati.

## DirectKafkaSource struttura

Specifica un archivio dati Apache Kafka.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome del archivio dati.
- `StreamingOptions`: un oggetto KafkaStreamingSourceOptions.

  Specifica le opzioni di streaming.
- `WindowSize`: numero (intero), non superiore a Nessuno.

  La quantità di tempo da dedicare all'elaborazione di ciascun micro batch.
- `DetectSchema`: booleano.

  Se determinare automaticamente o meno lo schema dai dati in entrata.
- `DataPreviewOptions`: un oggetto StreamingDataPreviewOptions.

  Specifica le opzioni relative all'anteprima dei dati per la visualizzazione di un campione dei dati.

## KafkaStreamingSourceOptions struttura

Opzioni aggiuntive per lo streaming.

Campi

- `BootstrapServers`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Un elenco di server di bootstrap URLs, ad esempio, come`b-1.vpc-test-2.o4q88o.c6.kafka.us-east-1.amazonaws.com:9094`. Questa opzione deve essere specificata nella chiamata API o definita nei metadati della tabella in catalogo dati.
- `SecurityProtocol`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il protocollo utilizzato per comunicare con i broker. I valori possibili sono `"SSL"` o `"PLAINTEXT"`.
- `ConnectionName`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome della connessione.
- `TopicName`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome dell'argomento come specificato in Apache Kafka. Devi specificare almeno uno tra `"topicName"`, `"assign"` o `"subscribePattern"`.

- `Assign`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Lo specifico `TopicPartitions` per consumare. Devi specificare almeno uno tra `"topicName"`, `"assign"` o `"subscribePattern"`.

- `SubscribePattern`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Uuna stringa regex Java che identifichi l'elenco degli argomenti a cui effettuare la sottoscrizione. Devi specificare almeno uno tra `"topicName"`, `"assign"` o `"subscribePattern"`.

- `Classification`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Una classificazione facoltativa.

- `Delimiter`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Specifica il carattere delimitatore.

- `StartingOffsets`: stringa UTF-8, corrispondente a Custom string pattern #61.

  La posizione di partenza nell'argomento Kafka da cui leggere i dati. I valori possibili sono `"earliest"` o `"latest"`. Il valore predefinito è `"latest"`.

- `EndingOffsets`: stringa UTF-8, corrispondente a Custom string pattern #61.

  L'endpoint al quale viene terminata una query batch. I valori possibili sono `"latest"` o una stringa JSON che specifica un offset finale per ogni `TopicPartition`.

- `PollTimeoutMs`: numero (long), non superiore a Nessuno.

  Il timeout in millisecondi per il polling dei dati da Kafka negli esecutori del processo Spark. Il valore predefinito è `512`.

- `NumRetries`: numero (intero), non superiore a Nessuno.

  Il numero di tentativi prima di non riuscire a recuperare gli offset Kafka. Il valore di default è `3`.

- `RetryIntervalMs`: numero (long), non superiore a Nessuno.

  Il tempo di attesa in millisecondi prima di riprovare a recuperare gli offset Kafka. Il valore di default è `10`.

- `MaxOffsetsPerTrigger`: numero (long), non superiore a Nessuno.

  Il limite di velocità sul numero massimo di offset elaborati per intervallo di attivazione. Il numero totale di offset specificato viene suddiviso proporzionalmente tra `topicPartitions` di diversi

volumi. Il valore di default è null, il che significa che il consumer legge tutti gli offset fino all'ultimo offset noto.

- `MinPartitions`: numero (intero), non superiore a Nessuno.

  Il numero minimo desiderato di partizioni da leggere da Kafka. Il valore di default è null, il che significa che il numero di partizioni Spark è uguale al numero di partizioni Kafka.

- `IncludeHeaders`: booleano.

  Se includere le intestazioni di Kafka. Quando l'opzione è impostata su "true", l'output dei dati conterrà una colonna aggiuntiva denominata "glue_streaming_kafka_headers" con tipo `Array[Struct(key: String, value: String)]`. Il valore di default è "false". Questa opzione è disponibile solo nella AWS Glue versione 3.0 o successiva.

- `AddRecordTimestamp`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Quando questa opzione è impostata su "true", l'output dei dati conterrà una colonna aggiuntiva denominata "__src_timestamp" che indica l'ora in cui il record corrispondente è stato ricevuto dall'argomento. Il valore predefinito è "false". Questa opzione è supportata nella AWS Glue versione 4.0 o successiva.

- `EmitConsumerLagMetrics`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Quando questa opzione è impostata su «true», per ogni batch emette le metriche relative alla durata compresa tra il record più vecchio ricevuto dall'argomento e il momento in AWS Glue cui arriva. CloudWatch Il nome della metrica è «glue.driver.streaming. maxConsumerLagInMs». Il valore predefinito è "false". Questa opzione è supportata in AWS Glue versione 4.0 o successive.

- `StartingTimestamp`: stringa UTF-8.

  Il timestamp del record nell'argomento Kinesis da cui iniziare la lettura dei dati. I valori possibili sono una stringa di timestamp in formato UTC del modello `yyyy-mm-ddTHH:MM:SSZ`, dove Z rappresenta uno scostamento del fuso orario UTC con un segno +/- (ad esempio: "2023-04-04T 08:00:00+08:00").

  Deve essere impostato solo un valore tra `StartingTimestamp` e `StartingOffsets`.

## RedshiftSource struttura

Specifica un archivio dati Amazon Redshift.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome dell'archivio dati Amazon Redshift.
- `Database` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il database da cui leggere.
- `Table` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  La tabella del database da cui leggere.
- `RedshiftTmpDir`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il percorso Amazon S3 in cui i dati temporanei possono essere caricati durante la copia dal database.
- `TmpDirIAMRole`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il ruolo IAM con autorizzazioni.

## AmazonRedshiftSource struttura

Il nome della connessione è l'origine Amazon Redshift.

Campi

- `Name`: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome dell'origine Amazon Redshift.
- `Data`: un oggetto AmazonRedshiftNodeData.

  Specifica i dati del nodo di origine Amazon Redshift.

## AmazonRedshiftNodeData struttura

Specifica un nodo Amazon Redshift.

Campi

- `AccessType`: stringa UTF-8, corrispondente a Custom string pattern #60.

Il tipo di accesso per la connessione Redshift. Può essere una connessione diretta o una al catalogo.

- `SourceType`: stringa UTF-8, corrispondente a Custom string pattern #60.

Il tipo di origine per specificare se una tabella specifica è l'origine o una query personalizzata.

- `Connection`: un oggetto Opzione.

La AWS Glue connessione al cluster Redshift.

- `Schema`: un oggetto Opzione.

Il nome dello schema Redshift quando si lavora con una connessione diretta.

- `Table`: un oggetto Opzione.

Il nome della tabella Redshift quando si lavora con una connessione diretta.

- `CatalogDatabase`: un oggetto Opzione.

Il nome del database AWS Glue Data Catalog quando si lavora con un catalogo di dati.

- `CatalogTable`: un oggetto Opzione.

Il nome della tabella AWS Glue Data Catalog quando si lavora con un catalogo di dati.

- `CatalogRedshiftSchema`: stringa UTF-8.

Il nome dello schema Redshift quando si lavora con un catalogo dati.

- `CatalogRedshiftTable`: stringa UTF-8.

La tabella del database da cui leggere.

- `TempDir`: stringa UTF-8, corrispondente a Custom string pattern #61.

Il percorso Amazon S3 in cui i dati temporanei possono essere caricati durante la copia dal database.

- `IamRole`: un oggetto Opzione.

Opzionale. Il nome del ruolo utilizzato durante la connessione a S3. Se lasciato vuoto, il ruolo IAM assumerà per impostazione predefinita il ruolo nel processo.

- `AdvancedOptions`: una matrice di oggetti AmazonRedshiftAdvancedOption.

Valori facoltativi durante la connessione al cluster Redshift.

- `SampleQuery`: stringa UTF-8.

  L'SQL utilizzato per recuperare i dati da una fonte Redshift quando SourceType è 'query'.

- `PreAction`: stringa UTF-8.

  L'SQL utilizzato prima di un'esecuzione di MERGE o APPEND con upsert.

- `PostAction`: stringa UTF-8.

  L'SQL utilizzato prima di un'esecuzione di MERGE o APPEND con upsert.

- `Action`: stringa UTF-8.

  Specifica come verrà eseguita la scrittura su un cluster Redshift.

- `TablePrefix`: stringa UTF-8, corrispondente a Custom string pattern #60.

  Specifica il prefisso di una tabella.

- `Upsert`: booleano.

  L'operazione utilizzata in un sink Redshift quando si esegue un APPEND.

- `MergeAction`: stringa UTF-8, corrispondente a Custom string pattern #60.

  L'operazione utilizzata per determinare come verrà gestito un MERGE in un sink Redshift.

- `MergeWhenMatched`: stringa UTF-8, corrispondente a Custom string pattern #60.

  L'operazione utilizzata per determinare come verrà gestito un MERGE in un sink Redshift quando un record esistente corrisponde a un nuovo record.

- `MergeWhenNotMatched`: stringa UTF-8, corrispondente a Custom string pattern #60.

  L'operazione utilizzata per determinare come verrà gestito un MERGE in un sink Redshift quando un record esistente non corrisponde a un nuovo record.

- `MergeClause`: stringa UTF-8.

  L'SQL utilizzato in un merge personalizzato per gestire i record corrispondenti.

- `CrawlerConnection`: stringa UTF-8.

  Specifica il nome della connessione associata alla tabella del catalogo utilizzata.

- `TableSchema`: una matrice di oggetti Opzione.

  L'array di output dello schema per un determinato nodo.

- `StagingTable`: stringa UTF-8.

  Il nome della tabella intermedia temporanea utilizzata quando si esegue un MERGE o un APPEND con upsert.
- `SelectedColumns`: una matrice di oggetti Opzione.

  L'elenco dei nomi di colonna utilizzati per determinare un record corrispondente quando si esegue un MERGE o un APPEND con upsert.

## AmazonRedshiftAdvancedOption struttura

Specifica un valore facoltativo per la connessione al cluster Redshift.

Campi

- `Key`: stringa UTF-8.

  La chiave dell'opzione di connessione aggiuntiva.
- `Value`: stringa UTF-8.

  Il valore dell'opzione di connessione aggiuntiva.

## Struttura Option

Specifica il valore di un'opzione.

Campi

- `Value`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Specifica il valore dell'opzione.
- `Label`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Specifica l'etichetta dell'opzione.
- `Description`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Specifica la descrizione dell'opzione.

## struttura S3 CatalogSource

Speciifica un data store Amazon S3 nel Data Catalog AWS Glue .

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome del archivio dati.

- `Database` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il database da cui leggere.

- `Table` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  La tabella del database da cui leggere.

- `PartitionPredicate`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Le partizioni che soddisfano questo predicato vengono eliminate. I file all'interno del periodo di conservazione in queste partizioni non vengono eliminati. Impostato su `""`: vuoto per impostazione predefinita.

- `AdditionalOptions`: un oggetto S3 SourceAdditionalOptions.

  Specifica opzioni di connessione aggiuntive.

## Struttura S3 SourceAdditionalOptions

Specifica opzioni di connessione aggiuntive per l'archivio dati Amazon S3.

Campi

- `BoundedSize`: numero (lungo).

  Imposta il limite superiore per la dimensione di destinazione del set di dati in byte che verranno elaborati.

- `BoundedFiles`: numero (lungo).

  Imposta il limite superiore per il numero di file di destinazione che verranno elaborati.

# Struttura S3 CsvSource

Specifica un archivio dati CSV (valori delimitati da comandi) archiviati in Amazon S3.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome del archivio dati.

- `Paths` - obbligatorio: una matrice di stringhe UTF-8.

  Un elenco dei percorsi Amazon S3 da cui leggere.

- `CompressionType`: stringa UTF-8 (valori validi: `gzip="GZIP"` | `bzip2="BZIP2"`).

  Specifica il modo in cui i dati sono compressi. In genere questo non è necessario se i dati hanno un'estensione del file standard. I valori possibili sono `"gzip"` e `"bzip"`).

- `Exclusions`: una matrice di stringhe UTF-8.

  Una stringa contenente un elenco di JSON di modelli glob in stile Unix da escludere. Ad esempio "[\"**.pdf \"]" esclude tutti i file PDF.

- `GroupSize`: stringa UTF-8, corrispondente a Custom string pattern #61.

  La dimensione del gruppo di destinazione in byte. Il valore di default viene calcolato in base alla dimensione dei dati di input e alle dimensioni del cluster. Quando sono presenti meno di 50.000 file di input, `"groupFiles"` deve essere impostato su `"inPartition"` per rendere effettiva la modifica.

- `GroupFiles`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Quando l'input contiene più di 50.000 file, il raggruppamento di file è attivato per impostazione predefinita. Per attivare il raggruppamento con meno di 50.000 file, imposta questo parametro su “inPartition”. Per disabilitare il raggruppamento in presenza di più di 50.000 file, imposta il parametro su `"none"`.

- `Recurse`: booleano.

  Se è impostato su “Vero”, legge i file in modo ricorsivo in tutte le sottodirectory dei percorsi specificati.

- `MaxBand`: numero (intero), non superiore a Nessuno.

Questa opzione controlla la durata in millisecondi dopo la quale è probabile che l'elenco s3 sia coerente. I file con timestamp di modifica che rientrano negli ultimi millisecondi MaxBand vengono tracciati appositamente quando vengono utilizzati per tenere conto JobBookmarks della coerenza finale di Amazon S3. Per la maggior parte degli utenti non è necessario impostare questa opzione. Il valore di default è 900.000 millisecondi o 15 minuti.

- `MaxFilesInBand`: numero (intero), non superiore a Nessuno.

  Questa opzione specifica il numero massimo di file da salvare negli ultimi secondi maxBand. Se si supera questo valore, i file aggiuntivi vengono saltati e solo elaborati nella successiva esecuzione del processo.

- `AdditionalOptions`: un oggetto S3 DirectSourceAdditionalOptions.

  Specifica opzioni di connessione aggiuntive.

- `Separator` - obbligatorio: stringa UTF-8 (valori validi: `comma="COMMA"` | `ctrla="CTRLA"` | `pipe="PIPE"` | `semicolon="SEMICOLON"` | `tab="TAB"`).

  Specifica il carattere delimitatore. Il valore di default è una virgola: ",", ma è possibile specificare qualsiasi altro carattere.

- `Escaper`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Specifica un carattere di escape. Questa opzione viene utilizzata solo durante la lettura di file CSV. Il valore predefinito è `none`. Se questa opzione è abilitata, il carattere immediatamente seguente viene usato così come è, ad eccezione di un piccolo set di caratteri di escape ben noti (`\n`, `\r`, `\t` e `\0`).

- `QuoteChar` - obbligatorio: stringa UTF-8 (valori validi: `quote="QUOTE"` | `quillemet="QUILLEMET"` | `single_quote="SINGLE_QUOTE"` | `disabled="DISABLED"`).

  Specifica il carattere da usare per le virgolette. Per impostazione predefinita vengono usate le virgolette doppie: `'"'`. Imposta questo valore su `-1` per disattivare completamente le virgolette.

- `Multiline`: booleano.

  Un valore booleano che specifica se un singolo registro può estendersi su più righe. Ciò può accadere quando un campo contiene un carattere di nuova riga tra virgolette. Imposta questa opzione su "Vero" se un qualsiasi registro si estende su più righe. Il valore di default è `False`, che consente una divisione dei file più netta durante l'analisi.

- `WithHeader`: booleano.

Un valore booleano che specifica se trattare la prima riga come intestazione. Il valore predefinito è `False`.

- `WriteHeader`: booleano.

  Un valore booleano che specifica se scrivere l'intestazione nell'output. Il valore predefinito è `True`.

- `SkipFirst`: booleano.

  Un valore booleano che specifica se ignorare la prima riga di dati. Il valore predefinito è `False`.

- `OptimizePerformance`: booleano.

  Un valore booleano che specifica se utilizzare il lettore SIMD CSV avanzato insieme ai formati di memoria colonnare basati su Apache Arrow. AWS Glue Disponibile solo nella versione 3.0.

- `OutputSchemas`: una matrice di oggetti GlueSchema.

  Specifica lo schema di dati per l'origine CSV S3 personalizzata.

## JDBCSource Struttura diretta

Specifica la connessione diretta all'origine JDBC.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome della connessione di origine JDBC.

- `Database` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il database della connessione di origine JDBC.

- `Table` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  La tabella della connessione di origine JDBC.

- `ConnectionName` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome della connessione dell'origine JDBC.

- `ConnectionType` - obbligatorio: stringa UTF-8 (valori validi: `sqlserver` | `mysql` | `oracle` | `postgresql` | `redshift`).

  Il tipo di connessione dell'origine JDBC.

- `RedshiftTmpDir`: stringa UTF-8, corrispondente a Custom string pattern #61.

  La directory temporanea dell'origine JDBC Redshift.
- `OutputSchemas`: una matrice di oggetti GlueSchema.

  Specifica lo schema di dati per l'origine JDBC diretta.

## Struttura S3 DirectSourceAdditionalOptions

Specifica opzioni di connessione aggiuntive per l'archivio dati Amazon S3.

Campi

- `BoundedSize`: numero (lungo).

  Imposta il limite superiore per la dimensione di destinazione del set di dati in byte che verranno elaborati.
- `BoundedFiles`: numero (lungo).

  Imposta il limite superiore per il numero di file di destinazione che verranno elaborati.
- `EnableSamplePath`: booleano.

  Imposta l'opzione per abilitare un percorso di esempio.
- `SamplePath`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Se abilitato, specifica il percorso di esempio.

## Struttura S3 JsonSource

Specifica un archivio dati JSON in Amazon S3.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome del archivio dati.
- `Paths` - obbligatorio: una matrice di stringhe UTF-8.

  Un elenco dei percorsi Amazon S3 da cui leggere.

- `CompressionType`: stringa UTF-8 (valori validi: `gzip="GZIP"` | `bzip2="BZIP2"`).

  Specifica il modo in cui i dati sono compressi. In genere questo non è necessario se i dati hanno un'estensione del file standard. I valori possibili sono `"gzip"` e `"bzip"`).

- `Exclusions`: una matrice di stringhe UTF-8.

  Una stringa contenente un elenco di JSON di modelli glob in stile Unix da escludere. Ad esempio "[\"**.pdf \"]" esclude tutti i file PDF.

- `GroupSize`: stringa UTF-8, corrispondente a Custom string pattern #61.

  La dimensione del gruppo di destinazione in byte. Il valore di default viene calcolato in base alla dimensione dei dati di input e alle dimensioni del cluster. Quando sono presenti meno di 50.000 file di input, `"groupFiles"` deve essere impostato su `"inPartition"` per rendere effettiva la modifica.

- `GroupFiles`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Quando l'input contiene più di 50.000 file, il raggruppamento di file è attivato per impostazione predefinita. Per attivare il raggruppamento con meno di 50.000 file, imposta questo parametro su “inPartition”. Per disabilitare il raggruppamento in presenza di più di 50.000 file, imposta il parametro su `"none"`.

- `Recurse`: booleano.

  Se è impostato su “Vero”, legge i file in modo ricorsivo in tutte le sottodirectory dei percorsi specificati.

- `MaxBand`: numero (intero), non superiore a Nessuno.

  Questa opzione controlla la durata in millisecondi dopo la quale è probabile che l'elenco s3 sia coerente. I file con timestamp di modifica che rientrano negli ultimi millisecondi MaxBand vengono tracciati appositamente quando vengono utilizzati per tenere conto JobBookmarks della coerenza finale di Amazon S3. Per la maggior parte degli utenti non è necessario impostare questa opzione. Il valore di default è 900.000 millisecondi o 15 minuti.

- `MaxFilesInBand`: numero (intero), non superiore a Nessuno.

  Questa opzione specifica il numero massimo di file da salvare negli ultimi secondi maxBand. Se si supera questo valore, i file aggiuntivi vengono saltati e solo elaborati nella successiva esecuzione del processo.

- `AdditionalOptions`: un oggetto S3 DirectSourceAdditionalOptions.

Specifica opzioni di connessione aggiuntive.

- `JsonPath`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Una stringa che definisce i dati JSON. JsonPath

- `Multiline`: booleano.

  Un valore booleano che specifica se un singolo registro può estendersi su più righe. Ciò può accadere quando un campo contiene un carattere di nuova riga tra virgolette. Imposta questa opzione su "Vero" se un qualsiasi registro si estende su più righe. Il valore di default è `False`, che consente una divisione dei file più netta durante l'analisi.

- `OutputSchemas`: una matrice di oggetti GlueSchema.

  Specifica lo schema di dati per l'origine JSON S3 personalizzata.

## Struttura S3 ParquetSource

Specifica un archivio dati di Apache Parquet archiviato in Amazon S3.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome del archivio dati.

- `Paths` - obbligatorio: una matrice di stringhe UTF-8.

  Un elenco dei percorsi Amazon S3 da cui leggere.

- `CompressionType`: stringa UTF-8 (valori validi: `snappy="SNAPPY"` | `lzo="LZO"` | `gzip="GZIP"` | `brotli="BROTLI"` | `lz4="LZ4"` | `uncompressed="UNCOMPRESSED"` | `none="NONE"`).

  Specifica il modo in cui i dati sono compressi. In genere questo non è necessario se i dati hanno un'estensione del file standard. I valori possibili sono `"gzip"` e `"bzip"`).

- `Exclusions`: una matrice di stringhe UTF-8.

  Una stringa contenente un elenco di JSON di modelli glob in stile Unix da escludere. Ad esempio "[\"**.pdf \"]" esclude tutti i file PDF.

- `GroupSize`: stringa UTF-8, corrispondente a Custom string pattern #61.

La dimensione del gruppo di destinazione in byte. Il valore di default viene calcolato in base alla dimensione dei dati di input e alle dimensioni del cluster. Quando sono presenti meno di 50.000 file di input, `"groupFiles"` deve essere impostato su `"inPartition"` per rendere effettiva la modifica.

- `GroupFiles`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Quando l'input contiene più di 50.000 file, il raggruppamento di file è attivato per impostazione predefinita. Per attivare il raggruppamento con meno di 50.000 file, imposta questo parametro su “inPartition”. Per disabilitare il raggruppamento in presenza di più di 50.000 file, imposta il parametro su `"none"`.

- `Recurse`: booleano.

  Se è impostato su “Vero”, legge i file in modo ricorsivo in tutte le sottodirectory dei percorsi specificati.

- `MaxBand`: numero (intero), non superiore a Nessuno.

  Questa opzione controlla la durata in millisecondi dopo la quale è probabile che l'elenco s3 sia coerente. I file con timestamp di modifica che rientrano negli ultimi millisecondi MaxBand vengono tracciati appositamente quando vengono utilizzati per tenere conto JobBookmarks della coerenza finale di Amazon S3. Per la maggior parte degli utenti non è necessario impostare questa opzione. Il valore di default è 900.000 millisecondi o 15 minuti.

- `MaxFilesInBand`: numero (intero), non superiore a Nessuno.

  Questa opzione specifica il numero massimo di file da salvare negli ultimi secondi maxBand. Se si supera questo valore, i file aggiuntivi vengono saltati e solo elaborati nella successiva esecuzione del processo.

- `AdditionalOptions`: un oggetto S3 DirectSourceAdditionalOptions.

  Specifica opzioni di connessione aggiuntive.

- `OutputSchemas`: una matrice di oggetti GlueSchema.

  Specifica lo schema di dati per l'origine Parquet S3 personalizzata.

## Struttura S3 DeltaSource

Specifica un'origine dati Delta Lake memorizzata in. Amazon S3

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome dell'origine del Delta Lake.
- `Paths` - obbligatorio: una matrice di stringhe UTF-8.

  Un elenco dei percorsi Amazon S3 da cui leggere.
- `AdditionalDeltaOptions`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8 corrispondente al Custom string pattern #61.

  Ogni valore è una stringa UTF-8 corrispondente al Custom string pattern #61.

  Specifica opzioni di connessione aggiuntive.
- `AdditionalOptions`: un oggetto S3 DirectSourceAdditionalOptions.

  Specifica opzioni aggiuntive per il connettore.
- `OutputSchemas`: una matrice di oggetti GlueSchema.

  Specifica lo schema di dati per l'origine Delta Lake.

## Struttura S3 CatalogDeltaSource

Speciifica un'origine dati Delta Lake registrata nel AWS Glue Data Catalog. L'origine dati deve essere archiviata in Amazon S3.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome dell'origine dati Delta Lake.
- `Database` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome del database da cui leggere.
- `Table` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome della tabella nel database da cui leggere.
- `AdditionalDeltaOptions`: una matrice della mappa di coppie chiave-valore.

Ogni chiave è una stringa UTF-8 corrispondente al Custom string pattern #61.

Ogni valore è una stringa UTF-8 corrispondente al Custom string pattern #61.

Specifica opzioni di connessione aggiuntive.

- `OutputSchemas`: una matrice di oggetti GlueSchema.

  Specifica lo schema di dati per l'origine Delta Lake.

## CatalogDeltaSource struttura

Specifica un'origine dati Delta Lake registrata nel AWS Glue Data Catalog.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome dell'origine dati Delta Lake.
- `Database` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome del database da cui leggere.
- `Table` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome della tabella nel database da cui leggere.
- `AdditionalDeltaOptions`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8 corrispondente al Custom string pattern #61.

  Ogni valore è una stringa UTF-8 corrispondente al Custom string pattern #61.

  Specifica opzioni di connessione aggiuntive.
- `OutputSchemas`: una matrice di oggetti GlueSchema.

  Specifica lo schema di dati per l'origine Delta Lake.

## Struttura S3 HudiSource

Specifica una fonte di dati Hudi memorizzata in. Amazon S3

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome dell'origine Hudi.
- `Paths` - obbligatorio: una matrice di stringhe UTF-8.

  Un elenco dei percorsi Amazon S3 da cui leggere.
- `AdditionalHudiOptions`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8 corrispondente al Custom string pattern #61.

  Ogni valore è una stringa UTF-8 corrispondente al Custom string pattern #61.

  Specifica opzioni di connessione aggiuntive.
- `AdditionalOptions`: un oggetto S3 DirectSourceAdditionalOptions.

  Specifica opzioni aggiuntive per il connettore.
- `OutputSchemas`: una matrice di oggetti GlueSchema.

  Specifica lo schema di dati per l'origine Hudi.

## Struttura S3 CatalogHudiSource

Speciifica una fonte di dati Hudi registrata nel Data Catalog. AWS Glue L'origine dati Hudi deve essere archiviata in. Amazon S3

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome dell'origine dati Hudi.
- `Database` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome del database da cui leggere.
- `Table` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome della tabella nel database da cui leggere.
- `AdditionalHudiOptions`: una matrice della mappa di coppie chiave-valore.

Ogni chiave è una stringa UTF-8 corrispondente al Custom string pattern #61.

Ogni valore è una stringa UTF-8 corrispondente al Custom string pattern #61.

Specifica opzioni di connessione aggiuntive.

- `OutputSchemas`: una matrice di oggetti GlueSchema.

  Specifica lo schema di dati per l'origine Hudi.

## Struttura S3 ExcelSource

Specifica un'origine dati S3 Excel.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome dell'origine dati S3 Excel.
- `Paths` - obbligatorio: una matrice di stringhe UTF-8.

  I percorsi S3 in cui si trovano i file Excel.
- `CompressionType`: stringa UTF-8 (valori validi: `snappy="SNAPPY"` | `lzo="LZO"` | `gzip="GZIP"` | `brotli="BROTLI"` | `lz4="LZ4"` | `uncompressed="UNCOMPRESSED"` | `none="NONE"`).

  Il formato di compressione utilizzato per i file Excel.
- `Exclusions`: una matrice di stringhe UTF-8.

  Modelli per escludere file o percorsi specifici dall'elaborazione.
- `GroupSize`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Definisce la dimensione dei gruppi di file per l'elaborazione in batch.
- `GroupFiles`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Specifica come i file devono essere raggruppati per l'elaborazione.
- `Recurse`: booleano.

  Indica se elaborare le sottodirectory in modo ricorsivo.

- `MaxBand`: numero (intero), non superiore a Nessuno.

  Il numero massimo di bande di elaborazione da utilizzare.
- `MaxFilesInBand`: numero (intero), non superiore a Nessuno.

  Il numero massimo di file da elaborare in ogni banda.
- `AdditionalOptions`: un oggetto S3 DirectSourceAdditionalOptions.

  Opzioni di configurazione aggiuntive per l'elaborazione delle origini S3 dirette.
- `NumberRows`: numero (lungo).

  Il numero di righe da elaborare da ogni file Excel.
- `SkipFooter`: numero (intero), non superiore a Nessuno.

  Il numero di righe da saltare alla fine di ogni file Excel.
- `OutputSchemas`: una matrice di oggetti GlueSchema.

  Gli AWS Glue schemi da applicare ai dati elaborati.

## CatalogHudiSource struttura

Specifica un'origine dati Hudi registrata nel AWS Glue Data Catalog.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome dell'origine dati Hudi.
- `Database` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome del database da cui leggere.
- `Table` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome della tabella nel database da cui leggere.
- `AdditionalHudiOptions`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8 corrispondente al Custom string pattern #61.

  Ogni valore è una stringa UTF-8 corrispondente al Custom string pattern #61.

Specifica opzioni di connessione aggiuntive.

- `OutputSchemas`: una matrice di oggetti GlueSchema.

  Specifica lo schema di dati per l'origine Hudi.

## Struttura Dynamo DBCatalog Source

Specifica un'origine dati DynamoDB nel Data Catalog. AWS Glue

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome dell'origine dati.
- `Database` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome del database da cui leggere.
- `Table` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome della tabella nel database da cui leggere.
- `PitrEnabled`: booleano.

  Speciifica se Point-in-Time Recovery (PITR) è abilitato per la tabella DynamoDB. Se impostato su `true`, consente la lettura a partire da un point-in-time specifico. Il valore predefinito è `false`.
- `AdditionalOptions`: un oggetto DDBELTCatalogAdditionalOptions.

  Specifica opzioni di connessione aggiuntive per l'origine dati DynamoDB.

## RelationalCatalogSource struttura

Specifica un'origine dei dati del database relazionale nel Catalogo dati di AWS Glue .

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome dell'origine dati.
- `Database` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

Il nome del database da cui leggere.

- `Table` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome della tabella nel database da cui leggere.

## JDBCConnectorStruttura dell'obiettivo

Specifica una destinazioni di dati che scrive su Amazon S3 nell'archiviazione a colonne di Apache Parquet.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome di destinazione dati.

- `Inputs` - obbligatorio: una matrice di stringhe UTF-8, non inferiore o superiore a 1 stringa.

  I nodi che sono input per la destinazione di dati.

- `ConnectionName` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome della connessione associata al connettore.

- `ConnectionTable` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome della tabella nella destinazione di dati.

- `ConnectorName` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome di un connettore che verrà utilizzato.

- `ConnectionType` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il tipo di connessione, come marketplace.jdbc o custom.jdbc, che designa una connessione a una destinazione di dati JDBC.

- `AdditionalOptions`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8 corrispondente al Custom string pattern #61.

  Ogni valore è una stringa UTF-8 corrispondente al Custom string pattern #61.

  Opzioni di connessione aggiuntive per il connettore.

- `OutputSchemas`: una matrice di oggetti GlueSchema.

  Specifica lo schema dati per la destinazione JDBC.

## SparkConnectorTarget struttura

Specifica una destinazione che utilizza un connettore Apache Spark.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome di destinazione dati.

- `Inputs` - obbligatorio: una matrice di stringhe UTF-8, non inferiore o superiore a 1 stringa.

  I nodi che sono input per la destinazione di dati.

- `ConnectionName` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome di una connessione per un connettore Apache Spark.

- `ConnectorName` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome di un connettore Apache Spark.

- `ConnectionType` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il tipo di connessione, come marketplace.spark o custom.spark, che designa una connessione a un archivio dati di Apache Spark.

- `AdditionalOptions`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8 corrispondente al Custom string pattern #61.

  Ogni valore è una stringa UTF-8 corrispondente al Custom string pattern #61.

  Opzioni di connessione aggiuntive per il connettore.

- `OutputSchemas`: una matrice di oggetti GlueSchema.

  Specifica lo schema di dati per la destinazione Spark personalizzata.

## BasicCatalogTarget struttura

Speciifica una destinazione che utilizza una tabella del catalogo AWS Glue dati.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome della destinazione di dati.

- `Inputs` - obbligatorio: una matrice di stringhe UTF-8, non inferiore o superiore a 1 stringa.

  I nodi che sono input per la destinazione di dati.

- `PartitionKeys`: una matrice di stringhe UTF-8.

  Le chiavi di partizione utilizzate per distribuire i dati in più partizioni o shard in base a una chiave o a un set di chiavi specifici.

- `Database` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il database che contiene la tabella da utilizzare come destinazione. Questo database deve esistere già nel catalogo dati.

- `Table` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  La tabella che definisce lo schema dei dati di output. Questa tabella deve esistere già nel Data Catalog.

## La mia struttura SQLCatalog Target

Specifica una destinazione che utilizza MySQL.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome di destinazione dati.

- `Inputs` - obbligatorio: una matrice di stringhe UTF-8, non inferiore o superiore a 1 stringa.

  I nodi che sono input per la destinazione di dati.

- `Database` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

Il nome del database in cui scrivere.

- `Table` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome della tabella del database in cui scrivere.

## Struttura di Postgree Target SQLCatalog

Specifica una destinazione che utilizza Postgres SQL.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome di destinazione dati.

- `Inputs` - obbligatorio: una matrice di stringhe UTF-8, non inferiore o superiore a 1 stringa.

  I nodi che sono input per la destinazione di dati.

- `Database` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome del database in cui scrivere.

- `Table` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome della tabella del database in cui scrivere.

## Struttura di Oracle Target SQLCatalog

Specifica una destinazione che utilizza Oracle SQL.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome di destinazione dati.

- `Inputs` - obbligatorio: una matrice di stringhe UTF-8, non inferiore o superiore a 1 stringa.

  I nodi che sono input per la destinazione di dati.

- `Database` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

Il nome del database in cui scrivere.

- `Table` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome della tabella del database in cui scrivere.

## SQLServerCatalogTarget Struttura Microsoft

Specifica una destinazione che utilizza Microsoft SQL.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome di destinazione dati.

- `Inputs` - obbligatorio: una matrice di stringhe UTF-8, non inferiore o superiore a 1 stringa.

  I nodi che sono input per la destinazione di dati.

- `Database` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome del database in cui scrivere.

- `Table` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome della tabella del database in cui scrivere.

## RedshiftTarget struttura

Specifica una destinazione che utilizza Amazon Redshift.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome di destinazione dati.

- `Inputs` - obbligatorio: una matrice di stringhe UTF-8, non inferiore o superiore a 1 stringa.

  I nodi che sono input per la destinazione di dati.

- `Database` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

Il nome del database in cui scrivere.

- `Table` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome della tabella del database in cui scrivere.

- `RedshiftTmpDir`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il percorso Amazon S3 in cui i dati temporanei possono essere caricati durante la copia dal database.

- `TmpDirIAMRole`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il ruolo IAM con autorizzazioni.

- `UpsertRedshiftOptions`: un oggetto UpsertRedshiftTargetOptions.

  Il set di opzioni per configurare un'operazione di upsert durante la scrittura su una destinazione Redshift.

## AmazonRedshiftTarget struttura

Specifica una destinazione Amazon Redshift.

Campi

- `Name`: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome della tabella Amazon Redshift.

- `Data`: un oggetto AmazonRedshiftNodeData.

  Specifica i dati del nodo di destinazione Amazon Redshift.

- `Inputs`: un array di stringhe UTF-8, non inferiore o superiore a 1 stringa.

  I nodi che sono input per la destinazione di dati.

## UpsertRedshiftTargetOptions struttura

Le opzioni per configurare un'operazione di upsert durante la scrittura su una destinazione Redshift.

Campi

- `TableLocation`: stringa UTF-8, corrispondente a Custom string pattern #61.

  La posizione fisica della tabella Redshift.
- `ConnectionName`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome della connessione da usare per scrivere su Redshift.
- `UpsertKeys`: una matrice di stringhe UTF-8.

  Le chiavi utilizzate per determinare se eseguire un aggiornamento o un inserimento.

## struttura S3 CatalogTarget

Speciifica un target di dati che scrive su Amazon S3 utilizzando AWS Glue il Data Catalog.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome di destinazione dati.
- `Inputs` - obbligatorio: una matrice di stringhe UTF-8, non inferiore o superiore a 1 stringa.

  I nodi che sono input per la destinazione di dati.
- `PartitionKeys`: una matrice di stringhe UTF-8.

  Specifica il partizionamento nativo utilizzando una sequenza di chiavi.
- `Table` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome della tabella del database in cui scrivere.
- `Database` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome del database in cui scrivere.
- `SchemaChangePolicy`: un oggetto CatalogSchemaChangePolicy.

  Una policy che specifica i comportamenti di aggiornamento per il crawler.
- `AutoDataQuality`: un oggetto AutoDataQuality.

Specifica se abilitare automaticamente la valutazione della qualità dei dati per il catalogo S3 di destinazione. Se impostato su `true`, i controlli di qualità dei dati vengono eseguiti automaticamente durante l'operazione di scrittura.

## Struttura S3 GlueParquetTarget

Specifica una destinazioni di dati che scrive su Amazon S3 nell'archiviazione a colonne di Apache Parquet.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome di destinazione dati.

- `Inputs` - obbligatorio: una matrice di stringhe UTF-8, non inferiore o superiore a 1 stringa.

  I nodi che sono input per la destinazione di dati.

- `PartitionKeys`: una matrice di stringhe UTF-8.

  Specifica il partizionamento nativo utilizzando una sequenza di chiavi.

- `Path` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Un singolo percorso Amazon S3 su cui scrivere.

- `Compression`: stringa UTF-8 (valori validi: `snappy="SNAPPY"` | `lzo="LZO"` | `gzip="GZIP"` | `brotli="BROTLI"` | `lz4="LZ4"` | `uncompressed="UNCOMPRESSED"` | `none="NONE"`).

  Specifica il modo in cui i dati sono compressi. In genere questo non è necessario se i dati hanno un'estensione del file standard. I valori possibili sono `"gzip"` e `"bzip"`).

- `NumberTargetPartitions`: stringa UTF-8.

  Specifica il numero di partizioni di destinazione per i file Parquet durante la scrittura in Amazon S3 utilizzando AWS Glue.

- `SchemaChangePolicy`: un oggetto DirectSchemaChangePolicy.

  Una policy che specifica i comportamenti di aggiornamento per il crawler.

- `AutoDataQuality`: un oggetto AutoDataQuality.

Speciifica se abilitare automaticamente la valutazione della qualità dei dati per il target S3 Parquet AWS Glue . Se impostato su `true`, i controlli di qualità dei dati vengono eseguiti automaticamente durante l'operazione di scrittura.

## CatalogSchemaChangePolicy struttura

Una policy che specifica i comportamenti di aggiornamento per il crawler.

Campi

- `EnableUpdateCatalog`: booleano.

  Stabilisce se usare il comportamento di aggiornamento quando il crawler riscontra una variazione dello schema.
- `UpdateBehavior`: stringa UTF-8 (valori validi: `UPDATE_IN_DATABASE` | `LOG`).

  Il comportamento di aggiornamento quando il crawler riscontra una variazione dello schema.

## struttura S3 DirectTarget

Specifica una destinazione di dati che scrive su Amazon S3.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome di destinazione dati.
- `Inputs` - obbligatorio: una matrice di stringhe UTF-8, non inferiore o superiore a 1 stringa.

  I nodi che sono input per la destinazione di dati.
- `PartitionKeys`: una matrice di stringhe UTF-8.

  Specifica il partizionamento nativo utilizzando una sequenza di chiavi.
- `Path` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Un singolo percorso Amazon S3 su cui scrivere.
- `Compression`: stringa UTF-8, corrispondente a Custom string pattern #61.

Specifica il modo in cui i dati sono compressi. In genere questo non è necessario se i dati hanno un'estensione del file standard. I valori possibili sono `"gzip"` e `"bzip"`).

- `NumberTargetPartitions`: stringa UTF-8.

  Specifica il numero di partizioni di destinazione durante la scrittura dei dati direttamente in Amazon S3.

- `Format` - obbligatorio: stringa UTF-8 (valori validi: `json="JSON"` | `csv="CSV"` | `avro="AVRO"` | `orc="ORC"` | `parquet="PARQUET"` | `hudi="HUDI"` | `delta="DELTA"` | `iceberg="ICEBERG"` | `hyper="HYPER"` | `xml="XML"`).

  Specifica il formato di output dei dati per la destinazione.

- `SchemaChangePolicy`: un oggetto DirectSchemaChangePolicy.

  Una policy che specifica i comportamenti di aggiornamento per il crawler.

- `AutoDataQuality`: un oggetto AutoDataQuality.

  Specifica se abilitare automaticamente la valutazione della qualità dei dati per la destinazione diretta S3. Se impostato su `true`, i controlli di qualità dei dati vengono eseguiti automaticamente durante l'operazione di scrittura.

- `OutputSchemas`: una matrice di oggetti GlueSchema.

  Specifica lo schema di dati per la destinazione diretta S3.

## Struttura S3 HudiCatalogTarget

Specifica una destinazione che scrive su un'origine dati Hudi nel Data Catalog. AWS Glue

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome di destinazione dati.

- `Inputs` - obbligatorio: una matrice di stringhe UTF-8, non inferiore o superiore a 1 stringa.

  I nodi che sono input per la destinazione di dati.

- `PartitionKeys`: una matrice di stringhe UTF-8.

  Specifica il partizionamento nativo utilizzando una sequenza di chiavi.

- `Table` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome della tabella del database in cui scrivere.

- `Database` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome del database in cui scrivere.

- `AdditionalOptions` - obbligatorio: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8 corrispondente al Custom string pattern #61.

  Ogni valore è una stringa UTF-8 corrispondente al Custom string pattern #61.

  Specifica le opzioni di connessione aggiuntive per il connettore.

- `SchemaChangePolicy`: un oggetto CatalogSchemaChangePolicy.

  Una policy che specifica i comportamenti di aggiornamento per il crawler.

- `AutoDataQuality`: un oggetto AutoDataQuality.

  Specifica se abilitare automaticamente la valutazione della qualità dei dati per il catalogo S3 Hudi di destinazione. Se impostato su `true`, i controlli di qualità dei dati vengono eseguiti automaticamente durante l'operazione di scrittura.

- `OutputSchemas`: una matrice di oggetti GlueSchema.

  Specifica lo schema di dati per il catalogo S3 Hudi di destinazione.

## Struttura S3 HudiDirectTarget

Speciifica una destinazione che scrive su una fonte di dati Hudi in. Amazon S3

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome di destinazione dati.

- `Inputs` - obbligatorio: una matrice di stringhe UTF-8, non inferiore o superiore a 1 stringa.

  I nodi che sono input per la destinazione di dati.

- `Path` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

Il percorso Amazon S3 dell'origine dati Hudi su cui scrivere.

- `Compression` - obbligatorio: stringa UTF-8 (valori validi: `gzip="GZIP"` | `lzo="LZO"` | `uncompressed="UNCOMPRESSED"` | `snappy="SNAPPY"`).

  Specifica il modo in cui i dati sono compressi. In genere questo non è necessario se i dati hanno un'estensione del file standard. I valori possibili sono `"gzip"` e `"bzip"`).

- `NumberTargetPartitions`: stringa UTF-8.

  Specifica il numero di partizioni della destinazione per la distribuzione dei file del set di dati Hudi in Amazon S3.

- `PartitionKeys`: una matrice di stringhe UTF-8.

  Specifica il partizionamento nativo utilizzando una sequenza di chiavi.

- `Format` - obbligatorio:, stringa UTF-8 (valori validi: `json="JSON"` | `csv="CSV"` | `avro="AVRO"` | `orc="ORC"` | `parquet="PARQUET"` | `hudi="HUDI"` | `delta="DELTA"` | `iceberg="ICEBERG"` | `hyper="HYPER"` | `xml="XML"`).

  Specifica il formato di output dei dati per la destinazione.

- `AdditionalOptions` - obbligatorio: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8 corrispondente al Custom string pattern #61.

  Ogni valore è una stringa UTF-8 corrispondente al Custom string pattern #61.

  Specifica le opzioni di connessione aggiuntive per il connettore.

- `SchemaChangePolicy`: un oggetto DirectSchemaChangePolicy.

  Una policy che specifica i comportamenti di aggiornamento per il crawler.

- `AutoDataQuality`: un oggetto AutoDataQuality.

  Specifica se abilitare automaticamente la valutazione della qualità dei dati per la destinazione diretta S3 Hudi. Se impostato su `true`, i controlli di qualità dei dati vengono eseguiti automaticamente durante l'operazione di scrittura.

## Struttura S3 DeltaCatalogTarget

Speciifica una destinazione che scrive su un'origine dati Delta Lake nel AWS Glue Data Catalog.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome di destinazione dati.
- `Inputs` - obbligatorio: una matrice di stringhe UTF-8, non inferiore o superiore a 1 stringa.

  I nodi che sono input per la destinazione di dati.
- `PartitionKeys`: una matrice di stringhe UTF-8.

  Specifica il partizionamento nativo utilizzando una sequenza di chiavi.
- `Table` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome della tabella del database in cui scrivere.
- `Database` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome del database in cui scrivere.
- `AdditionalOptions`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8 corrispondente al Custom string pattern #61.

  Ogni valore è una stringa UTF-8 corrispondente al Custom string pattern #61.

  Specifica le opzioni di connessione aggiuntive per il connettore.
- `SchemaChangePolicy`: un oggetto CatalogSchemaChangePolicy.

  Una policy che specifica i comportamenti di aggiornamento per il crawler.
- `AutoDataQuality`: un oggetto AutoDataQuality.

  Specifica se abilitare automaticamente la valutazione della qualità dei dati per il catalogo S3 Delta di destinazione. Se impostato su `true`, i controlli di qualità dei dati vengono eseguiti automaticamente durante l'operazione di scrittura.
- `OutputSchemas`: una matrice di oggetti GlueSchema.

  Specifica lo schema di dati per il catalogo S3 Delta di destinazione.

## Struttura S3 DeltaDirectTarget

Speciifica una destinazione che scrive su un'origine dati Delta Lake in. Amazon S3

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome di destinazione dati.
- `Inputs` - obbligatorio: una matrice di stringhe UTF-8, non inferiore o superiore a 1 stringa.

  I nodi che sono input per la destinazione di dati.
- `PartitionKeys`: una matrice di stringhe UTF-8.

  Specifica il partizionamento nativo utilizzando una sequenza di chiavi.
- `Path` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il percorso Amazon S3 dell'origine dati Delta Lake su cui scrivere.
- `Compression` - obbligatorio: stringa UTF-8 (valori validi: `uncompressed="UNCOMPRESSED"` | `snappy="SNAPPY"`).

  Specifica il modo in cui i dati sono compressi. In genere questo non è necessario se i dati hanno un'estensione del file standard. I valori possibili sono `"gzip"` e `"bzip"`).
- `NumberTargetPartitions`: stringa UTF-8.

  Specifica il numero di partizioni della destinazione per la distribuzione dei file del set di dati Delta Lake in Amazon S3.
- `Format` - obbligatorio, stringa UTF-8 (valori validi: `json="JSON"` | `csv="CSV"` | `avro="AVRO"` | `orc="ORC"` | `parquet="PARQUET"` | `hudi="HUDI"` | `delta="DELTA"` | `iceberg="ICEBERG"` | `hyper="HYPER"` | `xml="XML"`).

  Specifica il formato di output dei dati per la destinazione.
- `AdditionalOptions`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8 corrispondente al Custom string pattern #61.

  Ogni valore è una stringa UTF-8 corrispondente al Custom string pattern #61.

  Specifica le opzioni di connessione aggiuntive per il connettore.
- `SchemaChangePolicy`: un oggetto DirectSchemaChangePolicy.

  Una policy che specifica i comportamenti di aggiornamento per il crawler.
- `AutoDataQuality`: un oggetto AutoDataQuality.

Specifica se abilitare automaticamente la valutazione della qualità dei dati per la destinazione diretta S3 Delta. Se impostato su `true`, i controlli di qualità dei dati vengono eseguiti automaticamente durante l'operazione di scrittura.

## Struttura S3 HyperDirectTarget

Speciifica un target di HyperDirect dati che scrive su Amazon S3.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  L'identificatore univoco per il nodo di destinazione. HyperDirect
- `Inputs` - obbligatorio: una matrice di stringhe UTF-8, non inferiore o superiore a 1 stringa.

  Specifica la fonte di input per la HyperDirect destinazione.
- `Format`: stringa UTF-8 (valori validi: `json="JSON"` | `csv="CSV"` | `avro="AVRO"` | `orc="ORC"` | `parquet="PARQUET"` | `hudi="HUDI"` | `delta="DELTA"` | `iceberg="ICEBERG"` | `hyper="HYPER"` | `xml="XML"`).

  Speciifica il formato di output dei dati per la HyperDirect destinazione.
- `PartitionKeys`: una matrice di stringhe UTF-8.

  Definisce la strategia di partizionamento per i dati di output.
- `Path` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  La posizione S3 dove verranno scritti i dati di output.
- `Compression`: stringa UTF-8 (valori validi: `uncompressed="UNCOMPRESSED"`).

  Il tipo di compressione da applicare ai dati di output.
- `SchemaChangePolicy`: un oggetto DirectSchemaChangePolicy.

  Definisce come vengono gestite le modifiche allo schema durante le operazioni di scrittura.
- `AutoDataQuality`: un oggetto AutoDataQuality.

  Specifica se abilitare automaticamente la valutazione della qualità dei dati per la destinazione diretta S3 Hyper. Se impostato su `true`, i controlli di qualità dei dati vengono eseguiti automaticamente durante l'operazione di scrittura.

- `OutputSchemas`: una matrice di oggetti GlueSchema.

  Specifica lo schema di dati per la destinazione diretta S3 Hyper.

## Struttura S3 IcebergDirectTarget

Specifica una destinazione che scrive su un'origine dati Iceberg in Amazon S3.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Specifica l'identificatore univoco per il nodo di destinazione Iceberg nella pipeline di dati.

- `Inputs` - obbligatorio: una matrice di stringhe UTF-8, non inferiore o superiore a 1 stringa.

  Definisce la singola origine di input che fornisce i dati a questa destinazione Iceberg.

- `PartitionKeys`: una matrice di stringhe UTF-8.

  Specifica le colonne utilizzate per partizionare i dati della tabella Iceberg in S3.

- `Path` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Definisce la posizione S3 in cui saranno archiviati i dati della tabella Iceberg.

- `Format` - obbligatorio: stringa UTF-8 (valori validi: `json="JSON"` | `csv="CSV"` | `avro="AVRO"` | `orc="ORC"` | `parquet="PARQUET"` | `hudi="HUDI"` | `delta="DELTA"` | `iceberg="ICEBERG"` | `hyper="HYPER"` | `xml="XML"`).

  Specifica il formato di file utilizzato per l'archiviazione dei dati della tabella Iceberg (ad es., Parquet, ORC).

- `AdditionalOptions`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8 corrispondente al Custom string pattern #61.

  Ogni valore è una stringa UTF-8 corrispondente al Custom string pattern #61.

  Fornisce opzioni di configurazione aggiuntive per la personalizzazione del comportamento della tabella Iceberg.

- `SchemaChangePolicy`: un oggetto DirectSchemaChangePolicy.

Definisce come vengono gestite le modifiche allo schema durante la scrittura dei dati nella tabella Iceberg.

- `Compression` - obbligatorio: stringa UTF-8 (valori validi: `gzip="GZIP"` | `lzo="LZO"` | `uncompressed="UNCOMPRESSED"` | `snappy="SNAPPY"`).

  Specifica il codec di compressione utilizzato per i file della tabella Iceberg in S3.

- `NumberTargetPartitions`: stringa UTF-8.

  Imposta il numero di partizioni della destinazione per la distribuzione dei file della tabella Iceberg in S3.

- `OutputSchemas`: una matrice di oggetti GlueSchema.

  Specifica lo schema di dati per la destinazione diretta S3 Iceberg.

## DirectSchemaChangePolicy struttura

Una policy che specifica i comportamenti di aggiornamento per il crawler.

Campi

- `EnableUpdateCatalog`: booleano.

  Stabilisce se usare il comportamento di aggiornamento quando il crawler riscontra una variazione dello schema.

- `UpdateBehavior`: stringa UTF-8 (valori validi: `UPDATE_IN_DATABASE` | `LOG`).

  Il comportamento di aggiornamento quando il crawler riscontra una variazione dello schema.

- `Table`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Specifica la tabella nel database a cui si applica la policy di modifica dello schema.

- `Database`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Specifica il database a cui si applica la policy di modifica dello schema.

## ApplyMapping struttura

Specifica una trasformazione che mappa le chiavi delle proprietà dei dati nell'origine dei dati alle chiavi delle proprietà dei dati nella destinazione. È possibile rinominare le chiavi, modificare i tipi di dati per le chiavi e scegliere le chiavi da eliminare dal set di dati.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome del nodo di trasformazione.
- `Inputs` - obbligatorio: una matrice di stringhe UTF-8, non inferiore o superiore a 1 stringa.

  Gli input di dati identificati dai nomi dei nodi.
- `Mapping` - obbligatorio: una matrice di oggetti Mapping.

  Specifica la mappatura delle chiavi delle proprietà dei dati nell'origine dei dati alle chiavi delle proprietà dei dati nella destinazione.

## Struttura mappatura

Specifica la mappatura delle chiavi della proprietà dati.

Campi

- `ToKey`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Dopo l'applicazione della mappatura, quale dovrebbe essere il nome della colonna. Può coincidere con `FromPath`.
- `FromPath`: una matrice di stringhe UTF-8.

  La tabella o la colonna da modificare.
- `FromType`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il tipo di dati da modificare.
- `ToType`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Tipo di dati che devono essere modificati.
- `Dropped`: booleano.

Se è true, la colonna viene rimossa.

- `Children`: una matrice di oggetti Mapping.

  Applicabile solo alle strutture dati nidificate. Se si desidera modificare la struttura padre, ma anche uno dei suoi figli, è possibile compilare questa struttura di dati. È anche `Mapping`, ma il suo `FromPath` sarà la struttura padre `FromPath` più il `FromPath` da questa struttura.

  Per la parte dei figli, supponiamo di avere la struttura:

  ```
  { "FromPath": "OuterStructure", "ToKey": "OuterStructure", "ToType":
  "Struct", "Dropped": false, "Chidlren": [{ "FromPath": "inner", "ToKey":
  "inner", "ToType": "Double", "Dropped": false, }] }
  ```

  Puoi specificare un `Mapping` con l'aspetto:

  ```
  { "FromPath": "OuterStructure", "ToKey": "OuterStructure", "ToType":
  "Struct", "Dropped": false, "Chidlren": [{ "FromPath": "inner", "ToKey":
  "inner", "ToType": "Double", "Dropped": false, }] }
  ```

## SelectFields struttura

Specifica una trasformazione che sceglie le chiavi della proprietà dati che si desidera conservare.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome del nodo di trasformazione.

- `Inputs` - obbligatorio: una matrice di stringhe UTF-8, non inferiore o superiore a 1 stringa.

  Gli input di dati identificati dai nomi dei nodi.

- `Paths` - obbligatorio: una matrice di stringhe UTF-8.

  Un percorso JSON a una variabile nella struttura dati.

## DropFields struttura

Specifica una trasformazione che sceglie le chiavi della proprietà dati che si desidera eliminare.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome del nodo di trasformazione.
- `Inputs` - obbligatorio: una matrice di stringhe UTF-8, non inferiore o superiore a 1 stringa.

  Gli input di dati identificati dai nomi dei nodi.
- `Paths` - obbligatorio: una matrice di stringhe UTF-8.

  Un percorso JSON a una variabile nella struttura dati.

## RenameField struttura

Specifica una trasformazione che rinominerà una singola chiave di proprietà dati.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome del nodo di trasformazione.
- `Inputs` - obbligatorio: una matrice di stringhe UTF-8, non inferiore o superiore a 1 stringa.

  Gli input di dati identificati dai nomi dei nodi.
- `SourcePath` - obbligatorio: una matrice di stringhe UTF-8.

  Un percorso JSON a una variabile nella struttura dati per i dati di origine.
- `TargetPath` - obbligatorio: una matrice di stringhe UTF-8.

  Un percorso JSON a una variabile nella struttura dati per i dati di destinazione.

## Struttura Spigot

Specifica una trasformazione che scrive campioni dei dati in un bucket Amazon S3.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome del nodo di trasformazione.

- `Inputs` - obbligatorio: una matrice di stringhe UTF-8, non inferiore o superiore a 1 stringa.

  Gli input di dati identificati dai nomi dei nodi.

- `Path` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Un percorso in Amazon S3 dove la trasformazione scriverà un sottoinsieme di registri dal set di dati in un file JSON in un bucket Amazon S3.

- `Topk`: numero (intero), non superiore a 100.

  Specifica un numero di registri da scrivere a partire dall'inizio del set di dati.

- `Prob`: numero (doppio), non superiore a 1.

  La probabilità (un valore decimale con un valore massimo di 1) di scegliere un determinato registro. Il valore 1 indica che ogni riga letta dal set di dati deve essere inclusa nell'output del campione.

## Struttura join

Specifica una trasformazione che unisce due set di dati in un unico set di dati utilizzando una frase di confronto sulle chiavi di proprietà dei dati specificate. È possibile utilizzare inner, outer, left, right, left semi e left anti join.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome del nodo di trasformazione.

- `Inputs` - obbligatorio: una matrice di stringhe UTF-8, non inferiore a o superiore a 2 stringhe.

  Gli input di dati identificati dai nomi dei nodi.

- `JoinType` - obbligatorio:: stringa UTF-8 (valori validi: `equijoin="EQUIJOIN"` | `left="LEFT"` | `right="RIGHT"` | `outer="OUTER"` | `leftsemi="LEFT_SEMI"` | `leftanti="LEFT_ANTI"`).

  Specifica il tipo di join da eseguire sui set di dati.

- `Columns` - obbligatorio: una matrice di oggetti JoinColumn, non inferiore a o superiore a 2 strutture.

  Un elenco delle due colonne da unire.

## JoinColumn struttura

Specifica una colonna da unire.

Campi

- `From` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  La colonna da unire.
- `Keys` - obbligatorio: una matrice di stringhe UTF-8.

  La chiave della colonna da unire.

## SplitFields struttura

Specifica una trasformazione che divide le chiavi della proprietà dati in due `DynamicFrames`. L'output è una raccolta di `DynamicFrames`: uno con le chiavi di proprietà dei dati selezionate e uno con le chiavi di proprietà dei dati rimanenti.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome del nodo di trasformazione.
- `Inputs` - obbligatorio: una matrice di stringhe UTF-8, non inferiore o superiore a 1 stringa.

  Gli input di dati identificati dai nomi dei nodi.
- `Paths` - obbligatorio: una matrice di stringhe UTF-8.

  Un percorso JSON a una variabile nella struttura dati.

## SelectFromCollection struttura

Specifica una trasformazione che sceglie un `DynamicFrame` da una raccolta di `DynamicFrames`. L'output è il `DynamicFrame` selezionato

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

Il nome del nodo di trasformazione.

- `Inputs` - obbligatorio: una matrice di stringhe UTF-8, non inferiore o superiore a 1 stringa.

  Gli input di dati identificati dai nomi dei nodi.

- `Index` - obbligatorio: numero (intero), non superiore a Nessuno.

  L'indice per il DynamicFrame da selezionare.

## FillMissingValues struttura

Specifica una trasformazione che individua i registri nel set di dati che hanno valori mancanti e aggiunge un nuovo campo con un valore determinato dall'imputazione. Il set di dati di input viene utilizzato per addestrare il modello di machine learning che determina quale dovrebbe essere il valore mancante.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome del nodo di trasformazione.

- `Inputs` - obbligatorio: una matrice di stringhe UTF-8, non inferiore o superiore a 1 stringa.

  Gli input di dati identificati dai nomi dei nodi.

- `ImputedPath` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Un percorso JSON a una variabile nella struttura dati per il set di dati imputato.

- `FilledPath`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Un percorso JSON a una variabile nella struttura dati per il set di dati compilato.

## Struttura filtro

Specifica una trasformazione che divide un set di dati in due, in base a una condizione di filtro.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

Il nome del nodo di trasformazione.

- `Inputs` - obbligatorio: una matrice di stringhe UTF-8, non inferiore o superiore a 1 stringa.

  Gli input di dati identificati dai nomi dei nodi.

- `LogicalOperator` - obbligatorio: stringa UTF-8 (valori validi: `AND` | `OR`).

  L'operatore utilizzato per filtrare le righe confrontando il valore chiave con un valore specificato.

- `Filters` - obbligatorio: una matrice di oggetti FilterExpression.

  Specifica un'espressione di filtro.

## FilterExpression struttura

Specifica un'espressione di filtro.

Campi

- `Operation` - obbligatorio: stringa UTF-8 (valori validi: `EQ` | `LT` | `GT` | `LTE` | `GTE` | `REGEX` | `ISNULL`).

  Tipo di operazione da eseguire nell'espressione.

- `Negated`: booleano.

  Se l'espressione deve essere negata.

- `Values` - obbligatorio: una matrice di oggetti FilterValue.

  Un elenco di valori di filtro.

## FilterValue struttura

Rappresenta un'unica voce nell'elenco di valori di un `FilterExpression`.

Campi

- `Type` - obbligatorio: stringa UTF-8 (valori validi: `COLUMNEXTRACTED` | `CONSTANT`).

  Il tipo di valore del filtro.

- `Value` - obbligatorio: una matrice di stringhe UTF-8.

Il valore da associare.

## CustomCode struttura

Specifica una trasformazione che utilizza il codice personalizzato fornito per eseguire la trasformazione dei dati. L'output è una raccolta di DynamicFrames.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome del nodo di trasformazione.

- `Inputs` - obbligatorio: una matrice di stringhe UTF-8, almeno 1 stringa.

  Gli input di dati identificati dai nomi dei nodi.

- `Code` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #54.

  Il codice personalizzato utilizzato per eseguire la trasformazione dei dati.

- `ClassName` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome definito per la classe del nodo di codice personalizzato.

- `OutputSchemas`: una matrice di oggetti GlueSchema.

  Specifica lo schema di dati per la trasformazione del codice personalizzata.

## Struttura SparkSQL

Specifica una trasformazione in cui si inserisce una query SQL utilizzando la sintassi Spark SQL per trasformare i dati. L'output è un singolo `DynamicFrame`.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome del nodo di trasformazione.

- `Inputs` - obbligatorio: una matrice di stringhe UTF-8, almeno 1 stringa.

Gli input di dati identificati dai nomi dei nodi. È possibile associare un nome di tabella a ciascun nodo di input da utilizzare nella query SQL. Il nome scelto deve soddisfare le restrizioni sui nomi di Spark SQL.

- `SqlQuery` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #62.

  Query SQL che deve utilizzare la sintassi Spark SQL e restituire un singolo set di dati.

- `SqlAliases` - obbligatorio: una matrice di oggetti SqlAlias.

  Un elenco di alias. Un alias permette di specificare il nome da utilizzare nell'SQL per un determinato input. Ad esempio, hai una fonte di dati denominata "»MyDataSource. Se specifichi `From` as MyDataSource e `Alias` as SqlName, nel tuo SQL puoi fare:

  ```
  select * from SqlName
  ```

  e che ottiene dati da MyDataSource.

- `OutputSchemas`: una matrice di oggetti GlueSchema.

  Specifica lo schema di dati per la trasformazione SparkSQL.

## SqlAlias struttura

Rappresenta un'unica voce nell'elenco di valori per `SqlAliases`.

Campi

- `From` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #60.

  Una tabella o una colonna in una tabella.

- `Alias` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Un nome temporaneo dato a una tabella o a una colonna in una tabella.

## DropNullFields struttura

Specifica una trasformazione che rimuove le colonne dal set di dati se tutti i valori nella colonna sono “null”. Per impostazione predefinita, AWS Glue Studio riconosce gli oggetti nulli, ma alcuni valori come stringhe vuote, stringhe «nulle», numeri interi -1 o altri segnaposto come zeri, non vengono riconosciuti automaticamente come nulli.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome del nodo di trasformazione.
- `Inputs` - obbligatorio: una matrice di stringhe UTF-8, non inferiore o superiore a 1 stringa.

  Gli input di dati identificati dai nomi dei nodi.
- `NullCheckBoxList`: un oggetto NullCheckBoxList.

  Struttura che indica se determinati valori siano riconosciuti come valori nulli per la rimozione.
- `NullTextList`: una matrice di oggetti NullValueField, non superiore a 50 strutture.

  Una struttura che specifica un elenco di NullValueField strutture che rappresentano un valore nullo personalizzato come zero o un altro valore utilizzato come segnaposto nullo unico per il set di dati.

  La trasformazione `DropNullFields` rimuove i valori nulli personalizzati solo se sia il valore del segnaposto null che il tipo di dati corrispondono ai dati.

## NullCheckBoxList struttura

Indica se alcuni valori siano riconosciuti come valori nulli per la rimozione.

Campi

- `IsEmpty`: booleano.

  Specifica che una stringa vuota è considerata un valore nullo.
- `IsNullString`: booleano.

  Specifica che un valore che indica la parola "null" è considerato un valore nullo.
- `IsNegOne`: booleano.

  Specifica che un valore intero di -1 è considerato un valore nullo.

## NullValueField struttura

Rappresenta un valore nullo personalizzato, ad esempio uno zero o un altro valore utilizzato come segnaposto nullo univoco per il set di dati.

Campi

- `Value` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il valore del segnaposto nullo.
- `Datatype` - obbligatorio: un oggetto DataType.

  Il tipo di dati del valore.

## Struttura Datatype

Struttura che rappresenta il tipo di dati del valore.

Campi

- `Id` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #60.

  Il tipo di dati del valore.
- `Label` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #60.

  Etichetta assegnata al tipo di dati.

## Struttura Merge

Specifica una trasformazione che unisce `DynamicFrame` a con un `DynamicFrame` di staging basato sulle chiavi primarie specificate per identificare i registri. I registri duplicati (registri con le stesse chiavi primarie) non vengono deduplicati.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome del nodo di trasformazione.
- `Inputs` - obbligatorio: una matrice di stringhe UTF-8, non inferiore a o superiore a 2 stringhe.

  Gli input di dati identificati dai nomi dei nodi.
- `Source` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #60.

  L'origine `DynamicFrame` che sarà unita a `DynamicFrame` di staging.

- `PrimaryKeys` - obbligatorio: una matrice di stringhe UTF-8.

  L'elenco dei campi chiave primaria per abbinare i registri dall'origine e dai frame dinamici di staging.

## Struttura unione

Specifica una trasformazione che combina le righe di due o più set di dati in un unico risultato.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome del nodo di trasformazione.

- `Inputs` - obbligatorio: una matrice di stringhe UTF-8, non inferiore a o superiore a 2 stringhe.

  L'ID del nodo immette la trasformazione.

- `UnionType` - obbligatorio: stringa UTF-8 (valori validi: `ALL` | `DISTINCT`).

  Indica il tipo di trasformazione Union.

  `ALL`Specificare di unire tutte le righe dalle fonti di dati a quelle risultanti DynamicFrame. L'unione risultante non rimuove le righe duplicate.

  `DISTINCT`Specificare di rimuovere le righe duplicate nel risultato. DynamicFrame

## PIIDetection struttura

Specifica una trasformazione che identifica, rimuove o maschera i dati PII.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome del nodo di trasformazione.

- `Inputs` - obbligatorio: una matrice di stringhe UTF-8, non inferiore o superiore a 1 stringa.

  L'ID del nodo immette la trasformazione.

- `PiiType` - obbligatorio: stringa UTF-8 (valori validi: `RowAudit` | `RowHashing` | `RowMasking` | `RowPartialMasking` | `ColumnAudit` | `ColumnHashing` | `ColumnMasking`).

  Indica il tipo di PIIDetection trasformazione.

- `EntityTypesToDetect` - obbligatorio: una matrice di stringhe UTF-8.

  Indica i tipi di entità che la PIIDetection trasformazione identificherà come dati PII.

  Le entità di tipo PII includono: PERSON_NAME, DATE, USA_SNN, EMAIL, USA_ITIN, USA_PASSPORT_NUMBER, PHONE_NUMBER, BANK_ACCOUNT, IP_ADDRESS, MAC_ADDRESS, USA_CPT_CODE, USA_HCPCS_CODE, USA_NATIONAL_DRUG_CODE, USA_MEDICARE_BENEFICIARY_IDENTIFIER, USA_HEALTH_INSURANCE_CLAIM_NUMBER,CREDIT_CARD,USA_NATIONAL_PROVIDER_IDENTIFI

- `OutputColumnName`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Indica il nome della colonna di output che conterrà qualsiasi tipo di entità rilevato in quella riga.

- `SampleFraction`: numero (doppio), non superiore a 1.

  Indica la frazione dei dati da campionare durante la scansione di entità PII.

- `ThresholdFraction`: numero (doppio), non superiore a 1.

  Indica la frazione dei dati che devono essere soddisfatti per identificare una colonna come dati PII.

- `MaskValue`: stringa UTF-8, non superiore a 256 byte di lunghezza, corrispondente a Custom string pattern #58.

  Indica il valore che sostituirà l'entità rilevata.

- `RedactText`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Specifica se oscurare il testo contenente informazioni di identificazione personale (PII) rilevato. Se impostato su `true`, il contenuto con PII viene sostituito con caratteri di redazione.

- `RedactChar`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il carattere utilizzato per sostituire il contenuto con PII rilevato quando è attivato l'oscuramento. Il carattere di redazione predefinito è `*`.

- `MatchPattern`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Un modello di espressione regolare utilizzato per identificare il contenuto con PII aggiuntivo oltre gli algoritmi di rilevamento standard.

- `NumLeftCharsToExclude`: numero (intero), non superiore a Nessuno.

  Il numero di caratteri da escludere dalla redazione a sinistra del contenuto con PII rilevato. In questo modo è possibile preservare il contesto attorno ai dati sensibili.

- `NumRightCharsToExclude`: numero (intero), non superiore a Nessuno.

  Il numero di caratteri da escludere dalla redazione a destra del contenuto con PII rilevato. In questo modo è possibile preservare il contesto attorno ai dati sensibili.

- `DetectionParameters`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Parametri aggiuntivi per la configurazione del comportamento di rilevamento delle PII e delle impostazioni di sensibilità.

- `DetectionSensitivity`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il livello di sensibilità per il rilevamento delle PII. Livelli di sensibilità più elevati rilevano più PII potenziali, ma possono dare luogo a più falsi positivi.

## Struttura aggregata

Specifica una trasformazione che raggruppa le righe in base ai campi scelti e calcola il valore aggregato in base alla funzione specificata.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome del nodo di trasformazione.

- `Inputs` - obbligatorio: una matrice di stringhe UTF-8, non inferiore o superiore a 1 stringa.

  Specifica i campi e le righe da utilizzare come input per la trasformazione aggregata.

- `Groups` - obbligatorio: una matrice di stringhe UTF-8.

  Specifica i campi in base ai quali raggruppare.

- `Aggs` - obbligatorio: una matrice di oggetti AggregateOperation, non meno di 1 o più di 30 strutture.

  Specifica le funzioni di aggregazione da eseguire su campi specificati.

## DropDuplicates struttura

Specifica una trasformazione che rimuove le righe di dati ripetuti da un set di dati.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome del nodo di trasformazione.
- `Inputs` - obbligatorio: una matrice di stringhe UTF-8, non inferiore o superiore a 1 stringa.

  Gli input di dati identificati dai nomi dei nodi.
- `Columns`: una matrice di stringhe UTF-8.

  Il nome delle colonne da unire o rimuovere in caso di ripetizione.

## GovernedCatalogTarget struttura

Speciifica un target di dati che scrive su Amazon S3 utilizzando AWS Glue il Data Catalog.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome di destinazione dati.
- `Inputs` - obbligatorio: una matrice di stringhe UTF-8, non inferiore o superiore a 1 stringa.

  I nodi che sono input per la destinazione di dati.
- `PartitionKeys`: una matrice di stringhe UTF-8.

  Specifica il partizionamento nativo utilizzando una sequenza di chiavi.
- `Table` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome della tabella del database in cui scrivere.
- `Database` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome del database in cui scrivere.
- `SchemaChangePolicy`: un oggetto CatalogSchemaChangePolicy.

  Una policy che specifica il comportamento di aggiornamento per il catalogo governato.

## GovernedCatalogSource struttura

Speciifica l'archivio dati nel AWS Glue Data Catalog governato.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome del archivio dati.
- `Database` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il database da cui leggere.
- `Table` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  La tabella del database da cui leggere.
- `PartitionPredicate`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Le partizioni che soddisfano questo predicato vengono eliminate. I file all'interno del periodo di conservazione in queste partizioni non vengono eliminati. Impostato su `""`: vuoto per impostazione predefinita.
- `AdditionalOptions`: un oggetto S3 SourceAdditionalOptions.

  Specifica opzioni di connessione aggiuntive.

## AggregateOperation struttura

Specifica il set di parametri necessari per eseguire l'aggregazione nella trasformazione di aggregazione.

Campi

- `Column` - obbligatorio: una matrice di stringhe UTF-8.

  Specifica la colonna sul set di dati su cui verrà applicata la funzione di aggregazione.
- `AggFunc` - obbligatorio: stringa UTF-8 (valori validi:`avg` | `countDistinct` | `count` | `first` | `last` | `kurtosis` | `max` | `min` | `skewness` | `stddev_samp` | `stddev_pop` | `sum` | `sumDistinct` | `var_samp` | `var_pop`).

  Specifica la funzione di aggregazione da applicare.

Le possibili funzioni di aggregazione includono: avg countDistinct, count, first, last, kurtosis, max, min, skewness, stddev_samp, stddev_pop, sum, sumDistinct, var_samp, var_pop

## GlueSchema struttura

Specifica uno schema definito dall'utente quando uno schema non può essere determinato da AWS Glue.

Campi

- `Columns`: una matrice di oggetti GlueStudioSchemaColumn.

  Specifica le definizioni delle colonne che compongono uno AWS Glue schema.

## GlueStudioSchemaColumn struttura

Specifica una singola colonna in una definizione AWS Glue dello schema.

Campi

- `Name` - obbligatorio: stringa UTF-8, non più lunga di 1024 byte, corrispondente al Single-line string pattern.

  Il nome della colonna nello schema di AWS Glue Studio.
- `Type`: stringa UTF-8, non superiore a 131072 byte di lunghezza, corrispondente a Single-line string pattern.

  Il tipo di hive per questa colonna nello schema di AWS Glue Studio.
- `GlueStudioType`: stringa UTF-8, non superiore a 131072 byte di lunghezza, corrispondente a Single-line string pattern.

  Il tipo di dati della colonna come definito in AWS Glue Studio.

## GlueStudioColumn struttura

Specifica una singola colonna in AWS Glue Studio.

Campi

- `Key` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  La chiave della colonna in AWS Glue Studio.

- `FullPath` - obbligatorio: una matrice di stringhe UTF-8.

  TThe URL completo della colonna in AWS Glue Studio.

- `Type` - obbligatorio: stringa UTF-8 (valori validi: `array="ARRAY" | bigint="BIGINT" | bigint array="BIGINT_ARRAY" | binary="BINARY" | binary array="BINARY_ARRAY" | boolean="BOOLEAN" | boolean array="BOOLEAN_ARRAY" | byte="BYTE" | byte array="BYTE_ARRAY" | char="CHAR" | char array="CHAR_ARRAY" | choice="CHOICE" | choice array="CHOICE_ARRAY" | date="DATE" | date array="DATE_ARRAY" | decimal="DECIMAL" | decimal array="DECIMAL_ARRAY" | double="DOUBLE" | double array="DOUBLE_ARRAY" | enum="ENUM" | enum array="ENUM_ARRAY" | float="FLOAT" | float array="FLOAT_ARRAY" | int="INT" | int array="INT_ARRAY" | interval="INTERVAL" | interval array="INTERVAL_ARRAY" | long="LONG" | long array="LONG_ARRAY" | object="OBJECT" | short="SHORT" | short array="SHORT_ARRAY" | smallint="SMALLINT" | smallint array="SMALLINT_ARRAY" | string="STRING" | string array="STRING_ARRAY" | timestamp="TIMESTAMP" | timestamp array="TIMESTAMP_ARRAY" | tinyint="TINYINT" | tinyint array="TINYINT_ARRAY" | varchar="VARCHAR" | varchar array="VARCHAR_ARRAY" | null="NULL" | unknown="UNKNOWN" | unknown array="UNKNOWN_ARRAY"`).

  TThe tipo di colonna in AWS Glue Studio.

- `Children`: un array di strutture.

  TThe figli della colonna principale in AWS Glue Studio.

- `GlueStudioType`: stringa UTF-8 (valori validi: `array="ARRAY" | bigint="BIGINT" | bigint array="BIGINT_ARRAY" | binary="BINARY" | binary array="BINARY_ARRAY" | boolean="BOOLEAN" | boolean array="BOOLEAN_ARRAY" | byte="BYTE" | byte array="BYTE_ARRAY" | char="CHAR" | char array="CHAR_ARRAY" | choice="CHOICE" | choice array="CHOICE_ARRAY" | date="DATE" | date array="DATE_ARRAY" | decimal="DECIMAL" | decimal array="DECIMAL_ARRAY" | double="DOUBLE" | double array="DOUBLE_ARRAY" | enum="ENUM" | enum array="ENUM_ARRAY" | float="FLOAT" | float array="FLOAT_ARRAY" | int="INT" | int array="INT_ARRAY" | interval="INTERVAL" | interval array="INTERVAL_ARRAY" | long="LONG"`

| long array="LONG_ARRAY" | object="OBJECT" | short="SHORT" | short array="SHORT_ARRAY" | smallint="SMALLINT" | smallint array="SMALLINT_ARRAY" | string="STRING" | string array="STRING_ARRAY" | timestamp="TIMESTAMP" | timestamp array="TIMESTAMP_ARRAY" | tinyint="TINYINT" | tinyint array="TINYINT_ARRAY" | varchar="VARCHAR" | varchar array="VARCHAR_ARRAY" | null="NULL" | unknown="UNKNOWN" | unknown array="UNKNOWN_ARRAY").

Il tipo di dati della colonna come definito in AWS Glue Studio.

## DynamicTransform struttura

Specifica il set di parametri necessari per eseguire la trasformazione dinamica.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Specifica il nome della trasformazione dinamica.
- `TransformName` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Speciifica il nome della trasformazione dinamica così come appare nell'editor visivo di AWS Glue Studio.
- `Inputs` - obbligatorio: una matrice di stringhe UTF-8, non inferiore o superiore a 1 stringa.

  Specifica gli input necessari per la trasformazione dinamica.
- `Parameters`: una matrice di oggetti TransformConfigParameter.

  Specifica i parametri della trasformazione dinamica.
- `FunctionName` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Specifica il nome della funzione della trasformazione dinamica.
- `Path` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Specifica il percorso dei file sorgente e di configurazione della trasformazione dinamica.
- `Version`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Questo campo non è utilizzato e verrà dichiarato obsoleto in una versione futura.
- `OutputSchemas`: una matrice di oggetti GlueSchema.

Specifica lo schema di dati per la trasformazione dinamica.

## TransformConfigParameter struttura

Specifica i parametri nel file di configurazione della trasformazione dinamica.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Specifica il nome del parametro nel file di configurazione della trasformazione dinamica.

- `Type` - obbligatorio: stringa UTF-8 (valori validi: `str="STR"` | `int="INT"` | `float="FLOAT"` | `complex="COMPLEX"` | `bool="BOOL"` | `list="LIST"` | `null="NULL"`).

  Specifica il tipo di parametro nel file di configurazione della trasformazione dinamica.

- `ValidationRule`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Specifica la regola di convalida nel file di configurazione della trasformazione dinamica.

- `ValidationMessage`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Specifica il messaggio di convalida nel file di configurazione della trasformazione dinamica.

- `Value`: una matrice di stringhe UTF-8.

  Specifica il valore del parametro nel file di configurazione della trasformazione dinamica.

- `ListType`: stringa UTF-8 (valori validi: `str="STR"` | `int="INT"` | `float="FLOAT"` | `complex="COMPLEX"` | `bool="BOOL"` | `list="LIST"` | `null="NULL"`).

  Specifica il tipo di elenco del parametro nel file di configurazione della trasformazione dinamica.

- `IsOptional`: booleano.

  Specifica se il parametro è facoltativo o meno nel file di configurazione della trasformazione dinamica.

## EvaluateDataQuality struttura

Specifica i criteri di valutazione della qualità dei dati.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome della valutazione della qualità dei dati.
- `Inputs` - obbligatorio: una matrice di stringhe UTF-8, non inferiore o superiore a 1 stringa.

  Gli input della valutazione della qualità dei dati.
- `Ruleset` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 65.536 byte di lunghezza, corrispondente a Custom string pattern #59.

  Il set di regole per la valutazione della qualità dei dati.
- `Output`: stringa UTF-8 (valori validi: `PrimaryInput` | `EvaluationResults`).

  L'output della valutazione della qualità dei dati.
- `PublishingOptions`: un oggetto DQResultsPublishingOptions.

  Opzioni per configurare la modalità di pubblicazione dei risultati.
- `StopJobOnFailureOptions`: un oggetto DQStopJobOnFailureOptions.

  Opzioni per configurare come si interromperà il processo se la valutazione della qualità dei dati fallisce.

## DQResultsPublishingOptions struttura

Opzioni per configurare la modalità di pubblicazione dei risultati della valutazione della qualità dei dati.

Campi

- `EvaluationContext`: stringa UTF-8, corrispondente a Custom string pattern #60.

  Il contesto della valutazione.
- `ResultsS3Prefix`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il prefisso Amazon S3 aggiunto all'inizio dei risultati.
- `CloudWatchMetricsEnabled`: booleano.

  Abilita i parametri per i risultati della qualità dei dati.

- `ResultsPublishingEnabled`: booleano.

  Abilita la pubblicazione per i risultati della qualità dei dati.

## DQStopJobOnFailureOptions struttura

Opzioni per configurare come si interromperà il processo se la valutazione della qualità dei dati fallisce.

Campi

- `StopJobOnFailureTiming`: stringa UTF-8 (valori validi: `Immediate` | `AfterDataLoad`).

  Quando interrompere il processo se la valutazione della qualità dei dati fallisce. Le opzioni sono Immediate o AfterDataLoad.

## EvaluateDataQualityMultiFrame struttura

Specifica i criteri di valutazione della qualità dei dati.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome della valutazione della qualità dei dati.
- `Inputs` - obbligatorio: una matrice di stringhe UTF-8, almeno 1 stringa.

  Gli input della valutazione della qualità dei dati. Il primo input in questo elenco è l'origine dati primaria.
- `AdditionalDataSources`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8 corrispondente al Custom string pattern #63.

  Ogni valore è una stringa UTF-8 corrispondente al Custom string pattern #61.

  Gli alias di tutte le origini dati, tranne quella primaria.
- `Ruleset` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 65.536 byte di lunghezza, corrispondente a Custom string pattern #59.

  Il set di regole per la valutazione della qualità dei dati.

- `PublishingOptions`: un oggetto DQResultsPublishingOptions.

  Opzioni per configurare la modalità di pubblicazione dei risultati.
- `AdditionalOptions`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8 (valori validi: `performanceTuning.caching="CacheOption" | observations.scope="ObservationsOption" | compositeRuleEvaluation.method="CompositeOption"`).

  Ogni valore è una stringa UTF-8.

  Opzioni per configurare il comportamento di runtime della trasformazione.
- `StopJobOnFailureOptions`: un oggetto DQStopJobOnFailureOptions.

  Opzioni per configurare come si interromperà il processo se la valutazione della qualità dei dati fallisce.

## Struttura Recipe

Un nodo AWS Glue Studio che utilizza una AWS Glue DataBrew ricetta nei AWS Glue lavori.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome del nodo AWS Glue Studio.
- `Inputs` - obbligatorio: una matrice di stringhe UTF-8, non inferiore o superiore a 1 stringa.

  I nodi che costituiscono gli input del nodo della ricetta, identificati dal rispettivo ID.
- `RecipeReference`: un oggetto RecipeReference.

  Un riferimento alla DataBrew ricetta usata dal nodo.
- `RecipeSteps`: una matrice di oggetti RecipeStep.

  Passaggi di trasformazione utilizzati nel nodo della ricetta.

## RecipeReference struttura

Un riferimento a una AWS Glue DataBrew ricetta.

Campi

- `RecipeArn` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  L'ARN della ricetta. DataBrew
- `RecipeVersion` - obbligatorio: stringa UTF-8, lunghezza non inferiore a 1 o non superiore a 16 byte.

  L' RecipeVersion origine della DataBrew ricetta.

## SnowflakeNodeData struttura

Specifica la configurazione per i nodi Snowflake in Studio. AWS Glue

Campi

- `SourceType`: stringa UTF-8, corrispondente a Custom string pattern #60.

  Specifica come vengono specificati i dati recuperati. Valori validi: `"table"`, `"query"`.
- `Connection`: un oggetto Opzione.

  Specifica una connessione al catalogo AWS Glue dati a un endpoint Snowflake.
- `Schema`: stringa UTF-8.

  Specifica uno schema di database Snowflake da utilizzare per il nodo.
- `Table`: stringa UTF-8.

  Specifica una tabella Snowflake da utilizzare per il nodo.
- `Database`: stringa UTF-8.

  Specifica un database Snowflake da utilizzare per il nodo.
- `TempDir`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Attualmente non utilizzato.
- `IamRole`: un oggetto Opzione.

  Attualmente non utilizzato.
- `AdditionalOptions`: una matrice della mappa di coppie chiave-valore.

Ogni chiave è una stringa UTF-8 corrispondente al Custom string pattern #61.

Ogni valore è una stringa UTF-8 corrispondente al Custom string pattern #61.

Specifica le opzioni aggiuntive trasmesse al connettore Snowflake. Se altre opzioni sono specificate altrove in questo nodo, esse avranno la precedenza.

- `SampleQuery`: stringa UTF-8.

  Una stringa SQL utilizzata per recuperare i dati con il tipo di origine `query`.

- `PreAction`: stringa UTF-8.

  Una stringa SQL eseguita prima che il connettore Snowflake esegua le operazioni standard.

- `PostAction`: stringa UTF-8.

  Una stringa SQL eseguita dopo che il connettore Snowflake esegua le operazioni standard.

- `Action`: stringa UTF-8.

  Specifica l'operazione da intraprendere quando si scrive su una tabella con dati preesistenti. Valori validi: `append`, `merge`, `truncate`, `drop`.

- `Upsert`: booleano.

  Utilizzato quando Operazione è `append`. Specifica il comportamento di risoluzione quando esiste già una riga. Se impostato su true, le righe preesistenti verranno aggiornate. Se false, verranno inserite quelle righe.

- `MergeAction`: stringa UTF-8, corrispondente a Custom string pattern #60.

  Specifica un'operazione di unione. Valori validi: `simple`, `custom`. Se semplice, il comportamento di unione è definito da `MergeWhenMatched` e `MergeWhenNotMatched`. Se personalizzato, definito da `MergeClause`.

- `MergeWhenMatched`: stringa UTF-8, corrispondente a Custom string pattern #60.

  Specifica come risolvere i record che corrispondono a dati preesistenti durante l'unione. Valori validi: `update`, `delete`.

- `MergeWhenNotMatched`: stringa UTF-8, corrispondente a Custom string pattern #60.

  Specifica come elaborare i record che non corrispondono a dati preesistenti durante l'unione. Valori validi: `insert`, `none`.

- `MergeClause`: stringa UTF-8.

  Un'istruzione SQL che specifica un comportamento di merge personalizzato.

- `StagingTable`: stringa UTF-8.

  Il nome di una tabella intermedia utilizzata durante le operazioni `merge` o `append` con upsert. I dati vengono scritti in questa tabella, quindi spostati in `table` da un'azione successiva (PostAction) generata.

- `SelectedColumns`: una matrice di oggetti Opzione.

  Specifica le colonne combinate per identificare un record quando vengono rilevate corrispondenze per i merge e gli upsert. Un elenco di strutture con chiavi `value`, `label` e `description`. Ogni struttura descrive una colonna.

- `AutoPushdown`: booleano.

  Specifica se il pushdown automatico delle query è abilitato. Se il pushdown è abilitato, quando su Spark viene eseguita una query, se una parte di essa può essere "trasferita" al server Snowflake, viene sottoposta a pushdown. Ciò migliora le prestazioni di alcune query.

- `TableSchema`: una matrice di oggetti Opzione.

  Definisce manualmente lo schema di destinazione per il nodo. Un elenco di strutture con chiavi `value`, `label` e `description`. Ogni struttura definisce una colonna.

## SnowflakeSource struttura

Specifica un'origine dati Snowflake.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome dell'origine dati Snowflake.

- `Data` - obbligatorio: un oggetto SnowflakeNodeData.

  Configurazione per l'origine dati Snowflake.

- `OutputSchemas`: una matrice di oggetti GlueSchema.

  Specifica gli schemi definiti dall'utente per i dati di output.

## SnowflakeTarget struttura

Specifica una destinazione Snowflake.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome della destinazione Snowflake.
- `Data` - obbligatorio: un oggetto SnowflakeNodeData.

  Specifica i dati del nodo di destinazione Snowflake.
- `Inputs`: un array di stringhe UTF-8, non inferiore o superiore a 1 stringa.

  I nodi che sono input per la destinazione di dati.

## ConnectorDataSource struttura

Specifica un'origine generata con opzioni di connessione standard.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome del nodo di origine.
- `ConnectionType` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il`connectionType`, come fornito alla AWS Glue libreria sottostante. Il tipo di nodo supporta i tipi di connessione seguenti:
  - `opensearch`
  - `azuresql`
  - `azurecosmos`
  - `bigquery`
  - `saphana`
  - `teradata`
  - `vertica`
- `Data` - obbligatorio: una matrice della mappa di coppie chiave-valore.

Ogni chiave è una stringa UTF-8.

Ogni valore è una stringa UTF-8.

Una mappa che specifica le opzioni di connessione per il nodo. È possibile trovare le opzioni di connessione standard per il tipo di connessione corrispondente nella sezione Parametri di connessione della AWS Glue documentazione.

- `OutputSchemas`: una matrice di oggetti GlueSchema.

  Specifica lo schema di dati per questa origine.

## ConnectorDataTarget struttura

Specifica un a destinazione generata con opzioni di connessione standard.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome del nodo di destinazione.

- `ConnectionType` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il`ConnectionType`, come fornito alla AWS Glue libreria sottostante. Il tipo di nodo supporta i tipi di connessione seguenti:

  - `opensearch`
  - `azuresql`
  - `azurecosmos`
  - `bigquery`
  - `saphana`
  - `teradata`
  - `vertica`

- `Data` - obbligatorio: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8.

  Ogni valore è una stringa UTF-8.

Una mappa che specifica le opzioni di connessione per il nodo. È possibile trovare le opzioni di connessione standard per il tipo di connessione corrispondente nella sezione Parametri di connessione della AWS Glue documentazione.

- `Inputs`: un array di stringhe UTF-8, non inferiore o superiore a 1 stringa.

  I nodi che sono input per la destinazione di dati.

## RecipeStep struttura

Una fase della ricetta utilizzata in un nodo di ricetta per la preparazione dei dati di AWS Glue Studio.

Campi

- `Action` - obbligatorio: un oggetto RecipeAction.

  L'operazione di trasformazione del passaggio della ricetta.
- `ConditionExpressions`: una matrice di oggetti ConditionExpression.

  Le espressioni di condizione per il passaggio della ricetta.

## RecipeAction struttura

Azioni definite nel nodo della ricetta di preparazione dei dati di AWS Glue Studio.

Campi

- `Operation` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza, corrispondente a Custom string pattern #56.

  Il funzionamento dell'operazione della ricetta.
- `Parameters`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza, corrispondente a Custom string pattern #57.

  Ogni valore è una stringa UTF-8, non inferiore a 1 o superiore a 32768 byte di lunghezza.

  I parametri dell'operazione della ricetta.

## ConditionExpression struttura

Espressione della condizione definita nel nodo della ricetta di preparazione dei dati di AWS Glue Studio.

Campi

- `Condition` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza, corrispondente a Custom string pattern #56.

  La condizione dell'espressione di condizione.
- `Value`: stringa UTF-8, non superiore a 1.024 byte di lunghezza.

  Il valore dell'espressione di condizione.
- `TargetColumn` - obbligatorio: stringa UTF-8, lunghezza non inferiore a 1 o non superiore a 1024 byte.

  La colonna di destinazione delle espressioni di condizione.

## Struttura S3 CatalogIcebergSource

Speciifica un'origine dati Apache Iceberg registrata nel Data Catalog. AWS Glue L'origine dati Iceberg deve essere archiviata in. Amazon S3

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome dell'origine dati Iceberg.
- `Database` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome del database da cui leggere.
- `Table` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome della tabella nel database da cui leggere.
- `AdditionalIcebergOptions`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8 corrispondente al Custom string pattern #61.

  Ogni valore è una stringa UTF-8 corrispondente al Custom string pattern #61.

Specifica opzioni di connessione aggiuntive per l'origine dati Iceberg.

- `OutputSchemas`: una matrice di oggetti GlueSchema.

  Specifica lo schema di dati per l'origine Iceberg.

## CatalogIcebergSource struttura

Speciifica un'origine dati Apache Iceberg registrata nel Data Catalog. AWS Glue

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome dell'origine dati Iceberg.
- `Database` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome del database da cui leggere.
- `Table` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome della tabella nel database da cui leggere.
- `AdditionalIcebergOptions`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8 corrispondente al Custom string pattern #61.

  Ogni valore è una stringa UTF-8 corrispondente al Custom string pattern #61.

  Specifica opzioni di connessione aggiuntive per l'origine dati Iceberg.
- `OutputSchemas`: una matrice di oggetti GlueSchema.

  Specifica lo schema di dati per l'origine Iceberg.

## Struttura S3 IcebergCatalogTarget

Speciifica un target di catalogo Apache Iceberg che scrive dati Amazon S3 e registra la tabella nel Data Catalog. AWS Glue

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

Il nome del catalogo Iceberg di destinazione.

- `Inputs` - obbligatorio: una matrice di stringhe UTF-8, non inferiore o superiore a 1 stringa.

  La connessione di input del catalogo Iceberg di destinazione.

- `PartitionKeys`: una matrice di stringhe UTF-8.

  Un elenco di chiavi di partizione per la tabella Iceberg.

- `Table` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome della tabella da scrivere nel catalogo.

- `Database` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome del database in cui scrivere.

- `AdditionalOptions`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8 corrispondente al Custom string pattern #61.

  Ogni valore è una stringa UTF-8 corrispondente al Custom string pattern #61.

  Specifica opzioni di connessione aggiuntive per il catalogo Iceberg di destinazione.

- `SchemaChangePolicy`: un oggetto CatalogSchemaChangePolicy.

  La policy per la gestione delle modifiche allo schema nel catalogo di destinazione.

- `AutoDataQuality`: un oggetto AutoDataQuality.

  Specifica se abilitare automaticamente la valutazione della qualità dei dati per il catalogo S3 Iceberg di destinazione. Se impostato su `true`, i controlli di qualità dei dati vengono eseguiti automaticamente durante l'operazione di scrittura.

## DBELTConnectorStruttura Dynamo Source

Specifica un connettore ELT DynamoDB di origine per l'estrazione dei dati dalle tabelle DynamoDB.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome del connettore ELT DynamoDB di origine.

- `ConnectionOptions`: un oggetto DDBELTConnectionOpzioni.

  Le opzioni di connessione per il connettore DynamoDB ELT di origine.

- `OutputSchemas`: una matrice di oggetti GlueSchema.

  Specifica lo schema di dati per il connettore DynamoDB ELT di origine.

## DDBELTConnectionStruttura delle opzioni

Specifica le opzioni di connessione per le operazioni DynamoDB ELT (estrazione, caricamento, trasformazione). Questa struttura contiene parametri di configurazione per la connessione a e l'estrazione di dati da tabelle DynamoDB utilizzando il connettore ELT.

Campi

- `DynamodbExport`: stringa UTF-8 (valori validi: `ddb` | `s3`).

  Specifica il tipo di esportazione per l'estrazione di dati DynamoDB. Questo parametro determina il modo di cui i dati vengono esportati dalla tabella DynamoDB durante il processo ELT.

- `DynamodbUnnestDDBJson`: booleano.

  Un valore booleano che specifica se annullare la nidificazione del formato DynamoDB JSON durante l'estrazione dei dati. Se impostato su `true`, il connettore appiattirà le strutture JSON annidate negli elementi DynamoDB. Se impostato su `false`, la struttura JSON DynamoDB originale viene preservata.

- `DynamodbTableArn` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome della risorsa Amazon (ARN) della tabella DynamoDB da cui estrarre i dati. Questo parametro specifica la tabella di origine per l'operazione ELT.

- `DynamodbS3Bucket`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il nome del bucket Amazon S3 utilizzato per l'archiviazione intermedia durante il processo ELT DynamoDB. Questo bucket viene utilizzato per archiviare temporaneamente i dati DynamoDB esportati prima che vengano elaborati dal processo ELT.

- `DynamodbS3Prefix`: stringa UTF-8, corrispondente a Custom string pattern #61.

Il prefisso della chiave dell'oggetto S3 per i file archiviati nel bucket S3 intermedio durante il processo ELT DynamoDB. Questo prefisso aiuta a organizzare e identificare i file temporanei creati durante l'estrazione dei dati.

- `DynamodbS3BucketOwner`: stringa UTF-8, corrispondente a Custom string pattern #61.

  L'ID dell' AWS account del proprietario del bucket S3 specificato in. `DynamodbS3Bucket` Questo parametro è richiesto quando il bucket S3 è di proprietà di un AWS account diverso da quello che esegue il job ELT e consente l'accesso tra account diversi al bucket di archiviazione intermedio.
- `DynamodbStsRoleArn`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Il ruolo Amazon Resource Name (ARN) del AWS Security Token Service (STS) da assumere per l'accesso alle risorse DynamoDB e S3 durante l'operazione ELT. Questo ruolo deve disporre delle autorizzazioni necessarie per leggere dalla tabella DynamoDB e scrivere nel bucket S3 intermedio.

## DDBELTCatalogAdditionalOptions struttura

Specifica opzioni aggiuntive per le operazioni di catalogo ELT DynamoDB.

Campi

- `DynamodbExport`: stringa UTF-8, corrispondente a Custom string pattern #61.

  Specifica la configurazione di esportazione DynamoDB per l'operazione ETL.
- `DynamodbUnnestDDBJson`: booleano.

  Specifica se annullare la nidificazione del formato JSON DynamoDB. Se impostato su `true`, le strutture JSON annidate negli elementi DynamoDB vengono appiattite.

## Struttura Route

Specifica un nodo di routing che indirizza i dati verso diversi percorsi di output in base a condizioni di filtraggio definite.

Campi

- `Name` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #63.

  Il nome del nodo di routing.

- `Inputs` - obbligatorio: una matrice di stringhe UTF-8, non inferiore o superiore a 1 stringa.

  La connessione di ingresso del nodo di routing.
- `GroupFiltersList` - obbligatorio: una matrice di oggetti GroupFilters.

  Un elenco di filtri di gruppo che definiscono le condizioni di routing e i criteri per l'indirizzamento dei dati in diversi percorsi di output.

## GroupFilters struttura

Specifica un gruppo di filtri con un operatore logico che stabilisce come i filtri vengono combinati per valutare le condizioni di routing.

Campi

- `GroupName` - obbligatorio: stringa UTF-8, corrispondente a Custom string pattern #60.

  Il nome del gruppo di filtri.
- `Filters` - obbligatorio: una matrice di oggetti FilterExpression.

  Un elenco di espressioni di filtro che definiscono le condizioni per questo gruppo.
- `LogicalOperator` - obbligatorio: stringa UTF-8 (valori validi: `AND` | `OR`).

  L'operatore logico utilizzato per combinare i filtri in questo gruppo. Determina se tutti i filtri devono corrispondere (AND) o può corrispondere uno qualsiasi dei filtri (OR).

## AutoDataQuality struttura

Speciifica le opzioni di configurazione per la valutazione automatica della qualità dei dati nei AWS Glue lavori. Questa struttura abilita controlli e monitoraggio automatizzati della qualità dei dati durante le operazioni ETL, contribuendo a garantire l'integrità e l'affidabilità dei dati senza interventi manuali.

Campi

- `IsEnabled`: booleano.

  Specifica se è abilitata la valutazione automatica della qualità dei dati. Se impostato su `true`, i controlli di qualità dei dati vengono eseguiti automaticamente.
- `EvaluationContext`: stringa UTF-8, corrispondente a Custom string pattern #61.

Il contesto di valutazione per i controlli di qualità dei dati automatici. Definisce l'ambito e i parametri per la valutazione della qualità dei dati.

# API dei processi

L'API Jobs descrive i tipi di dati relativi APIs ai job e contiene informazioni su come utilizzare job, job run e trigger. AWS Glue

Argomenti

- Jobs
- Esecuzioni di processi
- Triggers

## Jobs

L'API Jobs descrive i tipi di dati e l'API relativi alla creazione, all'aggiornamento, all'eliminazione o alla visualizzazione di lavori in. AWS Glue

### Tipi di dati

- Struttura del processo
- ExecutionProperty struttura
- NotificationProperty struttura
- JobCommand struttura
- ConnectionsList struttura
- JobUpdate struttura
- SourceControlDetails struttura

### Struttura del processo

Specifica una definizione del processo.

Campi

- `Name`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome assegnato alla definizione del processo.

- `JobMode`: stringa UTF-8 (valori validi: `SCRIPT=""` | `VISUAL=""` | `NOTEBOOK=""`).

  Una modalità che descrive come è stato creato un processo. I valori validi sono:

  - `SCRIPT`- Il lavoro è stato creato utilizzando l'editor di script AWS Glue Studio.
  - `VISUAL`- Il lavoro è stato creato utilizzando l'editor visivo di AWS Glue Studio.
  - `NOTEBOOK`: il processo è stato creato utilizzando un notebook con sessioni interattive.

  Quando il campo `JobMode` è mancante o null, viene assegnato `SCRIPT` come valore predefinito.

- `JobRunQueuingEnabled`: booleano.

  Specifica se l'accodamento dell'esecuzione dei processi è abilitato per le esecuzioni di questo processo.

  Il valore true indica che l'accodamento delle esecuzioni di processi è abilitato. Se il valore è false o non popolato, le esecuzioni del processo non saranno prese in considerazione per l'accodamento.

  Se questo campo non corrisponde al set di valori impostato nell'esecuzione del processo, verrà utilizzato il valore presente nel campo di esecuzione del processo.

- `Description`: stringa di descrizione, non superiore a 2048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  Descrizione del processo.

- `LogUri`: stringa UTF-8.

  Questo campo è riservato per uso futuro.

- `Role`: stringa UTF-8.

  Il nome o ARN (Amazon Resource Name) del ruolo IAM associato a questo processo.

- `CreatedOn`: timestamp.

  La data e l'ora in cui è stata creata la specifica della definizione del processo.

- `LastModifiedOn`: timestamp.

L'ultimo point-in-time in cui è stata modificata la definizione del processo.

- `ExecutionProperty`: un oggetto ExecutionProperty.

  `ExecutionProperty` che specifica il numero massimo di esecuzioni simultanee consentite per il processo.

- `Command`: un oggetto JobCommand.

  Il `JobCommand` che esegue questo lavoro.

- `DefaultArguments`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8.

  Ogni valore è una stringa UTF-8.

  Gli argomenti predefiniti per ciascuna esecuzione del processo, specificati come coppie nome-valore.

  Qui è possibile specificare gli argomenti utilizzati dal proprio script di esecuzione del lavoro, nonché gli argomenti utilizzati dal proprio script di esecuzione AWS Glue del lavoro.

  Gli argomenti del processo potrebbero essere registrati. Non passare segreti in testo chiaro come argomenti. Recupera i segreti da una AWS Glue connessione Gestione dei segreti AWS o da un altro meccanismo di gestione dei segreti se intendi mantenerli all'interno del Job.

  Per informazioni su come specificare e utilizzare i propri argomenti Job, consultate l'argomento Calling AWS Glue APIs in Python nella guida per sviluppatori.

  Per informazioni sugli argomenti che puoi fornire a questo campo durante la configurazione dei processi Spark, consulta la pagina Special Parameters Used by AWS Glue nella Guida per gli sviluppatori.

  Per informazioni sugli argomenti che puoi fornire a questo campo durante la configurazione dei processi Ray, consulta la pagina Using job parameters in Ray jobs nella Guida per gli sviluppatori.

- `NonOverridableArguments`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8.

  Ogni valore è una stringa UTF-8.

Gli argomenti per questo processo che non vengono sovrascritti quando si forniscono argomenti di processo in un'esecuzione di processo, specificati come coppie nome-valore.

- `Connections`: un oggetto ConnectionsList.

  Le connessioni utilizzate per questo processo.

- `MaxRetries`: numero (intero).

  Il numero massimo di volte in cui è possibile riprovare questo processo dopo un JobRun errore.

- `AllocatedCapacity`: numero (intero).

  in quanto obsoleto. Usare invece `MaxCapacity`.

  Il numero di unità di elaborazione AWS Glue dati (DPUs) assegnate alle esecuzioni di questo processo. È possibile allocarne almeno 2 DPUs; l'impostazione predefinita è 10. Una DPU è una misura relativa della potenza di elaborazione costituita da 4 V di capacità CPUs di elaborazione e 16 GB di memoria. Per ulteriori informazioni, consultare la pagina dei prezzi di AWS Glue.

- `Timeout`: numero (intero), almeno 1.

  Timeout del processo in minuti. Indica il tempo massimo durante cui l'esecuzione di un processo può utilizzare le risorse prima di essere terminata e passare allo stato `TIMEOUT`.

  I processi devono avere valori di timeout inferiori a 7 giorni o 10080 minuti. In caso contrario, i processi genereranno un'eccezione.

  Quando il valore viene lasciato vuoto, il timeout è predefinito a 2.880 minuti per Glue versione 4.0 e precedenti o 480 minuti per Glue versione 5.0 e successive.

  Tutti i AWS Glue lavori esistenti con un valore di timeout superiore a 7 giorni verranno impostati come impostazione predefinita su 7 giorni. Ad esempio, se un processo in batch ha impostato un timeout di 20 giorni, sarà interrotto al settimo giorno.

  Per i processi di streaming, se è stata impostata una finestra di manutenzione, il processo sarà riavviato durante tale finestra dopo 7 giorni.

- `MaxCapacity`: numero (doppio).

  Per i job Glue versione 1.0 o precedente, utilizzando il tipo di worker standard, il numero di unità di elaborazione AWS Glue dati (DPUs) che possono essere allocate durante l'esecuzione di

questo processo. Una DPU è una misura relativa della potenza di elaborazione costituita da 4 V di capacità CPUs di elaborazione e 16 GB di memoria. Per ulteriori informazioni, consultare la pagina dei prezzi di AWS Glue.

Per i processi Glue versione 2.0 e successive, non è possibile specificare il valore `Maximum capacity`. Si deve invece specificare un `Worker type` e un `Number of workers`.

Non impostare `MaxCapacity` se usi `WorkerType` e `NumberOfWorkers`.

Il valore che è possibile allocare per `MaxCapacity` varia a seconda che si esegua un processo shell di Python, un processo ETL di Apache Spark o un processo ETL di streaming di Apache Spark:

- Quando si specifica un processo shell di Python (`JobCommand.Name`="pythonshell"), è possibile allocare 0,0625 o 1 DPU. Il valore di default è 0,0625 DPU.
- Quando specificate un job ETL di Apache Spark (`JobCommand.Name`="glueetl») o un job ETL di streaming Apache Spark (="gluestreaming»), potete allocare da 2 a 100. `JobCommand.Name` DPUs L'impostazione predefinita è 10. DPUs Questo tipo di processo non può avere un'allocazione DPU frazionata.

- `WorkerType`— Stringa UTF-8 (valori validi: `Standard=""` | `G.025X=""` | `G.1X=""` | `G.2X=""` | `G.4X=""` | `G.8X=""` | `G.12X=""` | `G.16X=""` | `R.1X=""` | | `R.2X=""` `R.4X=""` `R.8X=""` |`Z.2X=""`).

  Il tipo di worker predefinito allocato quando viene eseguito un processo.

  AWS Glue fornisce diversi tipi di lavoratori per soddisfare diversi requisiti di carico di lavoro:

  Worker di tipo G (worker di calcolo per uso generico):

  - G.025X: 0,25 DPU (2 vCPUs, 4 GB di memoria, disco da 84 GB)
  - G.1X: 1 DPU (4 v, 16 GB di memoria, disco da 94 GB) CPUs
  - G.2X: 2 DPU (8 vCPUs, 32 GB di memoria, disco da 138 GB)
  - G.4X: 4 DPU (16 vCPUs, 64 GB di memoria, disco da 256 GB)
  - G.8X: 8 DPU (32 vCPUs, 128 GB di memoria, disco da 512 GB)
  - G.12X: 12 DPU (48 vCPUs, 192 GB di memoria, disco da 768 GB)
  - G.16X: 16 DPU (64 vCPUs, 256 GB di memoria, disco da 1024 GB)

  Worker di tipo R (worker ottimizzati per la memoria):

  - R.1X: 1 DPU (4 vCPUs, 32 GB di memoria, disco da 94 GB)

  - R.2X: 2 DPU (8 vCPUs, 64 GB di memoria, disco da 128 GB)
  - R.4X: 4 DPU (16 vCPUs, 128 GB di memoria, disco da 256 GB)
  - R.8X: 8 DPU (32 vCPUs, 256 GB di memoria, disco da 512 GB)
- `NumberOfWorkers`: numero (intero).

  Il numero di worker di un `workerType` specifico allocati quando viene eseguito un processo.
- `SecurityConfiguration`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome della struttura `SecurityConfiguration` da usare con questo processo.
- `NotificationProperty`: un oggetto NotificationProperty.

  Specifica le proprietà di configurazione di una notifica di processo.
- `Running`: booleano.

  Questo campo è riservato per uso futuro.
- `GlueVersion`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Custom string pattern #48.

  Nei job Spark, `GlueVersion` determina le versioni di Apache Spark e Python disponibili in un job. AWS Glue La versione Python indica la versione supportata per i processi di tipo Spark.

  I processi Ray devono impostare il valore di `GlueVersion` su `4.0` o superiore. Tuttavia, le versioni di Ray, Python e le librerie aggiuntive disponibili nel processo Ray sono determinate dal parametro `Runtime` del comando del processo.

  Per ulteriori informazioni sulle AWS Glue versioni disponibili e sulle versioni corrispondenti di Spark e Python, consulta la versione Glue nella guida per sviluppatori.

  I lavori creati senza specificare una versione di Glue predefinita è Glue 5.1.
- `CodeGenConfigurationNodes`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8 corrispondente al Custom string pattern #60.

  Ogni valore è un oggetto CodeGenConfigurationNode.

  La rappresentazione di un grafico aciclico diretto su cui si basano sia il componente visivo che la generazione di codice di Glue Studio.

- `ExecutionClass`: una stringa UTF-8, non superiore a 16 byte di lunghezza (valori validi: `FLEX=""` | `STANDARD=""`).

  Indica se il processo viene eseguito con una classe di esecuzione standard o flessibile. La classe di esecuzione standard è ideale per carichi di lavoro sensibili al tempo che richiedono un avvio rapido dei processi e risorse dedicate.

  La classe di esecuzione flessibile è appropriata per i processi non sensibili al tempo i cui tempi di inizio e completamento possono variare.

  Solo i lavori con AWS Glue versione 3.0 e successive e il tipo di comando `glueetl` potranno essere impostati su`ExecutionClass`. `FLEX` La classe di esecuzione flessibile è disponibile per i processi Spark.

- `SourceControlDetails`: un oggetto SourceControlDetails.

  I dettagli per una configurazione di controllo di origine per un processo, che consente la sincronizzazione degli artefatti del processo da o verso un repository remoto.

- `MaintenanceWindow`: stringa UTF-8, corrispondente a Custom string pattern #34.

  Questo campo specifica un giorno della settimana e un'ora per una finestra di manutenzione per i lavori di streaming. AWS Glue esegue periodicamente attività di manutenzione. Durante queste finestre di manutenzione, AWS Glue sarà necessario riavviare i processi di streaming.

  AWS Glue riavvierà il lavoro entro 3 ore dalla finestra di manutenzione specificata. Ad esempio, se la finestra di manutenzione viene impostata per lunedì alle 10:00 GMT, i processi verranno riavviati tra le 10:00 GMT e le 13:00 GMT.

- `ProfileName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome di un profilo di AWS Glue utilizzo associato al lavoro.

## ExecutionProperty struttura

Una proprietà di esecuzione di un processo.

Campi

- `MaxConcurrentRuns`: numero (intero).

Il numero massimo di esecuzioni simultanee consentite per il processo. Il valore di default è 1. Viene restituito un errore al raggiungimento della soglia. Il valore massimo che è possibile specificare è controllato da un limite di servizio.

## NotificationProperty struttura

Specifica le proprietà di configurazione di una notifica.

Campi

- `NotifyDelayAfter`: numero (intero), almeno 1.

  Dopo l'inizio dell'esecuzione di un processo, la quantità di minuti da attendere prima di inviare una notifica di ritardo dell'esecuzione di un processo.

## JobCommand struttura

Specifica il codice eseguito quando viene eseguito un processo.

Campi

- `Name`: stringa UTF-8.

  Il nome del comando del processo. Per un processo ETL Apache Spark, deve essere `glueetl`. Per un processo shell Python, deve essere `pythonshell`. Per un processo ETL di streaming Apache Spark, deve essere `gluestreaming`. Per un processo Ray, questo deve essere `glueray`.
- `ScriptLocation`: stringa UTF-8, non superiore a 400000 byte di lunghezza.

  Specifica il percorso di Amazon Simple Storage Service (Amazon S3) per uno script che esegue un processo.
- `PythonVersion`: stringa UTF-8, corrispondente a Custom string pattern #49.

  La versione Python utilizzata per eseguire un processo shell Python. I valori consentiti sono 2 o 3.
- `Runtime`: stringa UTF-8, non superiore a 64 byte di lunghezza, corrispondente a Custom string pattern #33.

Nei processi Ray, Runtime viene utilizzato per specificare le versioni di Ray, Python e librerie aggiuntive disponibili nell'ambiente. Questo campo non viene utilizzato in altri tipi di processo. Per i valori dell'ambiente di runtime supportati, consultate Supported Ray runtime Environments nella AWS Glue Developer Guide.

## ConnectionsList struttura

Specifica le connessioni utilizzate da un processo.

Campi

- `Connections`— Un array di stringhe UTF-8, non più di 1000 stringhe.

  Un elenco di connessioni utilizzate dal processo.

## JobUpdate struttura

Specifica le informazioni utilizzate per aggiornare una definizione del processo esistente. La precedente definizione di processo viene completamente sovrascritta da questa informazione.

Campi

- `JobMode`: stringa UTF-8 (valori validi: `SCRIPT=""` | `VISUAL=""` | `NOTEBOOK=""`).

  Una modalità che descrive come è stato creato un processo. I valori validi sono:

  - `SCRIPT`- Il lavoro è stato creato utilizzando l'editor di script AWS Glue Studio.
  - `VISUAL`- Il lavoro è stato creato utilizzando l'editor visivo di AWS Glue Studio.
  - `NOTEBOOK`: il processo è stato creato utilizzando un notebook con sessioni interattive.

  Quando il campo `JobMode` è mancante o null, viene assegnato `SCRIPT` come valore predefinito.
- `JobRunQueuingEnabled`: booleano.

  Specifica se l'accodamento dell'esecuzione dei processi è abilitato per le esecuzioni di questo processo.

  Il valore true indica che l'accodamento delle esecuzioni di processi è abilitato. Se il valore è false o non popolato, le esecuzioni del processo non saranno prese in considerazione per l'accodamento.

Se questo campo non corrisponde al set di valori impostato nell'esecuzione del processo, verrà utilizzato il valore presente nel campo di esecuzione del processo.

- `Description`: stringa di descrizione, non superiore a 2048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  Descrizione del processo da definire.

- `LogUri`: stringa UTF-8.

  Questo campo è riservato per uso futuro.

- `Role`: stringa UTF-8.

  Il nome o ARN (Amazon Resource Name) del ruolo IAM associato a questo processo (richiesto).

- `ExecutionProperty`: un oggetto ExecutionProperty.

  `ExecutionProperty` che specifica il numero massimo di esecuzioni simultanee consentite per il processo.

- `Command`: un oggetto JobCommand.

  `JobCommand` che esegue il processo (richiesto).

- `DefaultArguments`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8.

  Ogni valore è una stringa UTF-8.

  Gli argomenti predefiniti per ciascuna esecuzione del processo, specificati come coppie nome-valore.

  Qui è possibile specificare gli argomenti utilizzati dal proprio script di esecuzione del lavoro, nonché gli argomenti utilizzati dal proprio script di esecuzione AWS Glue del lavoro.

  Gli argomenti del processo potrebbero essere registrati. Non passare segreti in testo chiaro come argomenti. Recupera i segreti da una AWS Glue connessione Gestione dei segreti AWS o da un altro meccanismo di gestione dei segreti se intendi mantenerli all'interno del Job.

  Per informazioni su come specificare e utilizzare i propri argomenti Job, consultate l'argomento Calling AWS Glue APIs in Python nella guida per sviluppatori.

Per informazioni sugli argomenti che puoi fornire a questo campo durante la configurazione dei processi Spark, consulta la pagina Special Parameters Used by AWS Glue nella Guida per gli sviluppatori.

Per informazioni sugli argomenti che puoi fornire a questo campo durante la configurazione dei processi Ray, consulta la pagina Using job parameters in Ray jobs nella Guida per gli sviluppatori.

- `NonOverridableArguments`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8.

  Ogni valore è una stringa UTF-8.

  Gli argomenti per questo processo che non vengono sovrascritti quando si forniscono argomenti di processo in un'esecuzione di processo, specificati come coppie nome-valore.

- `Connections`: un oggetto ConnectionsList.

  Le connessioni utilizzate per questo processo.

- `MaxRetries`: numero (intero).

  Il numero massimo di tentativi per riprovare il processo se ha esito negativo.

- `AllocatedCapacity`: numero (intero).

  in quanto obsoleto. Usare invece `MaxCapacity`.

  Il numero di unità di elaborazione AWS Glue dati (DPUs) da allocare a questo lavoro. È possibile allocarne almeno 2 DPUs; l'impostazione predefinita è 10. Una DPU è una misura relativa della potenza di elaborazione costituita da 4 V di capacità CPUs di elaborazione e 16 GB di memoria. Per ulteriori informazioni, consultare la pagina dei prezzi di AWS Glue.

- `Timeout`: numero (intero), almeno 1.

  Timeout del processo in minuti. Indica il tempo massimo durante cui l'esecuzione di un processo può utilizzare le risorse prima di essere terminata e passare allo stato `TIMEOUT`.

  I processi devono avere valori di timeout inferiori a 7 giorni o 10080 minuti. In caso contrario, i processi genereranno un'eccezione.

  Quando il valore viene lasciato vuoto, il timeout è predefinito a 2.880 minuti per Glue versione 4.0 e precedenti o 480 minuti per Glue versione 5.0 e successive.

Tutti i AWS Glue lavori esistenti con un valore di timeout superiore a 7 giorni verranno impostati come impostazione predefinita su 7 giorni. Ad esempio, se un processo in batch ha impostato un timeout di 20 giorni, sarà interrotto al settimo giorno.

Per i processi di streaming, se è stata impostata una finestra di manutenzione, il processo sarà riavviato durante tale finestra dopo 7 giorni.

- `MaxCapacity`: numero (doppio).

  Per i job Glue versione 1.0 o precedente, utilizzando il tipo di worker standard, il numero di unità di elaborazione AWS Glue dati (DPUs) che possono essere allocate durante l'esecuzione di questo processo. Una DPU è una misura relativa della potenza di elaborazione costituita da 4 V di capacità CPUs di elaborazione e 16 GB di memoria. Per ulteriori informazioni, consultare la pagina dei prezzi di AWS Glue.

  Per i processi Glue versione 2.0 e successive, non è possibile specificare il valore `Maximum capacity`. Si deve invece specificare un `Worker type` e un `Number of workers`.

  Non impostare `MaxCapacity` se usi `WorkerType` e `NumberOfWorkers`.

  Il valore che è possibile allocare per `MaxCapacity` varia a seconda che si esegua un processo shell di Python, un processo ETL di Apache Spark o un processo ETL di streaming di Apache Spark:

  - Quando si specifica un processo shell di Python (`JobCommand.Name`="pythonshell"), è possibile allocare 0,0625 o 1 DPU. Il valore di default è 0,0625 DPU.
  - Quando specificate un job ETL di Apache Spark (`JobCommand.Name`="glueetl») o un job ETL di streaming Apache Spark (="gluestreaming»), potete allocare da 2 a 100. `JobCommand.Name` DPUs L'impostazione predefinita è 10. DPUs Questo tipo di processo non può avere un'allocazione DPU frazionata.
- `WorkerType`— Stringa UTF-8 (valori validi: `Standard=""` | `G.025X=""` | `G.1X=""` | `G.2X=""` | `G.4X=""` | `G.8X=""` | `G.12X=""` | `G.16X=""` | `R.1X=""` | | `R.2X=""` `R.4X=""` `R.8X=""` |`Z.2X=""`).

  Il tipo di worker predefinito allocato quando viene eseguito un processo. Accetta un valore di G.025X, G.1X, G.2X, G.4X, G.8X, G.12X, G.16X, R.1X, R.2X, R.4X o R.8X per i job Spark. Accetta il valore Z.2X per i processi Ray. Per ulteriori informazioni, consultare Definire le proprietà di processo per i processi Spark
- `NumberOfWorkers`: numero (intero).

Il numero di worker di un `workerType` specifico allocati quando viene eseguito un processo.

- `SecurityConfiguration`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome della struttura `SecurityConfiguration` da usare con questo processo.

- `NotificationProperty`: un oggetto NotificationProperty.

  Specifica le proprietà di configurazione di una notifica di un processo.

- `GlueVersion`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Custom string pattern #48.

  Nei job Spark, `GlueVersion` determina le versioni di Apache Spark e Python disponibili in un job. AWS Glue La versione Python indica la versione supportata per i processi di tipo Spark.

  I processi Ray devono impostare il valore di `GlueVersion` su `4.0` o superiore. Tuttavia, le versioni di Ray, Python e le librerie aggiuntive disponibili nel processo Ray sono determinate dal parametro `Runtime` del comando del processo.

  Per ulteriori informazioni sulle AWS Glue versioni disponibili e sulle versioni corrispondenti di Spark e Python, consulta la versione Glue nella guida per sviluppatori.

  Se non viene fornita nella richiesta di aggiornamento, la versione Glue mantiene il valore della definizione del lavoro esistente.

- `CodeGenConfigurationNodes`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8 corrispondente al Custom string pattern #60.

  Ogni valore è un oggetto CodeGenConfigurationNode.

  La rappresentazione di un grafico aciclico diretto su cui si basano sia il componente visivo che la generazione di codice di Glue Studio.

- `ExecutionClass`: una stringa UTF-8, non superiore a 16 byte di lunghezza (valori validi: `FLEX=""` | `STANDARD=""`).

  Indica se il processo viene eseguito con una classe di esecuzione standard o flessibile. La classe di esecuzione standard è ideale per carichi di lavoro sensibili al tempo che richiedono un avvio rapido dei processi e risorse dedicate.

La classe di esecuzione flessibile è appropriata per i processi non sensibili al tempo i cui tempi di inizio e completamento possono variare.

Solo i lavori con AWS Glue versione 3.0 e successive e il tipo di comando `glueetl` potranno essere impostati su`ExecutionClass`. `FLEX` La classe di esecuzione flessibile è disponibile per i processi Spark.

- `SourceControlDetails`: un oggetto SourceControlDetails.

  I dettagli per una configurazione di controllo di origine per un processo, che consente la sincronizzazione degli artefatti del processo da o verso un repository remoto.
- `MaintenanceWindow`: stringa UTF-8, corrispondente a Custom string pattern #34.

  Questo campo specifica un giorno della settimana e un'ora per una finestra di manutenzione per i lavori di streaming. AWS Glue esegue periodicamente attività di manutenzione. Durante queste finestre di manutenzione, AWS Glue sarà necessario riavviare i processi di streaming.

  AWS Glue riavvierà il lavoro entro 3 ore dalla finestra di manutenzione specificata. Ad esempio, se la finestra di manutenzione viene impostata per lunedì alle 10:00 GMT, i processi verranno riavviati tra le 10:00 GMT e le 13:00 GMT.
- `ProfileName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome di un profilo di AWS Glue utilizzo associato al lavoro.

## SourceControlDetails struttura

I dettagli per una configurazione di controllo di origine per un processo, che consente la sincronizzazione degli artefatti del processo da o verso un repository remoto.

Campi

- `Provider`: stringa UTF-8 (valori validi: `GITHUB` | `AWS_CODE_COMMIT`).

  Il provider per il repository remoto.
- `Repository`: stringa UTF-8, non inferiore a 1 o superiore a 512 byte di lunghezza.

  Il nome del repository remoto che contiene gli artefatti del processo.
- `Owner`: stringa UTF-8, non inferiore a 1 o superiore a 512 byte di lunghezza.

Il proprietario del repository remoto che contiene gli artefatti del processo.

- `Branch`: stringa UTF-8, non inferiore a 1 o superiore a 512 byte di lunghezza.

  Un ramo opzionale nel repository remoto.

- `Folder`: stringa UTF-8, non inferiore a 1 o superiore a 512 byte di lunghezza.

  Una cartella opzionale nel repository remoto.

- `LastCommitId`: stringa UTF-8, non inferiore a 1 o superiore a 512 byte di lunghezza.

  L'ultimo ID di commit per un commit nel repository remoto.

- `LastSyncTimestamp`: stringa UTF-8, non inferiore a 1 o superiore a 512 byte di lunghezza.

  La data e l'ora in cui è stata eseguita l'ultima sincronizzazione di processo.

- `AuthStrategy`: stringa UTF-8 (valori validi: `PERSONAL_ACCESS_TOKEN` | `AWS_SECRETS_MANAGER`).

  Il tipo di autenticazione, che può essere un token di autenticazione memorizzato in AWS Secrets Manager o un token di accesso personale.

- `AuthToken`: stringa UTF-8, non inferiore a 1 o superiore a 512 byte di lunghezza.

  Il valore di un token di autorizzazione.

## Operazioni

- CreateJob azione (Python: create_job)
- UpdateJob azione (Python: update_job)
- GetJob azione (Python: get_job)
- GetJobs azione (Python: get_jobs)
- DeleteJob azione (Python: delete_job)
- ListJobs azione (Python: list_jobs)
- BatchGetJobs azione (Python: batch_get_jobs)

## CreateJob azione (Python: create_job)

Crea una nuova definizione del processo.

Richiesta

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome assegnato alla definizione del processo. Deve essere univoco all'interno dell'account .

- `JobMode`: stringa UTF-8 (valori validi: `SCRIPT=""` | `VISUAL=""` | `NOTEBOOK=""`).

  Una modalità che descrive come è stato creato un processo. I valori validi sono:

  - `SCRIPT`- Il lavoro è stato creato utilizzando l'editor di script di AWS Glue Studio.
  - `VISUAL`- Il lavoro è stato creato utilizzando l'editor visivo di AWS Glue Studio.
  - `NOTEBOOK`: il processo è stato creato utilizzando un notebook con sessioni interattive.

  Quando il campo `JobMode` è mancante o null, viene assegnato `SCRIPT` come valore predefinito.

- `JobRunQueuingEnabled`: booleano.

  Specifica se l'accodamento dell'esecuzione dei processi è abilitato per le esecuzioni di questo processo.

  Il valore true indica che l'accodamento delle esecuzioni di processi è abilitato. Se il valore è false o non popolato, le esecuzioni del processo non saranno prese in considerazione per l'accodamento.

  Se questo campo non corrisponde al set di valori impostato nell'esecuzione del processo, verrà utilizzato il valore presente nel campo di esecuzione del processo.

- `Description`: stringa di descrizione, non superiore a 2048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  Descrizione del processo da definire.

- `LogUri`: stringa UTF-8.

  Questo campo è riservato per uso futuro.

- `Role` - obbligatorio: stringa UTF-8.

  Il nome o ARN (Amazon Resource Name) del ruolo IAM associato a questo processo.

- `ExecutionProperty`: un oggetto ExecutionProperty.

  `ExecutionProperty` che specifica il numero massimo di esecuzioni simultanee consentite per il processo.

- `Command` - obbligatorio: un oggetto JobCommand.

  Il `JobCommand` che esegue questo lavoro.

- `DefaultArguments`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8.

  Ogni valore è una stringa UTF-8.

  Gli argomenti predefiniti per ciascuna esecuzione del processo, specificati come coppie nome-valore.

  Qui è possibile specificare gli argomenti utilizzati dal proprio script di esecuzione del lavoro, nonché gli argomenti utilizzati dal proprio script di esecuzione AWS Glue del lavoro.

  Gli argomenti del processo potrebbero essere registrati. Non passare segreti in testo chiaro come argomenti. Recupera i segreti da una AWS Glue connessione Gestione dei segreti AWS o da un altro meccanismo di gestione dei segreti se intendi mantenerli all'interno del Job.

  Per informazioni su come specificare e utilizzare i propri argomenti Job, consultate l'argomento Calling AWS Glue APIs in Python nella guida per sviluppatori.

  Per informazioni sugli argomenti che puoi fornire a questo campo durante la configurazione dei processi Spark, consulta la pagina Special Parameters Used by AWS Glue nella Guida per gli sviluppatori.

  Per informazioni sugli argomenti che puoi fornire a questo campo durante la configurazione dei processi Ray, consulta la pagina Using job parameters in Ray jobs nella Guida per gli sviluppatori.

- `NonOverridableArguments`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8.

  Ogni valore è una stringa UTF-8.

  Gli argomenti per questo processo che non vengono sovrascritti quando si forniscono argomenti di processo in un'esecuzione di processo, specificati come coppie nome-valore.

- `Connections`: un oggetto ConnectionsList.

  Le connessioni utilizzate per questo processo.

- `MaxRetries`: numero (intero).

Il numero massimo di tentativi per riprovare il processo se ha esito negativo.

- `AllocatedCapacity`: numero (intero).

  Questo parametro è obsoleto. Usare invece `MaxCapacity`.

  Il numero di unità di elaborazione AWS Glue dati (DPUs) da allocare a questo Job. È possibile allocarne almeno 2 DPUs; l'impostazione predefinita è 10. Una DPU è una misura relativa della potenza di elaborazione costituita da 4 V di capacità CPUs di elaborazione e 16 GB di memoria. Per ulteriori informazioni, consultare la pagina dei prezzi di AWS Glue.

- `Timeout`: numero (intero), almeno 1.

  Timeout del processo in minuti. Indica il tempo massimo durante cui l'esecuzione di un processo può utilizzare le risorse prima di essere terminata e passare allo stato `TIMEOUT`.

  I processi devono avere valori di timeout inferiori a 7 giorni o 10080 minuti. In caso contrario, i processi genereranno un'eccezione.

  Quando il valore viene lasciato vuoto, il timeout è predefinito a 2.880 minuti per Glue versione 4.0 e precedenti o 480 minuti per Glue versione 5.0 e successive.

  Tutti i AWS Glue lavori esistenti con un valore di timeout superiore a 7 giorni verranno impostati come impostazione predefinita su 7 giorni. Ad esempio, se un processo in batch ha impostato un timeout di 20 giorni, sarà interrotto al settimo giorno.

  Per i processi di streaming, se è stata impostata una finestra di manutenzione, il processo sarà riavviato durante tale finestra dopo 7 giorni.

- `MaxCapacity`: numero (doppio).

  Per i job Glue versione 1.0 o precedente, utilizzando il tipo di worker standard, il numero di unità di elaborazione AWS Glue dati (DPUs) che possono essere allocate durante l'esecuzione di questo processo. Una DPU è una misura relativa della potenza di elaborazione costituita da 4 V di capacità CPUs di elaborazione e 16 GB di memoria. Per ulteriori informazioni, consultare la pagina dei prezzi di AWS Glue.

  Per i processi Glue versione 2.0 e successive, non è possibile specificare il valore `Maximum capacity`. Si deve invece specificare un `Worker type` e un `Number of workers`.

  Non impostare `MaxCapacity` se usi `WorkerType` e `NumberOfWorkers`.

Il valore che è possibile allocare per `MaxCapacity` varia a seconda che si esegua un processo shell di Python, un processo ETL di Apache Spark o un processo ETL di streaming di Apache Spark:

- Quando si specifica un processo shell di Python (`JobCommand.Name`="pythonshell"), è possibile allocare 0,0625 o 1 DPU. Il valore di default è 0,0625 DPU.
- Quando specificate un job ETL di Apache Spark (`JobCommand.Name`="glueetl») o un job ETL di streaming Apache Spark (="gluestreaming»), potete allocare da 2 a 100. `JobCommand.Name` DPUs L'impostazione predefinita è 10. DPUs Questo tipo di processo non può avere un'allocazione DPU frazionata.

- `SecurityConfiguration`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome della struttura `SecurityConfiguration` da usare con questo processo.
- `Tags`: una matrice di mappe con coppie chiave-valore, non superiore alle 50 coppie.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Ogni valore è una stringa UTF-8, lunga non più di 256 byte.

  I tag da usare con questo processo. Puoi usare i tag per limitare l'accesso al processo. Per ulteriori informazioni sui tag in AWS Glue, consulta AWS Tags AWS Glue in nella guida per sviluppatori.
- `NotificationProperty`: un oggetto NotificationProperty.

  Specifica le proprietà di configurazione di una notifica di processo.
- `GlueVersion`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Custom string pattern #48.

  Nei job Spark, `GlueVersion` determina le versioni di Apache Spark e Python disponibili in un job. AWS Glue La versione Python indica la versione supportata per i processi di tipo Spark.

  I processi Ray devono impostare il valore di `GlueVersion` su `4.0` o superiore. Tuttavia, le versioni di Ray, Python e le librerie aggiuntive disponibili nel processo Ray sono determinate dal parametro `Runtime` del comando del processo.

  Per ulteriori informazioni sulle AWS Glue versioni disponibili e sulle versioni corrispondenti di Spark e Python, consulta la versione Glue nella guida per sviluppatori.

  I lavori creati senza specificare una versione di Glue predefinita è Glue 5.1.

- `NumberOfWorkers`: numero (intero).

  Il numero di worker di un `workerType` specifico allocati quando viene eseguito un processo.

- `WorkerType`— Stringa UTF-8 (valori validi: `Standard=""` | `G.025X=""` | `G.1X=""` | | `G.2X=""` | `G.4X=""` | `G.8X=""` | `G.12X=""` | `G.16X=""` | | `R.1X=""` `R.2X=""` `R.4X=""` |`R.8X=""`). `Z.2X=""`

  Il tipo di worker predefinito allocato quando viene eseguito un processo. Accetta un valore di G.025X, G.1X, G.2X, G.4X, G.8X, G.12X, G.16X, R.1X, R.2X, R.4X o R.8X per i job Spark. Accetta il valore Z.2X per i processi Ray.

  - Per il tipo di worker, ogni `G.025X` worker esegue il mapping a 0,25 DPU (2 v, 4 GB di memoria) con disco da 84 GB e fornisce 1 esecutore per lavoratore. CPUs Consigliamo questo tipo di worker per i processi di streaming a basso volume. Questo tipo di worker è disponibile solo per i lavori di streaming AWS Glue versione 3.0 o successiva.
  - Per il tipo di `G.1X` worker, ogni worker esegue il mapping a 1 DPU (4 vCPUs, 16 GB di memoria) con disco da 94 GB e fornisce 1 esecutore per lavoratore. Questi tipi di worker sono raccomandati per carichi di lavoro come trasformazioni di dati, join e query, in quanto offrono un modo scalabile ed economico per eseguire la maggior parte dei processi.
  - Per il tipo di `G.2X` worker, ogni worker esegue il mapping a 2 DPU (8 vCPUs, 32 GB di memoria) con disco da 138 GB e fornisce 1 esecutore per lavoratore. Questi tipi di worker sono raccomandati per carichi di lavoro come trasformazioni di dati, join e query, in quanto offrono un modo scalabile ed economico per eseguire la maggior parte dei processi.
  - Per il tipo di `G.4X` worker, ogni worker esegue il mapping a 4 DPU (16 vCPUs, 64 GB di memoria) con disco da 256 GB e fornisce 1 esecutore per lavoratore. Questi tipi di worker sono raccomandati per i processi i cui carichi di lavoro contengono trasformazioni, aggregazioni, join e query con i requisiti più elevati. Questo tipo di lavoratore è disponibile solo per i lavori Spark ETL AWS Glue versione 3.0 o successiva AWS nelle seguenti regioni: Stati Uniti orientali (Ohio), Stati Uniti orientali (Virginia settentrionale), Stati Uniti occidentali (California settentrionale), Stati Uniti occidentali (Oregon), Asia Pacifico (Mumbai), Asia Pacifico (Seoul), Asia Pacifico (Singapore), Asia Pacifico (Sydney), Asia Pacifico (Tokyo), Canada (Centrale), Europa (Francoforte), Europa (Irlanda), Europa (Londra), Europa (Spagna), Europa (Stoccolma) e
  - Per il tipo di `G.8X` worker, ogni worker esegue il mapping a 8 DPU (32 vCPUs, 128 GB di memoria) con disco da 512 GB e fornisce 1 esecutore per lavoratore. Questi tipi di worker sono raccomandati per i processi i cui carichi di lavoro contengono trasformazioni, aggregazioni, join e query con i requisiti più elevati. Questo tipo di worker è disponibile solo per i job Spark ETL AWS

Glue versione 3.0 o successiva, nelle stesse AWS regioni supportate per il tipo di lavoratore. `G.4X`

- Per il tipo di `G.12X` worker, ogni worker esegue il mapping a 12 DPU (48 vCPUs, 192 GB di memoria) con disco da 768 GB e fornisce 1 esecutore per lavoratore. Consigliamo questo tipo di lavoratore per lavori con carichi di lavoro molto elevati. Questo tipo di worker è disponibile solo per i job Spark ETL AWS Glue versione 4.0 o successiva.
- Per il tipo di `G.16X` worker, ogni worker esegue il mapping a 16 DPU (64 vCPUs, 256 GB di memoria) con disco da 1024 GB e fornisce 1 esecutore per lavoratore. Consigliamo questo tipo di lavoratore per lavori con carichi di lavoro molto elevati. Questo tipo di worker è disponibile solo per i job Spark ETL AWS Glue versione 4.0 o successiva.
- Per il tipo di `R.1X` worker, ogni worker esegue il mapping a 1 DPU (4 vCPUs, 32 GB di memoria) con disco da 94 GB e fornisce 1 esecutore per lavoratore. Consigliamo questo tipo di worker per carichi di lavoro che richiedono molta memoria. Questo tipo di worker è disponibile solo per i job AWS Glue Spark ETL versione 4.0 o successiva.
- Per il tipo di `R.2X` worker, ogni worker esegue il mapping su 2 DPU (8 vCPUs, 64 GB di memoria) con disco da 128 GB e fornisce 1 esecutore per lavoratore. Consigliamo questo tipo di worker per carichi di lavoro che richiedono molta memoria. Questo tipo di worker è disponibile solo per i job AWS Glue Spark ETL versione 4.0 o successiva.
- Per il tipo di `R.4X` worker, ogni worker esegue il mapping su 4 DPU (16 vCPUs, 128 GB di memoria) con disco da 256 GB e fornisce 1 esecutore per lavoratore. Consigliamo questo tipo di worker per carichi di lavoro che richiedono molta memoria. Questo tipo di worker è disponibile solo per i job AWS Glue Spark ETL versione 4.0 o successiva.
- Per il tipo di `R.8X` worker, ogni worker esegue il mapping a 8 DPU (32 vCPUs, 256 GB di memoria) con disco da 512 GB e fornisce 1 esecutore per lavoratore. Consigliamo questo tipo di worker per carichi di lavoro che richiedono molta memoria. Questo tipo di worker è disponibile solo per i job AWS Glue Spark ETL versione 4.0 o successiva.
- Per il tipo di `Z.2X` worker, ogni worker esegue il mapping su 2 M-DPU (8vCPUs, 64 GB di memoria) con disco da 128 GB e fornisce fino a 8 Ray worker in base all'autoscaler.

- `CodeGenConfigurationNodes`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8 corrispondente al Custom string pattern #60.

  Ogni valore è un oggetto CodeGenConfigurationNode.

  La rappresentazione di un grafico aciclico diretto su cui si basano sia il componente visivo che la generazione di codice di Glue Studio.

- `ExecutionClass`: una stringa UTF-8, non superiore a 16 byte di lunghezza (valori validi: `FLEX=""` | `STANDARD=""`).

  Indica se il processo viene eseguito con una classe di esecuzione standard o flessibile. La classe di esecuzione standard è ideale per carichi di lavoro sensibili al tempo che richiedono un avvio rapido dei processi e risorse dedicate.

  La classe di esecuzione flessibile è appropriata per i processi non sensibili al tempo i cui tempi di inizio e completamento possono variare.

  Solo i lavori con AWS Glue versione 3.0 e successive e il tipo di comando `glueetl` potranno essere impostati su. `ExecutionClass` `FLEX` La classe di esecuzione flessibile è disponibile per i processi Spark.
- `SourceControlDetails`: un oggetto SourceControlDetails.

  I dettagli per una configurazione di controllo di origine per un processo, che consente la sincronizzazione degli artefatti del processo da o verso un repository remoto.
- `MaintenanceWindow`: stringa UTF-8, corrispondente a Custom string pattern #34.

  Questo campo specifica un giorno della settimana e un'ora per una finestra di manutenzione per i lavori di streaming. AWS Glue esegue periodicamente attività di manutenzione. Durante queste finestre di manutenzione, AWS Glue sarà necessario riavviare i processi di streaming.

  AWS Glue riavvierà il lavoro entro 3 ore dalla finestra di manutenzione specificata. Ad esempio, se la finestra di manutenzione viene impostata per lunedì alle 10:00 GMT, i processi verranno riavviati tra le 10:00 GMT e le 13:00 GMT.
- `ProfileName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome di un profilo di AWS Glue utilizzo associato al lavoro.

Risposta

- `Name`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome univoco assegnato alla definizione del processo.

Errori

- `InvalidInputException`
- `IdempotentParameterMismatchException`
- `AlreadyExistsException`
- `InternalServiceException`
- `OperationTimeoutException`
- `ResourceNumberLimitExceededException`
- `ConcurrentModificationException`

## UpdateJob azione (Python: update_job)

Aggiorna la definizione di un processo esistente. La precedente definizione di processo viene completamente sovrascritta da questa informazione.

Richiesta

- `JobName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome della definizione del processo da aggiornare.
- `JobUpdate` - obbligatorio: un oggetto JobUpdate.

  Specifica i valori con cui aggiornare la definizione del processo. La configurazione non specificata viene rimossa o ripristinata ai valori predefiniti.
- `ProfileName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome di un profilo di AWS Glue utilizzo associato al lavoro.

Risposta

- `JobName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Restituisce il nome della definizione aggiornata del processo.

Errori

- `InvalidInputException`
- `EntityNotFoundException`
- `InternalServiceException`
- `OperationTimeoutException`
- `ConcurrentModificationException`

## GetJob azione (Python: get_job)

Recupera la definizione di un processo esistente.

Richiesta

- `JobName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome della definizione del processo da recuperare.

Risposta

- `Job`: un oggetto Processo.

  La definizione del processo richiesta.

Errori

- `InvalidInputException`
- `EntityNotFoundException`
- `InternalServiceException`
- `OperationTimeoutException`

## GetJobs azione (Python: get_jobs)

Recupera tutte le attuali definizioni del processo.

Richiesta

- `NextToken`: stringa UTF-8.

  Un token di continuazione, se si tratta di una chiamata di continuazione.
- `MaxResults`: numero (intero), non inferiore a 1 o superiore a 1.000.

  La dimensione massima della risposta.

Risposta

- `Jobs`: una matrice di oggetti Processo.

  Un elenco di definizioni del processo.
- `NextToken`: stringa UTF-8.

  Un token di continuazione, se non sono ancora state restituite tutte le definizioni del processo.

Errori

- `InvalidInputException`
- `EntityNotFoundException`
- `InternalServiceException`
- `OperationTimeoutException`

## DeleteJob azione (Python: delete_job)

Elimina una specifica definizione del processo. Se la definizione del processo non viene trovata, non viene generata alcuna eccezione.

Richiesta

- `JobName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome della definizione del processo da eliminare.

Risposta

- `JobName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome della definizione del processo eliminata.

Errori

- `InvalidInputException`
- `InternalServiceException`
- `OperationTimeoutException`

## ListJobs azione (Python: list_jobs)

Recupera i nomi di tutte le risorse lavorative in questo AWS account o le risorse con il tag specificato. Questa operazione consente di vedere quali risorse sono disponibili nel proprio account e i relativi nomi.

L'operazione accetta il campo facoltativo `Tags` che si può utilizzare come filtro per la risposta in modo che le risorse con tag possano essere recuperate come gruppo. Se si sceglie di utilizzare il filtro dei tag, potranno essere recuperate solo le risorse con tag.

Richiesta

- `NextToken`: stringa UTF-8.

  Token di continuazione, se si tratta di una richiesta di continuazione.
- `MaxResults`: numero (intero), non inferiore a 1 o superiore a 1.000.

  La dimensione massima di un elenco da restituire.
- `Tags`: una matrice di mappe con coppie chiave-valore, non superiore alle 50 coppie.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Ogni valore è una stringa UTF-8, lunga non più di 256 byte.

  Specifica che vengono restituite solo le risorse con tag.

Risposta

- `JobNames`: una matrice di stringhe UTF-8.

  I nomi di tutti i processi nell'account oppure i processi con i tag specificati.
- `NextToken`: stringa UTF-8.

  Token di continuazione, se l'elenco restituito non contiene l'ultimo parametro disponibile.

Errori

- `InvalidInputException`
- `EntityNotFoundException`
- `InternalServiceException`
- `OperationTimeoutException`

## BatchGetJobs azione (Python: batch_get_jobs)

Restituisce un elenco di metadati di risorse per un determinato elenco di nomi di processi. Dopo aver chiamato l'operazione `ListJobs`, puoi chiamare questa operazione per accedere ai dati a cui sono state concesse le autorizzazioni. Questa operazione supporta tutte le autorizzazioni IAM, tra cui le condizioni di autorizzazione che utilizzano i tag.

Richiesta

- `JobNames` - obbligatorio: matrice di stringhe UTF-8.

  L'elenco dei nomi di processo, che potrebbero essere i nomi restituiti dall'operazione `ListJobs`.

Risposta

- `Jobs`: una matrice di oggetti Processo.

  Un elenco di definizioni del processo.
- `JobsNotFound`: una matrice di stringhe UTF-8.

  Un elenco di nomi di processi non trovati.

Errori

- `InternalServiceException`
- `OperationTimeoutException`
- `InvalidInputException`

## Esecuzioni di processi

L'API Jobs Runs descrive i tipi di dati e l'API relativi all'avvio, all'arresto o alla visualizzazione delle esecuzioni di job e alla reimpostazione dei segnalibri dei processi, in. AWS Glue La cronologia di esecuzione dei processi è accessibile per 90 giorni per il flusso di lavoro e l'esecuzione del processo.

### Tipi di dati

- JobRun struttura
- Struttura Predecessor
- JobBookmarkEntry struttura
- BatchStopJobRunSuccessfulSubmission struttura
- BatchStopJobRunError struttura
- NotificationProperty struttura

### JobRun struttura

Contiene informazioni su una esecuzione di processo.

Campi

- `Id`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID di questa esecuzione di processo.
- `Attempt`: numero (intero).

  Il numero di tentativi di esecuzione di questo processo.
- `PreviousRunId`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

L'ID dell'esecuzione precedente di questo processo. Ad esempio, il `JobRunId` specificato nell'operazione `StartJobRun`.

- `TriggerName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del trigger che ha avviato questa esecuzione progetto.

- `JobName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome della definizione di processo in uso in questa esecuzione.

- `JobMode`: stringa UTF-8 (valori validi: `SCRIPT=""` | `VISUAL=""` | `NOTEBOOK=""`).

  Una modalità che descrive come è stato creato un processo. I valori validi sono:

  - `SCRIPT`- Il lavoro è stato creato utilizzando l'editor di script AWS Glue Studio.
  - `VISUAL`- Il lavoro è stato creato utilizzando l'editor visivo di AWS Glue Studio.
  - `NOTEBOOK`: il processo è stato creato utilizzando un notebook con sessioni interattive.

  Quando il campo `JobMode` è mancante o null, viene assegnato `SCRIPT` come valore predefinito.

- `JobRunQueuingEnabled`: booleano.

  Specifica se l'accodamento dell'esecuzione dei processi è abilitato per l'esecuzione del processo.

  Il valore true indica che l'accodamento delle esecuzioni di processi è abilitato per l'esecuzione. Se il valore è false o non popolato, l'esecuzione del processo non sarà presa in considerazione per l'accodamento.

- `StartedOn`: timestamp.

  La data e ora in cui questa esecuzione di processo è stata avviata.

- `LastModifiedOn`: timestamp.

  L'ultima volta in cui questa esecuzione di processo è stata modificata.

- `CompletedOn`: timestamp.

  La data e ora in cui questa elaborazione di processo è stata completata.

- `JobRunState`: stringa UTF-8 (valori validi: `STARTING` | `RUNNING` | `STOPPING` | `STOPPED` | `SUCCEEDED` | `FAILED` | `TIMEOUT` | `ERROR` | `WAITING` | `EXPIRED`).

Lo stato attuale del processo eseguito. Per ulteriori informazioni sugli stati dei processi terminati in modo anomalo, consulta AWS Glue Stati di esecuzione dei processi di .

- `Arguments`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8.

  Ogni valore è una stringa UTF-8.

  Gli argomenti del processo associati a questa esecuzione. Per questa esecuzione di processo, sostituiscono gli argomenti predefiniti impostati nella definizione del processo stessa.

  Qui è possibile specificare gli argomenti utilizzati dal proprio script di esecuzione del lavoro, nonché gli argomenti utilizzati dal proprio script di esecuzione AWS Glue del lavoro.

  Gli argomenti del processo potrebbero essere registrati. Non passare segreti in testo chiaro come argomenti. Recupera i segreti da una AWS Glue connessione Gestione dei segreti AWS o da un altro meccanismo di gestione dei segreti se intendi mantenerli all'interno del Job.

  Per informazioni su come specificare e utilizzare i propri argomenti Job, consultate l'argomento Calling AWS Glue APIs in Python nella guida per sviluppatori.

  Per informazioni sugli argomenti che puoi fornire a questo campo durante la configurazione dei processi Spark, consulta la pagina Special Parameters Used by AWS Glue nella Guida per gli sviluppatori.

  Per informazioni sugli argomenti che puoi fornire a questo campo durante la configurazione dei processi Ray, consulta la pagina Using job parameters in Ray jobs nella Guida per gli sviluppatori.

- `ErrorMessage`: stringa UTF-8.

  Un messaggio di errore associato a questa esecuzione di processo.

- `PredecessorRuns`: una matrice di oggetti Predecessor.

  Un elenco di predecessori di questa esecuzione di processo.

- `AllocatedCapacity`: numero (intero).

  in quanto obsoleto. Usare invece `MaxCapacity`.

  Il numero di unità di elaborazione AWS Glue dati (DPUs) assegnate a questo. JobRun DPUs È possibile allocare da 2 a 100; l'impostazione predefinita è 10. Una DPU è una misura relativa della

potenza di elaborazione costituita da 4 V di capacità CPUs di elaborazione e 16 GB di memoria. Per ulteriori informazioni, consultare la pagina dei prezzi di AWS Glue.

- `ExecutionTime`: numero (intero).

  Quantità di tempo (in secondi) durante cui l'esecuzione del processo ha utilizzato le risorse.

- `Timeout`: numero (intero), almeno 1.

  Timeout di `JobRun` (in minuti). Indica il tempo massimo durante cui l'esecuzione di un processo può utilizzare le risorse prima di essere terminata e passare allo stato `TIMEOUT`. Questo valore sostituisce il valore di timeout impostato nel processo padre.

  I processi devono avere valori di timeout inferiori a 7 giorni o 10080 minuti. In caso contrario, i processi genereranno un'eccezione.

  Quando il valore viene lasciato vuoto, il timeout è predefinito a 2.880 minuti per Glue versione 4.0 e precedenti o 480 minuti per Glue versione 5.0 e successive.

  Tutti i AWS Glue lavori esistenti con un valore di timeout superiore a 7 giorni verranno impostati come impostazione predefinita su 7 giorni. Ad esempio, se un processo in batch ha impostato un timeout di 20 giorni, sarà interrotto al settimo giorno.

  Per i processi di streaming, se è stata impostata una finestra di manutenzione, il processo sarà riavviato durante tale finestra dopo 7 giorni.

- `MaxCapacity`: numero (doppio).

  Per i job Glue versione 1.0 o precedente, utilizzando il tipo di worker standard, il numero di unità di elaborazione AWS Glue dati (DPUs) che possono essere allocate durante l'esecuzione di questo processo. Una DPU è una misura relativa della potenza di elaborazione costituita da 4 V di capacità CPUs di elaborazione e 16 GB di memoria. Per ulteriori informazioni, consultare la pagina dei prezzi di AWS Glue.

  Per i processi Glue versione 2.0 e successive, non è possibile specificare il valore `Maximum capacity`. Si deve invece specificare un `Worker type` e un `Number of workers`.

  Non impostare `MaxCapacity` se usi `WorkerType` e `NumberOfWorkers`.

  Il valore che è possibile allocare per `MaxCapacity` varia a seconda che si esegua un processo shell di Python, un processo ETL di Apache Spark o un processo ETL di streaming di Apache Spark:

- Quando si specifica un processo shell di Python (`JobCommand.Name`="pythonshell"), è possibile allocare 0,0625 o 1 DPU. Il valore di default è 0,0625 DPU.
- Quando specificate un job ETL di Apache Spark (`JobCommand.Name`="glueetl») o un job ETL di streaming Apache Spark (="gluestreaming»), potete allocare da 2 a 100. `JobCommand.Name` DPUs L'impostazione predefinita è 10. DPUs Questo tipo di processo non può avere un'allocazione DPU frazionata.

- `WorkerType`— Stringa UTF-8 (valori validi: `Standard=""` | `G.025X=""` | `G.1X=""` | `G.2X=""` | `G.4X=""` | `G.8X=""` | `G.12X=""` | `G.16X=""` | `R.1X=""` | | `R.2X=""` `R.4X=""` `R.8X=""` |`Z.2X=""`).

  Il tipo di worker predefinito allocato quando viene eseguito un processo. Accetta un valore di G.025X, G.1X, G.2X, G.4X, G.8X, G.12X, G.16X, R.1X, R.2X, R.4X o R.8X per i job Spark. Accetta il valore Z.2X per i processi Ray.

  - Per il tipo di worker, ogni `G.1X` worker esegue il mapping a 1 DPU (4 v, 16 GB di memoria) con disco da 94 GB e fornisce 1 esecutore per lavoratore. CPUs Questi tipi di worker sono raccomandati per carichi di lavoro come trasformazioni di dati, join e query, in quanto offrono un modo scalabile ed economico per eseguire la maggior parte dei processi.
  - Per il tipo di `G.2X` worker, ogni worker esegue il mapping a 2 DPU (8 vCPUs, 32 GB di memoria) con disco da 138 GB e fornisce 1 esecutore per lavoratore. Questi tipi di worker sono raccomandati per carichi di lavoro come trasformazioni di dati, join e query, in quanto offrono un modo scalabile ed economico per eseguire la maggior parte dei processi.
  - Per il tipo di `G.4X` worker, ogni worker esegue il mapping a 4 DPU (16 vCPUs, 64 GB di memoria) con disco da 256 GB e fornisce 1 esecutore per lavoratore. Questi tipi di worker sono raccomandati per i processi i cui carichi di lavoro contengono trasformazioni, aggregazioni, join e query con i requisiti più elevati. Questo tipo di lavoratore è disponibile solo per i job Spark ETL AWS Glue versione 3.0 o successiva AWS nelle seguenti regioni: Stati Uniti orientali (Ohio), Stati Uniti orientali (Virginia settentrionale), Stati Uniti occidentali (Oregon), Asia Pacifico (Singapore), Asia Pacifico (Sydney), Asia Pacifico (Tokyo), Canada (Centrale), Europa (Francoforte), Europa (Irlanda) ed Europa (Stoccolma).
  - Per il tipo di `G.8X` worker, ogni worker esegue il mapping a 8 DPU (32 vCPUs, 128 GB di memoria) con disco da 512 GB e fornisce 1 esecutore per lavoratore. Questi tipi di worker sono raccomandati per i processi i cui carichi di lavoro contengono trasformazioni, aggregazioni, join e query con i requisiti più elevati. Questo tipo di worker è disponibile solo per i job Spark ETL AWS Glue versione 3.0 o successiva, nelle stesse AWS regioni supportate per il tipo di lavoratore. `G.4X`

- Per il tipo di `G.12X` worker, ogni worker esegue il mapping a 12 DPU (48 vCPUs, 192 GB di memoria) con disco da 768 GB e fornisce 1 esecutore per lavoratore. Consigliamo questo tipo di lavoratore per lavori con carichi di lavoro molto elevati. Questo tipo di worker è disponibile solo per i job Spark ETL AWS Glue versione 4.0 o successiva.
- Per il tipo di `G.16X` worker, ogni worker esegue il mapping a 16 DPU (64 vCPUs, 256 GB di memoria) con disco da 1024 GB e fornisce 1 esecutore per lavoratore. Consigliamo questo tipo di lavoratore per lavori con carichi di lavoro molto elevati. Questo tipo di worker è disponibile solo per i job Spark ETL AWS Glue versione 4.0 o successiva.
- Per il tipo di `G.025X` worker, ogni worker esegue il mapping a 0,25 DPU (2 vCPUs, 4 GB di memoria) con disco da 84 GB e fornisce 1 esecutore per lavoratore. Consigliamo questo tipo di worker per i processi di streaming a basso volume. Questo tipo di worker è disponibile solo per i lavori di streaming AWS Glue versione 3.0 o successiva.
- Per il tipo di `R.1X` worker, ogni worker esegue il mapping a 1 DPU (4 vCPUs, 32 GB di memoria) con disco da 94 GB e fornisce 1 esecutore per lavoratore. Consigliamo questo tipo di worker per carichi di lavoro che richiedono molta memoria. Questo tipo di worker è disponibile solo per i job AWS Glue Spark ETL versione 4.0 o successiva.
- Per il tipo di `R.2X` worker, ogni worker esegue il mapping su 2 DPU (8 vCPUs, 64 GB di memoria) con disco da 128 GB e fornisce 1 esecutore per lavoratore. Consigliamo questo tipo di worker per carichi di lavoro che richiedono molta memoria. Questo tipo di worker è disponibile solo per i job AWS Glue Spark ETL versione 4.0 o successiva.
- Per il tipo di `R.4X` worker, ogni worker esegue il mapping su 4 DPU (16 vCPUs, 128 GB di memoria) con disco da 256 GB e fornisce 1 esecutore per lavoratore. Consigliamo questo tipo di worker per carichi di lavoro che richiedono molta memoria. Questo tipo di worker è disponibile solo per i job AWS Glue Spark ETL versione 4.0 o successiva.
- Per il tipo di `R.8X` worker, ogni worker esegue il mapping a 8 DPU (32 vCPUs, 256 GB di memoria) con disco da 512 GB e fornisce 1 esecutore per lavoratore. Consigliamo questo tipo di worker per carichi di lavoro che richiedono molta memoria. Questo tipo di worker è disponibile solo per i job AWS Glue Spark ETL versione 4.0 o successiva.
- Per il tipo di `Z.2X` worker, ogni worker esegue il mapping su 2 M-DPU (8vCPUs, 64 GB di memoria) con disco da 128 GB e fornisce fino a 8 Ray worker in base all'autoscaler.

- `NumberOfWorkers`: numero (intero).

  Il numero di worker di un `workerType` specifico allocati quando viene eseguito un processo.

- `SecurityConfiguration`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

Nome della struttura `SecurityConfiguration` da usare con questa esecuzione del processo.

- `LogGroupName`: stringa UTF-8.

  Il nome del gruppo di log per la registrazione sicura che può essere crittografato lato server in Amazon utilizzando. CloudWatch AWS KMS Questo nome può essere `/aws-glue/jobs/` e in questo caso la crittografia di default è `NONE`. Se si aggiunge un nome di ruolo e il nome `SecurityConfiguration` (in altre parole, `/aws-glue/jobs-yourRoleName-yourSecurityConfigurationName/`), la configurazione di sicurezza viene utilizzata per crittografare il gruppo di log.

- `NotificationProperty`: un oggetto NotificationProperty.

  Specifica le proprietà di configurazione di una notifica di esecuzione di un processo.

- `GlueVersion`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Custom string pattern #48.

  Nei job Spark, `GlueVersion` determina le versioni di Apache Spark e Python disponibili in un job. AWS Glue La versione Python indica la versione supportata per i processi di tipo Spark.

  I processi Ray devono impostare il valore di `GlueVersion` su `4.0` o superiore. Tuttavia, le versioni di Ray, Python e le librerie aggiuntive disponibili nel processo Ray sono determinate dal parametro `Runtime` del comando del processo.

  Per ulteriori informazioni sulle AWS Glue versioni disponibili e sulle versioni corrispondenti di Spark e Python, consulta la versione Glue nella guida per sviluppatori.

  I lavori creati senza specificare una versione di Glue predefinita è Glue 5.1.

- `DPUSeconds`: numero (doppio).

  Questo campo può essere impostato per esecuzioni dei processi con classe di esecuzione `FLEX` o quando è abilitato il dimensiona--mento automatico e rappresenta il tempo totale di esecuzione di ciascun esecutore durante il ciclo di vita dell'esecuzione di un processo in secondi, moltiplicato per un fattore DPU (1 per worker `G.1X`, 2 per worker `G.2X` o 0,25 per worker `G.025X`). Questo valore potrebbe essere diverso da quello `executionEngineRuntime` * `MaxCapacity` come nel caso dei processi di Auto Scaling, poiché il numero di esecutori in esecuzione in un determinato momento potrebbe essere inferiore a `MaxCapacity`. Pertanto, è possibile che il valore di `DPUSeconds` sia minore di `executionEngineRuntime` * `MaxCapacity`.

- `ExecutionClass`: una stringa UTF-8, non superiore a 16 byte di lunghezza (valori validi: `FLEX=""` | `STANDARD=""`).

  Indica se il processo viene eseguito con una classe di esecuzione standard o flessibile. La classe di esecuzione standard è ideale per carichi di lavoro sensibili al tempo che richiedono un avvio rapido dei processi e risorse dedicate.

  La classe di esecuzione flessibile è appropriata per i processi non sensibili al tempo i cui tempi di inizio e completamento possono variare.

  Solo i lavori con AWS Glue versione 3.0 e successive e il tipo di comando `glueetl` potranno essere impostati su`ExecutionClass`. `FLEX` La classe di esecuzione flessibile è disponibile per i processi Spark.
- `MaintenanceWindow`: stringa UTF-8, corrispondente a Custom string pattern #34.

  Questo campo specifica un giorno della settimana e un'ora per una finestra di manutenzione per i lavori di streaming. AWS Glue esegue periodicamente attività di manutenzione. Durante queste finestre di manutenzione, AWS Glue sarà necessario riavviare i processi di streaming.

  AWS Glue riavvierà il lavoro entro 3 ore dalla finestra di manutenzione specificata. Ad esempio, se la finestra di manutenzione viene impostata per lunedì alle 10:00 GMT, i processi verranno riavviati tra le 10:00 GMT e le 13:00 GMT.
- `ProfileName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome di un profilo di AWS Glue utilizzo associato all'esecuzione del processo.
- `StateDetail`: stringa UTF-8, non superiore a 400000 byte di lunghezza.

  Questo campo contiene dettagli relativi allo stato dell'esecuzione di un processo. Il campo ammette valori null.

  Ad esempio, quando l'esecuzione di un processo si trova nello stato WAITING a causa dell'accodamento, il campo contiene il motivo per cui l'esecuzione si trova in tale stato.
- `ExecutionRoleSessionPolicy`— Stringa UTF-8, lunga non meno di 2 o più di 2048 byte.

  Questa policy di sessione integrata nell' StartJobRun API consente di limitare dinamicamente le autorizzazioni del ruolo di esecuzione specificato per l'ambito del lavoro, senza richiedere la creazione di ruoli IAM aggiuntivi.

## Struttura Predecessor

Un'esecuzione di processo che è stata usata nel predicato di un trigger condizionale che ha attivato l'esecuzione di processo corrente.

Campi

- `JobName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome della definizione di processo usata dall'esecuzione del processo predecessore.
- `RunId`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID dell'esecuzione di processo dell'esecuzione processo predecessore.

## JobBookmarkEntry struttura

Definisce un punto nel quale un processo può riprendere l'elaborazione.

Campi

- `JobName`: stringa UTF-8.

  Il nome del processo in questione.
- `Version`: numero (intero).

  Versione del processo.
- `Run`: numero (intero).

  Il numero di ID dell'esecuzione.
- `Attempt`: numero (intero).

  Il numero di ID del tentativo.
- `PreviousRunId`: stringa UTF-8.

  Identificatore di esecuzione univoco associato all'esecuzione del processo precedente.
- `RunId`: stringa UTF-8.

  Il numero di ID dell'esecuzione.

- `JobBookmark`: stringa UTF-8.

  Il segnalibro stesso.

## BatchStopJobRunSuccessfulSubmission struttura

Registra una richiesta di arresto riuscita per un `JobRun` specificato.

Campi

- `JobName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome della definizione di processo usata nell'esecuzione del processo che è stata arrestata.
- `JobRunId`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Oggetto `JobRunId` dell'esecuzione del processo arrestata.

## BatchStopJobRunError struttura

Registra un errore che si è verificato durante il tentativo di arrestare un'esecuzione di un processo specifica.

Campi

- `JobName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome della definizione di processo usata nell'esecuzione del processo in questione.
- `JobRunId`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  `JobRunId` dell'esecuzione del processo in questione.
- `ErrorDetail`: un oggetto ErrorDetail.

  Specifica dettagli relativi all'errore che si è verificato.

## NotificationProperty struttura

Specifica le proprietà di configurazione di una notifica.

Campi

- `NotifyDelayAfter`: numero (intero), almeno 1.

  Dopo l'inizio dell'esecuzione di un processo, la quantità di minuti da attendere prima di inviare una notifica di ritardo dell'esecuzione di un processo.

## Operazioni

- StartJobRun azione (Python: start_job_run)
- BatchStopJobRun azione (Python: batch_stop_job_run)
- GetJobRun azione (Python: get_job_run)
- GetJobRuns azione (Python: get_job_runs)
- GetJobBookmark azione (Python: get_job_bookmark)
- GetJobBookmarks azione (Python: get_job_bookmarks)
- ResetJobBookmark azione (Python: reset_job_bookmark)

## StartJobRun azione (Python: start_job_run)

Avvia un'esecuzione di un processo usando una definizione di processo.

Richiesta

- `JobName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome della definizione di processo da usare.
- `JobRunQueuingEnabled`: booleano.

  Specifica se l'accodamento dell'esecuzione dei processi è abilitato per l'esecuzione del processo.

  Il valore true indica che l'accodamento delle esecuzioni di processi è abilitato per l'esecuzione. Se il valore è false o non popolato, l'esecuzione del processo non sarà presa in considerazione per l'accodamento.

- `JobRunId`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID di un precedente `JobRun` da ripetere.

- `Arguments`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8.

  Ogni valore è una stringa UTF-8.

  Gli argomenti del processo associati a questa esecuzione. Per questa esecuzione di processo, sostituiscono gli argomenti predefiniti impostati nella definizione del processo stessa.

  Qui è possibile specificare gli argomenti utilizzati dal proprio script di esecuzione del lavoro, nonché gli argomenti utilizzati dal proprio script di esecuzione del lavoro. AWS Glue

  Gli argomenti del processo potrebbero essere registrati. Non passare segreti in testo chiaro come argomenti. Recupera i segreti da una AWS Glue connessione Gestione dei segreti AWS o da un altro meccanismo di gestione dei segreti se intendi mantenerli all'interno del Job.

  Per informazioni su come specificare e utilizzare i propri argomenti Job, consultate l'argomento Calling AWS Glue APIs in Python nella guida per sviluppatori.

  Per informazioni sugli argomenti che puoi fornire a questo campo durante la configurazione dei processi Spark, consulta la pagina Special Parameters Used by AWS Glue nella Guida per gli sviluppatori.

  Per informazioni sugli argomenti che puoi fornire a questo campo durante la configurazione dei processi Ray, consulta la pagina Using job parameters in Ray jobs nella Guida per gli sviluppatori.

- `AllocatedCapacity`: numero (intero).

  in quanto obsoleto. Usare invece `MaxCapacity`.

  Il numero di unità di elaborazione AWS Glue dati (DPUs) da assegnare a questo. JobRun È possibile allocarne almeno 2 DPUs; l'impostazione predefinita è 10. Una DPU è una misura relativa della potenza di elaborazione costituita da 4 V di capacità CPUs di elaborazione e 16 GB di memoria. Per ulteriori informazioni, consultare la pagina dei prezzi di AWS Glue.

- `Timeout`: numero (intero), almeno 1.

Timeout di `JobRun` (in minuti). Indica il tempo massimo durante cui l'esecuzione di un processo può utilizzare le risorse prima di essere terminata e passare allo stato `TIMEOUT`. Questo valore sostituisce il valore di timeout impostato nel processo padre.

I processi devono avere valori di timeout inferiori a 7 giorni o 10080 minuti. In caso contrario, i processi genereranno un'eccezione.

Quando il valore viene lasciato vuoto, il timeout è predefinito a 2.880 minuti per Glue versione 4.0 e precedenti o 480 minuti per Glue versione 5.0 e successive.

Tutti i AWS Glue lavori esistenti con un valore di timeout superiore a 7 giorni verranno impostati come impostazione predefinita su 7 giorni. Ad esempio, se un processo in batch ha impostato un timeout di 20 giorni, sarà interrotto al settimo giorno.

Per i processi di streaming, se è stata impostata una finestra di manutenzione, il processo sarà riavviato durante tale finestra dopo 7 giorni.

- `MaxCapacity`: numero (doppio).

  Per i job Glue versione 1.0 o precedente, utilizzando il tipo di worker standard, il numero di unità di elaborazione AWS Glue dati (DPUs) che possono essere allocate durante l'esecuzione di questo processo. Una DPU è una misura relativa della potenza di elaborazione costituita da 4 V di capacità CPUs di elaborazione e 16 GB di memoria. Per ulteriori informazioni, consultare la pagina dei prezzi di AWS Glue.

  Per i processi Glue versione 2.0 e successive, non è possibile specificare il valore `Maximum capacity`. Si deve invece specificare un `Worker type` e un `Number of workers`.

  Non impostare `MaxCapacity` se usi `WorkerType` e `NumberOfWorkers`.

  Il valore che è possibile allocare per `MaxCapacity` varia a seconda che si esegua un processo shell di Python, un processo ETL di Apache Spark o un processo ETL di streaming di Apache Spark:

  - Quando si specifica un processo shell di Python (`JobCommand.Name`="pythonshell"), è possibile allocare 0,0625 o 1 DPU. Il valore di default è 0,0625 DPU.
  - Quando specificate un job ETL di Apache Spark (`JobCommand.Name`="glueetl») o un job ETL di streaming Apache Spark (="gluestreaming»), potete allocare da 2 a 100. `JobCommand.Name` DPUs L'impostazione predefinita è 10. DPUs Questo tipo di processo non può avere un'allocazione DPU frazionata.

- `SecurityConfiguration`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome della struttura `SecurityConfiguration` da usare con questa esecuzione del processo.

- `NotificationProperty`: un oggetto NotificationProperty.

  Specifica le proprietà di configurazione di una notifica di esecuzione di un processo.

- `WorkerType`: stringa UTF-8 (valori validi: `Standard=""` | `G.1X=""` | `G.2X=""` | `G.025X=""` | `G.4X=""` | `G.8X=""` | `Z.2X=""`).

  Il tipo di worker predefinito allocato quando viene eseguito un processo. Accetta un valore di G.1X, G.2X, G.4X, G.8X o G.025X per i processi Spark. Accetta il valore Z.2X per i processi Ray.

  - Per il tipo di `G.1X` worker, ogni worker esegue il mapping a 1 DPU (4 vCPUs, 16 GB di memoria) con disco da 94 GB e fornisce 1 esecutore per lavoratore. Questi tipi di worker sono raccomandati per carichi di lavoro come trasformazioni di dati, join e query, in quanto offrono un modo scalabile ed economico per eseguire la maggior parte dei processi.
  - Per il tipo di `G.2X` worker, ogni worker esegue il mapping a 2 DPU (8 vCPUs, 32 GB di memoria) con disco da 138 GB e fornisce 1 esecutore per lavoratore. Questi tipi di worker sono raccomandati per carichi di lavoro come trasformazioni di dati, join e query, in quanto offrono un modo scalabile ed economico per eseguire la maggior parte dei processi.
  - Per il tipo di `G.4X` worker, ogni worker esegue il mapping a 4 DPU (16 vCPUs, 64 GB di memoria) con disco da 256 GB e fornisce 1 esecutore per lavoratore. Questi tipi di worker sono raccomandati per i processi i cui carichi di lavoro contengono trasformazioni, aggregazioni, join e query con i requisiti più elevati. Questo tipo di lavoratore è disponibile solo per i job Spark ETL AWS Glue versione 3.0 o successiva AWS nelle seguenti regioni: Stati Uniti orientali (Ohio), Stati Uniti orientali (Virginia settentrionale), Stati Uniti occidentali (Oregon), Asia Pacifico (Singapore), Asia Pacifico (Sydney), Asia Pacifico (Tokyo), Canada (Centrale), Europa (Francoforte), Europa (Irlanda) ed Europa (Stoccolma).
  - Per il tipo di `G.8X` worker, ogni worker esegue il mapping a 8 DPU (32 vCPUs, 128 GB di memoria) con disco da 512 GB e fornisce 1 esecutore per lavoratore. Questi tipi di worker sono raccomandati per i processi i cui carichi di lavoro contengono trasformazioni, aggregazioni, join e query con i requisiti più elevati. Questo tipo di worker è disponibile solo per i job Spark ETL AWS Glue versione 3.0 o successiva, nelle stesse AWS regioni supportate per il tipo di lavoratore. `G.4X`
  - Per il tipo di `G.12X` worker, ogni worker esegue il mapping a 12 DPU (48 vCPUs, 192 GB di memoria) con disco da 768 GB e fornisce 1 esecutore per lavoratore. Consigliamo questo tipo di

lavoratore per lavori con carichi di lavoro molto elevati. Questo tipo di worker è disponibile solo per i job Spark ETL AWS Glue versione 4.0 o successiva.

- Per il tipo di `G.16X` worker, ogni worker esegue il mapping a 16 DPU (64 vCPUs, 256 GB di memoria) con disco da 1024 GB e fornisce 1 esecutore per lavoratore. Consigliamo questo tipo di lavoratore per lavori con carichi di lavoro molto elevati. Questo tipo di worker è disponibile solo per i job Spark ETL AWS Glue versione 4.0 o successiva.
- Per il tipo di `G.025X` worker, ogni worker esegue il mapping a 0,25 DPU (2 vCPUs, 4 GB di memoria) con disco da 84 GB e fornisce 1 esecutore per lavoratore. Consigliamo questo tipo di worker per i processi di streaming a basso volume. Questo tipo di worker è disponibile solo per i lavori di streaming AWS Glue versione 3.0 o successiva.
- Per il tipo di `R.1X` worker, ogni worker esegue il mapping a 1 DPU (4 vCPUs, 32 GB di memoria) con disco da 94 GB e fornisce 1 esecutore per lavoratore. Consigliamo questo tipo di worker per carichi di lavoro che richiedono molta memoria. Questo tipo di worker è disponibile solo per i job AWS Glue Spark ETL versione 4.0 o successiva.
- Per il tipo di `R.2X` worker, ogni worker esegue il mapping su 2 DPU (8 vCPUs, 64 GB di memoria) con disco da 128 GB e fornisce 1 esecutore per lavoratore. Consigliamo questo tipo di worker per carichi di lavoro che richiedono molta memoria. Questo tipo di worker è disponibile solo per i job AWS Glue Spark ETL versione 4.0 o successiva.
- Per il tipo di `R.4X` worker, ogni worker esegue il mapping su 4 DPU (16 vCPUs, 128 GB di memoria) con disco da 256 GB e fornisce 1 esecutore per lavoratore. Consigliamo questo tipo di worker per carichi di lavoro che richiedono molta memoria. Questo tipo di worker è disponibile solo per i job AWS Glue Spark ETL versione 4.0 o successiva.
- Per il tipo di `R.8X` worker, ogni worker esegue il mapping a 8 DPU (32 vCPUs, 256 GB di memoria) con disco da 512 GB e fornisce 1 esecutore per lavoratore. Consigliamo questo tipo di worker per carichi di lavoro che richiedono molta memoria. Questo tipo di worker è disponibile solo per i job AWS Glue Spark ETL versione 4.0 o successiva.
- Per il tipo di `Z.2X` worker, ogni worker esegue il mapping su 2 M-DPU (8vCPUs, 64 GB di memoria) con disco da 128 GB e fornisce fino a 8 Ray worker in base all'autoscaler.

- `NumberOfWorkers`: numero (intero).

  Il numero di worker di un `workerType` specifico allocati quando viene eseguito un processo.

- `ExecutionClass`: una stringa UTF-8, non superiore a 16 byte di lunghezza (valori validi: `FLEX=""` | `STANDARD=""`).

Indica se il processo viene eseguito con una classe di esecuzione standard o flessibile. La classe di esecuzione standard è ideale per carichi di lavoro sensibili al tempo che richiedono un avvio rapido dei processi e risorse dedicate.

La classe di esecuzione flessibile è appropriata per i processi non sensibili al tempo i cui tempi di inizio e completamento possono variare.

Solo i lavori con AWS Glue versione 3.0 e successive e il tipo di comando `glueetl` potranno essere impostati su. `ExecutionClass` `FLEX` La classe di esecuzione flessibile è disponibile per i processi Spark.

- `ProfileName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome di un profilo di AWS Glue utilizzo associato all'esecuzione del processo.
- `ExecutionRoleSessionPolicy`— Stringa UTF-8, lunga non meno di 2 o più di 2048 byte.

  Questa policy di sessione integrata nell' StartJobRun API consente di limitare dinamicamente le autorizzazioni del ruolo di esecuzione specificato per l'ambito del lavoro, senza richiedere la creazione di ruoli IAM aggiuntivi.

Risposta

- `JobRunId`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID assegnato a questa esecuzione processo.

Errori

- `InvalidInputException`
- `EntityNotFoundException`
- `InternalServiceException`
- `OperationTimeoutException`
- `ResourceNumberLimitExceededException`
- `ConcurrentRunsExceededException`

## BatchStopJobRun azione (Python: batch_stop_job_run)

Arresta una o più esecuzioni del processo per una definizione di processo specificata.

Richiesta

- `JobName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome della definizione di processo per cui arrestare le esecuzioni del processo.
- `JobRunIds` obbligatorio: una matrice di stringhe UTF-8, non inferiore a 1 o superiore a 25 stringhe.

  Elenco degli oggetti `JobRunIds` che dovrebbero essere arrestati per la definizione di processo.

Risposta

- `SuccessfulSubmissions`: una matrice di oggetti BatchStopJobRunSuccessfulSubmission.

  Un elenco di quelli che sono stati inviati correttamente per l'interruzione. JobRuns
- `Errors`: una matrice di oggetti BatchStopJobRunError.

  Un elenco degli errori rilevati nel tentativo di arrestare `JobRuns`, incluso il `JobRunId` per il quale si è verificato ciascun errore e i dettagli sull'errore stesso.

Errori

- `InvalidInputException`
- `InternalServiceException`
- `OperationTimeoutException`

## GetJobRun azione (Python: get_job_run)

Recupera i metadati per una determinata esecuzione di processo. La cronologia di esecuzione dei processi è accessibile per 365 giorni per flusso di lavoro ed esecuzione del processo.

Richiesta

- `JobName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome della definizione di processo in esecuzione.
- `RunId` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID dell'esecuzione processo.
- `PredecessorsIncluded`: booleano.

  True se un elenco delle esecuzioni predecessore deve essere restituito.

Risposta

- `JobRun`: un oggetto JobRun.

  I metadati di esecuzione del processo richiesti.

Errori

- `InvalidInputException`
- `EntityNotFoundException`
- `InternalServiceException`
- `OperationTimeoutException`

## GetJobRuns azione (Python: get_job_runs)

Recupera i metadati per tutte le esecuzioni di una definizione di processo specifica.

`GetJobRuns` restituisce le esecuzioni dei processi in ordine cronologico, con i processi più recenti restituiti per primi.

Richiesta

- `JobName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

Nome della definizione di processo per cui recuperare tutte le esecuzioni del processo.

- `NextToken`: stringa UTF-8.

  Un token di continuazione, se si tratta di una chiamata di continuazione.

- `MaxResults`: numero (intero), non inferiore a 1 o superiore a 200.

  La dimensione massima della risposta.

Risposta

- `JobRuns`: una matrice di oggetti JobRun.

  Un elenco di oggetti metadati esecuzione processo.

- `NextToken`: stringa UTF-8.

  Un token di continuazione, se non tutte le esecuzioni di processo richieste sono state restituite.

Errori

- `InvalidInputException`
- `EntityNotFoundException`
- `InternalServiceException`
- `OperationTimeoutException`

## GetJobBookmark azione (Python: get_job_bookmark)

Restituisce informazioni su una voce del segnalibro di processo.

Per ulteriori informazioni sull'abilitazione e l'utilizzo dei segnalibri di processo, consulta:

- Monitoraggio dei dati elaborati mediante segnalibri di processo
- Parametri Job utilizzati da AWS Glue
- Struttura del processo

Richiesta

- `JobName` - obbligatorio: stringa UTF-8.

  Il nome del processo in questione.

- `Version`: numero (intero).

  Versione del processo.

- `RunId`: stringa UTF-8.

  L'identificatore univoco dell'esecuzione associato a questa esecuzione di processo.

Risposta

- `JobBookmarkEntry`: un oggetto JobBookmarkEntry.

  Struttura che definisce un punto in cui un processo può riprendere l'elaborazione.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `InternalServiceException`
- `OperationTimeoutException`
- `ValidationException`

## GetJobBookmarks azione (Python: get_job_bookmarks)

Restituisce informazioni sulle voci del segnalibro di processo. L'elenco è ordinato sui numeri di versione decrescenti.

Per ulteriori informazioni sull'abilitazione e l'utilizzo dei segnalibri di processo, consulta:

- Monitoraggio dei dati elaborati mediante segnalibri di processo
- Parametri Job utilizzati da AWS Glue
- Struttura del processo

Richiesta

- `JobName` - obbligatorio: stringa UTF-8.

  Il nome del processo in questione.
- `MaxResults`: numero (intero).

  La dimensione massima della risposta.
- `NextToken`: numero (intero).

  Un token di continuazione, se si tratta di una chiamata di continuazione.

Risposta

- `JobBookmarkEntries`: una matrice di oggetti JobBookmarkEntry.

  Elenco di voci del segnalibro di processo che definisce un punto in cui un processo può riprendere l'elaborazione.
- `NextToken`: numero (intero).

  Un token di continuazione, che ha un valore pari a 1 se vengono restituite tutte le voci, oppure >1 se non vengono restituite tutte le esecuzioni di processo richieste.

Errori

- `InvalidInputException`
- `EntityNotFoundException`
- `InternalServiceException`
- `OperationTimeoutException`

## ResetJobBookmark azione (Python: reset_job_bookmark)

Ripristina una voce segnalibro.

Per ulteriori informazioni sull'abilitazione e l'utilizzo dei segnalibri di processo, consulta:

- Monitoraggio dei dati elaborati mediante segnalibri di processo
- Parametri Job utilizzati da AWS Glue

- Struttura del processo

Richiesta

- `JobName` - obbligatorio: stringa UTF-8.

  Il nome del processo in questione.
- `RunId`: stringa UTF-8.

  L'identificatore univoco dell'esecuzione associato a questa esecuzione di processo.

Risposta

- `JobBookmarkEntry`: un oggetto JobBookmarkEntry.

  La voce di ripristino del segnalibro.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `InternalServiceException`
- `OperationTimeoutException`

## Triggers

L'API Triggers descrive i tipi di dati e l'API relativi alla creazione, all'aggiornamento o all'eliminazione e all'avvio e all'arresto dei job trigger in. AWS Glue

### Tipi di dati

- Struttura trigger
- TriggerUpdate struttura
- Struttura predicato
- Struttura condizione
- Struttura operazione

- EventBatchingCondition struttura

## Struttura trigger

Informazioni su un trigger specifico.

Campi

- `Name`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del trigger.

- `WorkflowName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del flusso di lavoro associato al trigger.

- `Id`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Riservato per uso futuro.

- `Type`: stringa UTF-8 (valori validi: `SCHEDULED` | `CONDITIONAL` | `ON_DEMAND` | `EVENT`).

  Il tipo di trigger.

- `State`: stringa UTF-8 (valori validi: `CREATING` | `CREATED` | `ACTIVATING` | `ACTIVATED` | `DEACTIVATING` | `DEACTIVATED` | `DELETING` | `UPDATING`).

  Lo stato corrente del trigger.

- `Description`: stringa di descrizione, non superiore a 2048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  Una descrizione di questo trigger.

- `Schedule`: stringa UTF-8.

  Espressione `cron` usata per specificare la pianificazione (consulta Pianificazioni basate sul tempo per processi e crawler. Ad esempio, per eseguire un processo ogni giorno alle 12:15 UTC, devi specificare: `cron(15 12 * * ? *)`.

- `Actions`: una matrice di oggetti Azione.

Le operazioni avviate da questo trigger.

- `Predicate`: un oggetto Predicate.

  Il predicato di questo trigger, che definisce quando verrà attivato.

- `EventBatchingCondition`: un oggetto EventBatchingCondition.

  Condizione del batch che deve essere soddisfatta (numero specificato di eventi ricevuti o finestra temporale del batch scaduta) prima che venga attivato EventBridge l'attivazione dell'evento.

## TriggerUpdate struttura

Una struttura utilizzata per fornire informazioni per l'aggiornamento di un trigger. Questo oggetto aggiorna la definizione trigger precedente sovrascrivendola completamente.

Campi

- `Name`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Riservato per uso futuro.

- `Description`: stringa di descrizione, non superiore a 2048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  Una descrizione di questo trigger.

- `Schedule`: stringa UTF-8.

  Espressione `cron` usata per specificare la pianificazione (consulta Pianificazioni basate sul tempo per processi e crawler. Ad esempio, per eseguire un processo ogni giorno alle 12:15 UTC, devi specificare: `cron(15 12 * * ? *)`.

- `Actions`: una matrice di oggetti Azione.

  Le operazioni avviate da questo trigger.

- `Predicate`: un oggetto Predicate.

  Il predicato di questo trigger, che definisce quando verrà attivato.

- `EventBatchingCondition`: un oggetto EventBatchingCondition.

Condizione del batch che deve essere soddisfatta (numero specificato di eventi ricevuti o finestra temporale del batch scaduta) prima che venga attivato EventBridge l'attivazione dell'evento.

## Struttura predicato

Definisce il predicato del trigger, che determina il momento in cui viene attivato.

Campi

- `Logical`: stringa UTF-8 (valori validi: `AND` | `ANY`).

  Campo opzionale se è elencata una sola condizione. Se sono elencate più condizioni, questo campo è obbligatorio.
- `Conditions`— Una serie di Condizione oggetti, non più di 500 strutture.

  Un elenco delle condizioni che determinano il momento in cui il trigger verrà attivato.

## Struttura condizione

Definisce una condizione nella quale un trigger si attiva.

Campi

- `LogicalOperator`: stringa UTF-8 (valori validi: `EQUALS`).

  Un operatore logico.
- `JobName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del processo ali cui `JobRuns` si applica questa condizione e su cui attende questo trigger.
- `State`: stringa UTF-8 (valori validi: `STARTING` | `RUNNING` | `STOPPING` | `STOPPED` | `SUCCEEDED` | `FAILED` | `TIMEOUT` | `ERROR` | `WAITING` | `EXPIRED`).

  Lo stato della condizione. Attualmente, gli unici processi che stabiliscono che un trigger può essere ascoltato sono `SUCCEEDED`, `STOPPED`, `FAILED` e `TIMEOUT`. Gli unici crawler che stabiliscono che un trigger può ascoltare sono `SUCCEEDED`, `FAILED` e `CANCELLED`.
- `CrawlerName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

Il nome del crawler a cui si applica questa condizione.

- `CrawlState`: stringa UTF-8 (valori validi: `RUNNING` | `CANCELLING` | `CANCELLED` | `SUCCEEDED` | `FAILED` | `ERROR`).

  Lo stato del crawler a cui si applica questa condizione.

## Struttura operazione

Definisce un'operazione che deve essere avviata da un trigger.

Campi

- `JobName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome del processo che viene eseguito.
- `Arguments`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8.

  Ogni valore è una stringa UTF-8.

  Gli argomenti del processo utilizzati quando viene attivato il trigger. Per questa esecuzione di processo, sostituiscono gli argomenti predefiniti impostati nella definizione del processo stessa.

  Qui è possibile specificare gli argomenti utilizzati dal proprio script di esecuzione del lavoro, nonché gli argomenti utilizzati a sua volta. AWS Glue

  Per informazioni su come specificare e utilizzare i propri argomenti Job, consultate l'argomento Calling AWS Glue APIs in Python nella guida per sviluppatori.

  Per informazioni sulle coppie chiave-valore utilizzate per configurare il job, AWS Glue consultate l' AWS Glue argomento Parametri speciali usati da nella guida per sviluppatori.
- `Timeout`: numero (intero), almeno 1.

  Timeout di `JobRun` (in minuti). Indica il tempo massimo durante cui l'esecuzione di un processo può utilizzare le risorse prima di essere terminata e passare allo stato `TIMEOUT`. Questo valore sostituisce il valore di timeout impostato nel processo padre.

I processi devono avere valori di timeout inferiori a 7 giorni o 10.080 minuti. In caso contrario, i processi genereranno un'eccezione.

Quando il valore viene lasciato vuoto, il timeout è predefinito a 2.880 minuti per Glue versione 4.0 e precedenti o 480 minuti per Glue versione 5.0 e successive.

Tutti i AWS Glue lavori esistenti con un valore di timeout superiore a 7 giorni verranno impostati come impostazione predefinita su 7 giorni. Ad esempio, se un processo in batch ha impostato un timeout di 20 giorni, sarà interrotto al settimo giorno.

Per i processi di streaming, se è stata impostata una finestra di manutenzione, il processo sarà riavviato durante tale finestra dopo 7 giorni.

- `SecurityConfiguration`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome della struttura `SecurityConfiguration` da usare con questa operazione.

- `NotificationProperty`: un oggetto NotificationProperty.

  Specifica le proprietà di configurazione di una notifica di esecuzione di un processo.

- `CrawlerName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del crawler da usare con questa operazione.

## EventBatchingCondition struttura

Condizione del batch che deve essere soddisfatta (numero specificato di eventi ricevuti o finestra temporale del batch scaduta) prima che venga attivato EventBridge l'attivazione dell'evento.

Campi

- `BatchSize` - obbligatorio: numero (intero), non inferiore a 1 o superiore a 100.

  Numero di eventi che devono essere ricevuti da Amazon EventBridge prima che si EventBridge verifichi l'evento.

- `BatchWindow`: numero (intero), non inferiore a 1 o superiore a 900.

Intervallo di tempo in secondi dopo il quale si attiva EventBridge l'attivazione dell'evento. La finestra inizia quando viene ricevuto il primo evento.

## Operazioni

- CreateTrigger azione (Python: create_trigger)
- StartTrigger azione (Python: start_trigger)
- GetTrigger azione (Python: get_trigger)
- GetTriggers azione (Python: get_triggers)
- UpdateTrigger azione (Python: update_trigger)
- StopTrigger azione (Python: stop_trigger)
- DeleteTrigger azione (Python: delete_trigger)
- ListTriggers azione (Python: list_triggers)
- BatchGetTriggers azione (Python: batch_get_triggers)

## CreateTrigger azione (Python: create_trigger)

Crea un nuovo trigger.

Gli argomenti del processo potrebbero essere registrati. Non passare segreti in testo chiaro come argomenti. Recupera i segreti da AWS Glue Connection, AWS Secrets Manager o altro meccanismo di gestione dei segreti se intendi mantenerli all'interno del Job.

Richiesta

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del trigger.
- `WorkflowName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del flusso di lavoro associato al trigger.
- `Type` - obbligatorio: stringa UTF-8 (valori validi: `SCHEDULED` | `CONDITIONAL` | `ON_DEMAND` | `EVENT`).

Il tipo del nuovo trigger.

- `Schedule`: stringa UTF-8.

  Espressione `cron` usata per specificare la pianificazione (consulta Pianificazioni basate sul tempo per processi e crawler. Ad esempio, per eseguire un processo ogni giorno alle 12:15 UTC, devi specificare: `cron(15 12 * * ? *)`.

  Questo campo è obbligatorio quando il tipo di trigger è SCHEDULED (PIANIFICATO).

- `Predicate`: un oggetto Predicate.

  Un predicato per specificare quando occorre attivare il nuovo trigger.

  Questo campo è obbligatorio quando il tipo di trigger è `CONDITIONAL`.

- `Actions` - obbligatorio: una matrice di oggetti Azione.

  Le operazioni avviate da questo trigger al momento dell'attivazione.

- `Description`: stringa di descrizione, non superiore a 2048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  Una descrizione del nuovo trigger.

- `StartOnCreation`: booleano.

  Imposta su `true` per avviare i trigger `SCHEDULED` e `CONDITIONAL` al momento della creazione. True non è supportato per i trigger `ON_DEMAND`.

- `Tags`: una matrice di mappe con coppie chiave-valore, non superiore alle 50 coppie.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Ogni valore è una stringa UTF-8, lunga non più di 256 byte.

  I tag da usare con questo trigger. Puoi usare i tag per limitare l'accesso al trigger. Per ulteriori informazioni sui tag in AWS Glue, consulta AWS Tags in AWS Glue nella guida per sviluppatori.

- `EventBatchingCondition`: un oggetto EventBatchingCondition.

  Condizione del batch che deve essere soddisfatta (numero specificato di eventi ricevuti o finestra temporale del batch scaduta) prima che venga attivato EventBridge l'attivazione dell'evento.

Risposta

- Name: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del trigger.

Errori

- AlreadyExistsException
- EntityNotFoundException
- InvalidInputException
- IdempotentParameterMismatchException
- InternalServiceException
- OperationTimeoutException
- ResourceNumberLimitExceededException
- ConcurrentModificationException

## StartTrigger azione (Python: start_trigger)

Avvia un trigger esistente. Consulta la sezione Avvio dei processi per informazioni sull'avvio dei diversi tipi di trigger.

Richiesta

- Name - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del trigger da avviare.

Risposta

- Name: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del trigger avviato.

Errori

- `InvalidInputException`
- `InternalServiceException`
- `EntityNotFoundException`
- `OperationTimeoutException`
- `ResourceNumberLimitExceededException`
- `ConcurrentRunsExceededException`

## GetTrigger azione (Python: get_trigger)

Recupera la definizione di un trigger.

Richiesta

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome del trigger da recuperare.

Risposta

- `Trigger`: un oggetto Trigger.

  La definizione del trigger richiesta.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `InternalServiceException`
- `OperationTimeoutException`

## GetTriggers azione (Python: get_triggers)

Ottiene tutti i trigger associati a un processo.

Richiesta

- `NextToken`: stringa UTF-8.

  Un token di continuazione, se si tratta di una chiamata di continuazione.

- `DependentJobName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del processo per cui recuperare i trigger. Il trigger che può avviare questo processo viene restituito e, se non esiste nessun trigger di questo tipo, vengono restituiti tutti i trigger.

- `MaxResults`: numero (intero), non inferiore a 1 o superiore a 200.

  La dimensione massima della risposta.

Risposta

- `Triggers`: una matrice di oggetti Trigger.

  Un elenco di trigger per il processo specificato.

- `NextToken`: stringa UTF-8.

  Un token di continuazione, se non sono ancora stati restituiti tutti i trigger richiesti.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `InternalServiceException`
- `OperationTimeoutException`

## UpdateTrigger azione (Python: update_trigger)

Aggiorna una definizione del trigger.

Gli argomenti del processo potrebbero essere registrati. Non passare segreti in testo chiaro come argomenti. Recupera i segreti da AWS Glue Connection, AWS Secrets Manager o altro meccanismo di gestione dei segreti se intendi mantenerli all'interno del Job.

Richiesta

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del trigger da aggiornare.
- `TriggerUpdate` - obbligatorio: un oggetto TriggerUpdate.

  I nuovi valori con cui aggiornare il trigger.

Risposta

- `Trigger`: un oggetto Trigger.

  La definizione del trigger risultante.

Errori

- `InvalidInputException`
- `InternalServiceException`
- `EntityNotFoundException`
- `OperationTimeoutException`
- `ConcurrentModificationException`

## StopTrigger azione (Python: stop_trigger)

Arresta un trigger specificato.

Richiesta

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del trigger da arrestare.

Risposta

- Name: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del trigger che è stato arrestato.

Errori

- InvalidInputException
- InternalServiceException
- EntityNotFoundException
- OperationTimeoutException
- ConcurrentModificationException

## DeleteTrigger azione (Python: delete_trigger)

Elimina un trigger specificato. Se il trigger non viene trovato, non viene generata alcuna eccezione.

Richiesta

- Name - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del trigger da eliminare.

Risposta

- Name: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del trigger che è stato eliminato.

Errori

- InvalidInputException
- InternalServiceException

- `OperationTimeoutException`
- `ConcurrentModificationException`

## ListTriggers azione (Python: list_triggers)

Recupera i nomi di tutte le risorse di attivazione in questo AWS account o delle risorse con il tag specificato. Questa operazione consente di vedere quali risorse sono disponibili nel proprio account e i relativi nomi.

L'operazione accetta il campo facoltativo `Tags` che si può utilizzare come filtro per la risposta in modo che le risorse con tag possano essere recuperate come gruppo. Se si sceglie di utilizzare il filtro dei tag, potranno essere recuperate solo le risorse con tag.

Richiesta

- `NextToken`: stringa UTF-8.

  Token di continuazione, se si tratta di una richiesta di continuazione.
- `DependentJobName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del processo per cui recuperare i trigger. Viene restituito il trigger che può avviare questo processo. Se non esiste un trigger di questo tipo, vengono restituiti tutti i trigger.
- `MaxResults`: numero (intero), non inferiore a 1 o superiore a 200.

  La dimensione massima di un elenco da restituire.
- `Tags`: una matrice di mappe con coppie chiave-valore, non superiore alle 50 coppie.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Ogni valore è una stringa UTF-8, lunga non più di 256 byte.

  Specifica che vengono restituite solo le risorse con tag.

Risposta

- `TriggerNames`: una matrice di stringhe UTF-8.

  I nomi di tutti i trigger nell'account oppure i trigger con i tag specificati.

- `NextToken`: stringa UTF-8.

  Token di continuazione, se l'elenco restituito non contiene l'ultimo parametro disponibile.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `InternalServiceException`
- `OperationTimeoutException`

## BatchGetTriggers azione (Python: batch_get_triggers)

Restituisce un elenco di metadati di risorse per un determinato elenco di nomi di trigger. Dopo aver chiamato l'operazione `ListTriggers`, puoi chiamare questa operazione per accedere ai dati a cui sono state concesse le autorizzazioni. Questa operazione supporta tutte le autorizzazioni IAM, tra cui le condizioni di autorizzazione che utilizzano i tag.

Richiesta

- `TriggerNames` - obbligatorio: matrice di stringhe UTF-8.

  L'elenco dei nomi di trigger che potrebbero essere i nomi restituiti dall'operazione `ListTriggers`.

Risposta

- `Triggers`: una matrice di oggetti Trigger.

  Un elenco di definizioni di trigger.
- `TriggersNotFound`: una matrice di stringhe UTF-8.

  Un elenco di nomi di trigger non trovati.

Errori

- `InternalServiceException`
- `OperationTimeoutException`

- `InvalidInputException`

# Integrazione APIs in AWS Glue

## Tipi di dati

- Struttura di integrazione
- IntegrationConfig struttura
- IntegrationPartition struttura
- IntegrationError struttura
- IntegrationFilter struttura
- InboundIntegration struttura
- SourceProcessingProperties struttura
- TargetProcessingProperties struttura
- SourceTableConfig struttura
- TargetTableConfig struttura

## Struttura di integrazione

Descrive un'integrazione Zero-ETL.

Campi

- `SourceArn` - obbligatorio:, stringa UTF-8, lunghezza non inferiore a 1 o non superiore a 128 byte.

  L'ARN dell'origine dell'integrazione.
- `TargetArn` - obbligatorio:, stringa UTF-8, lunghezza non inferiore a 1 o non superiore a 128 byte.

  L'ARN della destinazione dell'integrazione.
- `Description`: stringa UTF-8, non superiore a 1000 byte di lunghezza, corrispondente a Custom string pattern #12.

  Una descrizione dell'integrazione.
- `IntegrationName` - obbligatorio:, stringa UTF-8, lunghezza non inferiore a 1 o non superiore a 128 byte.

Un nome univoco dell'integrazione.

- `IntegrationArn` - obbligatorio:, stringa UTF-8, lunghezza non inferiore a 1 o non superiore a 128 byte.

  Il nome della risorsa Amazon (ARN) dell'integrazione.

- `KmsKeyId`: stringa UTF-8, non inferiore a 1 o superiore a 2048 byte di lunghezza.

  L'ARN di una chiave KMS utilizzata per crittografare il canale.

- `AdditionalEncryptionContext`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8.

  Ogni valore è una stringa UTF-8.

  Un set facoltativo di coppie chiave-valore che contengono ulteriori informazioni contestuali per la crittografia. Può essere indicato solo se viene fornito `KMSKeyId`.

- `Tags`: una matrice di oggetti Tag.

  I metadati assegnati alla risorsa costituita da un elenco di coppie chiave-valore.

- `Status` - obbligatorio: stringa UTF-8 (valori validi: `CREATING` | `ACTIVE` | `MODIFYING` | `FAILED` | `DELETING` | `SYNCING` | `NEEDS_ATTENTION`).

  I possibili stati sono:

  - CREATING: l'integrazione è in fase di creazione.
  - ACTIVE: la creazione dell'integrazione ha esito positivo.
  - MODIFYING: l'integrazione è in fase di modifica.
  - FAILED: la creazione dell'integrazione ha esito negativo.
  - DELETING: l'integrazione viene eliminata.
  - SYNCING: l'integrazione è in fase di sincronizzazione.
  - NEEDS_ATTENTION: l'integrazione richiede attenzione, ad esempio per la sincronizzazione.

- `CreateTime` - obbligatorio: timestamp.

  L'ora di creazione dell'integrazione in fuso orario UTC.

- `IntegrationConfig`: un oggetto IntegrationConfig.

  Proprietà associate all'integrazione.

- `Errors`: una matrice di oggetti IntegrationError.

  Un elenco di errori associati all'integrazione.

- `DataFilter`: stringa UTF-8, non inferiore a 1 o superiore a 2048 byte di lunghezza.

  Seleziona le tabelle di origine per l'integrazione con la sintassi del filtro Maxwell.

## IntegrationConfig struttura

Proprietà associate all'integrazione.

Campi

- `RefreshInterval`: stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Specifica la frequenza con cui avvengono le estrazioni dell'acquisizione dei dati di modifica (CDC) o si verificano carichi incrementali. Questo parametro offre la flessibilità necessaria per allineare la frequenza di aggiornamento ai modelli di aggiornamento dei dati specifici, alle considerazioni sul carico del sistema e agli obiettivi di ottimizzazione delle prestazioni. L'incremento di tempo può essere impostato su un valore da 15 minuti a 8640 minuti (sei giorni).

- `SourceProperties`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8.

  Ogni valore è una stringa UTF-8.

  Una raccolta di coppie chiave-valore che specificano proprietà aggiuntive dell'origine dell'integrazione. Queste proprietà forniscono opzioni di configurazione che possono essere utilizzate per personalizzare il comportamento dell'origine ODB durante le operazioni di integrazione dei dati.

- `ContinuousSync`: booleano.

  Abilita la sincronizzazione continua per le estrazioni di dati su richiesta da: 1) applicazioni SaaS a servizi dati AWS come Amazon Redshift e 2) DynamoDB to. Amazon S3 Amazon S3

## IntegrationPartition struttura

Una struttura che descrive come vengono partizionati i dati nella destinazione.

Campi

- `FieldName`: stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Il nome del campo utilizzato per partizionare i dati nella destinazione. Evita di utilizzare colonne con valori univoci per ogni riga (ad esempio, LastModifiedTimestamp '', SystemModTimeStamp ') come colonna di partizione. Queste colonne non sono adatte al partizionamento poiché creano un numero elevato di piccole partizioni; ciò può causare problemi di prestazioni.
- `FunctionSpec`: stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Specifica la funzione utilizzata per partizionare i dati nella destinazione. I valori accettati per questo parametro sono:
  - `identity`: utilizza i valori di origine direttamente, senza trasformazione
  - `year`: estrae l'anno dai valori di timestamp (ad es., 2023)
  - `month`: estrae il mese dai valori di timestamp (ad es., 2023-01)
  - `day`: estrae il giorno dai valori di timestamp (ad es., 2023-01-15)
  - `hour`: estrae l'ora dai valori di timestamp (ad es., 2023-01-15-14)
- `ConversionSpec`: stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Specifica il formato di timestamp dei dati di origine. I valori validi sono:
  - `epoch_sec`: timestamp epoch Unix in secondi
  - `epoch_milli`: timestamp epoch Unix in millisecondi
  - `iso`: timestamp in formato ISO 8601

  > **Note**
  >
  > Specificare solo `ConversionSpec` quando si utilizzano funzioni di partizione basate su timestamp (anno, mese, giorno o ora). AWS Glue Zero-ETL utilizza questo parametro per trasformare correttamente i dati di origine in formato timestamp prima del partizionamento. Non utilizzare colonne ad alta cardinalità con la funzione di partizione `identity`. Tra le colonne ad alta cardinalità vi sono:
  >
  > - Chiavi primarie
  > - Campi timestamp (come `LastModifiedTimestamp`, `CreatedDate`)
  > - Timestamp generati dal sistema

L'utilizzo di colonne ad alta cardinalità con il partizionamento per identità crea tante piccole partizioni, che possono ridurre significativamente le prestazioni di acquisizione.

## IntegrationError struttura

Un errore associato a un'integrazione Zero-ETL.

Campi

- `ErrorCode`: stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Il codice associato a questo errore.
- `ErrorMessage`: stringa UTF-8, non inferiore a 1 o superiore a 2048 byte di lunghezza.

  Messaggio che descrive l'errore.

## IntegrationFilter struttura

Un filtro che può essere utilizzato quando si invoca una richiesta `DescribeIntegrations`.

Campi

- `Name`: stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Il nome del filtro.
- `Values`: una matrice di stringhe UTF-8.

  Un elenco di valori di filtro.

## InboundIntegration struttura

Una struttura di un'integrazione che scrive dati in una risorsa.

Campi

- `SourceArn` - obbligatorio: stringa UTF-8, lunghezza non inferiore a 1 o non superiore a 128 byte.

  L'ARN della risorsa di origine dell'integrazione.

- `TargetArn` - obbligatorio: stringa UTF-8, lunghezza non inferiore a 1 o non superiore a 128 byte.

  L'ARN della risorsa di destinazione dell'integrazione.
- `IntegrationArn` - obbligatorio: stringa UTF-8, lunghezza non inferiore a 1 o non superiore a 128 byte.

  L'ARN dell'integrazione Zero-ETL.
- `Status` - obbligatorio: stringa UTF-8 (valori validi: `CREATING` | `ACTIVE` | `MODIFYING` | `FAILED` | `DELETING` | `SYNCING` | `NEEDS_ATTENTION`).

  I possibili stati sono:
  - CREATING: l'integrazione è in fase di creazione.
  - ACTIVE: la creazione dell'integrazione ha esito positivo.
  - MODIFYING: l'integrazione è in fase di modifica.
  - FAILED: la creazione dell'integrazione ha esito negativo.
  - DELETING: l'integrazione viene eliminata.
  - SYNCING: l'integrazione è in fase di sincronizzazione.
  - NEEDS_ATTENTION: l'integrazione richiede attenzione, ad esempio per la sincronizzazione.
- `CreateTime` - obbligatorio: timestamp.

  L'ora di creazione dell'integrazione in fuso orario UTC.
- `IntegrationConfig`: un oggetto IntegrationConfig.

  Proprietà associate all'integrazione.
- `Errors`: una matrice di oggetti IntegrationError.

  Un elenco di errori associati all'integrazione.

## SourceProcessingProperties struttura

Le proprietà della risorsa associate all'origine dell'integrazione.

Campi

- `RoleArn`: stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Il ruolo IAM per accedere alla AWS Glue connessione.

## TargetProcessingProperties struttura

Le proprietà della risorsa associate alla destinazione dell'integrazione.

Campi

- `RoleArn`: stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Il ruolo IAM per accedere al AWS Glue database.
- `KmsArn`: stringa UTF-8, non inferiore a 1 o superiore a 2048 byte di lunghezza.

  L'ARN della chiave KMS utilizzata per la crittografia.
- `ConnectionName`: stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  La connessione AWS Glue di rete per configurare il AWS Glue job in esecuzione nel VPC del cliente.
- `EventBusArn`: stringa UTF-8, non inferiore a 1 o superiore a 2048 byte di lunghezza.

  L'ARN di un router di eventi Eventbridge per ricevere la notifica dello stato dell'integrazione.

## SourceTableConfig struttura

Proprietà utilizzate dal leg di origine per elaborare i dati dall'origine.

Campi

- `Fields`: una matrice di stringhe UTF-8.

  Un elenco di campi utilizzati per il filtraggio a livello di colonna. Attualmente non è supportata.
- `FilterPredicate`: stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Una clausola di condizione utilizzata per il filtraggio a livello di riga. Attualmente non è supportata.
- `PrimaryKey`: una matrice di stringhe UTF-8.

  Fornire il set di chiavi primarie per questa tabella. Al momento supportato specificamente per le entità SAP `Entity0f` su richiesta. Contatta l' AWS assistenza per rendere disponibile questa funzionalità.
- `RecordUpdateField`: stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

Campo di acquisizione incrementale basata su timestamp. Attualmente non è supportata.

## TargetTableConfig struttura

Proprietà utilizzate dal leg di destinazione per partizionare i dati nella destinazione.

Campi

- `UnnestSpec`: stringa UTF-8 (valori validi: `TOPLEVEL` | `FULL` | `NOUNNEST`).

  Specifica in che modo gli oggetti annidati vengono ridotti a oggetti di primo livello. I valori validi sono: "TOPLEVEL", "FULL" o "NOUNNEST".
- `PartitionSpec`: una matrice di oggetti IntegrationPartition.

  Determina il layout del file nella destinazione.
- `TargetTableName`: stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Il nome facoltativo di una tabella di destinazione.

## Operazioni

- CreateIntegration azione (Python: create_integration)
- ModifyIntegration azione (Python: modify_integration)
- DescribeIntegrations azione (Python: describe_integrations)
- DeleteIntegration azione (Python: delete_integration)
- DescribeInboundIntegrations azione (Python: describe_inbound_integrazioni)
- CreateIntegrationTableProperties azione (Python: create_integration_table_properties)
- UpdateIntegrationTableProperties azione (Python: update_integration_table_properties)
- GetIntegrationTableProperties azione (Python: get_integration_table_properties)
- DeleteIntegrationTableProperties azione (Python: delete_integration_table_properties)
- CreateIntegrationResourceProperty azione (Python: create_integration_resource_property)
- UpdateIntegrationResourceProperty azione (Python: update_integration_resource_property)
- GetIntegrationResourceProperty azione (Python: get_integration_resource_property)
- UntagResource azione (Python: untag_resource)

- ListTagsForResource azione (Python: list_tags_for_resource)

## CreateIntegration azione (Python: create_integration)

Crea un'integrazione zero-ETL nell'account del chiamante tra due risorse con Amazon Resource Names (ARNs): la e. `SourceArn` `TargetArn`

Richiesta

- `IntegrationName` - obbligatorio: stringa UTF-8, lunghezza non inferiore a 1 o non superiore a 128 byte.

  Un nome univoco per un'integrazione in AWS Glue.
- `SourceArn` - obbligatorio: stringa UTF-8, lunghezza non inferiore a 1 o non superiore a 128 byte.

  L'ARN della risorsa di origine dell'integrazione.
- `TargetArn` - obbligatorio: stringa UTF-8, lunghezza non inferiore a 1 o non superiore a 128 byte.

  L'ARN della risorsa di destinazione dell'integrazione.
- `Description`: stringa UTF-8, non superiore a 1000 byte di lunghezza, corrispondente a Custom string pattern #12.

  Una descrizione dell'integrazione.
- `DataFilter`: stringa UTF-8, non inferiore a 1 o superiore a 2048 byte di lunghezza.

  Seleziona le tabelle di origine per l'integrazione con la sintassi del filtro Maxwell.
- `KmsKeyId`: stringa UTF-8, non inferiore a 1 o superiore a 2048 byte di lunghezza.

  L'ARN di una chiave KMS utilizzata per crittografare il canale.
- `AdditionalEncryptionContext`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8.

  Ogni valore è una stringa UTF-8.

  Un set facoltativo di coppie chiave-valore che contengono ulteriori informazioni contestuali per la crittografia. Può essere indicato solo se viene fornito `KMSKeyId`.
- `Tags`: una matrice di oggetti Tag.

I metadati assegnati alla risorsa costituita da un elenco di coppie chiave-valore.

- `IntegrationConfig`: un oggetto IntegrationConfig.

  Le impostazioni di configurazione.

Risposta

- `SourceArn` - obbligatorio: stringa UTF-8, lunghezza non inferiore a 1 o non superiore a 128 byte.

  L'ARN della risorsa di origine dell'integrazione.
- `TargetArn` - obbligatorio: stringa UTF-8, lunghezza non inferiore a 1 o non superiore a 128 byte.

  L'ARN della risorsa di destinazione dell'integrazione.
- `IntegrationName` - obbligatorio: stringa UTF-8, lunghezza non inferiore a 1 o non superiore a 128 byte.

  Un nome univoco per un'integrazione in AWS Glue.
- `Description`: stringa UTF-8, non superiore a 1000 byte di lunghezza, corrispondente a Custom string pattern #12.

  Una descrizione dell'integrazione.
- `IntegrationArn` - obbligatorio: stringa UTF-8, lunghezza non inferiore a 1 o non superiore a 128 byte.

  Il nome della risorsa Amazon (ARN) dell'integrazione creata.
- `KmsKeyId`: stringa UTF-8, non inferiore a 1 o superiore a 2048 byte di lunghezza.

  L'ARN di una chiave KMS utilizzata per crittografare il canale.
- `AdditionalEncryptionContext`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8.

  Ogni valore è una stringa UTF-8.

  Un set facoltativo di coppie chiave-valore che contengono ulteriori informazioni contestuali per la crittografia.
- `Tags`: una matrice di oggetti Tag.

I metadati assegnati alla risorsa costituita da un elenco di coppie chiave-valore.

- `Status` - obbligatorio: stringa UTF-8 (valori validi: `CREATING` | `ACTIVE` | `MODIFYING` | `FAILED` | `DELETING` | `SYNCING` | `NEEDS_ATTENTION`).

  Lo stato dell'integrazione in fase di creazione.

  I possibili stati sono:

  - CREATING: l'integrazione è in fase di creazione.
  - ACTIVE: la creazione dell'integrazione ha esito positivo.
  - MODIFYING: l'integrazione è in fase di modifica.
  - FAILED: la creazione dell'integrazione ha esito negativo.
  - DELETING: l'integrazione viene eliminata.
  - SYNCING: l'integrazione è in fase di sincronizzazione.
  - NEEDS_ATTENTION: l'integrazione richiede attenzione, ad esempio per la sincronizzazione.
- `CreateTime` - obbligatorio: timestamp.

  L'ora di creazione dell'integrazione, in fuso orario UTC.
- `Errors`: una matrice di oggetti IntegrationError.

  Un elenco di errori associati alla creazione dell'integrazione.
- `DataFilter`: stringa UTF-8, non inferiore a 1 o superiore a 2048 byte di lunghezza.

  Seleziona le tabelle di origine per l'integrazione con la sintassi del filtro Maxwell.
- `IntegrationConfig`: un oggetto IntegrationConfig.

  Le impostazioni di configurazione.

Errori

- `ValidationException`
- `AccessDeniedException`
- `ResourceNotFoundException`
- `InternalServerException`
- `IntegrationConflictOperationFault`

- `IntegrationQuotaExceededFault`
- `KMSKeyNotAccessibleFault`
- `EntityNotFoundException`
- `InternalServiceException`
- `ConflictException`
- `ResourceNumberLimitExceededException`
- `InvalidInputException`

## ModifyIntegration azione (Python: modify_integration)

Modifica un'integrazione zero-ETL nell'account del chiamante.

Richiesta

- `IntegrationIdentifier` - obbligatorio: stringa UTF-8, lunghezza non inferiore a 1 o non superiore a 128 byte.

  Il nome della risorsa Amazon (ARN) dell'integrazione.
- `Description`: stringa UTF-8, non superiore a 1000 byte di lunghezza, corrispondente a Custom string pattern #12.

  Una descrizione dell'integrazione.
- `DataFilter`: stringa UTF-8, non inferiore a 1 o superiore a 2048 byte di lunghezza.

  Seleziona le tabelle di origine per l'integrazione con la sintassi del filtro Maxwell.
- `IntegrationConfig`: un oggetto IntegrationConfig.

  Le impostazioni di configurazione dell'integrazione. Attualmente, solo il RefreshInterval può essere modificato.
- `IntegrationName`: stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Un nome univoco per un'integrazione in AWS Glue.

Risposta

- `SourceArn` - obbligatorio: stringa UTF-8, lunghezza non inferiore a 1 o non superiore a 128 byte.

L'ARN dell'origine dell'integrazione.

- `TargetArn` - obbligatorio: stringa UTF-8, lunghezza non inferiore a 1 o non superiore a 128 byte.

  L'ARN della destinazione dell'integrazione.

- `IntegrationName` - obbligatorio: stringa UTF-8, lunghezza non inferiore a 1 o non superiore a 128 byte.

  Un nome univoco per un'integrazione in AWS Glue.

- `Description`: stringa UTF-8, non superiore a 1000 byte di lunghezza, corrispondente a Custom string pattern #12.

  Una descrizione dell'integrazione.

- `IntegrationArn` - obbligatorio: stringa UTF-8, lunghezza non inferiore a 1 o non superiore a 128 byte.

  Il nome della risorsa Amazon (ARN) dell'integrazione.

- `KmsKeyId`: stringa UTF-8, non inferiore a 1 o superiore a 2048 byte di lunghezza.

  L'ARN di una chiave KMS utilizzata per crittografare il canale.

- `AdditionalEncryptionContext`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8.

  Ogni valore è una stringa UTF-8.

  Un set facoltativo di coppie chiave-valore che contengono ulteriori informazioni contestuali per la crittografia.

- `Tags`: una matrice di oggetti Tag.

  I metadati assegnati alla risorsa costituita da un elenco di coppie chiave-valore.

- `Status` - obbligatorio: stringa UTF-8 (valori validi: `CREATING` | `ACTIVE` | `MODIFYING` | `FAILED` | `DELETING` | `SYNCING` | `NEEDS_ATTENTION`).

  Lo stato dell'integrazione in fase di modifica.

  I possibili stati sono:

  - CREATING: l'integrazione è in fase di creazione.

  - ACTIVE: la creazione dell'integrazione ha esito positivo.
  - MODIFYING: l'integrazione è in fase di modifica.
  - FAILED: la creazione dell'integrazione ha esito negativo.
  - DELETING: l'integrazione viene eliminata.
  - SYNCING: l'integrazione è in fase di sincronizzazione.
  - NEEDS_ATTENTION: l'integrazione richiede attenzione, ad esempio per la sincronizzazione.
- `CreateTime` - obbligatorio: timestamp.

  L'ora di creazione dell'integrazione, in fuso orario UTC.
- `Errors`: una matrice di oggetti IntegrationError.

  Un elenco di errori associati alla modifica dell'integrazione.
- `DataFilter`: stringa UTF-8, non inferiore a 1 o superiore a 2048 byte di lunghezza.

  Seleziona le tabelle di origine per l'integrazione con la sintassi del filtro Maxwell.
- `IntegrationConfig`: un oggetto IntegrationConfig.

  Le impostazioni di configurazione dell'integrazione aggiornate.

Errori

- `ValidationException`
- `AccessDeniedException`
- `InternalServerException`
- `IntegrationNotFoundFault`
- `IntegrationConflictOperationFault`
- `InvalidIntegrationStateFault`
- `EntityNotFoundException`
- `InternalServiceException`
- `ConflictException`
- `InvalidStateException`
- `InvalidInputException`

# DescribeIntegrations azione (Python: describe_integrations)

L'API viene utilizzata per recuperare un elenco di integrazioni.

Richiesta

- `IntegrationIdentifier`: stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Il nome della risorsa Amazon (ARN) dell'integrazione.
- `Marker`: stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Un valore che indica il punto di partenza per il set di record di risposta successivo in una richiesta seguente.
- `MaxRecords`: numero (intero).

  Il numero totale di elementi da restituire nell'output.
- `Filters`: una matrice di oggetti IntegrationFilter.

  Un elenco di chiavi e valori per filtrare i risultati. Le chiavi supportate sono «Status», "IntegrationName«e"». SourceArn IntegrationName è limitato a un solo valore.

Risposta

- `Integrations`: una matrice di oggetti Integrazione.

  Un elenco di integrazioni Zero-ETL.
- `Marker`: stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Un valore che indica il punto di partenza per il set di record di risposta successivo in una richiesta seguente.

Errori

- `ValidationException`
- `AccessDeniedException`
- `InternalServerException`
- `IntegrationNotFoundFault`
- `EntityNotFoundException`

- `InternalServiceException`
- `InvalidInputException`

## DeleteIntegration azione (Python: delete_integration)

Elimina l'integrazione Zero-ETL specificata.

Richiesta

- `IntegrationIdentifier` - obbligatorio: stringa UTF-8, lunghezza non inferiore a 1 o non superiore a 128 byte.

  Il nome della risorsa Amazon (ARN) dell'integrazione.

Risposta

- `SourceArn` - obbligatorio: stringa UTF-8, lunghezza non inferiore a 1 o non superiore a 128 byte.

  L'ARN dell'origine dell'integrazione.
- `TargetArn` - obbligatorio: stringa UTF-8, lunghezza non inferiore a 1 o non superiore a 128 byte.

  L'ARN della destinazione dell'integrazione.
- `IntegrationName` - obbligatorio: stringa UTF-8, lunghezza non inferiore a 1 o non superiore a 128 byte.

  Un nome univoco per un'integrazione in AWS Glue.
- `Description`: stringa UTF-8, non superiore a 1000 byte di lunghezza, corrispondente a Custom string pattern #12.

  Una descrizione dell'integrazione.
- `IntegrationArn` - obbligatorio: stringa UTF-8, lunghezza non inferiore a 1 o non superiore a 128 byte.

  Il nome della risorsa Amazon (ARN) dell'integrazione.
- `KmsKeyId`: stringa UTF-8, non inferiore a 1 o superiore a 2048 byte di lunghezza.

  L'ARN di una chiave KMS utilizzata per crittografare il canale.
- `AdditionalEncryptionContext`: una matrice della mappa di coppie chiave-valore.

Ogni chiave è una stringa UTF-8.

Ogni valore è una stringa UTF-8.

Un set facoltativo di coppie chiave-valore che contengono ulteriori informazioni contestuali per la crittografia.

- `Tags`: una matrice di oggetti Tag.

  I metadati assegnati alla risorsa costituita da un elenco di coppie chiave-valore.

- `Status` - obbligatorio: stringa UTF-8 (valori validi: `CREATING` | `ACTIVE` | `MODIFYING` | `FAILED` | `DELETING` | `SYNCING` | `NEEDS_ATTENTION`).

  Lo stato dell'integrazione in fase di eliminazione.

  I possibili stati sono:

  - CREATING: l'integrazione è in fase di creazione.
  - ACTIVE: la creazione dell'integrazione ha esito positivo.
  - MODIFYING: l'integrazione è in fase di modifica.
  - FAILED: la creazione dell'integrazione ha esito negativo.
  - DELETING: l'integrazione viene eliminata.
  - SYNCING: l'integrazione è in fase di sincronizzazione.
  - NEEDS_ATTENTION: l'integrazione richiede attenzione, ad esempio per la sincronizzazione.

- `CreateTime` - obbligatorio: timestamp.

  L'ora di creazione dell'integrazione, in fuso orario UTC.

- `Errors`: una matrice di oggetti IntegrationError.

  Un elenco di errori associati all'integrazione.

- `DataFilter`: stringa UTF-8, non inferiore a 1 o superiore a 2048 byte di lunghezza.

  Seleziona le tabelle di origine per l'integrazione con la sintassi del filtro Maxwell.

Errori

- `ValidationException`
- `AccessDeniedException`

- `InternalServerException`
- `IntegrationNotFoundFault`
- `IntegrationConflictOperationFault`
- `InvalidIntegrationStateFault`
- `EntityNotFoundException`
- `InternalServiceException`
- `ConflictException`
- `InvalidStateException`
- `InvalidInputException`

## DescribeInboundIntegrations azione (Python: describe_inbound_integrazioni)

Restituisce un elenco di integrazioni in entrata per l'integrazione specificata.

Richiesta

- `IntegrationArn`: stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Il nome della risorsa Amazon (ARN) dell'integrazione.
- `Marker`: stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Token per specificare dove iniziare l'impaginazione. Si tratta del marcatore di una risposta precedentemente troncata.
- `MaxRecords`: numero (intero).

  Il numero totale di elementi da restituire nell'output.
- `TargetArn`: stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Il nome della risorsa Amazon (ARN) della risorsa di destinazione nell'integrazione.

Risposta

- `InboundIntegrations`: una matrice di oggetti InboundIntegration.

  Un elenco di integrazioni in entrata.

- `Marker`: stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Un valore che indica il punto di partenza per il set di record di risposta successivo in una richiesta seguente.

Errori

- `ValidationException`
- `AccessDeniedException`
- `InternalServerException`
- `IntegrationNotFoundFault`
- `TargetResourceNotFound`
- `OperationNotSupportedException`
- `EntityNotFoundException`
- `InternalServiceException`
- `InvalidInputException`

## CreateIntegrationTableProperties azione (Python: create_integration_table_properties)

Questa API viene utilizzata per fornire proprietà di override facoltative per le tabelle da replicare. Queste proprietà possono includere proprietà di filtraggio e partizionamento per le tabelle di origine e di destinazione. Per impostare sia le proprietà di origine che quelle di destinazione, è necessario richiamare la stessa API con l'ARN `ResourceArn` di AWS Glue connessione e l'`ResourceArn`ARN del AWS Glue database rispettivamente `SourceTableConfig` con. `TargetTableConfig`

Richiesta

- `ResourceArn` - obbligatorio: stringa UTF-8, lunghezza non inferiore a 1 o non superiore a 128 byte.

  Il nome della risorsa Amazon (ARN) della tabella di destinazione per cui creare le proprietà della tabella di integrazione. Attualmente, questa API supporta solo la creazione di proprietà della tabella di integrazione per le tabelle di destinazione e l'ARN fornito dovrebbe essere l'ARN della tabella di destinazione nel Data Catalog. AWS Glue Il supporto per la creazione di proprietà della tabella

di integrazione per le connessioni di origine (utilizzando l'ARN della connessione) non è ancora implementato e verrà aggiunto in una versione futura.

- `TableName` - obbligatorio: stringa UTF-8, lunghezza non inferiore a 1 o non superiore a 128 byte.

  Il nome della tabella da replicare.

- `SourceTableConfig`: un oggetto SourceTableConfig.

  Una struttura di configurazione della tabella di origine. Consultare la struttura `SourceTableConfig` per un elenco di proprietà di origine supportate.

- `TargetTableConfig`: un oggetto TargetTableConfig.

  Una struttura di configurazione della tabella di destinazione.

Risposta

- Nessun parametro di risposta.

Errori

- `ValidationException`
- `AccessDeniedException`
- `ResourceNotFoundException`
- `InternalServerException`
- `EntityNotFoundException`
- `InternalServiceException`
- `InvalidInputException`

## UpdateIntegrationTableProperties azione (Python: update_integration_table_properties)

Questa API viene utilizzata per fornire proprietà di override facoltative per le tabelle da replicare. Queste proprietà possono includere proprietà di filtraggio e partizionamento per le tabelle di origine e di destinazione. Per impostare sia le proprietà di origine che quelle di destinazione, è necessario richiamare la stessa API con l'ARN `ResourceArn` di AWS Glue connessione e l'`ResourceArn`ARN del AWS Glue database rispettivamente `SourceTableConfig` con. `TargetTableConfig`

L'override si rifletterà in tutte le integrazioni che utilizzano lo stesso `ResourceArn` e la stessa tabella di origine.

Richiesta

- `ResourceArn` - obbligatorio: stringa UTF-8, lunghezza non inferiore a 1 o non superiore a 128 byte.

  L'ARN di connessione dell'origine o l'ARN del database della destinazione.
- `TableName` - obbligatorio: stringa UTF-8, lunghezza non inferiore a 1 o non superiore a 128 byte.

  Il nome della tabella da replicare.
- `SourceTableConfig`: un oggetto SourceTableConfig.

  Una struttura di configurazione della tabella di origine.
- `TargetTableConfig`: un oggetto TargetTableConfig.

  Una struttura di configurazione della tabella di destinazione.

Risposta

- Nessun parametro di risposta.

Errori

- `ValidationException`
- `AccessDeniedException`
- `ResourceNotFoundException`
- `InternalServerException`
- `EntityNotFoundException`
- `InternalServiceException`
- `InvalidInputException`

# GetIntegrationTableProperties azione (Python: get_integration_table_properties)

Questa API viene utilizzata per recuperare le proprietà di override facoltative per le tabelle da replicare. Queste proprietà possono includere proprietà di filtraggio e partizione per le tabelle di origine e di destinazione.

Richiesta

- `ResourceArn` - obbligatorio: stringa UTF-8, lunghezza non inferiore a 1 o non superiore a 128 byte.

  Il nome della risorsa Amazon (ARN) della tabella di destinazione per cui recuperare le proprietà della tabella di integrazione. Attualmente, questa API supporta solo il recupero delle proprietà per le tabelle di destinazione e l'ARN fornito dovrebbe essere l'ARN della tabella di destinazione nel Data Catalog. AWS Glue Il supporto per il recupero delle proprietà della tabella di integrazione per le connessioni di origine (utilizzando l'ARN della connessione) non è ancora implementato e verrà aggiunto in una versione futura.
- `TableName` - obbligatorio: stringa UTF-8, lunghezza non inferiore a 1 o non superiore a 128 byte.

  Il nome della tabella da replicare.

Risposta

- `ResourceArn`: stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Il nome della risorsa Amazon (ARN) della tabella di destinazione per cui recuperare le proprietà della tabella di integrazione. Attualmente, questa API supporta solo il recupero delle proprietà per le tabelle di destinazione e l'ARN fornito dovrebbe essere l'ARN della tabella di destinazione nel Data Catalog. AWS Glue Il supporto per il recupero delle proprietà della tabella di integrazione per le connessioni di origine (utilizzando l'ARN della connessione) non è ancora implementato e verrà aggiunto in una versione futura.
- `TableName`: stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Il nome della tabella da replicare.
- `SourceTableConfig`: un oggetto SourceTableConfig.

  Una struttura di configurazione della tabella di origine.

- `TargetTableConfig`: un oggetto TargetTableConfig.

  Una struttura di configurazione della tabella di destinazione.

Errori

- `ValidationException`
- `AccessDeniedException`
- `ResourceNotFoundException`
- `InternalServerException`
- `EntityNotFoundException`
- `InternalServiceException`
- `InvalidInputException`

## DeleteIntegrationTableProperties azione (Python: delete_integration_table_properties)

Elimina le proprietà della tabella create per le tabelle da replicare.

Richiesta

- `ResourceArn` - obbligatorio: stringa UTF-8, lunghezza non inferiore a 1 o non superiore a 128 byte.

  L'ARN di connessione dell'origine o l'ARN del database della destinazione.
- `TableName` - obbligatorio: stringa UTF-8, lunghezza non inferiore a 1 o non superiore a 128 byte.

  Il nome della tabella da replicare.

Risposta

- Nessun parametro di risposta.

Errori

- `ValidationException`

- `AccessDeniedException`
- `ResourceNotFoundException`
- `InternalServerException`
- `EntityNotFoundException`
- `InternalServiceException`
- `InvalidInputException`

## CreateIntegrationResourceProperty azione (Python: create_integration_resource_property)

Questa API può essere utilizzata per configurare la `ResourceProperty` AWS Glue connessione (per l'origine) o l'ARN del AWS Glue database (per la destinazione). Queste proprietà possono includere il ruolo per accedere alla connessione o al database. Per impostare sia le proprietà di origine che quelle di destinazione, è necessario richiamare la stessa API rispettivamente con la AWS Glue connessione ARN `ResourceArn` as `SourceProcessingProperties` with e il AWS Glue database `ResourceArn` ARN come with. `TargetProcessingProperties`

Richiesta

- `ResourceArn` - obbligatorio: stringa UTF-8, lunghezza non inferiore a 1 o non superiore a 128 byte.

  L'ARN di connessione dell'origine o l'ARN del database della destinazione.
- `SourceProcessingProperties`: un oggetto SourceProcessingProperties.

  Le proprietà della risorsa associate all'origine dell'integrazione.
- `TargetProcessingProperties`: un oggetto TargetProcessingProperties.

  Le proprietà della risorsa associate alla destinazione dell'integrazione.

Risposta

- `ResourceArn` - obbligatorio: stringa UTF-8, lunghezza non inferiore a 1 o non superiore a 128 byte.

  L'ARN di connessione dell'origine o l'ARN del database della destinazione.

- `SourceProcessingProperties`: un oggetto SourceProcessingProperties.

  Le proprietà della risorsa associate all'origine dell'integrazione.
- `TargetProcessingProperties`: un oggetto TargetProcessingProperties.

  Le proprietà della risorsa associate alla destinazione dell'integrazione.

Errori

- `ValidationException`
- `AccessDeniedException`
- `ConflictException`
- `InternalServerException`
- `ResourceNotFoundException`
- `EntityNotFoundException`
- `InternalServiceException`
- `InvalidInputException`

## UpdateIntegrationResourceProperty azione (Python: update_integration_resource_property)

Questa API può essere utilizzata per aggiornare la `ResourceProperty` AWS Glue connessione (per l'origine) o l'ARN del AWS Glue database (per la destinazione). Queste proprietà possono includere il ruolo per accedere alla connessione o al database. Dato che la stessa risorsa può essere utilizzata in più integrazioni, l'aggiornamento delle proprietà della risorsa influiranno su tutte le integrazioni che la utilizzano.

Richiesta

- `ResourceArn` - obbligatorio: stringa UTF-8, lunghezza non inferiore a 1 o non superiore a 128 byte.

  L'ARN di connessione dell'origine o l'ARN del database della destinazione.
- `SourceProcessingProperties`: un oggetto SourceProcessingProperties.

  Le proprietà della risorsa associate all'origine dell'integrazione.

- `TargetProcessingProperties`: un oggetto TargetProcessingProperties.

  Le proprietà della risorsa associate alla destinazione dell'integrazione.

Risposta

- `ResourceArn`: stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  L'ARN di connessione dell'origine o l'ARN del database della destinazione.
- `SourceProcessingProperties`: un oggetto SourceProcessingProperties.

  Le proprietà della risorsa associate all'origine dell'integrazione.
- `TargetProcessingProperties`: un oggetto TargetProcessingProperties.

  Le proprietà della risorsa associate alla destinazione dell'integrazione.

Errori

- `ValidationException`
- `AccessDeniedException`
- `InternalServerException`
- `ResourceNotFoundException`
- `EntityNotFoundException`
- `InternalServiceException`
- `InvalidInputException`

## GetIntegrationResourceProperty azione (Python: get_integration_resource_property)

Questa API viene utilizzata per recuperare l'ARN `ResourceProperty` della AWS Glue connessione (per l'origine) o del AWS Glue database (per la destinazione)

Richiesta

- `ResourceArn` - obbligatorio: stringa UTF-8, lunghezza non inferiore a 1 o non superiore a 128 byte.

L'ARN di connessione dell'origine o l'ARN del database della destinazione.

Risposta

- `ResourceArn`: stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  L'ARN di connessione dell'origine o l'ARN del database della destinazione.
- `SourceProcessingProperties`: un oggetto SourceProcessingProperties.

  Le proprietà della risorsa associate all'origine dell'integrazione.
- `TargetProcessingProperties`: un oggetto TargetProcessingProperties.

  Le proprietà della risorsa associate alla destinazione dell'integrazione.

Errori

- `ValidationException`
- `AccessDeniedException`
- `InternalServerException`
- `ResourceNotFoundException`
- `EntityNotFoundException`
- `InternalServiceException`
- `InvalidInputException`

## UntagResource azione (Python: untag_resource)

Rimuove i tag specificati da una risorsa di integrazione.

Richiesta

- `ResourceArn` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 10240 byte di lunghezza, corrispondente a Custom string pattern #50.

  Il nome della risorsa Amazon (ARN) della risorsa di integrazione.
- `TagsToRemove` - obbligatorio: una matrice di stringhe UTF-8, non superiore a 50 stringhe.

Un elenco di tag di metadati da rimuovere dalla risorsa.

Risposta

- Nessun parametro di risposta.

Errori

- `ResourceNotFoundException`

## ListTagsForResource azione (Python: list_tags_for_resource)

Elenca i tag di metadati assegnati alla risorsa specificata.

Richiesta

- `ResourceARN` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 10240 byte di lunghezza, corrispondente a Custom string pattern #50.

  L'ARN della risorsa.

Risposta

- `Tags`: una serie di oggetti Tag, non superiore a 10 strutture.

  Un elenco di tag.

Errori

- `ResourceNotFoundException`

## Eccezioni

- ResourceNotFoundException struttura
- InternalServerException struttura
- IntegrationAlreadyExistsFault struttura

- IntegrationConflictOperationFault struttura
- IntegrationQuotaExceededFault struttura
- KMSKeyNotAccessibleFault struttura
- IntegrationNotFoundFault struttura
- TargetResourceNotFound struttura
- InvalidIntegrationStateFault struttura

## ResourceNotFoundException struttura

Impossibile trovare la risorsa.

Campi

- `Message`: stringa UTF-8.

  Messaggio che descrive il problema.

## InternalServerException struttura

Si è verificato un errore interno del server.

Campi

- `Message`: stringa UTF-8.

  Messaggio che descrive il problema.

## IntegrationAlreadyExistsFault struttura

L'integrazione specificata esiste già.

Campi

- `Message`: stringa UTF-8.

  Messaggio che descrive il problema.

## IntegrationConflictOperationFault struttura

L'operazione richiesta è in conflitto con un'altra operazione.

Campi

- `Message`: stringa UTF-8.

  Messaggio che descrive il problema.

## IntegrationQuotaExceededFault struttura

I dati elaborati tramite l'integrazione hanno superato la quota.

Campi

- `Message`: stringa UTF-8.

  Messaggio che descrive il problema.

## KMSKeyNotAccessibleFault struttura

La chiave KMS specificata non è accessibile.

Campi

- `Message`: stringa UTF-8.

  Messaggio che descrive il problema.

## IntegrationNotFoundFault struttura

Impossibile trovare l'integrazione specificata.

Campi

- `Message`: stringa UTF-8.

  Messaggio che descrive il problema.

## TargetResourceNotFound struttura

Impossibile trovare la risorsa di destinazione.

Campi

- `Message`: stringa UTF-8.

  Messaggio che descrive il problema.

## InvalidIntegrationStateFault struttura

L'integrazione si trova in uno stato non valido.

Campi

- `Message`: stringa UTF-8.

  Messaggio che descrive il problema.

# API Sessioni interattive

L'API delle sessioni interattive descrive l' AWS Glue API relativa all'utilizzo di sessioni AWS Glue interattive per creare e testare script di estrazione, trasformazione e caricamento (ETL) per l'integrazione dei dati.

## Tipi di dati

- Struttura sessione
- SessionCommand struttura
- Struttura istruzione
- StatementOutput struttura
- StatementOutputData struttura
- ConnectionsList struttura

## Struttura sessione

Il periodo in cui è in esecuzione un ambiente di runtime Spark remoto.

Campi

- `Id`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID della sessione.

- `CreatedOn`: timestamp.

  La data e l'ora di creazione della sessione.

- `Status`: stringa UTF-8 (valori validi: `PROVISIONING` | `READY` | `FAILED` | `TIMEOUT` | `STOPPING` | `STOPPED`).

  Lo stato della sessione.

- `ErrorMessage`: stringa di descrizione, non superiore a 2048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  Il messaggio di errore visualizzato durante la sessione.

- `Description`: stringa di descrizione, non superiore a 2048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  La descrizione della sessione.

- `Role`: stringa UTF-8, non inferiore a 20 o superiore a 2048 byte di lunghezza, corrispondente a Custom string pattern #30.

  Il nome o l'Amazon Resource Name (ARN) del ruolo IAM associato alla sessione.

- `Command`: un oggetto SessionCommand.

  Il comando Object.see. SessionCommand

- `DefaultArguments`: una matrice di mappe con coppie chiave-valore, non superiore alle 75 coppie.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza, corrispondente a Custom string pattern #31.

  Ogni valore è una stringa UTF-8, non superiore a 4096 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  Una matrice della mappa di coppie chiave-valore. Il massimo è 75 coppie.

- `Connections`: un oggetto ConnectionsList.

  Il numero di connessioni utilizzate per la sessione.

- `Progress`: numero (doppio).

  L'avanzamento dell'esecuzione del codice della sessione.

- `MaxCapacity`: numero (doppio).

  Il numero di unità di elaborazione AWS Glue dati (DPUs) che possono essere allocate durante l'esecuzione del processo. Una DPU è una misura relativa della potenza di elaborazione costituita da 4 V di capacità CPUs di elaborazione e 16 GB di memoria.

- `SecurityConfiguration`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome della SecurityConfiguration struttura da utilizzare con la sessione.

- `GlueVersion`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Custom string pattern #48.

  La AWS Glue versione determina le versioni di Apache Spark e Python supportate. AWS Glue GlueVersion Deve essere maggiore di 2.0.

- `DataAccessId`: stringa UTF-8, non inferiore a 1 o superiore a 36 byte di lunghezza.

  L'ID di accesso ai dati della sessione.

- `PartitionId`: stringa UTF-8, non inferiore a 1 o superiore a 36 byte di lunghezza.

  L'ID di partizione della sessione.

- `NumberOfWorkers`: numero (intero).

  Il numero di dipendenti di uno specifico `WorkerType` da utilizzare per la sessione.

- `WorkerType`: stringa UTF-8 (valori validi: `Standard=""` | `G.1X=""` | `G.2X=""` | `G.025X=""` | `G.4X=""` | `G.8X=""` | `Z.2X=""`).

  Il tipo di worker predefinito allocato quando viene eseguita una sessione. Accetta un valore di `G.1X`, `G.2X`, `G.4X` o `G.8X` per le sessioni Spark. Accetta il valore `Z.2X` per le sessioni Ray.

- `CompletedOn`: timestamp.

  La data e ora in cui questa sessione è stata completata.

- `ExecutionTime`: numero (doppio).

Il tempo totale di esecuzione della sessione.

- `DPUSeconds`: numero (doppio).

  Il DPUs consumo della sessione (formula: ExecutionTime * MaxCapacity).

- `IdleTimeout`: numero (intero).

  Il numero di minuti di inattività prima del timeout della sessione.

- `ProfileName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome di un profilo di AWS Glue utilizzo associato alla sessione.

## SessionCommand struttura

Il `SessionCommand` che esegue questo lavoro.

Campi

- `Name`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Speciifica il nome di. SessionCommand Può essere 'glueetl' o 'gluestreaming'.

- `PythonVersion`: stringa UTF-8, corrispondente a Custom string pattern #49.

  Specifica la versione di Python. La versione Python indica la versione supportata per i processi di tipo Spark.

## Struttura istruzione

La dichiarazione o la richiesta di un'operazione particolare in una sessione.

Campi

- `Id`: numero (intero).

  L'ID della dichiarazione.

- `Code`: stringa UTF-8.

Il codice di esecuzione della dichiarazione.

- `State`: stringa UTF-8 (valori validi: `WAITING` | `RUNNING` | `AVAILABLE` | `CANCELLING` | `CANCELLED` | `ERROR`).

  Lo stato mentre viene eseguita la richiesta.

- `Output`: un oggetto StatementOutput.

  L'output in JSON.

- `Progress`: numero (doppio).

  L'avanzamento dell'esecuzione del codice.

- `StartedOn`: numero (lungo).

  L'ora e la data unix in cui è stata avviata la definizione del processo.

- `CompletedOn`: numero (lungo).

  L'ora e la data unix in cui è stata completata la definizione del processo.

## StatementOutput struttura

Output dell'esecuzione del codice in formato JSON.

Campi

- `Data`: un oggetto StatementOutputData.

  L'output dell'esecuzione del codice.

- `ExecutionCount`: numero (intero).

  Il numero di esecuzioni dell'output.

- `Status`: stringa UTF-8 (valori validi: `WAITING` | `RUNNING` | `AVAILABLE` | `CANCELLING` | `CANCELLED` | `ERROR`).

  Lo stato dell'output di esecuzione del codice.

- `ErrorName`: stringa UTF-8.

  Il nome dell'errore nell'output.

- `ErrorValue`: stringa UTF-8.

  Il valore dell'errore dell'output.
- `Traceback`: una matrice di stringhe UTF-8.

  L'analisi dell'output.

## StatementOutputData struttura

Output dell'esecuzione del codice in formato JSON.

Campi

- `TextPlain`: stringa UTF-8.

  L'output dell'esecuzione del codice in formato testo.

## ConnectionsList struttura

Specifica le connessioni utilizzate da un processo.

Campi

- `Connections`— Un array di stringhe UTF-8, non più di 1000 stringhe.

  Un elenco di connessioni utilizzate dal processo.

## Operazioni

- CreateSession azione (Python: create_session)
- StopSession azione (Python: stop_session)
- DeleteSession azione (Python: delete_session)
- GetSession azione (Python: get_session)
- ListSessions azione (Python: list_sessions)
- RunStatement azione (Python: run_statement)
- CancelStatement azione (Python: cancel_statement)
- GetStatement azione (Python: get_statement)

- ListStatements azione (Python: list_statements)
- GetGlueIdentityCenterConfiguration azione (Python: get_glue_identity_center_configuration)
- UpdateGlueIdentityCenterConfiguration azione (Python: update_glue_identity_center_configuration)
- CreateGlueIdentityCenterConfiguration azione (Python: create_glue_identity_center_configuration)
- DeleteGlueIdentityCenterConfiguration azione (Python: delete_glue_identity_center_configuration)

## CreateSession azione (Python: create_session)

Crea una nuova sessione.

Richiesta

Richiedere la creazione di una nuova sessione.

- `Id` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID della richiesta della sessione.
- `Description`: stringa di descrizione, non superiore a 2048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  La descrizione della sessione.
- `Role` - obbligatorio: stringa UTF-8, non inferiore a 20 o superiore a 2048 byte di lunghezza, corrispondente a Custom string pattern #30.

  L'ARN del ruolo IAM
- `Command` - obbligatorio: un oggetto SessionCommand.

  Il `SessionCommand` che esegue questo lavoro.
- `Timeout`: numero (intero), almeno 1.

  Il numero di minuti prima che la sessione scada. Il valore predefinito per i processi Spark ETL è 48 ore (2.880 minuti). Consulta la documentazione per altri tipi di processo.
- `IdleTimeout`: numero (intero), almeno 1.

  Il numero di minuti di inattività prima del timeout della sessione. L'impostazione predefinita per i processi ETL di Spark è il valore di timeout. Consulta la documentazione per altri tipi di processo.

- `DefaultArguments`: una matrice di mappe con coppie chiave-valore, non superiore alle 75 coppie.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza, corrispondente a Custom string pattern #31.

  Ogni valore è una stringa UTF-8, non superiore a 4096 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  Una matrice della mappa di coppie chiave-valore. Il massimo è 75 coppie.

- `Connections`: un oggetto ConnectionsList.

  Il numero di connessioni da utilizzare per la sessione.

- `MaxCapacity`: numero (doppio).

  Il numero di unità di elaborazione AWS Glue dati (DPUs) che possono essere allocate durante l'esecuzione del processo. Una DPU è una misura relativa della potenza di elaborazione costituita da 4 V di capacità CPUs di elaborazione e 16 GB di memoria.

- `NumberOfWorkers`: numero (intero).

  Il numero di dipendenti di uno specifico `WorkerType` da utilizzare per la sessione.

- `WorkerType`: stringa UTF-8 (valori validi: `Standard=""` | `G.1X=""` | `G.2X=""` | `G.025X=""` | `G.4X=""` | `G.8X=""` | `Z.2X=""`).

  Il tipo di worker predefinito allocato quando viene eseguito un processo. Accetta un valore di G.1X, G.2X, G.4X o G.8X per i processi Spark. Accetta il valore Z.2X per i notebook Ray.

  - Per il tipo di `G.1X` worker, ogni worker esegue il mapping su 1 DPU (4 vCPUs, 16 GB di memoria) con disco da 94 GB e fornisce 1 esecutore per lavoratore. Questi tipi di worker sono raccomandati per carichi di lavoro come trasformazioni di dati, join e query, in quanto offrono un modo scalabile ed economico per eseguire la maggior parte dei processi.
  - Per il tipo di `G.2X` worker, ogni worker esegue il mapping a 2 DPU (8 vCPUs, 32 GB di memoria) con disco da 138 GB e fornisce 1 esecutore per lavoratore. Questi tipi di worker sono raccomandati per carichi di lavoro come trasformazioni di dati, join e query, in quanto offrono un modo scalabile ed economico per eseguire la maggior parte dei processi.
  - Per il tipo di `G.4X` worker, ogni worker esegue il mapping a 4 DPU (16 vCPUs, 64 GB di memoria) con disco da 256 GB e fornisce 1 esecutore per lavoratore. Questi tipi di worker sono raccomandati per i processi i cui carichi di lavoro contengono trasformazioni, aggregazioni, join

e query con i requisiti più elevati. Questo tipo di lavoratore è disponibile solo per i job Spark ETL AWS Glue versione 3.0 o successiva AWS nelle seguenti regioni: Stati Uniti orientali (Ohio), Stati Uniti orientali (Virginia settentrionale), Stati Uniti occidentali (Oregon), Asia Pacifico (Singapore), Asia Pacifico (Sydney), Asia Pacifico (Tokyo), Canada (Centrale), Europa (Francoforte), Europa (Irlanda) ed Europa (Stoccolma).

- Per il tipo di `G.8X` worker, ogni worker esegue il mapping a 8 DPU (32 vCPUs, 128 GB di memoria) con disco da 512 GB e fornisce 1 esecutore per lavoratore. Questi tipi di worker sono raccomandati per i processi i cui carichi di lavoro contengono trasformazioni, aggregazioni, join e query con i requisiti più elevati. Questo tipo di worker è disponibile solo per i job Spark ETL AWS Glue versione 3.0 o successiva, nelle stesse AWS regioni supportate per il tipo di lavoratore. `G.4X`
- Per il tipo di `Z.2X` worker, ogni worker esegue il mapping su 2 M-DPU (8vCPUs, 64 GB di memoria) con disco da 128 GB e fornisce fino a 8 Ray worker in base all'autoscaler.

- `SecurityConfiguration`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome della SecurityConfiguration struttura da utilizzare con la sessione

- `GlueVersion`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Custom string pattern #48.

  La AWS Glue versione determina le versioni di Apache Spark e Python supportate. AWS Glue GlueVersion Deve essere maggiore di 2.0.

- `DataAccessId`: stringa UTF-8, non inferiore a 1 o superiore a 36 byte di lunghezza.

  L'ID di accesso ai dati della sessione.

- `PartitionId`: stringa UTF-8, non inferiore a 1 o superiore a 36 byte di lunghezza.

  L'ID di partizione della sessione.

- `Tags`: una matrice di mappe con coppie chiave-valore, non superiore alle 50 coppie.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Ogni valore è una stringa UTF-8, lunga non più di 256 byte.

  La mappa delle coppie di valori chiave (tag) appartenenti alla sessione.

- `RequestOrigin`: stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza, corrispondente a Custom string pattern #31.

L'origine della richiesta.

- `ProfileName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome di un profilo di AWS Glue utilizzo associato alla sessione.

Risposta

- `Session`: un oggetto Sessione.

  Restituisce l'oggetto di sessione nella risposta.

Errori

- `AccessDeniedException`
- `IdempotentParameterMismatchException`
- `InternalServiceException`
- `OperationTimeoutException`
- `InvalidInputException`
- `ValidationException`
- `AlreadyExistsException`
- `ResourceNumberLimitExceededException`

## StopSession azione (Python: stop_session)

Interrompe la sessione.

Richiesta

- `Id` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID della sessione da interrompere.
- `RequestOrigin`: stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza, corrispondente a Custom string pattern #31.

L'origine della richiesta.

Risposta

- `Id`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Restituisce l'ID della sessione interrotta.

Errori

- `AccessDeniedException`
- `InternalServiceException`
- `OperationTimeoutException`
- `InvalidInputException`
- `IllegalSessionStateException`
- `ConcurrentModificationException`

## DeleteSession azione (Python: delete_session)

Elimina la sessione.

Richiesta

- `Id` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID della sessione da eliminare.
- `RequestOrigin`: stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza, corrispondente a Custom string pattern #31.

  Il nome dell'origine della richiesta di eliminazione della sessione.

Risposta

- `Id`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Restituisce l'ID della sessione eliminata.

Errori

- `AccessDeniedException`
- `InternalServiceException`
- `OperationTimeoutException`
- `InvalidInputException`
- `IllegalSessionStateException`
- `ConcurrentModificationException`

## GetSession azione (Python: get_session)

Recupera la sessione.

Richiesta

- `Id` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID della sessione.
- `RequestOrigin`: stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza, corrispondente a Custom string pattern #31.

  L'origine della richiesta.

Risposta

- `Session`: un oggetto Sessione.

  Salva l'oggetto di sessione restituito nella risposta.

Errori

- `AccessDeniedException`
- `EntityNotFoundException`
- `InternalServiceException`
- `OperationTimeoutException`
- `InvalidInputException`

## ListSessions azione (Python: list_sessions)

Recupera un elenco di sessioni.

Richiesta

- `NextToken`: stringa UTF-8, non superiore a 400000 byte di lunghezza.

  Il token per il successivo set di risultati oppure null se non ci sono altri risultati.
- `MaxResults`: numero (intero), non inferiore a 1 o superiore a 1.000.

  Il numero massimo di risultati.
- `Tags`: una matrice di mappe con coppie chiave-valore, non superiore alle 50 coppie.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Ogni valore è una stringa UTF-8, lunga non più di 256 byte.

  Tag appartenenti alla sessione.
- `RequestOrigin`: stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza, corrispondente a Custom string pattern #31.

  L'origine della richiesta.

Risposta

- `Ids`: una matrice di stringhe UTF-8.

  Restituisce l'ID della sessione.
- `Sessions`: una matrice di oggetti Sessione.

Restituisce l'oggetto di sessione.

- `NextToken`: stringa UTF-8, non superiore a 400000 byte di lunghezza.

  Il token per il successivo set di risultati oppure null se non ci sono altri risultati.

Errori

- `AccessDeniedException`
- `InvalidInputException`
- `InternalServiceException`
- `OperationTimeoutException`

## RunStatement azione (Python: run_statement)

Esegue l'istruzione.

Richiesta

- `SessionId` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID di sessione dell'istruzione da eseguire.
- `Code` - obbligatorio: stringa UTF-8, non superiore a 68000 byte di lunghezza.

  Il codice dell'istruzione da eseguire.
- `RequestOrigin`: stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza, corrispondente a Custom string pattern #31.

  L'origine della richiesta.

Risposta

- `Id`: numero (intero).

  Restituisce l'ID dell'istruzione che è stata eseguita.

Errori

- `EntityNotFoundException`
- `AccessDeniedException`
- `InternalServiceException`
- `OperationTimeoutException`
- `InvalidInputException`
- `ValidationException`
- `ResourceNumberLimitExceededException`
- `IllegalSessionStateException`

## CancelStatement azione (Python: cancel_statement)

Annulla l'istruzione.

Richiesta

- `SessionId` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID di sessione dell'istruzione da annullare.
- `Id` - obbligatorio: numero (intero).

  L'ID dell'istruzione da annullare.
- `RequestOrigin`: stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza, corrispondente a Custom string pattern #31.

  L'origine della richiesta di annullare l'istruzione.

Risposta

- Nessun parametro di risposta.

Errori

- `AccessDeniedException`

- `EntityNotFoundException`
- `InternalServiceException`
- `OperationTimeoutException`
- `InvalidInputException`
- `IllegalSessionStateException`

## GetStatement azione (Python: get_statement)

Recupera l'istruzione.

Richiesta

- `SessionId` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID sessione dell'istruzione.
- `Id` - obbligatorio: numero (intero).

  L'ID dell'istruzione.
- `RequestOrigin`: stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza, corrispondente a Custom string pattern #31.

  L'origine della richiesta.

Risposta

- `Statement`: un oggetto Dichiarazione.

  Restituisce l'istruzione.

Errori

- `AccessDeniedException`
- `EntityNotFoundException`
- `InternalServiceException`
- `OperationTimeoutException`

- `InvalidInputException`
- `IllegalSessionStateException`

## ListStatements azione (Python: list_statements)

Elenca le istruzioni per la sessione.

Richiesta

- `SessionId` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID sessione delle istruzioni.
- `RequestOrigin`: stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza, corrispondente a Custom string pattern #31.

  L'origine della richiesta di elencare le istruzioni.
- `NextToken`: stringa UTF-8, non superiore a 400000 byte di lunghezza.

  Un token di continuazione, se si tratta di una chiamata di continuazione.

Risposta

- `Statements`: una matrice di oggetti Dichiarazione.

  Restituisce l'elenco delle istruzioni.
- `NextToken`: stringa UTF-8, non superiore a 400000 byte di lunghezza.

  Un token di continuazione, se non sono ancora stati restituiti tutte le istruzioni.

Errori

- `AccessDeniedException`
- `EntityNotFoundException`
- `InternalServiceException`
- `OperationTimeoutException`
- `InvalidInputException`

- `IllegalSessionStateException`

## GetGlueIdentityCenterConfiguration azione (Python: get_glue_identity_center_configuration)

Recupera i dettagli della configurazione corrente di Identity Center, incluse le informazioni sull'istanza e sull'applicazione di AWS Glue Identity Center associata.

Richiesta

- Nessun parametro della richiesta.

Risposta

Risposta contenente i dettagli di configurazione AWS Glue dell'Identity Center.

- `ApplicationArn`— Stringa UTF-8, lunga non meno di 10 o più di 1224 byte.

  L'Amazon Resource Name (ARN) dell'applicazione Identity Center associata alla AWS Glue configurazione.
- `InstanceArn`— Stringa UTF-8, lunga non meno di 10 o più di 1224 byte.

  L'Amazon Resource Name (ARN) dell'istanza Identity Center associata alla AWS Glue configurazione.
- `Scopes`: una matrice di stringhe UTF-8.

  Un elenco di ambiti di Identity Center che definiscono le autorizzazioni e i livelli di accesso per la configurazione. AWS Glue

Errori

- `InvalidInputException`
- `EntityNotFoundException`
- `InternalServiceException`
- `OperationTimeoutException`
- `AccessDeniedException`
- `ConcurrentModificationException`

## UpdateGlueIdentityCenterConfiguration azione (Python: update_glue_identity_center_configuration)

Aggiorna la configurazione esistente dell' AWS Glue Identity Center, consentendo la modifica degli ambiti e delle autorizzazioni per l'integrazione.

Richiesta

Richiesta di aggiornamento di una configurazione esistente di AWS Glue Identity Center.

- `Scopes`: una matrice di stringhe UTF-8, non inferiore a 1 o superiore a 50 stringhe.

  Un elenco di ambiti dell'Identity Center che definiscono le autorizzazioni e i livelli di accesso aggiornati per la AWS Glue configurazione.

Risposta

- Nessun parametro di risposta.

Errori

- `InvalidInputException`
- `EntityNotFoundException`
- `InternalServiceException`
- `OperationTimeoutException`
- `AccessDeniedException`
- `ConcurrentModificationException`

## CreateGlueIdentityCenterConfiguration azione (Python: create_glue_identity_center_configuration)

Crea una nuova configurazione di Identity Center per consentire l'integrazione tra e IAM AWS Glue Identity Center per l'autenticazione e l'autorizzazione. AWS Glue AWS

Richiesta

Richiesta di creazione di una nuova configurazione di AWS Glue Identity Center.

- `InstanceArn` - obbligatorio: stringa UTF-8, lunghezza non inferiore a 10 o non superiore a 1224 byte.

  L'Amazon Resource Name (ARN) dell'istanza Identity Center da associare alla AWS Glue configurazione.
- `Scopes`: una matrice di stringhe UTF-8, non inferiore a 1 o superiore a 50 stringhe.

  Un elenco di ambiti di Identity Center che definiscono le autorizzazioni e i livelli di accesso per la configurazione. AWS Glue

Risposta

Risposta derivante dalla creazione di una nuova configurazione di AWS Glue Identity Center.

- `ApplicationArn`— Stringa UTF-8, lunga non meno di 10 o più di 1224 byte.

  L'Amazon Resource Name (ARN) dell'applicazione Identity Center creata per la AWS Glue configurazione.

Errori

- `InvalidInputException`
- `AlreadyExistsException`
- `InternalServiceException`
- `OperationTimeoutException`
- `AccessDeniedException`
- `ConcurrentModificationException`

## DeleteGlueIdentityCenterConfiguration azione (Python: delete_glue_identity_center_configuration)

Elimina la configurazione esistente di Identity Center, rimuovendo l'integrazione tra e AWS Glue IAM Identity Center. AWS Glue AWS

Richiesta

- Nessun parametro della richiesta.

Risposta

- Nessun parametro di risposta.

Errori

- `InvalidInputException`
- `EntityNotFoundException`
- `InternalServiceException`
- `OperationTimeoutException`
- `AccessDeniedException`
- `ConcurrentModificationException`

# API endpoint di sviluppo

L'API Development endpoints descrive l' AWS Glue API relativa ai test utilizzando una soluzione personalizzata DevEndpoint.

## Tipi di dati

- DevEndpoint struttura
- DevEndpointCustomLibraries struttura

## DevEndpoint struttura

Un endpoint di sviluppo in cui uno sviluppatore può eseguire in remoto il debug, la trasformazione e il caricamento degli script ETL.

Campi

- `EndpointName`: stringa UTF-8.

  Nome della `DevEndpoint`.
- `RoleArn`: stringa UTF-8, corrispondente a AWS IAM ARN string pattern.

  Amazon Resource Name (ARN) del ruolo IAM utilizzato in questo `DevEndpoint`.

- `SecurityGroupIds`: una matrice di stringhe UTF-8.

  Un elenco degli identificatori dei gruppi di sicurezza utilizzati in questo `DevEndpoint`.

- `SubnetId`: stringa UTF-8.

  La sottorete ID per questo `DevEndpoint`.

- `YarnEndpointAddress`: stringa UTF-8.

  L'indirizzo dell'endpoint YARN utilizzato da questo `DevEndpoint`.

- `PrivateAddress`: stringa UTF-8.

  Un indirizzo IP privato per accedere `DevEndpoint` all'interno di un VPC se in uno di essi viene creato `DevEndpoint` Il campo `PrivateAddress` è presente solo quando viene creato `DevEndpoint` all'interno del VPC.

- `ZeppelinRemoteSparkInterpreterPort`: numero (intero).

  La porta Apache Zeppelin per l'interprete Apache Spark remoto.

- `PublicAddress`: stringa UTF-8.

  L'indirizzo IP pubblico utilizzato da questo `DevEndpoint`. Il campo `PublicAddress` è presente solo quando si crea un `DevEndpoint` non VPC.

- `Status`: stringa UTF-8.

  Lo stato corrente di questo `DevEndpoint`.

- `WorkerType`: stringa UTF-8 (valori validi: `Standard=""` | `G.1X=""` | `G.2X=""` | `G.025X=""` | `G.4X=""` | `G.8X=""` | `Z.2X=""`).

  Il tipo di worker predefinito allocato all'endpoint di sviluppo. Accetta un valore Standard, G.1X o G.2X.

  - Per il tipo di worker `Standard`, ciascun worker fornisce 4 vCPU, 16 GB di memoria, disco da 50 GB e 2 esecutori.
  - Per il tipo di worker `G.1X`, ciascun worker si mappa a 1 DPU (4 vCPU, 16 GB di memoria, disco da 64 GB) e fornisce 1 esecutore. Consigliamo questo tipo di lavoratore per lavori che richiedono molta memoria.
  - Per il tipo di worker `G.2X`, ciascun worker si mappa a 2 DPU (8 vCPU, 32 GB di memoria, disco da 128 GB) e fornisce 1 esecutore. Consigliamo questo tipo di lavoratore per lavori che richiedono molta memoria.

Problema noto: quando viene creato un endpoint di sviluppo con la configurazione `G.2X` `WorkerType`, i driver Spark per l'endpoint di sviluppo verranno eseguiti su 4 vCPU, 16 GB di memoria e un disco da 64 GB.

- `GlueVersion`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Custom string pattern #48.

  La versione Glue determina le versioni di Apache Spark e Python supportate. AWS Glue La versione Python indica la versione supportata per l'esecuzione di script ETL su endpoint di sviluppo.

  Per ulteriori informazioni sulle AWS Glue versioni disponibili e sulle versioni corrispondenti di Spark e Python, consulta la versione Glue nella guida per sviluppatori.

  Endpoint di sviluppo creati senza specificare una versione di Glue predefinita su Glue 0.9.

  È possibile specificare una versione del supporto Python per gli endpoint di sviluppo utilizzando il `Arguments` parametro in o. `CreateDevEndpoint` `UpdateDevEndpoint` APIs Se non vengono forniti argomenti, per impostazione predefinita viene utilizzata la versione Python 2.

- `NumberOfWorkers`: numero (intero).

  Il numero di worker di un `workerType` definito allocati all'endpoint di sviluppo.

  Il numero massimo di worker che è possibile definire è 299 `G.1X` e 149 per `G.2X`.

- `NumberOfNodes`: numero (intero).

  Il numero di unità di elaborazione AWS Glue dati (DPUs) assegnate a questo. `DevEndpoint`

- `AvailabilityZone`: stringa UTF-8.

  La zona di AWS disponibilità in cui si `DevEndpoint` trova.

- `VpcId`: stringa UTF-8.

  L'ID del Virtual Private Cloud (VPC) utilizzato da questo `DevEndpoint`.

- `ExtraPythonLibsS3Path`: stringa UTF-8.

  Percorsi a una o più librerie Python in un bucket Amazon S3 che devono essere caricati nel `DevEndpoint`. I valori multipli devono essere percorsi completi separati dalla virgola.

> **Note**
>
> Con un `DevEndpoint` è possibile utilizzare solo librerie Python pure. Le librerie che si basano sulle estensioni C, come la libreria di analisi dati Python pandas, non sono ancora supportate.

- `ExtraJarsS3Path`: stringa UTF-8.

  Percorsi a uno o più file `.jar` Java in un bucket S3 che devono essere caricati nel `DevEndpoint`.

> **Note**
>
> Puoi usare Java/Scala librerie pure solo con a`DevEndpoint`.

- `FailureReason`: stringa UTF-8.

  Il motivo di un errore corrente in questo `DevEndpoint`.

- `LastUpdateStatus`: stringa UTF-8.

  Lo stato dell'ultimo aggiornamento.

- `CreatedTimestamp`: timestamp.

  Il momento in cui è DevEndpoint stato creato.

- `LastModifiedTimestamp`: timestamp.

  Il momento dell'ultima modifica di questo `DevEndpoint`.

- `PublicKey`: stringa UTF-8.

  La chiave pubblica che deve essere utilizzata da questo `DevEndpoint` per l'autenticazione. Questo attributo viene fornito per la compatibilità con le versioni precedenti, in quanto l'attributo consigliato da usare è quello delle chiavi pubbliche.

- `PublicKeys`: una matrice di stringhe UTF-8, non più di 5 stringhe.

  Elenco di chiavi pubbliche che devono essere utilizzate da `DevEndpoints` per l'autenticazione. L'uso di questo attributo è preferibile rispetto a una singola chiave pubblica, perché le chiavi pubbliche permettono di avere una chiave privata diversa per ogni client.

> **Note**
>
> Se è già stato creato un endpoint con una chiave pubblica, è necessario rimuovere tale chiave per poter impostare un elenco di chiavi pubbliche. Chiama l'operazione API `UpdateDevEndpoint` con il contenuto della chiave pubblica nell'attributo `deletePublicKeys` e l'elenco delle nuove chiavi nell'attributo `addPublicKeys`.

- `SecurityConfiguration`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome della struttura `SecurityConfiguration` da utilizzare con questo `DevEndpoint`.

- `Arguments` – Una matrice di mappe con coppie chiave-valore, non superiore alle 100 coppie.

  Ogni chiave è una stringa UTF-8.

  Ogni valore è una stringa UTF-8.

  Mappa di argomenti usati per configurare `DevEndpoint`.

  Argomenti validi sono:

  - `"--enable-glue-datacatalog": ""`

  È possibile specificare una versione del supporto Python per gli endpoint di sviluppo utilizzando il `Arguments` parametro in o. `CreateDevEndpoint` `UpdateDevEndpoint` APIs Se non vengono forniti argomenti, per impostazione predefinita viene utilizzata la versione Python 2.

## DevEndpointCustomLibraries struttura

Librerie personalizzate da caricare in un endpoint di sviluppo.

Campi

- `ExtraPythonLibsS3Path`: stringa UTF-8.

  I percorsi a una o più librerie Python in un bucket Amazon Simple Storage Service (Amazon S3) che devono essere caricati nel `DevEndpoint`. I valori multipli devono essere percorsi completi separati dalla virgola.

> **Note**
>
> Con un `DevEndpoint` è possibile utilizzare solo librerie Python pure. Le librerie che si basano sulle estensioni C, come la libreria di analisi dati Python pandas, non sono ancora supportate.

- `ExtraJarsS3Path`: stringa UTF-8.

  Percorsi a uno o più file `.jar` Java in un bucket S3 che devono essere caricati nel `DevEndpoint`.

> **Note**
>
> Puoi usare Java/Scala librerie pure solo con a`DevEndpoint`.

## Operazioni

- CreateDevEndpoint azione (Python: create_dev_endpoint)
- UpdateDevEndpoint azione (Python: update_dev_endpoint)
- DeleteDevEndpoint azione (Python: delete_dev_endpoint)
- GetDevEndpoint azione (Python: get_dev_endpoint)
- GetDevEndpoints azione (Python: get_dev_endpoints)
- BatchGetDevEndpoints azione (Python: batch_get_dev_endpoints)
- ListDevEndpoints azione (Python: list_dev_endpoints)

## CreateDevEndpoint azione (Python: create_dev_endpoint)

Crea un nuovo endpoint di sviluppo.

Richiesta

- `EndpointName` - obbligatorio: stringa UTF-8.

  Il nome da assegnare al nuovo `DevEndpoint`.

- `RoleArn`: obbligatorio: stringa UTF-8, corrispondente a AWS IAM ARN string pattern.

Il ruolo IAM per il `DevEndpoint`.

- `SecurityGroupIds`: una matrice di stringhe UTF-8.

  Gruppo di sicurezza IDs per i gruppi di sicurezza che verranno utilizzati dal nuovo. `DevEndpoint`

- `SubnetId`: stringa UTF-8.

  La sottorete ID per il nuovo `DevEndpoint` da utilizzare.

- `PublicKey`: stringa UTF-8.

  La chiave pubblica che deve essere utilizzata da questo `DevEndpoint` per l'autenticazione. Questo attributo viene fornito per la compatibilità con le versioni precedenti, in quanto l'attributo consigliato da usare è quello delle chiavi pubbliche.

- `PublicKeys`: una matrice di stringhe UTF-8, non più di 5 stringhe.

  Elenco di chiavi pubbliche che devono essere usate dagli endpoint di sviluppo per l'autenticazione. L'uso di questo attributo è preferibile rispetto a una singola chiave pubblica, perché le chiavi pubbliche permettono di avere una chiave privata diversa per ogni client.

  > **Note**
  >
  > Se è già stato creato un endpoint con una chiave pubblica, è necessario rimuovere tale chiave per poter impostare un elenco di chiavi pubbliche. Chiama l'API `UpdateDevEndpoint` con il contenuto della chiave pubblica nell'attributo `deletePublicKeys` e l'elenco delle nuove chiavi nell'attributo `addPublicKeys`.

- `NumberOfNodes`: numero (intero).

  Il numero di unità di elaborazione AWS Glue dati (DPUs) da assegnare a questo`DevEndpoint`.

- `WorkerType`: stringa UTF-8 (valori validi: `Standard=""` | `G.1X=""` | `G.2X=""` | `G.025X=""` | `G.4X=""` | `G.8X=""` | `Z.2X=""`).

  Il tipo di worker predefinito allocato all'endpoint di sviluppo. Accetta un valore Standard, G.1X o G.2X.

  - Per il tipo di worker `Standard`, ciascun worker fornisce 4 vCPU, 16 GB di memoria, disco da 50 GB e 2 esecutori.

  - Per il tipo di worker `G.1X`, ciascun worker si mappa a 1 DPU (4 vCPU, 16 GB di memoria, disco da 64 GB) e fornisce 1 esecutore. Consigliamo questo tipo di lavoratore per lavori che richiedono molta memoria.
  - Per il tipo di worker `G.2X`, ciascun worker si mappa a 2 DPU (8 vCPU, 32 GB di memoria, disco da 128 GB) e fornisce 1 esecutore. Consigliamo questo tipo di lavoratore per lavori che richiedono molta memoria.

  Problema noto: quando viene creato un endpoint di sviluppo con la configurazione `G.2X` `WorkerType`, i driver Spark per l'endpoint di sviluppo verranno eseguiti su 4 vCPU, 16 GB di memoria e un disco da 64 GB.

- `GlueVersion`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Custom string pattern #48.

  La versione Glue determina le versioni di Apache Spark e Python supportate. AWS Glue La versione Python indica la versione supportata per l'esecuzione di script ETL su endpoint di sviluppo.

  Per ulteriori informazioni sulle AWS Glue versioni disponibili e sulle versioni corrispondenti di Spark e Python, consulta la versione Glue nella guida per sviluppatori.

  Endpoint di sviluppo creati senza specificare una versione di Glue predefinita su Glue 0.9.

  È possibile specificare una versione del supporto Python per gli endpoint di sviluppo utilizzando il `Arguments` parametro in o. `CreateDevEndpoint` `UpdateDevEndpoint` APIs Se non vengono forniti argomenti, per impostazione predefinita viene utilizzata la versione Python 2.

- `NumberOfWorkers`: numero (intero).

  Il numero di worker di un `workerType` definito allocati all'endpoint di sviluppo.

  Il numero massimo di worker che è possibile definire è 299 `G.1X` e 149 per `G.2X`.

- `ExtraPythonLibsS3Path`: stringa UTF-8.

  Percorsi a una o più librerie Python in un bucket Amazon S3 che devono essere caricati nel `DevEndpoint`. I valori multipli devono essere percorsi completi separati dalla virgola.

> **Note**
>
> Con un `DevEndpoint` è possibile utilizzare solo librerie Python pure. Le librerie che si basano sulle estensioni C, come la libreria di analisi dati Python pandas, non sono ancora supportate.

- `ExtraJarsS3Path`: stringa UTF-8.

  Percorsi a uno o più file `.jar` Java in un bucket S3 che devono essere caricati nel `DevEndpoint`.

- `SecurityConfiguration`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome della struttura `SecurityConfiguration` da utilizzare con questo `DevEndpoint`.

- `Tags` – Una matrice di mappe con coppie chiave-valore, non superiore alle 50 coppie.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Ogni valore è una stringa UTF-8, lunga non più di 256 byte.

  I tag da usare con questo DevEndpoint. È possibile utilizzare i tag per limitare l'accesso a. DevEndpoint Per ulteriori informazioni sui tag in AWS Glue, consulta AWS Tags AWS Glue in nella guida per sviluppatori.

- `Arguments` – Una matrice di mappe con coppie chiave-valore, non superiore alle 100 coppie.

  Ogni chiave è una stringa UTF-8.

  Ogni valore è una stringa UTF-8.

  Mappa di argomenti usati per configurare `DevEndpoint`.

Risposta

- `EndpointName`: stringa UTF-8.

  Il nome assegnato al nuovo `DevEndpoint`.

- `Status`: stringa UTF-8.

  Lo stato corrente del nuovo `DevEndpoint`.

- `SecurityGroupIds`: una matrice di stringhe UTF-8.

  I gruppi di sicurezza assegnati al nuovo `DevEndpoint`.

- `SubnetId`: stringa UTF-8.

  L'ID di sottorete assegnato al nuovo `DevEndpoint`.

- `RoleArn`: stringa UTF-8, corrispondente a AWS IAM ARN string pattern.

  L'Amazon Resource Name (ARN) del ruolo assegnato al nuovo `DevEndpoint`.

- `YarnEndpointAddress`: stringa UTF-8.

  L'indirizzo dell'endpoint YARN utilizzato da questo `DevEndpoint`.

- `ZeppelinRemoteSparkInterpreterPort`: numero (intero).

  La porta Apache Zeppelin per l'interprete Apache Spark remoto.

- `NumberOfNodes`: numero (intero).

  Il numero di unità di elaborazione AWS Glue dati (DPUs) assegnate a questo DevEndpoint.

- `WorkerType`: stringa UTF-8 (valori validi: `Standard=""` | `G.1X=""` | `G.2X=""` | `G.025X=""` | `G.4X=""` | `G.8X=""` | `Z.2X=""`).

  Il tipo di worker predefinito allocato all'endpoint di sviluppo. Può essere un valore Standard, G.1X o G.2X.

- `GlueVersion`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Custom string pattern #48.

  La versione Glue determina le versioni di Apache Spark e Python supportate. AWS Glue La versione Python indica la versione supportata per l'esecuzione di script ETL su endpoint di sviluppo.

  Per ulteriori informazioni sulle AWS Glue versioni disponibili e sulle versioni corrispondenti di Spark e Python, consulta la versione Glue nella guida per sviluppatori.

- `NumberOfWorkers`: numero (intero).

  Il numero di worker di un `workerType` definito allocati all'endpoint di sviluppo.

- `AvailabilityZone`: stringa UTF-8.

  La zona di AWS disponibilità in cui si `DevEndpoint` trova.

- `VpcId`: stringa UTF-8.

  L'ID del Virtual Private Cloud (VPC) utilizzato da questo `DevEndpoint`.

- `ExtraPythonLibsS3Path`: stringa UTF-8.

  Percorsi a una o più librerie Python in un bucket S3 che verranno caricati nel `DevEndpoint`.

- `ExtraJarsS3Path`: stringa UTF-8.

  Percorsi a uno o più file `.jar` Java in un bucket S3 che devono essere caricati nel `DevEndpoint`.

- `FailureReason`: stringa UTF-8.

  Il motivo di un errore corrente in questo `DevEndpoint`.

- `SecurityConfiguration`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome della struttura `SecurityConfiguration` da utilizzare con questo `DevEndpoint`.

- `CreatedTimestamp`: timestamp.

  Il momento in cui questo `DevEndpoint` è stato creato.

- `Arguments` – Una matrice di mappe con coppie chiave-valore, non superiore alle 100 coppie.

  Ogni chiave è una stringa UTF-8.

  Ogni valore è una stringa UTF-8.

  Mappa di argomenti usati per configurare questo `DevEndpoint`.

  Argomenti validi sono:

  - `"--enable-glue-datacatalog": ""`

  È possibile specificare una versione del supporto Python per gli endpoint di sviluppo utilizzando il `Arguments` parametro in o. `CreateDevEndpoint` `UpdateDevEndpoint` APIs Se non vengono forniti argomenti, per impostazione predefinita viene utilizzata la versione Python 2.

Errori

- `AccessDeniedException`
- `AlreadyExistsException`

- `IdempotentParameterMismatchException`
- `InternalServiceException`
- `OperationTimeoutException`
- `InvalidInputException`
- `ValidationException`
- `ResourceNumberLimitExceededException`

## UpdateDevEndpoint azione (Python: update_dev_endpoint)

Aggiorna un endpoint di sviluppo specificato.

Richiesta

- `EndpointName` - obbligatorio: stringa UTF-8.

  Nome del `DevEndpoint` da aggiornare.
- `PublicKey`: stringa UTF-8.

  La chiave pubblica che deve essere utilizzata da `DevEndpoint`.
- `AddPublicKeys`: una matrice di stringhe UTF-8, non più di 5 stringhe.

  L'elenco delle chiavi pubbliche che devono essere utilizzate da `DevEndpoint`.
- `DeletePublicKeys`: una matrice di stringhe UTF-8, non più di 5 stringhe.

  Elenco di chiavi pubbliche da eliminare da `DevEndpoint`.
- `CustomLibraries`: un oggetto DevEndpointCustomLibraries.

  Librerie Python o Java personalizzate da caricare nel `DevEndpoint`.
- `UpdateEtlLibraries`: booleano.

  `True` se l'elenco di librerie personalizzate da caricare nell'endpoint di sviluppo deve essere aggiornato, in caso contrario `False`.
- `DeleteArguments`: una matrice di stringhe UTF-8.

  L'elenco delle chiavi di argomento da eliminare dalla mappa di argomenti utilizzati per configurare il `DevEndpoint`.
- `AddArguments` – Una matrice di mappe con coppie chiave-valore, non superiore alle 100 coppie.

Ogni chiave è una stringa UTF-8.

Ogni valore è una stringa UTF-8.

La mappa di argomenti da aggiungere alla mappa di argomenti utilizzati per configurare il `DevEndpoint`.

Argomenti validi sono:

- `"--enable-glue-datacatalog": ""`

È possibile specificare una versione del supporto Python per gli endpoint di sviluppo utilizzando il `Arguments` parametro in o. `CreateDevEndpoint` `UpdateDevEndpoint` APIs Se non vengono forniti argomenti, per impostazione predefinita viene utilizzata la versione Python 2.

Risposta

- Nessun parametro di risposta.

Errori

- `EntityNotFoundException`
- `InternalServiceException`
- `OperationTimeoutException`
- `InvalidInputException`
- `ValidationException`

## DeleteDevEndpoint azione (Python: delete_dev_endpoint)

Elimina un endpoint di sviluppo specificato.

Richiesta

- `EndpointName` - obbligatorio: stringa UTF-8.

  Nome della `DevEndpoint`.

Risposta

- Nessun parametro di risposta.

Errori

- `EntityNotFoundException`
- `InternalServiceException`
- `OperationTimeoutException`
- `InvalidInputException`

## GetDevEndpoint azione (Python: get_dev_endpoint)

Recupera informazioni su un endpoint di sviluppo specificato.

> **Note**
>
> Quando viene creato un endpoint di sviluppo in un virtual private cloud (VPC), AWS Glue restituisce solo un indirizzo IP privato e il campo dell'indirizzo IP pubblico non è popolato. Quando si crea un endpoint di sviluppo non VPC, AWS Glue restituisce solo un indirizzo IP pubblico.

Richiesta

- `EndpointName` - obbligatorio: stringa UTF-8.

  Nome del `DevEndpoint` per cui recuperare le informazioni.

Risposta

- `DevEndpoint`: un oggetto DevEndpoint.

  Una definizione del `DevEndpoint`.

Errori

- `EntityNotFoundException`

- `InternalServiceException`
- `OperationTimeoutException`
- `InvalidInputException`

## GetDevEndpoints azione (Python: get_dev_endpoints)

Recupera tutti gli endpoint di sviluppo in questo account. AWS

> **Note**
>
> Quando viene creato un endpoint di sviluppo in un virtual private cloud (VPC), AWS Glue restituisce solo un indirizzo IP pubblico e il campo dell'indirizzo IP pubblico non è popolato. Quando si crea un endpoint di sviluppo non VPC, AWS Glue restituisce solo un indirizzo IP pubblico.

Richiesta

- `MaxResults`: numero (intero), non inferiore a 1 o superiore a 1.000.

  La dimensione massima di informazioni da restituire.
- `NextToken`: stringa UTF-8.

  Un token di continuazione, se si tratta di una chiamata di continuazione.

Risposta

- `DevEndpoints`: una matrice di oggetti DevEndpoint.

  Un elenco di definizioni di `DevEndpoint`.
- `NextToken`: stringa UTF-8.

  Un token di continuazione, se non sono ancora state restituite tutte le definizioni di `DevEndpoint`.

Errori

- `EntityNotFoundException`

- `InternalServiceException`
- `OperationTimeoutException`
- `InvalidInputException`

## BatchGetDevEndpoints azione (Python: batch_get_dev_endpoints)

Restituisce un elenco di metadati di risorse per un elenco di nomi di endpoint di sviluppo. Dopo aver chiamato l'operazione `ListDevEndpoints`, puoi chiamare questa operazione per accedere ai dati a cui sono state concesse le autorizzazioni. Questa operazione supporta tutte le autorizzazioni IAM, tra cui le condizioni di autorizzazione che utilizzano i tag.

Richiesta

- `customerAccountId`: stringa UTF-8.

  AWS L'ID dell'account.
- `DevEndpointNames`. Obbligatorio: una serie di stringhe UTF-8, non inferiore a 1 o superiore a 25 stringhe.

  L'elenco dei nomi di `DevEndpoint` che potrebbero essere i nomi restituiti dall'operazione `ListDevEndpoint`.

Risposta

- `DevEndpoints`: una matrice di oggetti DevEndpoint.

  Un elenco di definizioni di `DevEndpoint`.
- `DevEndpointsNotFound` – Una serie di stringhe UTF-8, non inferiore a 1 o superiore a 25 stringhe.

  Un elenco di `DevEndpoints` non trovati.

Errori

- `AccessDeniedException`
- `InternalServiceException`
- `OperationTimeoutException`

- `InvalidInputException`

## ListDevEndpoints azione (Python: list_dev_endpoints)

Recupera i nomi di tutte le risorse `DevEndpoint` in questo account AWS oppure le risorse con il tag specificato. Questa operazione consente di vedere quali risorse sono disponibili nel proprio account e i relativi nomi.

L'operazione accetta il campo facoltativo `Tags` che si può utilizzare come filtro per la risposta in modo che le risorse con tag possano essere recuperate come gruppo. Se si sceglie di utilizzare il filtro dei tag, potranno essere recuperate solo le risorse con tag.

Richiesta

- `NextToken`: stringa UTF-8.

  Token di continuazione, se si tratta di una richiesta di continuazione.
- `MaxResults`: numero (intero), non inferiore a 1 o superiore a 1.000.

  La dimensione massima di un elenco da restituire.
- `Tags`: una matrice di mappe con coppie chiave-valore, non superiore alle 50 coppie.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Ogni valore è una stringa UTF-8, lunga non più di 256 byte.

  Specifica che vengono restituite solo le risorse con tag.

Risposta

- `DevEndpointNames`: una matrice di stringhe UTF-8.

  I nomi di tutti i `DevEndpoint` nell'account oppure i `DevEndpoint` con i tag specificati.
- `NextToken`: stringa UTF-8.

  Token di continuazione, se l'elenco restituito non contiene l'ultimo parametro disponibile.

Errori

- `InvalidInputException`
- `EntityNotFoundException`
- `InternalServiceException`
- `OperationTimeoutException`

# Registro degli schemi

L'API del registro Schema descrive i tipi di dati e le API relativi all'utilizzo degli schemi in AWS Glue.

## Tipi di dati

- RegistryId struttura
- RegistryListItem struttura
- MetadataInfo struttura
- OtherMetadataValueListItem struttura
- SchemaListItem struttura
- SchemaVersionListItem struttura
- MetadataKeyValuePair struttura
- SchemaVersionErrorItem struttura
- ErrorDetails struttura
- SchemaVersionNumber struttura
- SchemaId struttura

## RegistryId struttura

Una struttura wrapper che può contenere il nome del registro e l'Amazon Resource Name (ARN).

Campi

- `RegistryName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Custom string pattern #46.

Nome del registro. Utilizzato solo per la ricerca. Deve essere fornito `RegistryArn` o `RegistryName`.

- `RegistryArn` – stringa UTF-8, non inferiore a 1 o superiore a 10240 byte di lunghezza, corrispondente a Custom string pattern #50.

  Arn del registro da aggiornare. Deve essere fornito `RegistryArn` o `RegistryName`.

## RegistryListItem struttura

Una struttura contenente i dettagli di un registro.

Campi

- `RegistryName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Custom string pattern #46.

  Il nome del registro.

- `RegistryArn` – stringa UTF-8, non inferiore a 1 o superiore a 10240 byte di lunghezza, corrispondente a Custom string pattern #50.

  L'ARN (Amazon Resource Name) del registro.

- `Description`: stringa di descrizione, non superiore a 2048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  Una descrizione del registro.

- `Status`: stringa UTF-8 (valori validi: `AVAILABLE` | `DELETING`).

  Lo stato del registro.

- `CreatedTime`: stringa UTF-8.

  La data in cui è stato creato il registro.

- `UpdatedTime`: stringa UTF-8.

  La data in cui è stato aggiornato il registro.

## MetadataInfo struttura

Una struttura contenente informazioni sui metadati per una versione dello schema.

Campi

- `MetadataValue`: stringa UTF-8, non inferiore a 1 o superiore a 256 byte di lunghezza, corrispondente a Custom string pattern #14.

  Il valore corrispondente alla chiave di metadati.
- `CreatedTime`: stringa UTF-8.

  L'ora in cui è stata creata la voce.
- `OtherMetadataValueList`: una matrice di oggetti OtherMetadataValueListItem.

  Altri metadati appartenenti alla stessa chiave di metadati.

## OtherMetadataValueListItem struttura

Struttura contenente altri metadati per una versione dello schema appartenente alla stessa chiave di metadati.

Campi

- `MetadataValue` – stringa UTF-8, non inferiore a 1 o superiore a 256 byte di lunghezza, corrispondente a Custom string pattern #14.

  Il valore corrispondente della chiave di metadati per gli altri metadati appartenenti alla stessa chiave dei metadati.
- `CreatedTime`: stringa UTF-8.

  L'ora in cui è stata creata la voce.

## SchemaListItem struttura

Un oggetto che contiene dettagli minimi per uno schema.

Campi

- `RegistryName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Custom string pattern #46.

  Il nome del registro in cui si trova lo schema.

- `SchemaName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Custom string pattern #46.

  Il nome dello schema.

- `SchemaArn` – stringa UTF-8, non inferiore a 1 o superiore a 10240 byte di lunghezza, corrispondente a Custom string pattern #50.

  L'ARN (Amazon Resource Name) per lo schema.

- `Description`: stringa di descrizione, non superiore a 2048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  Una descrizione per lo schema.

- `SchemaStatus`: stringa UTF-8 (valori validi: `AVAILABLE` | `PENDING` | `DELETING`).

  Lo stato dello schema.

- `CreatedTime`: stringa UTF-8.

  La data e l'ora di creazione dello schema.

- `UpdatedTime`: stringa UTF-8.

  La data e l'ora di aggiornamento dello schema.

## SchemaVersionListItem struttura

Oggetto contenente i dettagli relativi a una versione dello schema.

Campi

- `SchemaArn` – stringa UTF-8, non inferiore a 1 o superiore a 10240 byte di lunghezza, corrispondente a Custom string pattern #50.

  L'ARN (Amazon Resource Name) dello schema.

- `SchemaVersionId` – stringa UTF-8, non inferiore a 36 o superiore a 36 byte di lunghezza, corrispondente a Custom string pattern #45.

  L'identificatore univoco della versione dello schema.
- `VersionNumber` – Numero (intero), non inferiore a 1 o superiore a 100000.

  Il numero di versione dello schema.
- `Status`: stringa UTF-8 (valori validi: `AVAILABLE` | `PENDING` | `FAILURE` | `DELETING`).

  Lo stato della versione dello schema.
- `CreatedTime`: stringa UTF-8.

  La data e l'ora in cui è stata creata la versione dello schema.

## MetadataKeyValuePair struttura

La struttura contenente una coppia chiave-valore per i metadati.

Campi

- `MetadataKey`: stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza, corrispondente a Custom string pattern #14.

  Una chiave di metadati.
- `MetadataValue` – stringa UTF-8, non inferiore a 1 o superiore a 256 byte di lunghezza, corrispondente a Custom string pattern #14.

  Il valore corrispondente a una chiave di metadati.

## SchemaVersionErrorItem struttura

Oggetto che contiene i dettagli di errore per un'operazione su una versione dello schema.

Campi

- `VersionNumber`: numero (intero), non inferiore a 1 o superiore a 100000.

  Il numero di versione dello schema.
- `ErrorDetails`: un oggetto ErrorDetails.

I dettagli dell'errore per la versione dello schema.

## ErrorDetails struttura

Un oggetto contenente dettagli di errore.

Campi

- `ErrorCode`: stringa UTF-8.

  Il codice di errore per un errore.
- `ErrorMessage`: stringa UTF-8.

  Il messaggio di errore relativo a un errore.

## SchemaVersionNumber struttura

Una struttura contenente le informazioni sulla versione dello schema.

Campi

- `LatestVersion`: booleano.

  La versione più recente disponibile per lo schema.
- `VersionNumber` – Numero (intero), non inferiore a 1 o superiore a 100000.

  Il numero di versione dello schema.

## SchemaId struttura

L'ID univoco dello schema nel registro degli AWS Glue schemi.

Campi

- `SchemaArn` – stringa UTF-8, non inferiore a 1 o superiore a 10240 byte di lunghezza, corrispondente a Custom string pattern #50.

  L'ARN (Amazon Resource Name) dello schema. Deve essere fornito `SchemaArn` o `SchemaName`.

- `SchemaName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Custom string pattern #46.

  Il nome dello schema. Deve essere fornito `SchemaArn` o `SchemaName`.

- `RegistryName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Custom string pattern #46.

  Il nome del registro degli schemi che contiene lo schema.

## Operazioni

- CreateRegistry azione (Python: create_registry)
- CreateSchema azione (Python: create_schema)
- GetSchema azione (Python: get_schema)
- ListSchemaVersions azione (Python: list_schema_versions)
- GetSchemaVersion azione (Python: get_schema_version)
- GetSchemaVersionsDiff azione (Python: get_schema_versions_diff)
- ListRegistries azione (Python: list_registries)
- ListSchemas azione (Python: list_schemas)
- RegisterSchemaVersion azione (Python: register_schema_version)
- UpdateSchema azione (Python: update_schema)
- CheckSchemaVersionValidity azione (Python: check_schema_version_idity)
- UpdateRegistry azione (Python: update_registry)
- GetSchemaByDefinition azione (Python: get_schema_by_definition)
- GetRegistry azione (Python: get_registry)
- PutSchemaVersionMetadata azione (Python: put_schema_version_metadata)
- QuerySchemaVersionMetadata azione (Python: query_schema_version_metadata)
- RemoveSchemaVersionMetadata azione (Python: remove_schema_version_metadata)
- DeleteRegistry azione (Python: delete_registry)
- DeleteSchema azione (Python: delete_schema)
- DeleteSchemaVersions azione (Python: delete_schema_versions)

# CreateRegistry azione (Python: create_registry)

Crea un nuovo registro che può essere utilizzato per contenere una raccolta di schemi.

Richiesta

- `RegistryName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Custom string pattern #46.

  Il nome del registro da creare, che può avere una lunghezza massima di 255 caratteri e può contenere solo lettere, numeri, trattino, trattino basso, simbolo del dollaro o cancelletto. Non sono ammessi spazi.
- `Description`: stringa di descrizione, non superiore a 2048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  Una descrizione del registro. Se la descrizione non viene fornita, non ci sarà alcun valore di default.
- `Tags` – Una matrice di mappe con coppie chiave-valore, non superiore alle 50 coppie.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Ogni valore è una stringa UTF-8, lunga non più di 256 byte.

  AWS tag che contengono una coppia chiave-valore e possono essere ricercati tramite console, riga di comando o API.

Risposta

- `RegistryArn` – stringa UTF-8, non inferiore a 1 o superiore a 10240 byte di lunghezza, corrispondente a Custom string pattern #50.

  L'Amazon Resource Name (ARN) del registro appena creato.
- `RegistryName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Custom string pattern #46.

  Il nome del registro.
- `Description`: stringa di descrizione, non superiore a 2048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  Una descrizione del registro.

- `Tags` – Una matrice di mappe con coppie chiave-valore, non superiore alle 50 coppie.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Ogni valore è una stringa UTF-8, lunga non più di 256 byte.

  I tag per il registro.

Errori

- `InvalidInputException`
- `AccessDeniedException`
- `AlreadyExistsException`
- `ResourceNumberLimitExceededException`
- `ConcurrentModificationException`
- `ThrottlingException`
- `InternalServiceException`

## CreateSchema azione (Python: create_schema)

Crea un nuovo set di schemi e registra la definizione dello schema. Restituisce un errore se il set di schemi esiste già senza registrare effettivamente la versione.

Quando viene creato il set di schemi, un checkpoint della versione verrà impostato sulla prima versione. La modalità di compatibilità "DISABLED" limita l'aggiunta di eventuali versioni aggiuntive dello schema dopo la prima versione. Per tutte le altre modalità di compatibilità, la convalida delle impostazioni di compatibilità verrà applicata solo a partire dalla seconda versione quando viene utilizzata l'API `RegisterSchemaVersion`.

Quando questa API viene chiamata senza un `RegistryId`, verrà creata una voce per un "registro predefinito" nelle tabelle del database del registro, se non è già presente.

Richiesta

- `RegistryId`: un oggetto RegistryId.

Si tratta di una forma wrapper che contiene i campi di identità del registro. Se non viene fornito, verrà utilizzato il registro predefinito. Il formato ARN per lo stesso sarà: `arn:aws:glue:us-east-2:<customer id>:registry/default-registry:random-5-letter-id`.

- `SchemaName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Custom string pattern #46.

  Il nome dello schema da creare, che può avere una lunghezza massima di 255 caratteri e può contenere solo lettere, numeri, trattini, carattere di sottolineatura, simbolo del dollaro o segno di hash. Non sono ammessi spazi.

- `DataFormat` - obbligatorio: stringa UTF-8 (valori validi: `AVRO` | `JSON` | `PROTOBUF`).

  Il formato dei dati della definizione dello schema. Al momento sono supportati `AVRO`, `JSON` e `PROTOBUF`.

- `Compatibility`: stringa UTF-8 (valori validi: `NONE` | `DISABLED` | `BACKWARD` | `BACKWARD_ALL` | `FORWARD` | `FORWARD_ALL` | `FULL` | `FULL_ALL`).

  La modalità di compatibilità dello schema. I valori possibili sono:

  - NONE: non si applica alcuna modalità di compatibilità. È possibile utilizzare questa opzione negli scenari di sviluppo o se non si conosce la modalità di compatibilità da applicare agli schemi. Qualsiasi nuova versione aggiunta sarà accettata senza essere sottoposta a un controllo di compatibilità.
  - DISABLED: questa scelta di compatibilità impedisce il controllo delle versioni per uno schema specifico. È possibile utilizzare questa opzione per impedire il controllo delle versioni future di uno schema.
  - BACKWARD: questa scelta di compatibilità è consigliata in quanto consente ai ricevitori di dati di leggere sia la versione attuale che quella precedente dello schema. Ciò significa che, ad esempio, una nuova versione dello schema non può eliminare i campi dati o modificarne il tipo, in modo che non possano essere letti dai lettori che utilizzano la versione precedente.
  - BACKWARD_ALL:: questa scelta di compatibilità consente ai ricevitori di dati di leggere sia la versione attuale che tutte quelle precedenti dello schema. È possibile utilizzare questa opzione quando è necessario eliminare campi o aggiungere campi facoltativi e verificare la compatibilità con tutte le versioni precedenti dello schema.
  - FORWARD: questa scelta di compatibilità consente ai ricevitori di dati di leggere sia la versione attuale che una versione successiva dello schema, ma non necessariamente le versioni più

recenti. È possibile utilizzare questa opzione quando è necessario aggiungere campi o eliminare campi facoltativi, ma solo verificare la compatibilità con l'ultima versione dello schema.

- FORWARD_ALL: questa scelta di compatibilità consente ai ricevitori di dati di leggere scritti dai produttori di qualsiasi nuovo schema registrato. È possibile utilizzare questa opzione quando è necessario aggiungere campi o eliminare campi facoltativi e verificare la compatibilità con tutte le versioni precedenti dello schema.
- FULL: questa scelta di compatibilità consente ai ricevitori di dati di leggere i dati scritti dai produttori utilizzando la versione precedente o successiva dello schema, ma non necessariamente versioni precedenti o successive. È possibile utilizzare questa opzione quando è necessario aggiungere o eliminare campi facoltativi, ma solo verificare la compatibilità con l'ultima versione dello schema.
- FULL_ALL: questa scelta di compatibilità consente ai ricevitori di dati di leggere i dati scritti dai produttori utilizzando tutte le versioni precedenti dello schema. È possibile utilizzare questa opzione quando è necessario aggiungere o eliminare campi facoltativi e verificare la compatibilità con tutte le versioni precedenti dello schema.

- `Description`: stringa di descrizione, non superiore a 2048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  Una descrizione facoltativa dello schema. Se la descrizione non viene fornita, non ci sarà alcun valore predefinito automatico.

- `Tags` – Una matrice di mappe con coppie chiave-valore, non superiore alle 50 coppie.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Ogni valore è una stringa UTF-8, lunga non più di 256 byte.

  AWS tag che contengono una coppia chiave-valore e possono essere ricercati tramite console, riga di comando o API. Se specificato, segue lo AWS tags-on-create schema.

- `SchemaDefinition`: stringa UTF-8, non inferiore a 1 o superiore a 170000 byte di lunghezza, corrispondente a Custom string pattern #13.

  La definizione dello schema che utilizza l'impostazione `DataFormat` per `SchemaName`.

Risposta

- `RegistryName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Custom string pattern #46.

Il nome del registro.

- `RegistryArn` – stringa UTF-8, non inferiore a 1 o superiore a 10240 byte di lunghezza, corrispondente a Custom string pattern #50.

  L'ARN (Amazon Resource Name) del registro.

- `SchemaName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Custom string pattern #46.

  Il nome dello schema.

- `SchemaArn` – stringa UTF-8, non inferiore a 1 o superiore a 10240 byte di lunghezza, corrispondente a Custom string pattern #50.

  L'ARN (Amazon Resource Name) dello schema.

- `Description`: stringa di descrizione, non superiore a 2048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  Una descrizione dello schema, se specificata al momento della creazione.

- `DataFormat`: stringa UTF-8 (valori validi: `AVRO` | `JSON` | `PROTOBUF`).

  Il formato dei dati della definizione dello schema. Al momento sono supportati `AVRO`, `JSON` e `PROTOBUF`.

- `Compatibility`: stringa UTF-8 (valori validi: `NONE` | `DISABLED` | `BACKWARD` | `BACKWARD_ALL` | `FORWARD` | `FORWARD_ALL` | `FULL` | `FULL_ALL`).

  La modalità di compatibilità dello schema.

- `SchemaCheckpoint` – Numero (intero), non inferiore a 1 o superiore a 100000.

  Il numero di versione del checkpoint (l'ultima volta che la modalità di compatibilità è stata modificata).

- `LatestSchemaVersion` – Numero (intero), non inferiore a 1 o superiore a 100000.

  La versione più recente dello schema associata alla definizione dello schema restituita.

- `NextSchemaVersion` – Numero (intero), non inferiore a 1 o superiore a 100000.

  La versione successiva dello schema associata alla definizione dello schema restituita.

- `SchemaStatus`: stringa UTF-8 (valori validi: `AVAILABLE` | `PENDING` | `DELETING`).

Lo stato dello schema.

- `Tags` – Una matrice di mappe con coppie chiave-valore, non superiore alle 50 coppie.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Ogni valore è una stringa UTF-8, lunga non più di 256 byte.

  I tag per lo schema.

- `SchemaVersionId` – stringa UTF-8, non inferiore a 36 o superiore a 36 byte di lunghezza, corrispondente a Custom string pattern #45.

  L'identificatore univoco della prima versione dello schema.

- `SchemaVersionStatus`: stringa UTF-8 (valori validi: `AVAILABLE` | `PENDING` | `FAILURE` | `DELETING`).

  Lo stato della prima versione dello schema creata.

Errori

- `InvalidInputException`
- `AccessDeniedException`
- `EntityNotFoundException`
- `AlreadyExistsException`
- `ResourceNumberLimitExceededException`
- `ConcurrentModificationException`
- `ThrottlingException`
- `InternalServiceException`

## GetSchema azione (Python: get_schema)

Descrive lo schema specificato nel dettaglio.

Richiesta

- `SchemaId` - obbligatorio: un oggetto SchemaId.

Si tratta di una struttura wrapper che contiene i campi di identità dello schema. La struttura include:

- SchemaId$SchemaArn: L'Amazon Resource Name (ARN) dello schema. Deve essere fornito `SchemaArn` oppure `SchemaName` e `RegistryName`.
- SchemaId$SchemaName: il nome dello schema. Deve essere fornito `SchemaArn` oppure `SchemaName` e `RegistryName`.

Risposta

- `RegistryName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Custom string pattern #46.

  Il nome del registro.
- `RegistryArn` – stringa UTF-8, non inferiore a 1 o superiore a 10240 byte di lunghezza, corrispondente a Custom string pattern #50.

  L'ARN (Amazon Resource Name) del registro.
- `SchemaName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Custom string pattern #46.

  Il nome dello schema.
- `SchemaArn` – stringa UTF-8, non inferiore a 1 o superiore a 10240 byte di lunghezza, corrispondente a Custom string pattern #50.

  L'ARN (Amazon Resource Name) dello schema.
- `Description`: stringa di descrizione, non superiore a 2048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  Una descrizione dello schema, se specificata al momento della creazione
- `DataFormat`: stringa UTF-8 (valori validi: `AVRO` | `JSON` | `PROTOBUF`).

  Il formato dei dati della definizione dello schema. Al momento sono supportati `AVRO`, `JSON` e `PROTOBUF`.
- `Compatibility`: stringa UTF-8 (valori validi: `NONE` | `DISABLED` | `BACKWARD` | `BACKWARD_ALL` | `FORWARD` | `FORWARD_ALL` | `FULL` | `FULL_ALL`).

  La modalità di compatibilità dello schema.

- `SchemaCheckpoint` – Numero (intero), non inferiore a 1 o superiore a 100000.

  Il numero di versione del checkpoint (l'ultima volta che la modalità di compatibilità è stata modificata).
- `LatestSchemaVersion` – Numero (intero), non inferiore a 1 o superiore a 100000.

  La versione più recente dello schema associata alla definizione dello schema restituita.
- `NextSchemaVersion` – Numero (intero), non inferiore a 1 o superiore a 100000.

  La versione successiva dello schema associata alla definizione dello schema restituita.
- `SchemaStatus`: stringa UTF-8 (valori validi: `AVAILABLE` | `PENDING` | `DELETING`).

  Lo stato dello schema.
- `CreatedTime`: stringa UTF-8.

  La data e l'ora di creazione dello schema.
- `UpdatedTime`: stringa UTF-8.

  La data e l'ora di aggiornamento dello schema.

Errori

- `InvalidInputException`
- `AccessDeniedException`
- `EntityNotFoundException`
- `ThrottlingException`
- `InternalServiceException`

## ListSchemaVersions azione (Python: list_schema_versions)

Restituisce un elenco delle versioni dello schema create dall'utente, con informazioni minime. Le versioni dello schema nello stato Deleted non verranno incluse nei risultati. Se non sono disponibili versioni dello schema, verranno restituiti risultati vuoti.

Richiesta

- `SchemaId` - obbligatorio: un oggetto SchemaId.

Si tratta di una struttura wrapper che contiene i campi di identità dello schema. La struttura include:

- SchemaId$SchemaArn: L'Amazon Resource Name (ARN) dello schema. Deve essere fornito `SchemaArn` oppure `SchemaName` e `RegistryName`.
- SchemaId$SchemaName: il nome dello schema. Deve essere fornito `SchemaArn` oppure `SchemaName` e `RegistryName`.

- `MaxResults` – Numero (intero), non inferiore a 1 o superiore a 100.

  Numero massimo di risultati richiesti per pagina. Se il valore non viene fornito, sarà impostato in modo predefinito su 25 per pagina.

- `NextToken`: stringa UTF-8.

  Un token di continuazione, se si tratta di una chiamata di continuazione.

Risposta

- `Schemas`: una matrice di oggetti SchemaVersionListItem.

  Una matrice di oggetti `SchemaVersionList` contenenti i dettagli di ogni versione dello schema.

- `NextToken`: stringa UTF-8.

  Un token di continuazione per impaginare l'elenco restituito di token, restituiti se il segmento corrente dell'elenco non è l'ultimo.

Errori

- `InvalidInputException`
- `AccessDeniedException`
- `EntityNotFoundException`
- `ThrottlingException`
- `InternalServiceException`

# GetSchemaVersion azione (Python: get_schema_version)

Ottiene lo schema specificato in base al relativo ID univoco assegnato alla creazione o alla registrazione di una versione dello schema. Le versioni dello schema nello stato Deleted non verranno incluse nei risultati.

Richiesta

- `SchemaId`: un oggetto SchemaId.

  Si tratta di una struttura wrapper che contiene i campi di identità dello schema. La struttura include:

  - SchemaId$SchemaArn: L'Amazon Resource Name (ARN) dello schema. Deve essere fornito `SchemaArn` oppure `SchemaName` e `RegistryName`.
  - SchemaId$SchemaName: il nome dello schema. Deve essere fornito `SchemaArn` oppure `SchemaName` e `RegistryName`.
- `SchemaVersionId` – stringa UTF-8, non inferiore a 36 o superiore a 36 byte di lunghezza, corrispondente a Custom string pattern #45.

  La versione `SchemaVersionId` dello schema. Questo campo è obbligatorio per il recupero in base all'ID dello schema. Deve essere fornito questo o il wrapper `SchemaId`.
- `SchemaVersionNumber`: un oggetto SchemaVersionNumber.

  Il numero di versione dello schema.

Risposta

- `SchemaVersionId` – stringa UTF-8, non inferiore a 36 o superiore a 36 byte di lunghezza, corrispondente a Custom string pattern #45.

  La versione `SchemaVersionId` dello schema.
- `SchemaDefinition` – stringa UTF-8, non inferiore a 1 o superiore a 170000 byte di lunghezza, corrispondente a Custom string pattern #13.

  La definizione dello schema per l'ID dello schema.
- `DataFormat`: stringa UTF-8 (valori validi: `AVRO` | `JSON` | `PROTOBUF`).

  Il formato dei dati della definizione dello schema. Al momento sono supportati `AVRO`, `JSON` e `PROTOBUF`.

- `SchemaArn` – stringa UTF-8, non inferiore a 1 o superiore a 10240 byte di lunghezza, corrispondente a Custom string pattern #50.

  L'ARN (Amazon Resource Name) dello schema.

- `VersionNumber` – Numero (intero), non inferiore a 1 o superiore a 100000.

  Il numero di versione dello schema.

- `Status`: stringa UTF-8 (valori validi: `AVAILABLE` | `PENDING` | `FAILURE` | `DELETING`).

  Lo stato della versione dello schema.

- `CreatedTime`: stringa UTF-8.

  La data e l'ora in cui è stata creata la versione dello schema.

Errori

- `InvalidInputException`
- `AccessDeniedException`
- `EntityNotFoundException`
- `ThrottlingException`
- `InternalServiceException`

## GetSchemaVersionsDiff azione (Python: get_schema_versions_diff)

Recupera la differenza di versione dello schema nel tipo di differenza specificato tra due versioni dello schema archiviate nel registro degli schemi.

Questa API consente di confrontare due versioni dello schema tra due definizioni dello schema nello stesso schema.

Richiesta

- `SchemaId` - obbligatorio: un oggetto SchemaId.

  Si tratta di una struttura wrapper che contiene i campi di identità dello schema. La struttura include:

  - SchemaId$SchemaArn: L'Amazon Resource Name (ARN) dello schema. Deve essere fornito `SchemaArn` o `SchemaName`.

  - SchemaId$SchemaName: il nome dello schema. Deve essere fornito `SchemaArn` o `SchemaName`.
- `FirstSchemaVersionNumber` - obbligatorio: un oggetto SchemaVersionNumber.

  La prima delle due versioni dello schema da confrontare.
- `SecondSchemaVersionNumber` - obbligatorio: un oggetto SchemaVersionNumber.

  La seconda delle due versioni dello schema da confrontare.
- `SchemaDiffType`. Obbligatorio: stringa UTF-8 (valori validi: `SYNTAX_DIFF`).

  Si riferisce a `SYNTAX_DIFF`, che è il tipo di diff attualmente supportato.

Risposta

- `Diff`: obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 340000 byte di lunghezza, corrispondente a Custom string pattern #13.

  La differenza tra gli schemi come stringa in JsonPatch formato.

Errori

- `InvalidInputException`
- `EntityNotFoundException`
- `AccessDeniedException`
- `ThrottlingException`
- `InternalServiceException`

## ListRegistries azione (Python: list_registries)

Restituisce un elenco di registri dall'utente, con informazioni minime. I registri nello stato `Deleting` non verranno inclusi nei risultati. Se non sono disponibili registri, verranno restituiti risultati vuoti.

Richiesta

- `MaxResults` – Numero (intero), non inferiore a 1 o superiore a 100.

Numero massimo di risultati richiesti per pagina. Se il valore non viene fornito, sarà impostato in modo predefinito su 25 per pagina.

- `NextToken`: stringa UTF-8.

  Un token di continuazione, se si tratta di una chiamata di continuazione.

Risposta

- `Registries`: una matrice di oggetti RegistryListItem.

  Una matrice di oggetti `RegistryDetailedListItem` contenenti dettagli minimi di ogni registro.
- `NextToken`: stringa UTF-8.

  Un token di continuazione per impaginare l'elenco restituito di token, restituiti se il segmento corrente dell'elenco non è l'ultimo.

Errori

- `InvalidInputException`
- `AccessDeniedException`
- `ThrottlingException`
- `InternalServiceException`

## ListSchemas azione (Python: list_schemas)

Restituisce un elenco di schemi con dettagli minimi. Gli schemi nello stato Deleting non verranno inclusi nei risultati. Se non sono disponibili schemi, verranno restituiti risultati vuoti.

Quando il `RegistryId` non viene fornito, tutti gli schemi nei registri faranno parte della risposta dell'API.

Richiesta

- `RegistryId`: un oggetto RegistryId.

  Una struttura wrapper che può contenere il nome del registro e l'Amazon Resource Name (ARN).
- `MaxResults` – Numero (intero), non inferiore a 1 o superiore a 100.

Numero massimo di risultati richiesti per pagina. Se il valore non viene fornito, sarà impostato in modo predefinito su 25 per pagina.

- `NextToken`: stringa UTF-8.

  Un token di continuazione, se si tratta di una chiamata di continuazione.

Risposta

- `Schemas`: una matrice di oggetti SchemaListItem.

  Una matrice di oggetti `SchemaListItem` contenenti i dettagli di ogni schema.
- `NextToken`: stringa UTF-8.

  Un token di continuazione per impaginare l'elenco restituito di token, restituiti se il segmento corrente dell'elenco non è l'ultimo.

Errori

- `InvalidInputException`
- `AccessDeniedException`
- `EntityNotFoundException`
- `ThrottlingException`
- `InternalServiceException`

## RegisterSchemaVersion azione (Python: register_schema_version)

Aggiunge una nuova versione allo schema esistente. Restituisce un errore se la nuova versione dello schema non soddisfa i requisiti di compatibilità del set di schemi. Questa API non creerà un nuovo set di schemi e restituirà un errore 404 se il set di schemi non è già presente nel registro degli schemi.

Se si tratta della prima definizione dello schema da registrare nel registro degli schemi, questa API archivierà la versione dello schema e restituirà immediatamente. In caso contrario, l'esecuzione di questa chiamata potrebbe durare più a lungo rispetto ad altre operazioni a causa delle modalità di compatibilità. È possibile chiamare l'API `GetSchemaVersion` con `SchemaVersionId` per controllare le modalità di compatibilità.

Se la stessa definizione di schema è già archiviata nel registro degli schemi come versione, viene restituito al chiamante l'ID dello schema esistente.

Richiesta

- `SchemaId` - obbligatorio: un oggetto SchemaId.

  Si tratta di una struttura wrapper che contiene i campi di identità dello schema. La struttura include:

  - SchemaId$SchemaArn: L'Amazon Resource Name (ARN) dello schema. Deve essere fornito `SchemaArn` oppure `SchemaName` e `RegistryName`.
  - SchemaId$SchemaName: il nome dello schema. Deve essere fornito `SchemaArn` oppure `SchemaName` e `RegistryName`.
- `SchemaDefinition`: obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 170000 byte di lunghezza, corrispondente a Custom string pattern #13.

  La definizione dello schema che utilizza l'impostazione `DataFormat` per `SchemaName`.

Risposta

- `SchemaVersionId` – stringa UTF-8, non inferiore a 36 o superiore a 36 byte di lunghezza, corrispondente a Custom string pattern #45.

  L'ID univoco che rappresenta la versione di questo schema.
- `VersionNumber` – Numero (intero), non inferiore a 1 o superiore a 100000.

  La versione di questo schema (solo per flusso di sincronizzazione, se si tratta della prima versione).
- `Status`: stringa UTF-8 (valori validi: `AVAILABLE` | `PENDING` | `FAILURE` | `DELETING`).

  Lo stato della versione dello schema.

Errori

- `InvalidInputException`
- `AccessDeniedException`
- `EntityNotFoundException`
- `ResourceNumberLimitExceededException`

- `ThrottlingException`
- `ConcurrentModificationException`
- `InternalServiceException`

## UpdateSchema azione (Python: update_schema)

Aggiorna la descrizione, l'impostazione di compatibilità o il checkpoint della versione per un set di schemi.

Per aggiornare l'impostazione di compatibilità, la chiamata non convaliderà la compatibilità per l'intero set di versioni dello schema con la nuova impostazione di compatibilità. Se il valore per `Compatibility` viene fornito, è necessario fornire anche `VersionNumber` (un checkpoint). L'API convaliderà il numero di versione del checkpoint per garantire coerenza.

Se viene fornito il valore per il `VersionNumber` (checkpoint), `Compatibility` è facoltativo e può essere utilizzato per set/reset un checkpoint per lo schema.

Questo aggiornamento verrà eseguito solo se lo schema si trova nello stato AVAILABLE.

Richiesta

- `SchemaId` - obbligatorio: un oggetto SchemaId.

  Si tratta di una struttura wrapper che contiene i campi di identità dello schema. La struttura include:

  - SchemaId$SchemaArn: L'Amazon Resource Name (ARN) dello schema. Deve essere fornito `SchemaArn` o `SchemaName`.
  - SchemaId$SchemaName: il nome dello schema. Deve essere fornito `SchemaArn` o `SchemaName`.
- `SchemaVersionNumber`: un oggetto SchemaVersionNumber.

  Numero di versione richiesto per il checkpoint. Deve essere fornito `VersionNumber` o `Compatibility`.
- `Compatibility`: stringa UTF-8 (valori validi: `NONE` | `DISABLED` | `BACKWARD` | `BACKWARD_ALL` | `FORWARD` | `FORWARD_ALL` | `FULL` | `FULL_ALL`).

  La nuova impostazione di compatibilità per lo schema.
- `Description`: stringa di descrizione, non superiore a 2048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

La nuova descrizione per lo schema.

Risposta

- `SchemaArn` – stringa UTF-8, non inferiore a 1 o superiore a 10240 byte di lunghezza, corrispondente a Custom string pattern #50.

  L'ARN (Amazon Resource Name) dello schema.

- `SchemaName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Custom string pattern #46.

  Il nome dello schema.

- `RegistryName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Custom string pattern #46.

  Il nome del registro che contiene lo schema.

Errori

- `InvalidInputException`
- `AccessDeniedException`
- `EntityNotFoundException`
- `ConcurrentModificationException`
- `ThrottlingException`
- `InternalServiceException`

## CheckSchemaVersionValidity azione (Python: check_schema_version_idity)

Convalida lo schema fornito. Questa chiamata non ha effetti collaterali, convalida semplicemente usando lo schema fornito, utilizzando `DataFormat` come formato. Poiché non prevede un nome di set di schemi, non vengono eseguiti controlli di compatibilità.

Richiesta

- `DataFormat` - obbligatorio: stringa UTF-8 (valori validi: `AVRO` | `JSON` | `PROTOBUF`).

Il formato dei dati della definizione dello schema. Al momento sono supportati `AVRO`, `JSON` e `PROTOBUF`.

- `SchemaDefinition` – Obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 170000 byte di lunghezza, corrispondente a Custom string pattern #13.

  La definizione dello schema che deve essere convalidato.

Risposta

- `Valid`: booleano.

  Se lo schema è valido restituisce true, in caso contrario false.
- `Error`: stringa UTF-8, non inferiore a 1 o superiore a 5000 byte di lunghezza.

  Un messaggio di errore di convalida.

Errori

- `InvalidInputException`
- `AccessDeniedException`
- `ThrottlingException`
- `InternalServiceException`

## UpdateRegistry azione (Python: update_registry)

Aggiorna un registro esistente che viene utilizzato per contenere una raccolta di schemi. Le proprietà aggiornate si riferiscono al registro e non modificano gli schemi all'interno del registro.

Richiesta

- `RegistryId` - obbligatorio: un oggetto RegistryId.

  Si tratta di una struttura wrapper che può contenere il nome del registro e l'Amazon Resource Name (ARN).
- `Description`: obbligatorio: stringa di descrizione, non superiore a 2048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

Una descrizione del registro. Se la descrizione non viene fornita, questo campo non verrà aggiornato.

Risposta

- `RegistryName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Custom string pattern #46.

  Il nome del registro aggiornato.
- `RegistryArn` – stringa UTF-8, non inferiore a 1 o superiore a 10240 byte di lunghezza, corrispondente a Custom string pattern #50.

  L'Amazon Resource name (ARN) del registro aggiornato.

Errori

- `InvalidInputException`
- `AccessDeniedException`
- `EntityNotFoundException`
- `ConcurrentModificationException`
- `ThrottlingException`
- `InternalServiceException`

## GetSchemaByDefinition azione (Python: get_schema_by_definition)

Recupera uno schema mediante la `SchemaDefinition`. La definizione dello schema viene inviata al registro degli schemi, canonicalizzata e sottoposta ad hashing. Se l'hash è abbinato nell'ambito del `SchemaName` o ARN (o il registro di default, se non viene fornito), vengono restituiti i metadati dello schema. In caso contrario, viene restituito un errore 404 o. NotFound Le versioni dello schema nello stato `Deleted` non verranno incluse nei risultati.

Richiesta

- `SchemaId` - obbligatorio: un oggetto SchemaId.

  Si tratta di una struttura wrapper che contiene i campi di identità dello schema. La struttura include:

  - SchemaId$SchemaArn: L'Amazon Resource Name (ARN) dello schema. Deve essere fornito `SchemaArn` o `SchemaName`.
  - SchemaId$SchemaName: il nome dello schema. Deve essere fornito `SchemaArn` o `SchemaName`.
- `SchemaDefinition`. Obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 170000 byte di lunghezza, corrispondente a Custom string pattern #13.

  La definizione dello schema per il quale sono necessari i dettagli dello schema.

Risposta

- `SchemaVersionId` – stringa UTF-8, non inferiore a 36 o superiore a 36 byte di lunghezza, corrispondente a Custom string pattern #45.

  L'ID dello schema della versione dello schema.
- `SchemaArn` – stringa UTF-8, non inferiore a 1 o superiore a 10240 byte di lunghezza, corrispondente a Custom string pattern #50.

  L'ARN (Amazon Resource Name) dello schema.
- `DataFormat`: stringa UTF-8 (valori validi: `AVRO` | `JSON` | `PROTOBUF`).

  Il formato dei dati della definizione dello schema. Al momento sono supportati `AVRO`, `JSON` e `PROTOBUF`.
- `Status`: stringa UTF-8 (valori validi: `AVAILABLE` | `PENDING` | `FAILURE` | `DELETING`).

  Lo stato della versione dello schema.
- `CreatedTime`: stringa UTF-8.

  La data e l'ora di creazione dello schema.

Errori

- `InvalidInputException`
- `AccessDeniedException`
- `EntityNotFoundException`
- `ThrottlingException`

- `InternalServiceException`

## GetRegistry azione (Python: get_registry)

Descrive il registro specificato nel dettaglio.

Richiesta

- `RegistryId` - obbligatorio: un oggetto RegistryId.

  Si tratta di una struttura wrapper che può contenere il nome del registro e l'Amazon Resource Name (ARN).

Risposta

- `RegistryName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Custom string pattern #46.

  Il nome del registro.
- `RegistryArn` – stringa UTF-8, non inferiore a 1 o superiore a 10240 byte di lunghezza, corrispondente a Custom string pattern #50.

  L'ARN (Amazon Resource Name) del registro.
- `Description`: stringa di descrizione, non superiore a 2048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  Una descrizione del registro.
- `Status`: stringa UTF-8 (valori validi: `AVAILABLE` | `DELETING`).

  Lo stato del registro.
- `CreatedTime`: stringa UTF-8.

  La data e l'ora di creazione del registro.
- `UpdatedTime`: stringa UTF-8.

  La data e l'ora di aggiornamento del registro.

Errori

- `InvalidInputException`
- `AccessDeniedException`
- `EntityNotFoundException`
- `ThrottlingException`
- `InternalServiceException`

## PutSchemaVersionMetadata azione (Python: put_schema_version_metadata)

Inserisce la coppia chiave-valore dei metadati per un ID versione dello schema specificato. Sarà consentito un massimo di 10 coppie chiave-valore per ciascuna versione dello schema. Possono essere aggiunte su una o più chiamate.

Richiesta

- `SchemaId`: un oggetto SchemaId.

  L'ID univoco dello schema.
- `SchemaVersionNumber`: un oggetto SchemaVersionNumber.

  Il numero di versione dello schema.
- `SchemaVersionId` – stringa UTF-8, non inferiore a 36 o superiore a 36 byte di lunghezza, corrispondente a Custom string pattern #45.

  L'ID versione univoco della versione dello schema.
- `MetadataKeyValue` - obbligatorio: un oggetto MetadataKeyValuePair.

  Il valore corrispondente a una chiave di metadati.

Risposta

- `SchemaArn` – stringa UTF-8, non inferiore a 1 o superiore a 10240 byte di lunghezza, corrispondente a Custom string pattern #50.

  L'ARN (Amazon Resource Name) per lo schema.

- `SchemaName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Custom string pattern #46.

  Il nome per lo schema.

- `RegistryName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Custom string pattern #46.

  Il nome per il registro.

- `LatestVersion`: booleano.

  La versione più recente dello schema.

- `VersionNumber` – Numero (intero), non inferiore a 1 o superiore a 100000.

  Il numero di versione dello schema.

- `SchemaVersionId` – stringa UTF-8, non inferiore a 36 o superiore a 36 byte di lunghezza, corrispondente a Custom string pattern #45.

  L'ID versione univoco della versione dello schema.

- `MetadataKey` – stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza, corrispondente a Custom string pattern #14.

  La chiave di metadati.

- `MetadataValue` – stringa UTF-8, non inferiore a 1 o superiore a 256 byte di lunghezza, corrispondente a Custom string pattern #14.

  Il valore della chiave di metadati.

Errori

- `InvalidInputException`
- `AccessDeniedException`
- `AlreadyExistsException`
- `EntityNotFoundException`
- `ThrottlingException`
- `ResourceNumberLimitExceededException`

# QuerySchemaVersionMetadata azione (Python: query_schema_version_metadata)

Query per le informazioni sui metadati della versione dello schema.

Richiesta

- `SchemaId`: un oggetto SchemaId.

  Una struttura wrapper che può contenere il nome dello schema e l'Amazon Resource Name (ARN).
- `SchemaVersionNumber`: un oggetto SchemaVersionNumber.

  Il numero di versione dello schema.
- `SchemaVersionId` – stringa UTF-8, non inferiore a 36 o superiore a 36 byte di lunghezza, corrispondente a Custom string pattern #45.

  L'ID versione univoco della versione dello schema.
- `MetadataList`: una matrice di oggetti MetadataKeyValuePair.

  Cerca coppie chiave-valore per i metadati, se non vengono forniti verranno recuperate tutte le informazioni sui metadati.
- `MaxResults`: numero (intero), non inferiore a 1 o superiore a 50.

  Numero massimo di risultati richiesti per pagina. Se il valore non viene fornito, sarà impostato in modo predefinito su 25 per pagina.
- `NextToken`: stringa UTF-8.

  Un token di continuazione, se si tratta di una chiamata di continuazione.

Risposta

- `MetadataInfoMap`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza, corrispondente a Custom string pattern #14.

  Ogni valore è un oggetto MetadataInfo.

  Una mappa di una chiave di metadati e dei valori associati.

- `SchemaVersionId` – stringa UTF-8, non inferiore a 36 o superiore a 36 byte di lunghezza, corrispondente a Custom string pattern #45.

  L'ID versione univoco della versione dello schema.
- `NextToken`: stringa UTF-8.

  Un token di continuazione per impaginare l'elenco restituito di token, restituiti se il segmento corrente dell'elenco non è l'ultimo.

Errori

- `InvalidInputException`
- `AccessDeniedException`
- `ThrottlingException`
- `EntityNotFoundException`

## RemoveSchemaVersionMetadata azione (Python: remove_schema_version_metadata)

Rimuove una coppia chiave-valore dai metadati della versione dello schema per l'ID versione dello schema specificato.

Richiesta

- `SchemaId`: un oggetto SchemaId.

  Una struttura wrapper che può contenere il nome dello schema e l'Amazon Resource Name (ARN).
- `SchemaVersionNumber`: un oggetto SchemaVersionNumber.

  Il numero di versione dello schema.
- `SchemaVersionId` – stringa UTF-8, non inferiore a 36 o superiore a 36 byte di lunghezza, corrispondente a Custom string pattern #45.

  L'ID versione univoco della versione dello schema.
- `MetadataKeyValue` - obbligatorio: un oggetto MetadataKeyValuePair.

  Il valore della chiave di metadati.

Risposta

- `SchemaArn` – stringa UTF-8, non inferiore a 1 o superiore a 10240 byte di lunghezza, corrispondente a Custom string pattern #50.

  L'ARN (Amazon Resource Name) dello schema.
- `SchemaName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Custom string pattern #46.

  Il nome dello schema.
- `RegistryName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Custom string pattern #46.

  Il nome del registro.
- `LatestVersion`: booleano.

  La versione più recente dello schema.
- `VersionNumber` – Numero (intero), non inferiore a 1 o superiore a 100000.

  Il numero di versione dello schema.
- `SchemaVersionId` – stringa UTF-8, non inferiore a 36 o superiore a 36 byte di lunghezza, corrispondente a Custom string pattern #45.

  L'ID versione per la versione dello schema.
- `MetadataKey` – stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza, corrispondente a Custom string pattern #14.

  La chiave di metadati.
- `MetadataValue` – stringa UTF-8, non inferiore a 1 o superiore a 256 byte di lunghezza, corrispondente a Custom string pattern #14.

  Il valore della chiave di metadati.

Errori

- `InvalidInputException`
- `AccessDeniedException`
- `ThrottlingException`

- `EntityNotFoundException`

## DeleteRegistry azione (Python: delete_registry)

Elimina l'intero registro, inclusi gli schemi e tutte le relative versioni. Per ottenere lo stato dell'operazione di eliminazione, è possibile chiamare l'API `GetRegistry` dopo la chiamata asincrona. L'eliminazione di un registro disattiverà tutte le operazioni online per il registro, come`UpdateRegistry`,, `CreateSchema` e. `UpdateSchema` `RegisterSchemaVersion` APIs

Richiesta

- `RegistryId` - obbligatorio: un oggetto RegistryId.

  Si tratta di una struttura wrapper che può contenere il nome del registro e l'Amazon Resource Name (ARN).

Risposta

- `RegistryName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Custom string pattern #46.

  Il nome del registro in fase di eliminazione.
- `RegistryArn` – stringa UTF-8, non inferiore a 1 o superiore a 10240 byte di lunghezza, corrispondente a Custom string pattern #50.

  L'Amazon Resource Name (ARN) del registro in fase di eliminazione.
- `Status`: stringa UTF-8 (valori validi: `AVAILABLE` | `DELETING`).

  Lo stato del registro. Un'operazione riuscita restituirà lo stato `Deleting`.

Errori

- `InvalidInputException`
- `EntityNotFoundException`
- `AccessDeniedException`
- `ThrottlingException`
- `ConcurrentModificationException`

# DeleteSchema azione (Python: delete_schema)

Elimina l'intero set di schemi, inclusi il set di schemi e tutte le relative versioni. Per ottenere lo stato dell'operazione di eliminazione, è possibile chiamare l'API `GetSchema` dopo la chiamata asincrona. L'eliminazione di un registro disattiverà tutte le operazioni online per lo schema, ad esempio, e. `GetSchemaByDefinition` `RegisterSchemaVersion` APIs

Richiesta

- `SchemaId` - obbligatorio: un oggetto SchemaId.

  Si tratta di una struttura wrapper che può contenere il nome dello schema e l'Amazon Resource Name (ARN).

Risposta

- `SchemaArn` – stringa UTF-8, non inferiore a 1 o superiore a 10240 byte di lunghezza, corrispondente a Custom string pattern #50.

  L'Amazon Resource Name (ARN) dello schema in fase di eliminazione.
- `SchemaName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Custom string pattern #46.

  Il nome dello schema in fase di eliminazione.
- `Status`: stringa UTF-8 (valori validi: `AVAILABLE` | `PENDING` | `DELETING`).

  Lo stato dello schema.

Errori

- `InvalidInputException`
- `EntityNotFoundException`
- `AccessDeniedException`
- `ThrottlingException`
- `ConcurrentModificationException`

# DeleteSchemaVersions azione (Python: delete_schema_versions)

Rimuove le versioni dallo schema specificato. Può essere fornito un numero di versione o un intervallo. Se la modalità di compatibilità impedisce l'eliminazione di una versione necessaria, ad esempio BACKWARDS_FULL, viene restituito un errore. Chiamare l'API `GetSchemaVersions` dopo questa chiamata elencherà lo stato delle versioni eliminate.

Quando l'intervallo di numeri di versione contiene la versione di checkpoint, l'API restituirà un conflitto 409 e non procederà con l'eliminazione. Prima di utilizzare questa API è necessario rimuovere il checkpoint usando l'API `DeleteSchemaCheckpoint`

Non è possibile utilizzare l'API `DeleteSchemaVersions` per eliminare la prima versione dello schema nel set di schemi. La prima versione dello schema può essere eliminata solo dall'API `DeleteSchema`. Questa operazione eliminerà anche i `SchemaVersionMetadata` collegati nelle versioni dello schema. Le eliminazioni definitive verranno applicate al database.

Se la modalità di compatibilità impedisce l'eliminazione di una versione necessaria, ad esempio BACKWARDS_FULL, viene restituito un errore.

Richiesta

- `SchemaId` - obbligatorio: un oggetto SchemaId.

  Si tratta di una struttura wrapper che può contenere il nome dello schema e l'Amazon Resource Name (ARN).

- `Versions`: obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 100000 byte di lunghezza, corrispondente a Custom string pattern #15.

  Può essere fornita un intervallo di versioni nel formato:

  - un unico numero di versione, 5
  - un intervallo, 5-8: elimina le versioni 5, 6, 7, 8

Risposta

- `SchemaVersionErrors`: una matrice di oggetti SchemaVersionErrorItem.

  Un elenco di oggetti `SchemaVersionErrorItem`, ciascuno contenente un errore e una versione dello schema.

Errori

- `InvalidInputException`
- `EntityNotFoundException`
- `AccessDeniedException`
- `ThrottlingException`
- `ConcurrentModificationException`

# Flussi di lavoro

L'API Workflows descrive i tipi di dati e l'API relativi alla creazione, all'aggiornamento o alla visualizzazione dei flussi di lavoro in. AWS Glue La cronologia di esecuzione dei processi è accessibile per 90 giorni per il flusso di lavoro e l'esecuzione del processo.

## Tipi di dati

- JobNodeDetails struttura
- CrawlerNodeDetails struttura
- TriggerNodeDetails struttura
- Struttura crawl
- Struttura nodo
- Struttura edge
- Struttura flusso di lavoro
- WorkflowGraph struttura
- WorkflowRun struttura
- WorkflowRunStatistics struttura
- StartingEventBatchCondition struttura
- Struttura blueprint
- BlueprintDetails struttura
- LastActiveDefinition struttura
- BlueprintRun struttura

## JobNodeDetails struttura

I dettagli di un nodo processo presenti nel flusso di lavoro.

Campi

- `JobRuns`: una matrice di oggetti JobRun.

  Le informazioni sulle esecuzioni del processo rappresentate dal nodo processo.

## CrawlerNodeDetails struttura

I dettagli di un nodo crawler presenti nel flusso di lavoro.

Campi

- `Crawls`: una matrice di oggetti Crawl.

  Un elenco di esecuzioni del crawler rappresentato dal nodo crawler.

## TriggerNodeDetails struttura

I dettagli di un nodo trigger presenti nel flusso di lavoro.

Campi

- `Trigger`: un oggetto Trigger.

  Le informazioni del trigger rappresentate dal nodo trigger.

## Struttura crawl

I dettagli di una esecuzione del crawler nel flusso di lavoro.

Campi

- `State`: stringa UTF-8 (valori validi: `RUNNING` | `CANCELLING` | `CANCELLED` | `SUCCEEDED` | `FAILED` | `ERROR`).

  Lo stato del crawler.

- `StartedOn`: timestamp.

  La data e l'ora in cui è stata avviata l'esecuzione del crawler.

- `CompletedOn`: timestamp.

  La data e l'ora in cui si è conclusa l'esecuzione del crawler.

- `ErrorMessage`: stringa di descrizione, non superiore a 2048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  Il messaggio di errore associato al crawler.

- `LogGroup`: stringa UTF-8, non inferiore a 1 o superiore a 512 byte di lunghezza, corrispondente a Log group string pattern.

  Il gruppo di log associato al crawler.

- `LogStream`: stringa UTF-8, non inferiore a 1 o superiore a 512 byte di lunghezza, corrispondente a Log-stream string pattern.

  Il flusso di log associato all'esecuzione del crawler.

## Struttura nodo

Un nodo rappresenta un AWS Glue componente (trigger, crawler o job) su un grafico del flusso di lavoro.

Campi

- `Type`: stringa UTF-8 (valori validi: `CRAWLER` | `JOB` | `TRIGGER`).

  Il tipo di AWS Glue componente rappresentato dal nodo.

- `Name`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del AWS Glue componente rappresentato dal nodo.

- `UniqueId`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID univoco assegnato al nodo all'interno del flusso di lavoro.

- `TriggerDetails`: un oggetto TriggerNodeDetails.

Le informazioni sul trigger quando il nodo rappresenta un trigger.

- `JobDetails`: un oggetto JobNodeDetails.

  Le informazioni sul processo quando il nodo rappresenta un processo.

- `CrawlerDetails`: un oggetto CrawlerNodeDetails.

  Dettagli del crawler quando il nodo rappresenta un crawler.

## Struttura edge

Un bordo rappresenta una connessione diretta tra due AWS Glue componenti che fanno parte del flusso di lavoro a cui appartiene il bordo.

Campi

- `SourceId`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'id univoco del nodo all'interno del flusso di lavoro in cui ha origine l'edge.

- `DestinationId`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'id univoco del nodo all'interno del flusso di lavoro in cui termina l'edge.

## Struttura flusso di lavoro

Un flusso di lavoro è una raccolta di più AWS Glue job e crawler dipendenti che vengono eseguiti per completare un'attività ETL complessa. Ogni flusso di lavoro gestisce l'esecuzione e il monitoraggio di tutti i suoi processi e crawler.

Campi

- `Name`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome del flusso di lavoro.

- `Description`: stringa UTF-8.

Una descrizione del flusso di lavoro.

- `DefaultRunProperties`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Ogni valore è una stringa UTF-8.

  Un insieme di proprietà da utilizzare come parte di ogni esecuzione del flusso di lavoro. Le proprietà di esecuzione vengono rese disponibili per ogni processo nel flusso di lavoro. Un processo può modificare le proprietà dei processi successivi nel flusso.

- `CreatedOn`: timestamp.

  La data e l'ora in cui il flusso di lavoro è stato creato.

- `LastModifiedOn`: timestamp.

  La data e l'ora più recenti in cui il flusso di lavoro è stato modificato.

- `LastRun`: un oggetto WorkflowRun.

  Le informazioni relative all'ultima esecuzione del flusso di lavoro.

- `Graph`: un oggetto WorkflowGraph.

  Il grafico che rappresenta tutti i AWS Glue componenti che appartengono al flusso di lavoro come nodi e le connessioni dirette tra di essi come bordi.

- `CreationStatus`: stringa UTF-8 (valori validi: `CREATING` | `CREATED` | `CREATION_FAILED`).

  Lo stato della creazione del flusso di lavoro.

- `MaxConcurrentRuns`: numero (intero).

  È possibile utilizzare questo parametro per impedire aggiornamenti multipli indesiderati dei dati, per controllare i costi o, in alcuni casi, per evitare il superamento del numero massimo di esecuzioni simultanee di uno qualsiasi dei processi componenti. Se si lascia questo parametro vuoto, non è previsto alcun limite al numero di esecuzioni simultanee del flusso di lavoro.

- `BlueprintDetails`: un oggetto BlueprintDetails.

  Questa struttura indica i dettagli del blueprint da cui viene creato questo particolare flusso di lavoro.

## WorkflowGraph struttura

Un diagramma del flusso di lavoro rappresenta il flusso di lavoro completo che contiene tutti i componenti di AWS Glue presenti nel flusso di lavoro e tutte le connessioni orientate esistenti tra essi.

Campi

- `Nodes`: una matrice di oggetti Nodo.

  Un elenco dei AWS Glue componenti appartengono al flusso di lavoro rappresentato come nodi.

- `Edges`: una matrice di oggetti Edge.

  Un elenco di tutte le connessioni orientate tra i nodi appartenenti al flusso di lavoro.

## WorkflowRun struttura

Un'esecuzione di un flusso di lavoro è costituita da tutte le informazioni di runtime sull'esecuzione del flusso stesso.

Campi

- `Name`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome del flusso di lavoro che è stato eseguito.

- `WorkflowRunId`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID di questa esecuzione del flusso di lavoro.

- `PreviousRunId`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID dell'esecuzione del flusso di lavoro precedente.

- `WorkflowRunProperties`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

Ogni valore è una stringa UTF-8.

Le proprietà di esecuzione del flusso di lavoro impostate durante l'esecuzione.

- `StartedOn`: timestamp.

  La data e l'ora in cui è stata avviata l'esecuzione del flusso di lavoro.

- `CompletedOn`: timestamp.

  La data e l'ora in cui si è conclusa l'esecuzione del flusso di lavoro.

- `Status`: stringa UTF-8 (valori validi: `RUNNING` | `COMPLETED` | `STOPPING` | `STOPPED` | `ERROR`).

  Lo stato dell'esecuzione del flusso di lavoro.

- `ErrorMessage`: stringa UTF-8.

  Questo messaggio di errore descrive qualsiasi errore che potrebbe essersi verificato durante l'avvio dell'esecuzione del flusso di lavoro. Attualmente l'unico messaggio di errore è "Esecuzioni simultanee superate per il flusso di lavoro: `foo`".

- `Statistics`: un oggetto WorkflowRunStatistics.

  Le statistiche dell'esecuzione.

- `Graph`: un oggetto WorkflowGraph.

  Il grafico che rappresenta tutti i AWS Glue componenti che appartengono al flusso di lavoro come nodi e le connessioni dirette tra di essi come bordi.

- `StartingEventBatchCondition`: un oggetto StartingEventBatchCondition.

  La condizione batch che ha avviato l'esecuzione del flusso di lavoro.

## WorkflowRunStatistics struttura

Le statistiche di esecuzione del flusso di lavoro forniscono le statistiche sull'esecuzione di quest'ultimo.

Campi

- `TotalActions`: numero (intero).

  Numero totale di operazioni nell'esecuzione del flusso di lavoro.

- `TimeoutActions`: numero (intero).

  Numero totale di operazioni andate in timeout.

- `FailedActions`: numero (intero).

  Numero totale di operazioni che non si sono concluse correttamente.

- `StoppedActions`: numero (intero).

  Numero totale di operazioni che sono state interrotte.

- `SucceededActions`: numero (intero).

  Numero totale di operazioni che si sono concluse correttamente.

- `RunningActions`: numero (intero).

  Numero totale di operazioni in stato di esecuzione.

- `ErroredActions`: numero (intero).

  Indica il numero di esecuzioni del processo nello stato ERRORE nell'esecuzione del flusso di lavoro.

- `WaitingActions`: numero (intero).

  Indica il numero di esecuzioni del processo nello stato IN ATTESA nell'esecuzione del flusso di lavoro.

## StartingEventBatchCondition struttura

La condizione batch che ha avviato l'esecuzione del flusso di lavoro. È arrivato il numero di eventi nella dimensione del batch, nel qual caso il BatchSize membro è diverso da zero, oppure la finestra del batch è scaduta, nel qual caso il BatchWindow membro è diverso da zero.

Campi

- `BatchSize`: numero (intero).

  Numero di eventi nel batch.

- `BatchWindow`: numero (intero).

  Durata del periodo di batch in secondi.

# Struttura blueprint

I dettagli di un blueprint.

Campi

- `Name`: stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza, corrispondente a Custom string pattern #31.

  Nome del blueprint.

- `Description`: stringa UTF-8, non inferiore a 1 o superiore a 512 byte di lunghezza.

  La descrizione del blueprint.

- `CreatedOn`: timestamp.

  La data e l'ora di registrazione del blueprint.

- `LastModifiedOn`: timestamp.

  La data e l'ora dell'ultima modifica apportata al blueprint.

- `ParameterSpec`: stringa UTF-8, non inferiore a 1 o superiore a 131072 byte di lunghezza.

  Una stringa JSON che indica l'elenco delle specifiche dei parametri per il blueprint.

- `BlueprintLocation`: stringa UTF-8.

  Specifica il percorso in Amazon S3 in cui è pubblicato il blueprint.

- `BlueprintServiceLocation`: stringa UTF-8.

  Specifica un percorso in Amazon S3 in cui il blueprint viene copiato quando si chiama `CreateBlueprint/UpdateBlueprint` per registrare il blueprint in AWS Glue.

- `Status`: stringa UTF-8 (valori validi: `CREATING` | `ACTIVE` | `UPDATING` | `FAILED`).

  Stato della registrazione del blueprint.

  - Creazione: la registrazione del blueprint è in corso.
  - Attivo: il blueprint è stato registrato correttamente.
  - Aggiornamento: è in corso un aggiornamento della registrazione del blueprint.
  - Non riuscito: registrazione del blueprint non riuscita.

- `ErrorMessage`: stringa UTF-8.

Un messaggio di errore.

- `LastActiveDefinition`: un oggetto LastActiveDefinition.

  Quando sono presenti più versioni di un blueprint e la versione più recente presenta alcuni errori, questo attributo indica l'ultima definizione del blueprint riuscita disponibile con il servizio.

## BlueprintDetails struttura

I dettagli di un blueprint.

Campi

- `BlueprintName`: stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza, corrispondente a Custom string pattern #31.

  Nome del blueprint.
- `RunId`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  ID esecuzione per questo blueprint.

## LastActiveDefinition struttura

Quando sono presenti più versioni di un blueprint e la versione più recente presenta alcuni errori, questo attributo indica l'ultima definizione del blueprint riuscita disponibile con il servizio.

Campi

- `Description`: stringa UTF-8, non inferiore a 1 o superiore a 512 byte di lunghezza.

  La descrizione del blueprint.
- `LastModifiedOn`: timestamp.

  La data e l'ora dell'ultima modifica apportata al blueprint.
- `ParameterSpec`: stringa UTF-8, non inferiore a 1 o superiore a 131072 byte di lunghezza.

  Una stringa JSON che specifica i parametri per il blueprint.
- `BlueprintLocation`: stringa UTF-8.

Speciifica un percorso in Amazon S3 in cui il blueprint viene pubblicato dallo sviluppatore. AWS Glue

- `BlueprintServiceLocation`: stringa UTF-8.

  Specifica un percorso in Amazon S3 in cui viene copiato il progetto quando crei o aggiorni il progetto.

## BlueprintRun struttura

I dettagli dell'esecuzione di un blueprint.

Campi

- `BlueprintName`: stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza, corrispondente a Custom string pattern #31.

  Nome del blueprint.

- `RunId`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID esecuzione per l'esecuzione del blueprint.

- `WorkflowName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome di un flusso di lavoro creato a seguito di un'esecuzione del blueprint riuscita. Se l'esecuzione di un blueprint presenta un errore, non verrà creato alcun flusso di lavoro.

- `State`: stringa UTF-8 (valori validi: `RUNNING` | `SUCCEEDED` | `FAILED` | `ROLLING_BACK`).

  Stato dell'esecuzione del blueprint. I valori possibili sono:

  - In esecuzione: l'esecuzione del blueprint è in corso.
  - Riuscito: l'esecuzione del blueprint è stata completata correttamente.
  - Non riuscito: l'esecuzione del blueprint non è riuscita e il ripristino dello stato precedente è completato.
  - Ripristino dello stato precedente: l'esecuzione del blueprint non è riuscita ed è in corso il ripristino dello stato precedente.

- `StartedOn`: timestamp.

La data e l'ora in cui è stata avviata l'esecuzione del blueprint.

- `CompletedOn`: timestamp.

  La data e l'ora in cui è stata completata l'esecuzione del blueprint.

- `ErrorMessage`: stringa UTF-8.

  Indica eventuali errori rilevati durante l'esecuzione del blueprint.

- `RollbackErrorMessage`: stringa UTF-8.

  Se ci sono errori durante la creazione delle entità di un flusso di lavoro, si tenta di ripristinare le entità create fino a quel punto ed eliminarle. Questo attributo indica gli errori rilevati durante il tentativo di eliminare le entità create.

- `Parameters` – stringa UTF-8, non inferiore a 1 o superiore a 131072 byte di lunghezza.

  I parametri del blueprint come stringa. Dovrai fornire un valore per ogni chiave richiesta dalla specifica del parametro, definita nella `Blueprint$ParameterSpec`.

- `RoleArn`: stringa UTF-8, non inferiore a 1 o superiore a 1024 byte di lunghezza, corrispondente a Custom string pattern #30.

  ARN del ruolo. Questo ruolo verrà assunto dal AWS Glue servizio e verrà utilizzato per creare il flusso di lavoro e altre entità di un flusso di lavoro.

## Operazioni

- CreateWorkflow azione (Python: create_workflow)
- UpdateWorkflow azione (Python: update_workflow)
- DeleteWorkflow azione (Python: delete_workflow)
- GetWorkflow azione (Python: get_workflow)
- ListWorkflows azione (Python: list_workflows)
- BatchGetWorkflows azione (Python: batch_get_workflows)
- GetWorkflowRun azione (Python: get_workflow_run)
- GetWorkflowRuns azione (Python: get_workflow_runs)
- GetWorkflowRunProperties azione (Python: get_workflow_run_properties)
- PutWorkflowRunProperties azione (Python: put_workflow_run_properties)

- CreateBlueprint azione (Python: create_blueprint)
- UpdateBlueprint azione (Python: update_blueprint)
- DeleteBlueprint azione (Python: delete_blueprint)
- ListBlueprints azione (Python: list_blueprints)
- BatchGetBlueprints azione (Python: batch_get_blueprints)
- StartBlueprintRun azione (Python: start_blueprint_run)
- GetBlueprintRun azione (Python: get_blueprint_run)
- GetBlueprintRuns azione (Python: get_blueprint_runs)
- StartWorkflowRun azione (Python: start_workflow_run)
- StopWorkflowRun azione (Python: stop_workflow_run)
- ResumeWorkflowRun azione (Python: resume_workflow_run)

## CreateWorkflow azione (Python: create_workflow)

Crea un nuovo flusso di lavoro.

Richiesta

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome da assegnare al nuovo flusso di lavoro. Deve essere univoco all'interno dell'account.
- `Description`— Stringa UTF-8, lunga non più di 120000 byte.

  Una descrizione del flusso di lavoro.
- `DefaultRunProperties`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Ogni valore è una stringa UTF-8.

  Un insieme di proprietà da utilizzare come parte di ogni esecuzione del flusso di lavoro.

  Le proprietà di esecuzione potrebbero essere registrate. Non passare segreti in testo semplice come proprietà. Recupera i segreti da AWS Glue Connection, AWS Secrets Manager o altro

meccanismo di gestione dei segreti se intendi utilizzarli all'interno dell'esecuzione del flusso di lavoro.

- `Tags`: una matrice di mappe con coppie chiave-valore, non superiore alle 50 coppie.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Ogni valore è una stringa UTF-8, lunga non più di 256 byte.

  I tag da utilizzare con questo flusso di lavoro.

- `MaxConcurrentRuns`: numero (intero).

  È possibile utilizzare questo parametro per impedire aggiornamenti multipli indesiderati dei dati, per controllare i costi o, in alcuni casi, per evitare il superamento del numero massimo di esecuzioni simultanee di uno qualsiasi dei processi componenti. Se si lascia questo parametro vuoto, non è previsto alcun limite al numero di esecuzioni simultanee del flusso di lavoro.

Risposta

- `Name`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome del flusso di lavoro fornito come parte della richiesta.

Errori

- `AlreadyExistsException`
- `InvalidInputException`
- `InternalServiceException`
- `OperationTimeoutException`
- `ResourceNumberLimitExceededException`
- `ConcurrentModificationException`

## UpdateWorkflow azione (Python: update_workflow)

Aggiorna un flusso di lavoro esistente.

Richiesta

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome del flusso di lavoro da aggiornare.

- `Description`— Stringa UTF-8, lunga non più di 120000 byte.

  La descrizione del flusso di lavoro.

- `DefaultRunProperties`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Ogni valore è una stringa UTF-8.

  Un insieme di proprietà da utilizzare come parte di ogni esecuzione del flusso di lavoro.

  Le proprietà di esecuzione potrebbero essere registrate. Non passare segreti in testo semplice come proprietà. Recupera i segreti da AWS Glue Connection, AWS Secrets Manager o altro meccanismo di gestione dei segreti se intendi utilizzarli all'interno dell'esecuzione del flusso di lavoro.

- `MaxConcurrentRuns`: numero (intero).

  È possibile utilizzare questo parametro per impedire aggiornamenti multipli indesiderati dei dati, per controllare i costi o, in alcuni casi, per evitare il superamento del numero massimo di esecuzioni simultanee di uno qualsiasi dei processi componenti. Se si lascia questo parametro vuoto, non è previsto alcun limite al numero di esecuzioni simultanee del flusso di lavoro.

Risposta

- `Name`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome del flusso di lavoro fornito nella richiesta.

Errori

- `InvalidInputException`

- `EntityNotFoundException`
- `InternalServiceException`
- `OperationTimeoutException`
- `ConcurrentModificationException`

## DeleteWorkflow azione (Python: delete_workflow)

Elimina un flusso di lavoro.

Richiesta

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome del flusso di lavoro da eliminare.

Risposta

- `Name`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome del flusso di lavoro fornito nella richiesta.

Errori

- `InvalidInputException`
- `InternalServiceException`
- `OperationTimeoutException`
- `ConcurrentModificationException`

## GetWorkflow azione (Python: get_workflow)

Recupera i metadati delle risorse per un flusso di lavoro.

Richiesta

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome del flusso di lavoro da recuperare.

- `IncludeGraph`: booleano.

  Specifica se includere un diagramma al momento della restituzione dei metadati delle risorse del flusso di lavoro.

Risposta

- `Workflow`: un oggetto Flusso di lavoro.

  I metadati delle risorse per il flusso di lavoro.

Errori

- `InvalidInputException`
- `EntityNotFoundException`
- `InternalServiceException`
- `OperationTimeoutException`

## ListWorkflows azione (Python: list_workflows)

Elenca i nomi dei flussi di lavoro creati nell'account.

Richiesta

- `NextToken`: stringa UTF-8.

  Token di continuazione, se si tratta di una richiesta di continuazione.

- `MaxResults`— Numero (intero), non inferiore a 1 o superiore a 25.

  La dimensione massima di un elenco da restituire.

Risposta

- `Workflows`: una matrice di stringhe UTF-8, non inferiore a 1 o superiore a 25 stringhe.

  Elenco dei nomi dei flussi di lavoro nell'account.
- `NextToken`: stringa UTF-8.

  Un token di continuazione, se non tutti i nomi di flussi di lavoro sono stati restituiti.

Errori

- `InvalidInputException`
- `InternalServiceException`
- `OperationTimeoutException`

## BatchGetWorkflows azione (Python: batch_get_workflows)

Restituisce un elenco di metadati di risorse per un elenco di nomi di flussi di lavoro. Dopo aver chiamato l'operazione `ListWorkflows`, puoi chiamare questa operazione per accedere ai dati a cui sono state concesse le autorizzazioni. Questa operazione supporta tutte le autorizzazioni IAM, tra cui le condizioni di autorizzazione che utilizzano i tag.

Richiesta

- `Names` - obbligatorio: una serie di stringhe UTF-8, non inferiore a 1 o superiore a 25 stringhe.

  L'elenco dei nomi di flussi di lavoro, che potrebbero essere i nomi restituiti dall'operazione `ListWorkflows`.
- `IncludeGraph`: booleano.

  Specifica se includere un diagramma al momento della restituzione dei metadati delle risorse del flusso di lavoro.

Risposta

- `Workflows`: una matrice di oggetti Flusso di lavoro, non inferiore a 1 o superiore a 25 strutture.

  Un elenco di metadati delle risorse del flusso di lavoro.

- `MissingWorkflows`: una serie di stringhe UTF-8, non inferiore a 1 o superiore a 25 stringhe.

  Un elenco di nomi di flussi di lavoro non trovati.

Errori

- `InternalServiceException`
- `OperationTimeoutException`
- `InvalidInputException`

## GetWorkflowRun azione (Python: get_workflow_run)

Consente di recuperare i metadati di una specifica esecuzione di un flusso di lavoro. La cronologia di esecuzione dei processi è accessibile per 90 giorni per flusso di lavoro ed esecuzione del processo.

Richiesta

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome del flusso di lavoro in esecuzione.
- `RunId` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID dell'esecuzione del flusso di lavoro.
- `IncludeGraph`: booleano.

  Specifica se includere o meno il diagramma del flusso di lavoro nella risposta.

Risposta

- `Run`: un oggetto WorkflowRun.

  I metadati dell'esecuzione del flusso di lavoro richiesti.

Errori

- `InvalidInputException`

- `EntityNotFoundException`
- `InternalServiceException`
- `OperationTimeoutException`

## GetWorkflowRuns azione (Python: get_workflow_runs)

Recupera i metadati di tutte le esecuzioni di un dato flusso di lavoro.

Richiesta

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome del flusso di lavoro i cui metadati di esecuzione devono essere restituiti.
- `IncludeGraph`: booleano.

  Specifica se includere o meno il diagramma del flusso di lavoro nella risposta.
- `NextToken`: stringa UTF-8.

  La dimensione massima della risposta.
- `MaxResults`: numero (intero), non inferiore a 1 o superiore a 1.000.

  Il numero massimo di esecuzioni del flusso di lavoro da includere nella risposta.

Risposta

- `Runs`: una matrice di oggetti WorkflowRun, non inferiore a 1 o superiore a 1.000 strutture.

  Un elenco oggetti che rappresentano i metadati dell'esecuzione di un flusso di lavoro.
- `NextToken`: stringa UTF-8.

  Un token di continuazione, se non tutte le esecuzioni del flusso di lavoro richieste sono state restituite.

Errori

- `InvalidInputException`
- `EntityNotFoundException`

- `InternalServiceException`
- `OperationTimeoutException`

## GetWorkflowRunProperties azione (Python: get_workflow_run_properties)

Recupera le proprietà dell'esecuzione del flusso di lavoro che sono state impostate durante l'esecuzione.

Richiesta

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome del flusso di lavoro che è stato eseguito.
- `RunId` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del flusso di lavoro le cui proprietà dell'esecuzione devono essere restituite.

Risposta

- `RunProperties`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Ogni valore è una stringa UTF-8.

  Le proprietà dell'esecuzione del flusso di lavoro che sono state impostate durante l'esecuzione specificata.

Errori

- `InvalidInputException`
- `EntityNotFoundException`
- `InternalServiceException`
- `OperationTimeoutException`

# PutWorkflowRunProperties azione (Python: put_workflow_run_properties)

Imposta le proprietà dell'esecuzione del flusso di lavoro specificate per la specifica esecuzione del flusso di lavoro. Se una proprietà esiste già per l'esecuzione specificata, il vecchio valore viene sovrascritto, altrimenti aggiunge la proprietà alle proprietà esistenti.

Richiesta

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome del flusso di lavoro che è stato eseguito.
- `RunId` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID dell'esecuzione del flusso di lavoro per il quale è necessario aggiornare le proprietà dell'esecuzione.
- `RunProperties` - obbligatorio: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Ogni valore è una stringa UTF-8.

  Le proprietà da impostare per l'esecuzione specificata.

  Le proprietà di esecuzione potrebbero essere registrate. Non passare segreti in testo semplice come proprietà. Recupera i segreti da AWS Glue Connection, AWS Secrets Manager o altro meccanismo di gestione dei segreti se intendi utilizzarli all'interno dell'esecuzione del flusso di lavoro.

Risposta

- Nessun parametro di risposta.

Errori

- `AlreadyExistsException`
- `EntityNotFoundException`

- `InvalidInputException`
- `InternalServiceException`
- `OperationTimeoutException`
- `ResourceNumberLimitExceededException`
- `ConcurrentModificationException`

## CreateBlueprint azione (Python: create_blueprint)

Registra un blueprint con. AWS Glue

Richiesta

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza, corrispondente a Custom string pattern #31.

  Nome del blueprint.
- `Description`: stringa UTF-8, non inferiore a 1 o superiore a 512 byte di lunghezza.

  Una descrizione del blueprint.
- `BlueprintLocation` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 8192 byte di lunghezza, corrispondente a Custom string pattern #32.

  Specifica il percorso in Amazon S3 in cui è pubblicato il blueprint.
- `Tags`: una matrice di mappe con coppie chiave-valore, non superiore alle 50 coppie.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Ogni valore è una stringa UTF-8, lunga non più di 256 byte.

  Tag da applicare a questo blueprint.

Risposta

- `Name`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Restituisce il nome del blueprint registrato.

Errori

- `AlreadyExistsException`
- `InvalidInputException`
- `OperationTimeoutException`
- `InternalServiceException`
- `ResourceNumberLimitExceededException`

## UpdateBlueprint azione (Python: update_blueprint)

Aggiorna un blueprint registrato.

Richiesta

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza, corrispondente a Custom string pattern #31.

  Nome del blueprint.
- `Description`: stringa UTF-8, non inferiore a 1 o superiore a 512 byte di lunghezza.

  Una descrizione del blueprint.
- `BlueprintLocation` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 8192 byte di lunghezza, corrispondente a Custom string pattern #32.

  Specifica il percorso in Amazon S3 in cui è pubblicato il blueprint.

Risposta

- `Name`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Restituisce il nome del blueprint aggiornato.

Errori

- `EntityNotFoundException`
- `ConcurrentModificationException`

- `InvalidInputException`
- `OperationTimeoutException`
- `InternalServiceException`
- `IllegalBlueprintStateException`

## DeleteBlueprint azione (Python: delete_blueprint)

Elimina un blueprint esistente.

Richiesta

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome del blueprint da eliminare.

Risposta

- `Name`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Restituisce il nome del blueprint eliminato.

Errori

- `InvalidInputException`
- `OperationTimeoutException`
- `InternalServiceException`

## ListBlueprints azione (Python: list_blueprints)

Elenca tutti i nomi dei blueprint in un account.

Richiesta

- `NextToken`: stringa UTF-8.

  Token di continuazione, se si tratta di una richiesta di continuazione.

- `MaxResults`— Numero (intero), non inferiore a 1 o superiore a 25.

  La dimensione massima di un elenco da restituire.

- `Tags`: una matrice di mappe con coppie chiave-valore, non superiore alle 50 coppie.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Ogni valore è una stringa UTF-8, lunga non più di 256 byte.

  Filtra l'elenco in base a un tag di AWS risorsa.

Risposta

- `Blueprints`: una matrice di stringhe UTF-8.

  Elenco dei nomi dei blueprint nell'account.

- `NextToken`: stringa UTF-8.

  Un token di continuazione, se non sono stati restituiti tutti i nomi di piani.

Errori

- `InvalidInputException`
- `InternalServiceException`
- `OperationTimeoutException`

## BatchGetBlueprints azione (Python: batch_get_blueprints)

Recupera le informazioni su un elenco dei blueprint.

Richiesta

- `Names` - obbligatorio: una serie di stringhe UTF-8, non inferiore a 1 o superiore a 25 stringhe.

  Un elenco di nomi di piani.

- `IncludeBlueprint`: booleano.

  Specifica se includere o meno il blueprint nella risposta.

- `IncludeParameterSpec`: booleano.

  Specifica se includere o meno i parametri, come stringa JSON, per il blueprint nella risposta.

Risposta

- `Blueprints`: una matrice di oggetti Blueprint.

  Restituisce un elenco di blueprint come oggetto `Blueprints`.
- `MissingBlueprints`: una matrice di stringhe UTF-8.

  Restituisce un elenco di `BlueprintNames` che non sono stati trovati.

Errori

- `InternalServiceException`
- `OperationTimeoutException`
- `InvalidInputException`

## StartBlueprintRun azione (Python: start_blueprint_run)

Avvia una nuova esecuzione del blueprint specificato.

Richiesta

- `BlueprintName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza, corrispondente a Custom string pattern #31.

  Nome del blueprint.
- `Parameters`: stringa UTF-8, non inferiore a 1 o superiore a 131072 byte di lunghezza.

  Specifica i parametri come oggetto `BlueprintParameters`.
- `RoleArn`: obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 1024 byte di lunghezza, corrispondente a Custom string pattern #30.

  Specifica il ruolo IAM utilizzato per creare il flusso di lavoro.

Risposta

- `RunId`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID esecuzione per l'esecuzione del blueprint.

Errori

- `InvalidInputException`
- `OperationTimeoutException`
- `InternalServiceException`
- `ResourceNumberLimitExceededException`
- `EntityNotFoundException`
- `IllegalBlueprintStateException`

## GetBlueprintRun azione (Python: get_blueprint_run)

Recupera i dettagli dell'esecuzione di un blueprint.

Richiesta

- `BlueprintName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza, corrispondente a Custom string pattern #31.

  Nome del blueprint.
- `RunId` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID esecuzione per l'esecuzione del blueprint che si intende recuperare.

Risposta

- `BlueprintRun`: un oggetto BlueprintRun.

  Restituisce un oggetto `BlueprintRun`.

Errori

- `EntityNotFoundException`
- `InternalServiceException`
- `OperationTimeoutException`

## GetBlueprintRuns azione (Python: get_blueprint_runs)

Recupera i dettagli delle esecuzioni del blueprint per un blueprint specificato.

Richiesta

- `BlueprintName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome del blueprint.
- `NextToken`: stringa UTF-8.

  Token di continuazione, se si tratta di una richiesta di continuazione.
- `MaxResults`: numero (intero), non inferiore a 1 o superiore a 1.000.

  La dimensione massima di un elenco da restituire.

Risposta

- `BlueprintRuns`: una matrice di oggetti BlueprintRun.

  Restituisce un elenco di oggetti `BlueprintRun`.
- `NextToken`: stringa UTF-8.

  Un token di continuazione, se non sono state restituite tutte le esecuzioni del blueprint.

Errori

- `EntityNotFoundException`
- `InternalServiceException`
- `OperationTimeoutException`

- `InvalidInputException`

## StartWorkflowRun azione (Python: start_workflow_run)

Avvia una nuova esecuzione del flusso di lavoro specificato.

Richiesta

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome del flusso di lavoro da avviare.
- `RunProperties`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Ogni valore è una stringa UTF-8.

  Le proprietà dell'esecuzione del flusso di lavoro per la nuova esecuzione del flusso di lavoro.

  Le proprietà di esecuzione potrebbero essere registrate. Non passare segreti in testo semplice come proprietà. Recupera i segreti da AWS Glue Connection, AWS Secrets Manager o altro meccanismo di gestione dei segreti se intendi utilizzarli all'interno dell'esecuzione del flusso di lavoro.

Risposta

- `RunId`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Un ID per la nuova esecuzione.

Errori

- `InvalidInputException`
- `EntityNotFoundException`
- `InternalServiceException`

- `OperationTimeoutException`
- `ResourceNumberLimitExceededException`
- `ConcurrentRunsExceededException`

## StopWorkflowRun azione (Python: stop_workflow_run)

Interrompe l'esecuzione del flusso di lavoro specificato.

Richiesta

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome del flusso di lavoro da arrestare.
- `RunId` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID dell'esecuzione del flusso di lavoro da arrestare.

Risposta

- Nessun parametro di risposta.

Errori

- `InvalidInputException`
- `EntityNotFoundException`
- `InternalServiceException`
- `OperationTimeoutException`
- `IllegalWorkflowStateException`

## ResumeWorkflowRun azione (Python: resume_workflow_run)

Riavvia i nodi selezionati di una precedente esecuzione del flusso di lavoro parzialmente completata e riprende l'esecuzione del flusso di lavoro. Vengono eseguiti i nodi selezionati e tutti i nodi che sono a valle dei nodi selezionati.

Richiesta

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome del flusso di lavoro da recuperare.

- `RunId` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID dell'esecuzione del flusso di lavoro da arrestare.

- `NodeIds` - obbligatorio: una matrice di stringhe UTF-8.

  Un elenco dei nodi IDs per i nodi che desideri riavviare. I nodi che devono essere riavviati devono avere un tentativo di esecuzione nell'esecuzione originale.

Risposta

- `RunId`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nuovo ID assegnato all'esecuzione del flusso di lavoro ripresa. Ogni ripresa dell'esecuzione del flusso di lavoro avrà un nuovo ID esecuzione.

- `NodeIds`: una matrice di stringhe UTF-8.

  Un elenco dei nodi IDs per i nodi che sono stati effettivamente riavviati.

Errori

- `InvalidInputException`
- `EntityNotFoundException`
- `InternalServiceException`
- `OperationTimeoutException`
- `ConcurrentRunsExceededException`
- `IllegalWorkflowStateException`

# Profili di utilizzo

L'API dei profili di utilizzo descrive i tipi di dati e l'API relativi alla creazione, all'aggiornamento o alla visualizzazione dei profili di utilizzo in AWS Glue.

## Tipi di dati

- ProfileConfiguration struttura
- ConfigurationObject struttura
- UsageProfileDefinition struttura

## ProfileConfiguration struttura

Speciifica i valori del processo e della sessione che un amministratore configura in un profilo di AWS Glue utilizzo.

Campi

- `SessionConfiguration`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Ogni valore è un oggetto ConfigurationObject.

  Una mappa chiave-valore dei parametri di configurazione per le sessioni. AWS Glue
- `JobConfiguration`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Ogni valore è un oggetto ConfigurationObject.

  Una mappa chiave-valore dei parametri di configurazione per i lavori. AWS Glue

## ConfigurationObject struttura

Speciifica i valori impostati da un amministratore per ogni parametro di processo o sessione configurato in un profilo di AWS Glue utilizzo.

Campi

- `DefaultValue`: stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza, corrispondente a Custom string pattern #35.

  Un valore predefinito del parametro.

- `AllowedValues`: una matrice di stringhe UTF-8.

  Un elenco di valori consentiti del parametro.

- `MinValue`: stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza, corrispondente a Custom string pattern #35.

  Un valore minimo consentito del parametro.

- `MaxValue`: stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza, corrispondente a Custom string pattern #35.

  Un valore massimo consentito del parametro.

## UsageProfileDefinition struttura

Descrive un profilo di AWS Glue utilizzo.

Campi

- `Name`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del profilo di utilizzo.

- `Description`: stringa di descrizione, non superiore a 2048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  Una descrizione del profilo di utilizzo.

- `CreatedOn`: timestamp.

  La data e l'ora di creazione del profilo di utilizzo.

- `LastModifiedOn`: timestamp.

  La data e l'ora dell'ultima modifica apportata al profilo di utilizzo.

## Operazioni

- CreateUsageProfile azione (Python: create_usage_profile)
- GetUsageProfile azione (Python: get_usage_profile)
- UpdateUsageProfile azione (Python: update_usage_profile)
- DeleteUsageProfile azione (Python: delete_usage_profile)
- ListUsageProfiles azione (Python: list_usage_profiles)

## CreateUsageProfile azione (Python: create_usage_profile)

Crea un profilo di utilizzo AWS Glue .

Richiesta

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del profilo di utilizzo.
- `Description`: stringa di descrizione, non superiore a 2048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  Una descrizione del profilo di utilizzo.
- `Configuration` - obbligatorio: un oggetto ProfileConfiguration.

  Un oggetto `ProfileConfiguration` che specifica i valori di processo e sessione del profilo.
- `Tags`: una matrice di mappe con coppie chiave-valore, non superiore alle 50 coppie.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Ogni valore è una stringa UTF-8, lunga non più di 256 byte.

  Un elenco di tag applicati al profilo di utilizzo.

Risposta

- `Name`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

Il nome del profilo di utilizzo creato.

Errori

- `InvalidInputException`
- `InternalServiceException`
- `AlreadyExistsException`
- `OperationTimeoutException`
- `ResourceNumberLimitExceededException`
- `OperationNotSupportedException`

## GetUsageProfile azione (Python: get_usage_profile)

Recupera informazioni sul profilo di utilizzo specificato. AWS Glue

Richiesta

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del profilo di utilizzo da recuperare.

Risposta

- `Name`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del profilo di utilizzo.
- `Description`: stringa di descrizione, non superiore a 2048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  Una descrizione del profilo di utilizzo.
- `Configuration`: un oggetto ProfileConfiguration.

  Un oggetto `ProfileConfiguration` che specifica i valori di processo e sessione del profilo.
- `CreatedOn`: timestamp.

La data e l'ora di creazione del profilo di utilizzo.

- `LastModifiedOn`: timestamp.

  La data e l'ora dell'ultima modifica apportata al profilo di utilizzo.

Errori

- `InvalidInputException`
- `InternalServiceException`
- `EntityNotFoundException`
- `OperationTimeoutException`
- `OperationNotSupportedException`

## UpdateUsageProfile azione (Python: update_usage_profile)

Aggiornare un profilo di utilizzo AWS Glue .

Richiesta

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del profilo di utilizzo.
- `Description`: stringa di descrizione, non superiore a 2048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  Una descrizione del profilo di utilizzo.
- `Configuration` - obbligatorio: un oggetto ProfileConfiguration.

  Un oggetto `ProfileConfiguration` che specifica i valori di processo e sessione del profilo.

Risposta

- `Name`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del profilo di utilizzo aggiornato.

Errori

- `InvalidInputException`
- `InternalServiceException`
- `EntityNotFoundException`
- `OperationTimeoutException`
- `OperationNotSupportedException`
- `ConcurrentModificationException`

## DeleteUsageProfile azione (Python: delete_usage_profile)

Elimina il profilo di utilizzo specificato. AWS Glue

Richiesta

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del profilo di utilizzo da eliminare.

Risposta

- Nessun parametro di risposta.

Errori

- `InvalidInputException`
- `InternalServiceException`
- `OperationTimeoutException`
- `OperationNotSupportedException`

## ListUsageProfiles azione (Python: list_usage_profiles)

Elenca tutti i profili di utilizzo. AWS Glue

Richiesta

- `NextToken`: stringa UTF-8, non superiore a 400000 byte di lunghezza.

  Un token di continuazione, incluso se si tratta di una chiamata di continuazione.
- `MaxResults`: numero (intero), non inferiore a 1 o superiore a 200.

  Il numero massimo di profili di utilizzo da restituire in una risposta singola.

Risposta

- `Profiles`: una matrice di oggetti UsageProfileDefinition.

  Un elenco di oggetti del profilo di utilizzo (`UsageProfileDefinition`).
- `NextToken`: stringa UTF-8, non superiore a 400000 byte di lunghezza.

  Un token di continuazione, presente se il segmento dell'elenco corrente non è l'ultimo.

Errori

- `InternalServiceException`
- `OperationTimeoutException`
- `InvalidInputException`
- `OperationNotSupportedException`

# API machine learning

L'API Machine learning descrive i tipi di dati relativi al machine learning e include le API per la creazione, l'eliminazione o l'aggiornamento di una trasformazione, oppure l'avvio dell'esecuzione di un'attività di machine learning.

## Tipi di dati

- TransformParameters struttura
- EvaluationMetrics struttura
- MLTransform struttura

- FindMatchesParameters struttura
- FindMatchesMetrics struttura
- ConfusionMatrix struttura
- GlueTable struttura
- TaskRun struttura
- TransformFilterCriteria struttura
- TransformSortCriteria struttura
- TaskRunFilterCriteria struttura
- TaskRunSortCriteria struttura
- TaskRunProperties struttura
- FindMatchesTaskRunProperties struttura
- ImportLabelsTaskRunProperties struttura
- ExportLabelsTaskRunProperties struttura
- LabelingSetGenerationTaskRunProperties struttura
- SchemaColumn struttura
- TransformEncryption struttura
- MLUserDataEncryption struttura
- ColumnImportance struttura

## TransformParameters struttura

I parametri specifici dell'algoritmo che sono associati alla trasformazione basata su machine learning.

Campi

- `TransformType` - Obbligatorio: stringa UTF-8 (valori validi: `FIND_MATCHES`).

  Il tipo di trasformazione basata su machine learning.

  Per ulteriori informazioni sui tipi di trasformazioni basate su machine learning, consultare Creazione di trasformazioni basate su machine learning.
- `FindMatchesParameters`: un oggetto FindMatchesParameters.

  I parametri dell'algoritmo di rilevamento delle corrispondenze.

## EvaluationMetrics struttura

I parametri di valutazione forniscono una stima della qualità della trasformazione basata su machine learning.

Campi

- `TransformType` - Obbligatorio: stringa UTF-8 (valori validi: `FIND_MATCHES`).

  Il tipo di trasformazione basata su machine learning.
- `FindMatchesMetrics`: un oggetto FindMatchesMetrics.

  I parametri di valutazione per l'algoritmo di rilevamento delle corrispondenze.

## MLTransform struttura

Una struttura per la trasformazione basata su machine learning.

Campi

- `TransformId`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID univoco della trasformazione generato per la trasformazione basata su machine learning. L'ID è garantito univoco e non si modifica nel tempo.
- `Name`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Un nome definito dall'utente per la trasformazione basata su machine learning. I nomi non sono garantite come univoci e possono essere modificati in qualsiasi momento.
- `Description`: stringa di descrizione, non superiore a 2048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  Una testo descrittivo esteso definito dall'utente per la trasformazione basata su machine learning. Le descrizioni non sono garantite come univoche e possono essere modificate in qualsiasi momento.
- `Status`: stringa UTF-8 (valori validi: `NOT_READY` | `READY` | `DELETING`).

  Lo stato corrente della trasformazione basata su machine learning.

- `CreatedOn`: timestamp.

  Un Timestamp. La data e l'ora di creazione di questa trasformazione basata su machine learning.

- `LastModifiedOn`: timestamp.

  Un Timestamp. L'ultimo istante temporale in cui questa trasformazione basata su machine learning è stata modificata.

- `InputRecordTables`: una matrice di oggetti GlueTable, non superiore a 10 strutture.

  Un elenco di definizioni di AWS Glue tabella utilizzate dalla trasformazione.

- `Parameters`: un oggetto TransformParameters.

  Oggetto `TransformParameters`. È possibile utilizzare i parametri per ottimizzare (personalizzare) il comportamento della trasformazione basata su machine learning specificando i dati da utilizzare per l'addestramento e le preferenze sui vari compromessi (ad esempio precisione vs. recupero o accuratezza vs. costo).

- `EvaluationMetrics`: un oggetto EvaluationMetrics.

  Oggetto `EvaluationMetrics`. I parametri di valutazione forniscono una stima della qualità della trasformazione basata su machine learning.

- `LabelCount`: numero (intero).

  Un identificatore di conteggio per i file di etichettatura generati da AWS Glue per questa trasformazione. Man mano che si crea una trasformazione migliore, è possibile scaricare, etichettare e caricare il file di etichettatura in modo iterativo.

- `Schema`: una matrice di oggetti SchemaColumn, non superiore a 100 strutture.

  Una mappa di coppie chiave-valore che rappresenta le colonne e i tipi di dati sui quali può essere eseguita questa trasformazione. È imposto un limite massimo di 100 colonne.

- `Role`: stringa UTF-8.

  Il nome o il nome della risorsa Amazon (ARN) del ruolo IAM con le autorizzazioni richieste. Le autorizzazioni richieste includono sia le autorizzazioni AWS Glue del ruolo di servizio per AWS Glue le risorse sia le autorizzazioni Amazon S3 richieste dalla trasformazione.

  - Questo ruolo richiede le autorizzazioni AWS Glue del ruolo di servizio per consentire l'accesso alle risorse in. AWS Glue Consulta Collegamento di una policy agli utenti IAM che accedono a AWS Glue.

  - Questo ruolo ha bisogno dell'autorizzazione per accedere a origini, destinazioni, cartella temporanea, script e librerie di Amazon Simple Storage Service (Amazon S3) utilizzate dall'esecuzione di questa attività di trasformazione.
- `GlueVersion`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Custom string pattern #48.

  Questo valore determina con quale versione di AWS Glue questa trasformazione di machine learning è compatibile. Glue 1.0 è consigliata per la maggior parte dei clienti. Se il valore non è impostato, la compatibilità di Glue è impostata per default su Glue 0.9. Per ulteriori informazioni, consulta Versioni di AWS Glue nella guida per gli sviluppatori.
- `MaxCapacity`: numero (doppio).

  Il numero di unità di elaborazione AWS Glue dati (DPUs) assegnate alle esecuzioni delle attività per questa trasformazione. È possibile allocare da 2 a 100 DPUs; l'impostazione predefinita è 10. Una DPU è una misura relativa della potenza di elaborazione costituita da 4 V di capacità CPUs di elaborazione e 16 GB di memoria. Per ulteriori informazioni, consultare la pagina dei prezzi di AWS Glue.

  `MaxCapacity` è un'opzione mutuamente esclusiva con `NumberOfWorkers` e `WorkerType`.
  - Se `NumberOfWorkers` o `WorkerType` è impostata, `MaxCapacity` può essere impostata.
  - Se `MaxCapacity` è impostata, né `NumberOfWorkers` né `WorkerType` possono essere impostate.
  - Se `WorkerType` è impostata, `NumberOfWorkers` è obbligatoria (e viceversa).
  - `MaxCapacity` e `NumberOfWorkers` devono essere entrambe almeno 1.

  Quando il campo `WorkerType` è impostato su un valore diverso da `Standard`, il campo `MaxCapacity` è impostato automaticamente e diventa di sola lettura.
- `WorkerType`: stringa UTF-8 (valori validi: `Standard=""` | `G.1X=""` | `G.2X=""` | `G.025X=""` | `G.4X=""` | `G.8X=""` | `Z.2X=""`).

  Il tipo di worker predefinito allocato al momento dell'esecuzione di un'attività di questa trasformazione. Accetta un valore Standard, G.1X o G.2X.
  - Per il tipo di worker `Standard`, ciascun worker fornisce 4 vCPU, 16 GB di memoria, disco da 50 GB e 2 esecutori.
  - Per il tipo di worker `G.1X`, ciascun worker fornisce 4 vCPU, 16 GB di memoria, disco da 64 GB e 1 esecutore.

  - Per il tipo di worker `G.2X`, ciascun worker fornisce 8 vCPU, 32 GB di memoria, disco da 128 GB e 1 esecutore.

  `MaxCapacity` è un'opzione mutuamente esclusiva con `NumberOfWorkers` e `WorkerType`.

  - Se `NumberOfWorkers` o `WorkerType` è impostata, `MaxCapacity` può essere impostata.
  - Se `MaxCapacity` è impostata, né `NumberOfWorkers` né `WorkerType` possono essere impostate.
  - Se `WorkerType` è impostata, `NumberOfWorkers` è obbligatoria (e viceversa).
  - `MaxCapacity` e `NumberOfWorkers` devono essere entrambe almeno 1.
- `NumberOfWorkers`: numero (intero).

  Il numero di worker di uno specifico `workerType` allocati al momento dell'esecuzione di un'attività della trasformazione.

  Se `WorkerType` è impostata, `NumberOfWorkers` è obbligatoria (e viceversa).
- `Timeout`: numero (intero), almeno 1.

  Il timeout in minuti della trasformazione basata su machine learning.
- `MaxRetries`: numero (intero).

  Il numero massimo di tentativi dopo la conclusione non corretta di un `MLTaskRun` della trasformazione basata su machine learning.
- `TransformEncryption`: un oggetto TransformEncryption.

  Le encryption-at-rest impostazioni della trasformazione che si applicano all'accesso ai dati dell'utente. Le trasformazioni di machine learning possono accedere ai dati utente crittografati in Amazon S3 utilizzando il servizio di gestione delle chiavi.

## FindMatchesParameters struttura

I parametri per configurare la trasformazione di rilevamento delle corrispondenze.

Campi

- `PrimaryKeyColumnName`: stringa UTF-8, non inferiore a 1 o superiore a 1024 byte di lunghezza, corrispondente a Single-line string pattern.

Il nome di una colonna che identifica in modo univoco le righe nella tabella di origine. Utilizzata per identificare i registri corrispondenti.

- `PrecisionRecallTradeoff`: numero (doppio), non superiore a 1,0.

  Il valore selezionato durante l'ottimizzazione della trasformazione per indicare la distribuzione tra precisione e recupero. Il valore 0,5 significa nessuna preferenza; il valore 1 significa una tendenza esclusiva verso la precisione e un valore di 0 significa una tendenza al recupero. Poiché si tratta di un compromesso, la scelta di valori prossimi a 1 significa recupero molto basso mentre la scelta di valori prossimi a 0 comporta una precisione molto bassa.

  Il parametro precisione indica la frequenza con cui il modello risulta corretto quando prevede una corrispondenza.

  Il parametro recupero indica quanto spesso il modello riesce a prevedere la corrispondenza quando questa è in effetti presente.

- `AccuracyCostTradeoff` – Numero (doppio), non superiore a 1,0.

  Il valore selezionato durante l'ottimizzazione della trasformazione per indicare la distribuzione tra accuratezza e costo. Il valore 0,5 significa che il sistema bilancia accuratezza e costi. Il valore 1 significa una tendenza esclusiva verso l'accuratezza, che spesso implica un costo superiore, talvolta notevolmente superiore. Il valore 0 significa una tendenza esclusiva verso il costo, che può portare a una trasformazione `FindMatches` meno accurata, talvolta con un livello di accuratezza inaccettabilmente basso.

  L'accuratezza misura la capacità della trasformazione di individuare veri positivi e veri negativi. L'incremento dell'accuratezza implica maggiori risorse di elaborazione e costi superiori. Tuttavia permette di raggiungere anche un livello maggiore di recupero.

  Il costo misura la quantità di risorse di elaborazione, e quindi di denaro, che viene utilizzata per eseguire la trasformazione.

- `EnforceProvidedLabels`: booleano.

  Il valore che server per attivare o disattivare la forzatura dell'output affinché corrisponda alle etichette fornite dagli utenti. Se il valore è `True`, la trasformazione `find matches` forza l'output affinché corrisponda alle etichette fornite. I risultati sostituiscono i normali risultati della combinazione. Se il valore è `False`, la trasformazione `find matches` non garantisce che tutte le etichette fornite siano rispettate e i risultati si basano sul modello addestrato.

Si noti che l'impostazione di questo valore su true può incrementare il tempo di esecuzione della combinazione.

## FindMatchesMetrics struttura

I parametri di valutazione per l'algoritmo di rilevamento delle corrispondenze. La qualità della trasformazione basata su machine learning è misurato chiedendo alla trasformazione di prevedere alcune corrispondenze e confrontando i risultati con alcune corrispondenze note dello stesso set di dati. I parametri di qualità sono basati su un sottoinsieme dei dati, perciò non sono assolutamente precisi.

Campi

- `AreaUnderPRCurve`: numero (doppio), non superiore a 1,0.

  L'area sotto la precision/recall curva (AUPRC) è un unico numero che misura la qualità complessiva della trasformazione, indipendentemente dalla scelta fatta per la precisione rispetto al richiamo. Valori più elevati indicano che si dispone di un compromesso tra precisione e recupero più interessante.

  Per ulteriori informazioni, consulta la voce Precisione e recupero su Wikipedia.

- `Precision`: numero (doppio), non superiore a 1,0.

  Il parametro precisione indica la frequenza con cui la trasformazione risulta corretta quando prevede una corrispondenza. Nello specifico, misura la capacità della trasformazione di individuare i veri positivi rispetto al totale dei veri positivi possibili.

  Per ulteriori informazioni, consulta la voce Precisione e recupero su Wikipedia.

- `Recall`: numero (doppio), non superiore a 1,0.

  Il parametro recupero indica quanto spesso la trasformazione riesce a prevedere la corrispondenza quando questa è in effetti presente. Nello specifico, misura la capacità della trasformazione di individuare i veri positivi rispetto al totale dei registri che compongono i dati di origine.

  Per ulteriori informazioni, consulta la voce Precisione e recupero su Wikipedia.

- `F1`: numero (doppio), non superiore a 1,0.

Il parametro F1 massimo indica l'accuratezza della trasformazione con un valore tra 0 e 1, dove 1 è la migliore precisione.

Per ulteriori informazioni, consulta la voce F1 score su Wikipedia.

- `ConfusionMatrix`: un oggetto ConfusionMatrix.

  La matrice di confusione mostra gli elementi che la trasformazione sta predicendo in modo accurato e quali tipi di errori sta commettendo.

  Per ulteriori informazioni, consulta la voce MAtrice di confusione su Wikipedia.

- `ColumnImportances`: una serie di oggetti ColumnImportance, non superiore a 100 strutture.

  Un elenco di strutture `ColumnImportance` contenenti parametri sull'importanza delle colonne, ordinate in ordine di importanza decrescente.

## ConfusionMatrix struttura

La matrice di confusione mostra gli elementi che la trasformazione sta predicendo in modo accurato e quali tipi di errori sta commettendo.

Per ulteriori informazioni, consulta la voce MAtrice di confusione su Wikipedia.

Campi

- `NumTruePositives`: numero (lungo).

  Il numero di corrispondenze nei dati correttamente rilevate dalla trasformazione, nella matrice di confusione della trasformazione.

- `NumFalsePositives`: numero (lungo).

  Il numero di mancate corrispondenze nei dati che la trasformazione ha erroneamente classificato come corrispondenza, nella matrice di confusione della trasformazione.

- `NumTrueNegatives`: numero (lungo).

  Il numero di mancate corrispondenze nei dati che la trasformazione ha correttamente rifiutato, nella matrice di confusione della trasformazione.

- `NumFalseNegatives`: numero (lungo).

Il numero di corrispondenze nei dati che la trasformazione non ha rilevato, nella matrice di confusione della trasformazione.

## GlueTable struttura

Il database e la AWS Glue Data Catalog tabella utilizzati per i dati di input o output.

Campi

- `DatabaseName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Un nome del database in AWS Glue Data Catalog.
- `TableName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Un nome della tabella in AWS Glue Data Catalog.
- `CatalogId`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Un identificatore univoco per AWS Glue Data Catalog.
- `ConnectionName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome della connessione a AWS Glue Data Catalog.
- `AdditionalOptions`: una matrice di mappe di coppie chiave-valore, non meno di 1 o più di 10 coppie.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Ogni valore è una stringa Description, non superiore a 2.048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  Opzioni aggiuntive per la tabella. Al momento sono supportate due chiavi:

  - `pushDownPredicate`: filtra le partizioni senza dover elencare e leggere tutti i file nel set di dati.

- `catalogPartitionPredicate`: per utilizzare l'eliminazione delle partizioni lato server utilizzando gli indici delle partizioni in AWS Glue Data Catalog.

## TaskRun struttura

I parametri di campionamento associati alla trasformazione basata su machine learning.

Campi

- `TransformId`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'identificatore univoco della trasformazione.
- `TaskRunId`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'identificatore univoco di questa esecuzione dell'attività.
- `Status`: stringa UTF-8 (valori validi: `RUNNING` | `FINISHED` | `FAILED` | `PENDING_EXECUTION` | `TIMED_OUT` | `CANCELING` | `CANCELED` | `RECEIVED_BY_TASKRUNNER`).

  Lo stato corrente dell'esecuzione dell'attività invocata.
- `LogGroupName`: stringa UTF-8.

  I nomi dei gruppi di log per la conservazione sicura dei log, associati a questa esecuzione dell'attività.
- `Properties`: un oggetto TaskRunProperties.

  Specifica le proprietà di configurazione associate a questa esecuzione dell'attività.
- `ErrorString`: stringa UTF-8.

  L'elenco delle stringhe di errore associate a questa esecuzione dell'attività.
- `StartedOn`: timestamp.

  La data e l'ora in cui è stata avviata questa esecuzione dell'attività.
- `LastModifiedOn`: timestamp.

  L'ultimo istante temporale in cui è stata modificata l'esecuzione dell'attività invocata.
- `CompletedOn`: timestamp.

L'ultimo istante temporale in cui è stata conclusa l'esecuzione dell'attività invocata.

- `ExecutionTime`: numero (intero).

  Quantità di tempo (in secondi) durante la quale l'esecuzione dell'attività ha utilizzato le risorse.

## TransformFilterCriteria struttura

I criteri utilizzati per filtrare la trasformazione basata su machine learning.

Campi

- `Name`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Un nome di trasformazione univoco utilizzato per filtrare la trasformazione basata su machine learning.

- `TransformType`: stringa UTF-8 (valori validi: `FIND_MATCHES`).

  Il tipo di trasformazione basata su machine learning utilizzata per filtrare le trasformazioni basate su machine learning.

- `Status`: stringa UTF-8 (valori validi: `NOT_READY` | `READY` | `DELETING`).

  Filtra l'elenco delle trasformazioni basate su machine learning in base all'ultimo stato della trasformazione (per valutare se una trasformazione può essere utilizzata o meno). Uno dei valori "NOT_READY", "READY" o "DELETING".

- `GlueVersion`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Custom string pattern #48.

  Questo valore determina con quale versione di AWS Glue questa trasformazione di apprendimento automatico è compatibile. Glue 1.0 è consigliata per la maggior parte dei clienti. Se il valore non è impostato, la compatibilità di Glue è impostata per default su Glue 0.9. Per ulteriori informazioni, consulta Versioni di AWS Glue nella guida per gli sviluppatori.

- `CreatedBefore`: timestamp.

  La data e l'ora prima della quale le trasformazioni sono state create.

- `CreatedAfter`: timestamp.

La data e l'ora dopo la quale le trasformazioni sono state create.

- `LastModifiedBefore`: timestamp.

  Filtra le trasformazioni la cui ultima modifica è avvenuta prima di questa data.

- `LastModifiedAfter`: timestamp.

  Filtra le trasformazioni la cui ultima modifica è avvenuta dopo questa data.

- `Schema`: una serie di oggetti SchemaColumn, non superiore a 100 strutture.

  Filtra i set di dati con uno specifico schema. L'oggetto `Map<Column, Type>` è una matrice di coppie chiave-valore che rappresenta lo schema accettato da questa trasformazione, dove `Column` è il nome di una colonna e `Type` è il tipo di dati, ad esempio un intero o una stringa. È imposto un limite massimo di 100 colonne.

## TransformSortCriteria struttura

I criteri di ordinamento associati alla trasformazione basata su machine learning.

Campi

- `Column` - Obbligatorio: stringa UTF-8 (valori validi: `NAME` | `TRANSFORM_TYPE` | `STATUS` | `CREATED` | `LAST_MODIFIED`).

  La colonna da utilizzare nel criterio di ordinamento associato alla trasformazione basata su machine learning.

- `SortDirection` - obbligatorio: stringa UTF-8 (valori validi: `DESCENDING` | `ASCENDING`).

  Il tipo di ordinamento da utilizzare nel criterio di ordinamento associato alla trasformazione basata su machine learning.

## TaskRunFilterCriteria struttura

I criteri che vengono utilizzati per filtrare le esecuzioni di attività della trasformazione basata su machine learning.

Campi

- `TaskRunType`: stringa UTF-8 (valori validi: `EVALUATION` | `LABELING_SET_GENERATION` | `IMPORT_LABELS` | `EXPORT_LABELS` | `FIND_MATCHES`).

  Il tipo di esecuzione dell'attività.

- `Status`: stringa UTF-8 (valori validi: `RUNNING` | `FINISHED` | `FAILED` | `PENDING_EXECUTION` | `TIMED_OUT` | `CANCELING` | `CANCELED` | `RECEIVED_BY_TASKRUNNER`).

  Lo stato attuale dell'esecuzione dell'attività.

- `StartedBefore`: timestamp.

  Filtra le esecuzioni delle attività avviate prima di questa data.

- `StartedAfter`: timestamp.

  Filtra le esecuzioni delle attività avviate dopo questa data.

## TaskRunSortCriteria struttura

I criteri di ordinamento utilizzati per ordinare l'elenco delle esecuzioni delle attività della trasformazione basata su machine learning.

Campi

- `Column` - obbligatorio: stringa UTF-8 (valori validi: `TASK_RUN_TYPE` | `STATUS` | `STARTED`).

  La colonna da usare per ordinare l'elenco delle esecuzioni delle attività della trasformazione basata su machine learning.

- `SortDirection` - obbligatorio: stringa UTF-8 (valori validi: `DESCENDING` | `ASCENDING`).

  Il tipo di ordinamento da usare per ordinare l'elenco delle esecuzioni delle attività della trasformazione basata su machine learning.

## TaskRunProperties struttura

Le proprietà di configurazione dell'esecuzione dell'attività.

Campi

- `TaskType`: stringa UTF-8 (valori validi: `EVALUATION` | `LABELING_SET_GENERATION` | `IMPORT_LABELS` | `EXPORT_LABELS` | `FIND_MATCHES`).

  Il tipo di esecuzione dell'attività.

- `ImportLabelsTaskRunProperties`: un oggetto ImportLabelsTaskRunProperties.

  Le proprietà di configurazione per l'esecuzione di un'attività di importazione di etichette.

- `ExportLabelsTaskRunProperties`: un oggetto ExportLabelsTaskRunProperties.

  Le proprietà di configurazione per l'esecuzione di un'attività di esportazione di etichette.

- `LabelingSetGenerationTaskRunProperties`: un oggetto LabelingSetGenerationTaskRunProperties.

  Le proprietà di configurazione per l'esecuzione di un'attività di generazione di un set di etichettatura.

- `FindMatchesTaskRunProperties`: un oggetto FindMatchesTaskRunProperties.

  Le proprietà di configurazione per l'esecuzione di un'attività di rilevamento delle corrispondenze.

## FindMatchesTaskRunProperties struttura

Specifica le proprietà di configurazione per l'esecuzione di un'attività di rilevamento delle corrispondenze.

Campi

- `JobId`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID del processo dell'esecuzione di un'attività di rilevamento delle corrispondenze.

- `JobName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome assegnato al processo dell'esecuzione di un'attività di rilevamento delle corrispondenze.

- `JobRunId`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

L'ID di esecuzione del processo dell'esecuzione di un'attività di rilevamento delle corrispondenze.

## ImportLabelsTaskRunProperties struttura

Specifica le proprietà di configurazione per l'esecuzione di un'attività di importazione delle etichette.

Campi

- `InputS3Path`: stringa UTF-8.

  Il percorso Amazon Simple Storage Service (Amazon S3) da dove saranno importate le etichette.
- `Replace`: booleano.

  Indica se sovrascrivere le etichette esistenti.

## ExportLabelsTaskRunProperties struttura

Specifica le proprietà di configurazione per l'esecuzione di un'attività di esportazione delle etichette.

Campi

- `OutputS3Path`: stringa UTF-8.

  Il percorso Amazon Simple Storage Service (Amazon S3) dove saranno esportate le etichette.

## LabelingSetGenerationTaskRunProperties struttura

Specifica le proprietà di configurazione per l'esecuzione di un'attività di generazione di un set di etichettatura.

Campi

- `OutputS3Path`: stringa UTF-8.

  Il percorso Amazon Simple Storage Service (Amazon S3) dove sarà generato il set di etichettatura.

## SchemaColumn struttura

Una coppia chiave-valore che rappresenta una colonna e un tipo di dati sui quali può essere eseguita questa trasformazione. Il parametro `Schema` di `MLTransform` può contenere fino a 100 di queste strutture.

Campi

- `Name`: stringa UTF-8, non inferiore a 1 o superiore a 1024 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome della colonna.

- `DataType`: stringa UTF-8, non superiore a 131072 byte di lunghezza, corrispondente a Single-line string pattern.

  Il tipo di dati della colonna.

## TransformEncryption struttura

Le encryption-at-rest impostazioni della trasformazione che si applicano all'accesso ai dati dell'utente. Le trasformazioni di machine learning possono accedere ai dati utente crittografati in Amazon S3 utilizzando il servizio di gestione delle chiavi.

Inoltre, le etichette importate e le trasformazioni addestrate possono ora essere crittografate utilizzando una chiave del servizio di gestione delle chiavi fornita dal cliente.

Campi

- `MlUserDataEncryption`: un oggetto MLUserDataEncryption.

  Un oggetto `MLUserDataEncryption` contenente la modalità di crittografia e l'ID chiave KMS fornito dal cliente.

- `TaskRunSecurityConfigurationName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome della configurazione di sicurezza.

## MLUserDataEncryption struttura

Le encryption-at-rest impostazioni della trasformazione che si applicano all'accesso ai dati dell'utente.

Campi

- `MlUserDataEncryptionMode` - obbligatorio: stringa UTF-8 (valori validi: `DISABLED` | `SSE-KMS="SSEKMS"`).

  La modalità di crittografia applicata ai dati utente. I valori validi sono:

  - DISABLED: la crittografia è disattivata
  - SSEKMS: utilizzo della crittografia lato server con AWS Key Management Service (SSE-KMS) per i dati utente archiviati in Amazon S3.
- `KmsKeyId`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID della chiave KMS fornita dal cliente.

## ColumnImportance struttura

Una struttura contenente il nome della colonna e il punteggio di importanza della colonna per una colonna.

L'importanza delle colonne consente di comprendere il modo in cui queste contribuiscono al modello, identificando quali colonne nei registri sono più importanti di altre.

Campi

- `ColumnName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome di una colonna.
- `Importance`: numero (doppio), non superiore a 1,0.

  Punteggio di importanza per la colonna, come numero decimale.

## Operazioni

- Crea MLTransform azione (Python: create_ml_transform)

- MLTransform Azione di aggiornamento (Python: update_ml_transform)
- MLTransform Azione di cancellazione (Python: delete_ml_transform)
- Ottieni MLTransform un'azione (Python: get_ml_transform)
- Ottieni MLTransforms un'azione (Python: get_ml_transforms)
- MLTransforms Azione elenco (Python: list_ml_transforms)
- Inizia MLEvaluation TaskRun l'azione (Python: start_ml_evaluation_task_run)
- Inizia MLLabeling SetGenerationTaskRun l'azione (Python: start_ml_labeling_set_generation_task_run)
- Azione Get MLTask Run (Python: get_ml_task_run)
- Azione Get MLTask Runs (Python: get_ml_task_runs)
- Annulla azione MLTask Esegui (Python: cancel_ml_task_run)
- StartExportLabelsTaskRun azione (Python: start_export_labels_task_run)
- StartImportLabelsTaskRun azione (Python: start_import_labels_task_run)

## Crea MLTransform azione (Python: create_ml_transform)

Crea una trasformazione di apprendimento automatico. AWS Glue Questa operazione crea la trasformazione e tutti i parametri necessari per l'addestramento.

Richiamare questa operazione come primo passo del processo di utilizzo di una trasformazione basata su machine learning (come ad esempio la trasformazione `FindMatches`) per la deduplicazione dei dati. È possibile fornire una `Description` facoltativa, nonché i parametri che si desiderano utilizzare per l'algoritmo.

È inoltre necessario specificare determinati parametri per le attività eseguite per conto dell'utente nell'ambito dell'apprendimento dai dati e della creazione di una trasformazione di apprendimento automatico di alta qualità. AWS Glue Questi parametri includono `Role` e, facoltativamente, `AllocatedCapacity`, `Timeout` e `MaxRetries`. Per ulteriori informazioni, consulta la pagina sui processi.

Richiesta

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

Il nome univoco assegnato alla trasformazione al momento della creazione.

- `Description`: stringa di descrizione, non superiore a 2048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  Una descrizione della trasformazione basata su machine learning che viene definita. L'impostazione predefinita è una stringa vuota.

- `InputRecordTables` - obbligatorio: una matrice di oggetti GlueTable, non superiore a 10 strutture.

  Un elenco di definizioni di AWS Glue tabella utilizzate dalla trasformazione.

- `Parameters` - obbligatorio: un oggetto TransformParameters.

  I parametri algoritmici specifici per il tipo di trasformazione usata. Condizionalmente dipendenti dal tipo di trasformazione.

- `Role` - obbligatorio: stringa UTF-8.

  Il nome o il nome della risorsa Amazon (ARN) del ruolo IAM con le autorizzazioni richieste. Le autorizzazioni richieste includono sia le autorizzazioni AWS Glue del ruolo di servizio per AWS Glue le risorse sia le autorizzazioni Amazon S3 richieste dalla trasformazione.

  - Questo ruolo richiede le autorizzazioni AWS Glue del ruolo di servizio per consentire l'accesso alle risorse in. AWS Glue Consulta Collegamento di una policy agli utenti IAM che accedono a AWS Glue.
  - Questo ruolo ha bisogno dell'autorizzazione per accedere a origini, destinazioni, cartella temporanea, script e librerie di Amazon Simple Storage Service (Amazon S3) utilizzate dall'esecuzione di questa attività di trasformazione.

- `GlueVersion`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Custom string pattern #48.

  Questo valore determina con quale versione di AWS Glue questa trasformazione di machine learning è compatibile. Glue 1.0 è consigliata per la maggior parte dei clienti. Se il valore non è impostato, la compatibilità di Glue è impostata per default su Glue 0.9. Per ulteriori informazioni, consulta Versioni di AWS Glue nella guida per gli sviluppatori.

- `MaxCapacity`: numero (doppio).

  Il numero di unità di elaborazione AWS Glue dati (DPUs) assegnate alle esecuzioni delle attività per questa trasformazione. È possibile allocare da 2 a 100 DPUs; l'impostazione predefinita è 10.

Una DPU è una misura relativa della potenza di elaborazione costituita da 4 V di capacità CPUs di elaborazione e 16 GB di memoria. Per ulteriori informazioni, consultare la pagina dei prezzi di AWS Glue.

`MaxCapacity` è un'opzione mutuamente esclusiva con `NumberOfWorkers` e `WorkerType`.

- Se `NumberOfWorkers` o `WorkerType` è impostata, `MaxCapacity` può essere impostata.
- Se `MaxCapacity` è impostata, né `NumberOfWorkers` né `WorkerType` possono essere impostate.
- Se `WorkerType` è impostata, `NumberOfWorkers` è obbligatoria (e viceversa).
- `MaxCapacity` e `NumberOfWorkers` devono essere entrambe almeno 1.

Quando il campo `WorkerType` è impostato su un valore diverso da `Standard`, il campo `MaxCapacity` è impostato automaticamente e diventa di sola lettura.

Quando il campo `WorkerType` è impostato su un valore diverso da `Standard`, il campo `MaxCapacity` è impostato automaticamente e diventa di sola lettura.

- `WorkerType`: stringa UTF-8 (valori validi: `Standard=""` | `G.1X=""` | `G.2X=""` | `G.025X=""` | `G.4X=""` | `G.8X=""` | `Z.2X=""`).

  Il tipo di worker predefinito allocato quando viene eseguita questa attività. Accetta un valore Standard, G.1X o G.2X.

  - Per il tipo di worker `Standard`, ciascun worker fornisce 4 vCPU, 16 GB di memoria, disco da 50 GB e 2 esecutori.
  - Per il tipo di worker `G.1X`, ciascun worker fornisce 4 vCPU, 16 GB di memoria, disco da 64 GB e 1 esecutore.
  - Per il tipo di worker `G.2X`, ciascun worker fornisce 8 vCPU, 32 GB di memoria, disco da 128 GB e 1 esecutore.

  `MaxCapacity` è un'opzione mutuamente esclusiva con `NumberOfWorkers` e `WorkerType`.

  - Se `NumberOfWorkers` o `WorkerType` è impostata, `MaxCapacity` può essere impostata.
  - Se `MaxCapacity` è impostata, né `NumberOfWorkers` né `WorkerType` possono essere impostate.
  - Se `WorkerType` è impostata, `NumberOfWorkers` è obbligatoria (e viceversa).
  - `MaxCapacity` e `NumberOfWorkers` devono essere entrambe almeno 1.

- `NumberOfWorkers`: numero (intero).

Il numero di worker di un `workerType` specifico allocati quando viene eseguita questa attività.

Se `WorkerType` è impostata, `NumberOfWorkers` è obbligatoria (e viceversa).

- `Timeout`: numero (intero), almeno 1.

  Il timeout dell'esecuzione dell'attività per questa trasformazione in minuti. Questo è il periodo di tempo massimo durante il quale un'attività in esecuzione per questa trasformazione può consumare risorse prima di essere terminata e impostata allo stato `TIMEOUT`. Il valore di default è 2.880 minuti (48 ore).

- `MaxRetries`: numero (intero).

  Il numero massimo di tentativi di un'attività della trasformazione dopo un'esecuzione conclusa con esito negativo.

- `Tags`: una matrice di mappe con coppie chiave-valore, non superiore alle 50 coppie.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Ogni valore è una stringa UTF-8, lunga non più di 256 byte.

  I tag da utilizzare con questa trasformazione basata su machine learning. È possibile utilizzare tag per limitare l'accesso alla trasformazione basata su machine learning. Per ulteriori informazioni sui tag in AWS Glue, consulta AWS Tags in AWS Glue nella guida per sviluppatori.

- `TransformEncryption`: un oggetto TransformEncryption.

  Le encryption-at-rest impostazioni della trasformazione che si applicano all'accesso ai dati dell'utente. Le trasformazioni di machine learning possono accedere ai dati utente crittografati in Amazon S3 utilizzando il servizio di gestione delle chiavi.

Risposta

- `TransformId`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Un identificatore univoco generato per la trasformazione.

Errori

- `AlreadyExistsException`

- `InvalidInputException`
- `OperationTimeoutException`
- `InternalServiceException`
- `AccessDeniedException`
- `ResourceNumberLimitExceededException`
- `IdempotentParameterMismatchException`

## MLTransform Azione di aggiornamento (Python: update_ml_transform)

Aggiorna una trasformazione basata su machine learning esistente. Richiamare questa operazione per ottimizzare i parametri dell'algoritmo al fine di ottenere risultati migliori.

Dopo aver invocato questa operazione, è possibile richiamare l'operazione `StartMLEvaluationTaskRun` per valutare in che modo i nuovi parametri hanno raggiunto gli obiettivi (ad esempio migliorare la qualità della trasformazione basata su machine learning o renderla più conveniente).

Richiesta

- `TransformId` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Un identificatore univoco generato al momento della creazione della trasformazione.
- `Name`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome univoco assegnato alla trasformazione al momento della creazione.
- `Description`: stringa di descrizione, non superiore a 2048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  Una descrizione della trasformazione. L'impostazione predefinita è una stringa vuota.
- `Parameters`: un oggetto TransformParameters.

  I parametri di configurazione specifici per il tipo di trasformazione (algoritmo) utilizzato. Condizionalmente dipendenti dal tipo di trasformazione.
- `Role`: stringa UTF-8.

Il nome o il nome della risorsa Amazon (ARN) del ruolo IAM con le autorizzazioni richieste.

- `GlueVersion`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Custom string pattern #48.

  Questo valore determina con quale versione di AWS Glue questa trasformazione di machine learning è compatibile. Glue 1.0 è consigliata per la maggior parte dei clienti. Se il valore non è impostato, la compatibilità di Glue è impostata per default su Glue 0.9. Per ulteriori informazioni, consulta Versioni di AWS Glue nella guida per gli sviluppatori.

- `MaxCapacity`: numero (doppio).

  Il numero di unità di elaborazione AWS Glue dati (DPUs) assegnate alle esecuzioni delle attività per questa trasformazione. È possibile allocare da 2 a 100 DPUs; l'impostazione predefinita è 10. Una DPU è una misura relativa della potenza di elaborazione costituita da 4 V di capacità CPUs di elaborazione e 16 GB di memoria. Per ulteriori informazioni, consultare la pagina dei prezzi di AWS Glue.

  Quando il campo `WorkerType` è impostato su un valore diverso da `Standard`, il campo `MaxCapacity` è impostato automaticamente e diventa di sola lettura.

- `WorkerType`: stringa UTF-8 (valori validi: `Standard=""` | `G.1X=""` | `G.2X=""` | `G.025X=""` | `G.4X=""` | `G.8X=""` | `Z.2X=""`).

  Il tipo di worker predefinito allocato quando viene eseguita questa attività. Accetta un valore Standard, G.1X o G.2X.

  - Per il tipo di worker `Standard`, ciascun worker fornisce 4 vCPU, 16 GB di memoria, disco da 50 GB e 2 esecutori.
  - Per il tipo di worker `G.1X`, ciascun worker fornisce 4 vCPU, 16 GB di memoria, disco da 64 GB e 1 esecutore.
  - Per il tipo di worker `G.2X`, ciascun worker fornisce 8 vCPU, 32 GB di memoria, disco da 128 GB e 1 esecutore.

- `NumberOfWorkers`: numero (intero).

  Il numero di worker di un `workerType` specifico allocati quando viene eseguita questa attività.

- `Timeout`: numero (intero), almeno 1.

  Il timeout dell'esecuzione dell'attività per questa trasformazione in minuti. Questo è il periodo di tempo massimo durante il quale un'attività in esecuzione per questa trasformazione può

consumare risorse prima di essere terminata e impostata allo stato `TIMEOUT`. Il valore di default è 2.880 minuti (48 ore).

- `MaxRetries`: numero (intero).

  Il numero massimo di tentativi di un'attività della trasformazione dopo un'esecuzione conclusa con esito negativo.

Risposta

- `TransformId`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Identificatore univoco della trasformazione che è stata aggiornata.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `OperationTimeoutException`
- `InternalServiceException`
- `AccessDeniedException`

## MLTransform Azione di cancellazione (Python: delete_ml_transform)

Elimina una trasformazione di apprendimento automatico. AWS Glue Le trasformazioni basate su machine learning sono un tipo speciale di trasformazione che utilizza il machine learning per interpretare i dettagli della trasformazione da eseguire imparando da esempi forniti da operatori umani. Queste trasformazioni vengono quindi salvate da. AWS Glue Se una trasformazione non è più necessaria, è possibile eliminarla invocando `DeleteMLTransforms`. Tuttavia, tutti i AWS Glue lavori che fanno ancora riferimento alla trasformazione eliminata non avranno più esito positivo.

Richiesta

- `TransformId` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'identificatore univoco della trasformazione da eliminare.

Risposta

- `TransformId`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'identificatore univoco della trasformazione che è stata eliminata.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `OperationTimeoutException`
- `InternalServiceException`

## Ottieni MLTransform un'azione (Python: get_ml_transform)

Ottiene un artefatto di trasformazione basato sull'apprendimento AWS Glue automatico e tutti i metadati corrispondenti. Le trasformazioni basate su machine learning sono un tipo speciale di trasformazione che utilizza il machine learning per interpretare i dettagli della trasformazione da eseguire imparando da esempi forniti da operatori umani. Queste trasformazioni vengono quindi salvate da. AWS GlueÈ possibile recuperare i metadati invocando `GetMLTransform`.

Richiesta

- `TransformId` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'identificatore univoco della trasformazione, generato al momento della creazione della trasformazione.

Risposta

- `TransformId`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'identificatore univoco della trasformazione, generato al momento della creazione della trasformazione.

- `Name`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome univoco assegnato alla trasformazione al momento della creazione.

- `Description`: stringa di descrizione, non superiore a 2048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  Una descrizione della trasformazione.

- `Status`: stringa UTF-8 (valori validi: `NOT_READY` | `READY` | `DELETING`).

  L'ultima stato noto della trasformazione (per indicare se può essere utilizzata o meno). Uno dei valori "NOT_READY", "READY" o "DELETING".

- `CreatedOn`: timestamp.

  La data e l'ora di creazione della trasformazione.

- `LastModifiedOn`: timestamp.

  La data e l'ora in cui la trasformazione è stata modificata l'ultima volta.

- `InputRecordTables`: una serie di oggetti GlueTable, non superiore a 10 strutture.

  Un elenco di definizioni di AWS Glue tabella utilizzate dalla trasformazione.

- `Parameters`: un oggetto TransformParameters.

  I parametri di configurazione specifici per l'algoritmo utilizzato.

- `EvaluationMetrics`: un oggetto EvaluationMetrics.

  I parametri di valutazione più recenti.

- `LabelCount`: numero (intero).

  Il numero di etichette disponibili per questa trasformazione.

- `Schema`: una serie di oggetti SchemaColumn, non superiore a 100 strutture.

  L'oggetto `Map<Column, Type>` che rappresenta lo schema accettato da questa trasformazione. È imposto un limite massimo di 100 colonne.

- `Role`: stringa UTF-8.

  Il nome o il nome della risorsa Amazon (ARN) del ruolo IAM con le autorizzazioni richieste.

- `GlueVersion`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Custom string pattern #48.

  Questo valore determina con quale versione di AWS Glue questa trasformazione di machine learning è compatibile. Glue 1.0 è consigliata per la maggior parte dei clienti. Se il valore non è impostato, la compatibilità di Glue è impostata per default su Glue 0.9. Per ulteriori informazioni, consulta Versioni di AWS Glue nella guida per gli sviluppatori.

- `MaxCapacity`: numero (doppio).

  Il numero di unità di elaborazione AWS Glue dati (DPUs) assegnate alle esecuzioni delle attività per questa trasformazione. È possibile allocare da 2 a 100 DPUs; l'impostazione predefinita è 10. Una DPU è una misura relativa della potenza di elaborazione costituita da 4 V di capacità CPUs di elaborazione e 16 GB di memoria. Per ulteriori informazioni, consultare la pagina dei prezzi di AWS Glue.

  Quando il campo `WorkerType` è impostato su un valore diverso da `Standard`, il campo `MaxCapacity` è impostato automaticamente e diventa di sola lettura.

- `WorkerType`: stringa UTF-8 (valori validi: `Standard=""` | `G.1X=""` | `G.2X=""` | `G.025X=""` | `G.4X=""` | `G.8X=""` | `Z.2X=""`).

  Il tipo di worker predefinito allocato quando viene eseguita questa attività. Accetta un valore Standard, G.1X o G.2X.

  - Per il tipo di worker `Standard`, ciascun worker fornisce 4 vCPU, 16 GB di memoria, disco da 50 GB e 2 esecutori.
  - Per il tipo di worker `G.1X`, ciascun worker fornisce 4 vCPU, 16 GB di memoria, disco da 64 GB e 1 esecutore.
  - Per il tipo di worker `G.2X`, ciascun worker fornisce 8 vCPU, 32 GB di memoria, disco da 128 GB e 1 esecutore.

- `NumberOfWorkers`: numero (intero).

  Il numero di worker di un `workerType` specifico allocati quando viene eseguita questa attività.

- `Timeout`: numero (intero), almeno 1.

  Il timeout dell'esecuzione dell'attività per questa trasformazione in minuti. Questo è il periodo di tempo massimo durante il quale un'attività in esecuzione per questa trasformazione può consumare risorse prima di essere terminata e impostata allo stato `TIMEOUT`. Il valore di default è 2.880 minuti (48 ore).

- `MaxRetries`: numero (intero).

  Il numero massimo di tentativi di un'attività della trasformazione dopo un'esecuzione conclusa con esito negativo.
- `TransformEncryption`: un oggetto TransformEncryption.

  Le encryption-at-rest impostazioni della trasformazione che si applicano all'accesso ai dati dell'utente. Le trasformazioni di machine learning possono accedere ai dati utente crittografati in Amazon S3 utilizzando il servizio di gestione delle chiavi.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `OperationTimeoutException`
- `InternalServiceException`

## Ottieni MLTransforms un'azione (Python: get_ml_transforms)

Ottiene un elenco ordinabile e filtrabile di trasformazioni di machine learning esistenti. AWS Glue Le trasformazioni basate su machine learning sono un tipo speciale di trasformazione che utilizza il machine learning per interpretare i dettagli della trasformazione da eseguire imparando da esempi forniti da operatori umani. Queste trasformazioni vengono quindi salvate da AWS Glue ed è possibile recuperarne i metadati chiamando. `GetMLTransforms`

Richiesta

- `NextToken`: stringa UTF-8.

  Un token di paginazione per partizionare i risultati.
- `MaxResults`: numero (intero), non inferiore a 1 o superiore a 1.000.

  Numero massimo di risultati da restituire.
- `Filter`: un oggetto TransformFilterCriteria.

  Il criterio di filtraggio della trasformazione.
- `Sort`: un oggetto TransformSortCriteria.

Il criterio di ordinamento.

Risposta

- `Transforms` - obbligatorio: una matrice di oggetti MLTransform.

  Un elenco di trasformazioni basate su machine learning.

- `NextToken`: stringa UTF-8.

  Un token di impaginazione, se sono disponibili altri risultati.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `OperationTimeoutException`
- `InternalServiceException`

## MLTransforms Azione elenco (Python: list_ml_transforms)

Recupera un elenco ordinabile e filtrabile delle trasformazioni di AWS Glue machine learning esistenti in questo AWS account o le risorse con il tag specificato. L'operazione accetta il campo facoltativo `Tags` che si può utilizzare come filtro per la risposta in modo che le risorse con tag possano essere recuperate come gruppo. Se si sceglie di utilizzare il filtro dei tag, potranno essere recuperate solo le risorse con tag.

Richiesta

- `NextToken`: stringa UTF-8.

  Token di continuazione, se si tratta di una richiesta di continuazione.

- `MaxResults`: numero (intero), non inferiore a 1 o superiore a 1.000.

  La dimensione massima di un elenco da restituire.

- `Filter`: un oggetto TransformFilterCriteria.

Un elemento `TransformFilterCriteria` utilizzato per filtrare la trasformazione basata su machine learning.

- `Sort`: un oggetto TransformSortCriteria.

  Un elemento `TransformSortCriteria` usato per ordinare le trasformazioni basate su machine learning.

- `Tags`: una matrice di mappe con coppie chiave-valore, non superiore alle 50 coppie.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Ogni valore è una stringa UTF-8, lunga non più di 256 byte.

  Specifica che vengono restituite solo le risorse con tag.

Risposta

- `TransformIds` - obbligatorio: una matrice di stringhe UTF-8.

  Gli identificatori di tutte le trasformazioni basate su machine learning nell'account o le trasformazioni basate su machine learning con i tag specificati.

- `NextToken`: stringa UTF-8.

  Token di continuazione, se l'elenco restituito non contiene l'ultimo parametro disponibile.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `OperationTimeoutException`
- `InternalServiceException`

## Inizia MLEvaluation TaskRun l'azione (Python: start_ml_evaluation_task_run)

Avvia un'attività per stimare la qualità della trasformazione.

Quando fornisci set di etichette come esempi di verità, l'apprendimento AWS Glue automatico utilizza alcuni di questi esempi per trarne insegnamenti. Le altre etichette sono impiegate come test per stimare la qualità.

Restituisce un identificatore univoco dell'esecuzione. È possibile invocare `GetMLTaskRun` per ottenere ulteriori informazioni sulle statistiche di `EvaluationTaskRun`.

Richiesta

- `TransformId` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'identificatore univoco della trasformazione basata su machine learning.

Risposta

- `TaskRunId`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'identificatore univoco associato a questa esecuzione.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `OperationTimeoutException`
- `InternalServiceException`
- `ConcurrentRunsExceededException`
- `MLTransformNotReadyException`

## Inizia MLLabeling SetGenerationTaskRun l'azione (Python: start_ml_labeling_set_generation_task_run)

Avvia il flusso di apprendimento attivo della trasformazione basata su machine learning per migliorare la qualità della trasformazione generando set di etichette e aggiungendo etichette.

Al termine di `StartMLLabelingSetGenerationTaskRun`, AWS Glue avrà generato un "set di etichettatura" o un set di domande a cui l'operatore umano è chiamato a rispondere.

Nel caso della trasformazione `FindMatches`, queste domande seguono la seguente struttura: "Qual è il modo corretto per raggruppare queste righe in gruppi costituiti interamente di registri corrispondenti?"

Dopo il completamento del processo di etichettatura, è possibile caricare le etichette con una chiamata a `StartImportLabelsTaskRun`. Al termine di `StartImportLabelsTaskRun`, tutte le esecuzioni successive della trasformazione basata su machine learning utilizzeranno le etichette nuove e migliorate ed eseguiranno una trasformazione di maggiore qualità.

Nota: il ruolo utilizzato per scrivere il set di etichettatura impostato sul `OutputS3Path` è il ruolo associato alla trasformazione basata su machine learning, specificato nell'API `CreateMLTransform`.

Richiesta

- `TransformId` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'identificatore univoco della trasformazione basata su machine learning.
- `OutputS3Path` - obbligatorio: stringa UTF-8.

  Il percorso Amazon Simple Storage Service (Amazon S3) dove si genera il set di etichettatura.

Risposta

- `TaskRunId`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'identificatore univoco dell'esecuzione associato a questa esecuzione di attività.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `OperationTimeoutException`
- `InternalServiceException`

- `ConcurrentRunsExceededException`

## Azione Get MLTask Run (Python: get_ml_task_run)

Recupera i dettagli di una specifica esecuzione di attività su una trasformazione basata su machine learning. Le attività di machine learning sono attività asincrone eseguite per conto dell'utente nell'ambito di vari flussi di lavoro di machine learning. AWS Glue È possibile verificare le statistiche di ogni esecuzione di attività invocando `GetMLTaskRun` con il `TaskRunID` e il `TransformID` della sua trasformazione padre.

Richiesta

- `TransformId` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'identificatore univoco della trasformazione basata su machine learning.
- `TaskRunId` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'identificatore univoco dell'esecuzione dell'attività.

Risposta

- `TransformId`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'identificatore univoco dell'esecuzione dell'attività.
- `TaskRunId`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'identificatore univoco dell'esecuzione associato a questa esecuzione di attività.
- `Status`: stringa UTF-8 (valori validi: `RUNNING` | `FINISHED` | `FAILED` | `PENDING_EXECUTION` | `TIMED_OUT` | `CANCELING` | `CANCELED` | `RECEIVED_BY_TASKRUNNER`).

  Lo stato di questa esecuzione dell'attività.
- `LogGroupName`: stringa UTF-8.

  I nomi dei gruppi di log associati all'esecuzione dell'attività.

- `Properties`: un oggetto TaskRunProperties.

  L'elenco delle proprietà associate all'esecuzione dell'attività.

- `ErrorString`: stringa UTF-8.

  Le stringhe di errore associate all'esecuzione dell'attività.

- `StartedOn`: timestamp.

  La data e l'ora in cui è stata avviata questa esecuzione dell'attività.

- `LastModifiedOn`: timestamp.

  La data e l'ora in questa esecuzione dell'attività è stata modificata l'ultima volta.

- `CompletedOn`: timestamp.

  La data e l'ora in cui eseguire questa esecuzione dell'attività è stata completata.

- `ExecutionTime`: numero (intero).

  Quantità di tempo (in secondi) durante la quale l'esecuzione dell'attività ha utilizzato le risorse.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `OperationTimeoutException`
- `InternalServiceException`

## Azione Get MLTask Runs (Python: get_ml_task_runs)

Restituisce un elenco di esecuzioni di una trasformazione basata su machine learning. Le attività di machine learning sono attività asincrone eseguite per conto dell'utente nell'ambito di vari flussi di lavoro di machine learning. AWS Glue È possibile ottenere un elenco filtrabile e ordinabile delle esecuzioni delle attività di machine learning invocando `GetMLTaskRuns` con il `TransformID` della trasformazione padre e altri parametri facoltativi come documentato in questa sezione.

Questa operazione restituisce un elenco di storico di esecuzioni e deve essere paginato.

Richiesta

- `TransformId` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'identificatore univoco della trasformazione basata su machine learning.

- `NextToken`: stringa UTF-8.

  Un token per l'impaginazione dei risultati. L'impostazione predefinita è vuota.

- `MaxResults`: numero (intero), non inferiore a 1 o superiore a 1.000.

  Numero massimo di risultati da restituire.

- `Filter`: un oggetto TaskRunFilterCriteria.

  I criteri di filtro, nella struttura `TaskRunFilterCriteria`, per l'esecuzione dell'attività.

- `Sort`: un oggetto TaskRunSortCriteria.

  I criteri di ordinamento, nella struttura `TaskRunSortCriteria`, per l'esecuzione dell'attività.

Risposta

- `TaskRuns`: una matrice di oggetti TaskRun.

  Un elenco delle esecuzioni di attività associate alla trasformazione.

- `NextToken`: stringa UTF-8.

  Un token di impaginazione, se sono disponibili altri risultati.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `OperationTimeoutException`
- `InternalServiceException`

# Annulla azione MLTask Esegui (Python: cancel_ml_task_run)

Annulla (interrompe) un'esecuzione dell'attività. Le attività di machine learning sono attività asincrone eseguite per conto dell'utente nell'ambito di vari flussi di lavoro di AWS Glue machine learning. È possibile annullare un'attività di machine learning in qualsiasi momento invocando `CancelMLTaskRun` con l'`TransformID` della trasformazione padre dell'esecuzione dell'attività e il `TaskRunId` dell'esecuzione dell'attività.

Richiesta

- `TransformId` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'identificatore univoco della trasformazione basata su machine learning.
- `TaskRunId` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Un identificatore univoco dell'esecuzione dell'attività.

Risposta

- `TransformId`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'identificatore univoco della trasformazione basata su machine learning.
- `TaskRunId`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'identificatore univoco dell'esecuzione dell'attività.
- `Status`: stringa UTF-8 (valori validi: `RUNNING` | `FINISHED` | `FAILED` | `PENDING_EXECUTION` | `TIMED_OUT` | `CANCELING` | `CANCELED` | `RECEIVED_BY_TASKRUNNER`).

  Lo stato di questa esecuzione.

Errori

- `EntityNotFoundException`
- `InvalidInputException`

- `OperationTimeoutException`
- `InternalServiceException`

## StartExportLabelsTaskRun azione (Python: start_export_labels_task_run)

Avvia un'operazione asincrona di esportazione di tutti i dati etichettati per una determinata trasformazione. Questa attività è l'unica chiamata API relativa alle etichette che non fa parte del tipico flusso di lavoro di addestramento attivo. Generalmente si utilizza `StartExportLabelsTaskRun` quando si desidera operare contemporaneamente su tutte le etichette, ad esempio quando si desidera rimuovere o modificare delle etichette che sono state indicate in precedenza come verità. Questa operazione API accetta il `TransformId` le cui etichette si desiderano esportare e un percorso su Amazon Simple Storage Service (Amazon S3) su cui esportare le etichette. L'operazione restituisce un `TaskRunId`. È possibile controllare lo stato dell'esecuzione dell'attività invocando l'API `GetMLTaskRun`.

Richiesta

- `TransformId` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'identificatore univoco della trasformazione basata su machine learning.
- `OutputS3Path` - Obbligatorio: stringa UTF-8.

  Il percorso di Amazon S3 su cui esportare le etichette.

Risposta

- `TaskRunId`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'identificatore univoco dell'esecuzione dell'attività.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `OperationTimeoutException`

- `InternalServiceException`

## StartImportLabelsTaskRun azione (Python: start_import_labels_task_run)

Consente di fornire ulteriori etichette (esempi di verità) da utilizzare per addestrare la trasformazione basata su machine learning e migliorarne la qualità. Questa operazione API viene in genere utilizzata come parte del flusso di lavoro di addestramento attivo che inizia con l'invocazione di `StartMLLabelingSetGenerationTaskRun` e che ha come risultato finale il miglioramento della qualità della trasformazione basata su machine learning.

Al completamento di `StartMLLabelingSetGenerationTaskRun`, il machine learning di AWS Glue avrà generato una serie di domande a cui l'operatore umano è chiamato a rispondere. (L'attività di risposta a tali domande è spesso denominata "etichettatura" all'interno dei flussi di lavoro di machine learning). Nel caso della trasformazione `FindMatches`, queste domande seguono la seguente struttura: "Qual è il modo corretto per raggruppare queste righe in gruppi costituiti interamente di registri corrispondenti?" Al termine del processo di etichettatura, gli utenti caricano la propria chiamata a. answers/labels `StartImportLabelsTaskRun` Al termine di `StartImportLabelsTaskRun`, tutte le esecuzioni successive della trasformazione basata su machine learning utilizzeranno le etichette nuove e migliorate ed eseguiranno una trasformazione di maggiore qualità.

Per impostazione predefinita, `StartMLLabelingSetGenerationTaskRun` apprende continuamente dalle etichette caricate e le combina a meno che il parametro `Replace` non sia impostato su "true". Se `Replace` è impostato su "true", `StartImportLabelsTaskRun` elimina e dimentica tutte le etichette caricate in precedenza e apprende solo dal set esatto appena caricato. La sostituzione delle etichette può essere utile se ci si rende conto di aver precedentemente caricato delle etichette errate e si ritiene che ciò possa avere ripercussioni negative sulla qualità della trasformazione.

È possibile controllare lo stato dell'esecuzione dell'attività invocando l'operazione `GetMLTaskRun`.

Richiesta

- `TransformId` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'identificatore univoco della trasformazione basata su machine learning.
- `InputS3Path` - obbligatorio: stringa UTF-8.

Il percorso Amazon Simple Storage Service (Amazon S3) da cui si importano le etichette.

- `ReplaceAllLabels`: booleano.

  Indica se sovrascrivere le etichette esistenti.

Risposta

- `TaskRunId`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'identificatore univoco dell'esecuzione dell'attività.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `OperationTimeoutException`
- `ResourceNumberLimitExceededException`
- `InternalServiceException`

# API di qualità dei dati

L'API di qualità dei dati descrive i tipi di dati relativi alla qualità dei dati e include l'API per la creazione, l'eliminazione o l'aggiornamento dei set di regole, le esecuzioni e le valutazioni della qualità.

## Tipi di dati

- DataSource struttura
- DataQualityRulesetListDetails struttura
- DataQualityTargetTable struttura
- DataQualityRulesetEvaluationRunDescription struttura
- DataQualityRulesetEvaluationRunFilter struttura
- DataQualityEvaluationRunAdditionalRunOptions struttura

- DataQualityRuleRecommendationRunDescription struttura
- DataQualityRuleRecommendationRunFilter struttura
- DataQualityResult struttura
- DataQualityAnalyzerResult struttura
- DataQualityObservation struttura
- MetricBasedObservation struttura
- DataQualityMetricValues struttura
- DataQualityRuleResult struttura
- DataQualityResultDescription struttura
- DataQualityResultFilterCriteria struttura
- DataQualityRulesetFilterCriteria struttura
- DataQualityAggregatedMetrics struttura
- StatisticAnnotation struttura
- TimestampedInclusionAnnotation struttura
- AnnotationError struttura
- DatapointInclusionAnnotation struttura
- StatisticSummaryList lista
- StatisticSummary struttura
- RunIdentifier struttura
- StatisticModelResult struttura
- DataQualityGlueTable struttura

## DataSource struttura

Una fonte di dati (una AWS Glue tabella) per la quale desideri ottenere risultati sulla qualità dei dati.

Campi

- `GlueTable`: un oggetto GlueTable.

  Una AWS Glue tabella.
- `DataQualityGlueTable`: un oggetto DataQualityGlueTable.

Una AWS Glue tabella per le operazioni sulla qualità dei dati.

## DataQualityRulesetListDetails struttura

Descrive un set di regole di qualità dei dati restituito da `GetDataQualityRuleset`.

Campi

- `Name`— UTF-8 stringa, lunga non meno di 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Il nome del set di regole di qualità dei dati.
- `Description`: stringa di descrizione, non superiore a 2048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  Una descrizione del set di regole di qualità dei dati.
- `CreatedOn`: timestamp.

  La data e l'ora di creazione del set di regole della qualità dei dati.
- `LastModifiedOn`: timestamp.

  La data e l'ora di modifica del set di regole della qualità dei dati.
- `TargetTable`: un oggetto DataQualityTargetTable.

  Un oggetto che rappresenta una AWS Glue tabella.
- `RecommendationRunId`— UTF-8 stringa, lunga non meno di 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Quando un set di regole è stato creato da un'esecuzione di raccomandazione, questo ID di esecuzione viene generato per collegare i due.
- `RuleCount`: numero (intero).

  Il numero di regole nel set di regole.

## DataQualityTargetTable struttura

Un oggetto che rappresenta una AWS Glue tabella.

Campi

- `TableName`— Obbligatorio: UTF-8 stringa, lunga almeno 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Il nome della AWS Glue tabella.

- `DatabaseName`— Obbligatorio: UTF-8 stringa, lunga almeno 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Il nome del database in cui esiste la AWS Glue tabella.

- `CatalogId`— UTF-8 stringa, lunga non meno di 1 o più di 255 byte, corrispondente a. Single-line string pattern

  L'ID del catalogo in cui si trova la AWS Glue tabella.

## DataQualityRulesetEvaluationRunDescription struttura

Descrive il risultato di un'esecuzione di valutazione del set di regole della qualità dei dati.

Campi

- `RunId`— UTF-8 stringa, lunga non meno di 1 o più di 255 byte, corrispondente a. Single-line string pattern

  L'identificatore univoco dell'esecuzione associato a questa esecuzione di attività.

- `Status`— UTF-8 stringa (valori validi: `RUNNING` | `FINISHED` | `FAILED` | `PENDING_EXECUTION` | | `TIMED_OUT` | `CANCELING` `CANCELED` |`RECEIVED_BY_TASKRUNNER`).

  Lo stato di questa esecuzione.

- `StartedOn`: timestamp.

  La data e l'ora di inizio dell'esecuzione.

- `DataSource`: un oggetto DataSource.

  L'origine dati (una AWS Glue tabella) associata all'esecuzione.

## DataQualityRulesetEvaluationRunFilter struttura

I criteri di filtro.

Campi

- `DataSource` - obbligatorio: un oggetto DataSource.

  Filtro basato su una fonte di dati (una AWS Glue tabella) associata all'esecuzione.
- `StartedBefore`: timestamp.

  Filtra i risultati in base alle esecuzioni iniziate prima di questo momento.
- `StartedAfter`: timestamp.

  Filtra i risultati in base alle esecuzioni iniziate dopo questo momento.
- `RulesetName`— UTF-8 stringa, lunga non meno di 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Filtra i risultati per esecuzioni associate a un nome di set di regole specifico.

## DataQualityEvaluationRunAdditionalRunOptions struttura

Opzioni di esecuzione aggiuntive che è possibile specificare per l'esecuzione di una valutazione.

Campi

- `CloudWatchMetricsEnabled`: booleano.

  Se abilitare o meno le CloudWatch metriche.
- `ResultsS3Prefix`— UTF-8 stringa.

  Prefisso per Amazon S3 per archiviare i risultati.
- `CompositeRuleEvaluationMethod`— UTF-8 stringa (valori validi: `COLUMN` | `ROW`).

  Imposta il metodo di valutazione per le regole composite nel set di regole su ROW/COLUMN
- `CustomLogGroupPrefix`— stringa UTF-8 .

  Un prefisso personalizzato per i nomi dei gruppi di CloudWatch log. Quando specificato, i registri delle esecuzioni di valutazione vengono scritti nei gruppi di `/aws-glue/data-quality/output`

log <CustomLogGroupPrefix>/error e predefiniti e <CustomLogGroupPrefix>/output anziché in quelli predefiniti/aws-glue/data-quality/error. Il nome totale del gruppo di log, incluso il suffisso (/erroro/output), non deve superare i 512 caratteri.

## DataQualityRuleRecommendationRunDescription struttura

Descrive il risultato dell'esecuzione di una raccomandazione per una regola di qualità dei dati.

Campi

- RunId— UTF-8 stringa, lunga non meno di 1 o più di 255 byte, corrispondente a. Single-line string pattern

  L'identificatore univoco dell'esecuzione associato a questa esecuzione di attività.
- Status— UTF-8 stringa (valori validi: RUNNING | FINISHED | FAILED | PENDING_EXECUTION | | TIMED_OUT | CANCELING CANCELED |RECEIVED_BY_TASKRUNNER).

  Lo stato di questa esecuzione.
- StartedOn: timestamp.

  La data e l'ora in cui è stata avviata questa esecuzione.
- DataSource: un oggetto DataSource.

  L'origine dati (AWS Glue tabella) associata all'esecuzione del consiglio.

## DataQualityRuleRecommendationRunFilter struttura

Un filtro per elencare le esecuzioni delle raccomandazioni per la qualità dei dati.

Campi

- DataSource - obbligatorio: un oggetto DataSource.

  Filtro basato su una fonte di dati specificata (AWS Glue tabella).
- StartedBefore: timestamp.

  Filtra in base all'ora per i risultati avviati prima dell'ora indicata.
- StartedAfter: timestamp.

Filtra in base all'ora per i risultati avviati dopo l'ora indicata.

## DataQualityResult struttura

Descrive un risultato di qualità dei dati.

Campi

- `ResultId`— UTF-8 stringa, lunga non meno di 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Un ID di risultato univoco per il risultato della qualità dei dati.
- `ProfileId`— UTF-8 stringa, di lunghezza non inferiore a 1 o più di 255 byte, corrispondente a. Single-line string pattern

  L'ID profilo del risultato di qualità dei dati.
- `Score`: numero (doppio), non superiore a 1.

  Un punteggio aggregato della qualità dei dati. Rappresenta il rapporto tra le regole inviate e il numero totale di regole.
- `DataSource`: un oggetto DataSource.

  La tabella associata al risultato della qualità dei dati, se presente.
- `RulesetName`— UTF-8 stringa, di lunghezza non inferiore a 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Il nome del set di regole associato al risultato della qualità dei dati.
- `EvaluationContext`— UTF-8 stringa.

  Nel contesto di un lavoro in AWS Glue Studio, a ogni nodo dell'area di disegno viene in genere assegnato un nome e i nodi di qualità dei dati avranno dei nomi. Nel caso di più nodi, `evaluationContext` può differenziare i nodi.
- `StartedOn`: timestamp.

  La data e ora di inizio di questa esecuzione della qualità dei dati.
- `CompletedOn`: timestamp.

  La data e ora di completamento dell'esecuzione della qualità dei dati.

- `JobName`— UTF-8 stringa, lunga non meno di 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Il nome del processo associato al risultato della qualità dei dati, se presente.

- `JobRunId`— UTF-8 stringa, di lunghezza non inferiore a 1 o più di 255 byte, corrispondente a. Single-line string pattern

  L'ID di esecuzione del processo associato al risultato della qualità dei dati, se presente.

- `RulesetEvaluationRunId`— UTF-8 stringa, di lunghezza non inferiore a 1 o più di 255 byte, corrispondente a. Single-line string pattern

  L'ID di esecuzione univoco per la valutazione del set di regole per questo risultato di qualità dei dati.

- `RuleResults`: una matrice di oggetti DataQualityRuleResult, non superiore a 2.000 strutture.

  Un elenco di oggetti `DataQualityRuleResult` che rappresentano i risultati per ogni regola.

- `AnalyzerResults`: una matrice di oggetti DataQualityAnalyzerResult, non superiore a 2.000 strutture.

  Un elenco di oggetti `DataQualityAnalyzerResult` che rappresentano i risultati per ogni analizzatore.

- `Observations`: una matrice di oggetti DataQualityObservation, non superiore a 50 strutture.

  Un elenco di oggetti `DataQualityObservation` che rappresentano le osservazioni generate dopo la valutazione di regole e analizzatori.

- `AggregatedMetrics`: un oggetto DataQualityAggregatedMetrics.

  Un riepilogo degli `DataQualityAggregatedMetrics` oggetti che mostra il conteggio totale delle righe e delle regole elaborate, comprese le relative pass/fail statistiche basate sui risultati a livello di riga.

## DataQualityAnalyzerResult struttura

Descrive il risultato della valutazione di uno strumento di analisi della qualità dei dati.

Campi

- `Name`— UTF-8 stringa, lunga non meno di 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Il nome dello strumento di analisi della qualità dei dati.

- `Description`— UTF-8 stringa, lunga non più di 2048 byte, corrispondente a. URI address multi-line string pattern

  Una descrizione dello strumento di analisi della qualità dei dati.

- `EvaluationMessage`— UTF-8 stringa, lunga non più di 2048 byte, corrispondente a. URI address multi-line string pattern

  Un messaggio di valutazione.

- `EvaluatedMetrics`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una UTF-8 stringa, lunga non meno di 1 o più di 255 byte, che corrisponde a. Single-line string pattern

  Ogni valore è un numero (doppio).

  Una mappa dei parametri associati alla valutazione dello strumento di analisi.

## DataQualityObservation struttura

Descrive l'osservazione generata dopo la valutazione delle regole e degli strumenti di analisi.

Campi

- `Description`— UTF-8 stringa, lunga non più di 2048 byte, corrispondente a. URI address multi-line string pattern

  Una descrizione dell'osservazione della qualità dei dati.

- `MetricBasedObservation`: un oggetto MetricBasedObservation.

  Un oggetto di tipo `MetricBasedObservation` che rappresenta l'osservazione basata sui parametri di qualità dei dati valutati.

## MetricBasedObservation struttura

Descrive l'osservazione basata su parametri generata in base a parametri di qualità dei dati valutati.

Campi

- `MetricName`— UTF-8 stringa, lunga non meno di 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Il nome del parametro di qualità dei dati utilizzato per generare l'osservazione.
- `StatisticId`— UTF-8 stringa, di lunghezza non inferiore a 1 o più di 255 byte, corrispondente a. Single-line string pattern

  L'ID statistica.
- `MetricValues`: un oggetto DataQualityMetricValues.

  Un oggetto di tipo `DataQualityMetricValues` che rappresenta l'analisi del valore del parametro di qualità dei dati.
- `NewRules`— Una serie di UTF-8 stringhe.

  Un elenco di nuove regole sulla qualità dei dati generate nell'ambito dell'osservazione basata sul valore dei parametri di qualità dei dati.

## DataQualityMetricValues struttura

Descrive il valore del parametro di qualità dei dati in base all'analisi dei dati storici.

Campi

- `ActualValue`: numero (doppio).

  Il valore effettivo del parametro di qualità dei dati.
- `ExpectedValue`: numero (doppio).

  Il valore previsto del parametro di qualità dei dati in base all'analisi dei dati storici.
- `LowerLimit`: numero (doppio).

  Il limite inferiore del valore del parametro di qualità dei dati in base all'analisi dei dati storici.
- `UpperLimit`: numero (doppio).

Il limite superiore del valore del parametro di qualità dei dati in base all'analisi dei dati storici.

## DataQualityRuleResult struttura

Descrive il risultato della valutazione del set di regole della qualità dei dati.

Campi

- `Name`— UTF-8 stringa, lunga non meno di 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Il nome della regola di qualità dei dati.

- `Description`— UTF-8 stringa, lunga non più di 2048 byte, corrispondente a. URI address multi-line string pattern

  Una descrizione della regola di qualità dei dati.

- `EvaluationMessage`— UTF-8 stringa, lunga non più di 2048 byte, corrispondente a. URI address multi-line string pattern

  Un messaggio di valutazione.

- `Result`— UTF-8 stringa (valori validi: `PASS` | `FAIL` |`ERROR`).

  Lo stato positivo o negativo per la regola.

- `EvaluatedMetrics`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una UTF-8 stringa, lunga non meno di 1 o più di 255 byte, che corrisponde a. Single-line string pattern

  Ogni valore è un numero (doppio).

  Una mappa dei parametri associati alla valutazione della regola.

- `EvaluatedRule`— UTF-8 stringa, lunga non più di 2048 byte, corrispondente a. URI address multi-line string pattern

  La regola oggetto di valutazione.

- `RuleMetrics`: una matrice della mappa di coppie chiave-valore.

Ogni chiave è una UTF-8 stringa, lunga non meno di 1 o più di 255 byte, che corrisponde a. Single-line string pattern

Ogni valore è un numero (doppio).

Una mappa contenente i parametri associati alla valutazione della regola in base ai risultati a livello di riga.

## DataQualityResultDescription struttura

Descrive un risultato di qualità dei dati.

Campi

- `ResultId`— UTF-8 stringa, lunga non meno di 1 o più di 255 byte, corrispondente a. Single-line string pattern

  L'ID del risultato univoco per questo risultato della qualità dei dati.
- `DataSource`: un oggetto DataSource.

  Il nome della tabella associata al risultato della qualità dei dati.
- `JobName`— UTF-8 stringa, di lunghezza non inferiore a 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Il nome del processo associato al risultato della qualità dei dati.
- `JobRunId`— UTF-8 stringa, di lunghezza non inferiore a 1 o più di 255 byte, corrispondente a. Single-line string pattern

  L'ID di esecuzione del processo associato al risultato della qualità dei dati.
- `StartedOn`: timestamp.

  L'ora di inizio dell'esecuzione per questo risultato di qualità dei dati.

## DataQualityResultFilterCriteria struttura

Criteri utilizzati per restituire i risultati della qualità dei dati.

Campi

- `DataSource`: un oggetto DataSource.

  Filtra i risultati in base all'origine dati specificata. Ad esempio, recuperare tutti i risultati per una AWS Glue tabella.

- `JobName`— UTF-8 stringa, lunga non meno di 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Filtra i risultati in base al nome del processo specificato.

- `JobRunId`— UTF-8 stringa, di lunghezza non inferiore a 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Filtra i risultati in base all'ID di esecuzione del processo specificato.

- `StartedAfter`: timestamp.

  Filtra i risultati in base alle esecuzioni iniziate dopo questo momento.

- `StartedBefore`: timestamp.

  Filtra i risultati in base alle esecuzioni iniziate prima di questo momento.

## DataQualityRulesetFilterCriteria struttura

I criteri utilizzati per filtrare i set di regole della qualità dei dati.

Campi

- `Name`— UTF-8 stringa, lunga non meno di 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Il nome del criterio di filtro del set di regole.

- `Description`: stringa di descrizione, non superiore a 2048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  La descrizione dei criteri di filtro del set di regole.

- `CreatedBefore`: timestamp.

  Filtra i set di regole creati prima di questa data.

- `CreatedAfter`: timestamp.

  Filtra i set di regole creati dopo questa data.

- `LastModifiedBefore`: timestamp.

  Filtra i set di regole modificati per l'ultima volta prima di questa data.

- `LastModifiedAfter`: timestamp.

  Filtra i set di regole modificati per l'ultima volta dopo questa data.

- `TargetTable`: un oggetto DataQualityTargetTable.

  Il nome e il nome del database della tabella di destinazione.

## DataQualityAggregatedMetrics struttura

Un riepilogo delle metriche che mostra il conteggio totale delle righe e delle regole elaborate, comprese le relative pass/fail statistiche basate sui risultati a livello di riga.

Campi

- `TotalRowsProcessed`: numero (doppio).

  Il numero totale di righe elaborate durante la valutazione della qualità dei dati.

- `TotalRowsPassed`: numero (doppio).

  Il numero totale di righe che hanno superato tutte le regole di qualità dei dati applicabili.

- `TotalRowsFailed`: numero (doppio).

  Il numero totale di righe che non hanno superato una o più regole di qualità dei dati.

- `TotalRulesProcessed`: numero (doppio).

  Il numero totale di regole di qualità dei dati valutate.

- `TotalRulesPassed`: numero (doppio).

  Il numero totale di regole di qualità dei dati che hanno superato i criteri di valutazione.

- `TotalRulesFailed`: numero (doppio).

  Il numero totale di regole di qualità dei dati che non hanno superato i criteri di valutazione.

## StatisticAnnotation struttura

Un'annotazione statistica.

Campi

- `ProfileId`— UTF-8 stringa, lunga non meno di 1 o più di 255 byte, corrispondente a. Single-line string pattern

  L'ID profilo.

- `StatisticId`— UTF-8 stringa, di lunghezza non inferiore a 1 o più di 255 byte, corrispondente a. Single-line string pattern

  L'ID statistica.

- `StatisticRecordedOn`: timestamp.

  Il timestamp del momento in cui è stata registrata la statistica annotata.

- `InclusionAnnotation`: un oggetto TimestampedInclusionAnnotation.

  L'annotazione di inclusione applicata alla statistica.

## TimestampedInclusionAnnotation struttura

Un'annotazione di inclusione con timestamp.

Campi

- `Value`— UTF-8 stringa (valori validi: `INCLUDE` | `EXCLUDE`).

  Il valore dell'annotazione di inclusione.

- `LastModifiedOn`: timestamp.

  Il timestamp del momento in cui è avvenuta l'ultima modifica dell'annotazione di inclusione.

## AnnotationError struttura

Un'annotazione non riuscita.

Campi

- `ProfileId`— UTF-8 stringa, lunga non meno di 1 o più di 255 byte, corrispondente a. Single-line string pattern

  L'ID profilo dell'annotazione non riuscita.
- `StatisticId`— UTF-8 stringa, di lunghezza non inferiore a 1 o più di 255 byte, corrispondente a. Single-line string pattern

  L'ID statistica dell'annotazione non riuscita.
- `FailureReason`: stringa di descrizione, non superiore a 2048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  Il motivo per cui l'annotazione non è riuscita.

## DatapointInclusionAnnotation struttura

Un'annotazione di inclusione.

Campi

- `ProfileId`— UTF-8 stringa, lunga non meno di 1 o più di 255 byte, corrispondente a. Single-line string pattern

  L'ID del profilo di qualità dei dati a cui appartiene la statistica.
- `StatisticId`— UTF-8 stringa, di lunghezza non inferiore a 1 o più di 255 byte, corrispondente a. Single-line string pattern

  L'ID statistica.
- `InclusionAnnotation`— UTF-8 stringa (valori validi: `INCLUDE` | `EXCLUDE`).

  Il valore dell'annotazione di inclusione da applicare alla statistica.

## StatisticSummaryList lista

elenco di `StatisticSummary`.

Un array di oggetti StatisticSummary.

elenco di `StatisticSummary`.

## StatisticSummary struttura

Informazioni di riepilogo relative a una statistica.

Campi

- `StatisticId`— UTF-8 stringa, lunga non meno di 1 o più di 255 byte, corrispondente a. Single-line string pattern

  L'ID statistica.

- `ProfileId`— UTF-8 stringa, di lunghezza non inferiore a 1 o più di 255 byte, corrispondente a. Single-line string pattern

  L'ID profilo.

- `RunIdentifier`: un oggetto RunIdentifier.

  L'identificatore dell'esecuzione

- `StatisticName`— UTF-8 stringa, di lunghezza non inferiore a 1 o più di 255 byte, corrispondente a. Custom string pattern #16

  Il nome della statistica.

- `DoubleValue`: numero (doppio).

  Il valore della statistica.

- `EvaluationLevel`— UTF-8 stringa (valori validi: `Dataset="DATASET"` | `Column="COLUMN"` |`Multicolumn="MULTICOLUMN"`).

  Il livello di valutazione della statistica. Valori possibili: `Dataset`, `Column`, `Multicolumn`.

- `ColumnsReferenced`— Una matrice di UTF-8 stringhe.

  L'elenco delle colonne a cui fa riferimento la statistica.

- `ReferencedDatasets`— Una serie di UTF-8 stringhe.

  L'elenco dei set di dati a cui fa riferimento la statistica.

- `StatisticProperties`: una matrice della mappa di coppie chiave-valore.

Ogni chiave è una UTF-8 stringa, lunga non meno di 1 o più di 255 byte, che corrisponde a. Single-line string pattern

Ogni valore è una stringa Description, non superiore a 2.048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

Una `StatisticPropertiesMap`, che contiene una `NameString` e una `DescriptionString`

- `RecordedOn`: timestamp.

  Il timestamp del momento in cui è stata registrata la statistica.

- `InclusionAnnotation`: un oggetto TimestampedInclusionAnnotation.

  L'annotazione di inclusione della statistica.

## RunIdentifier struttura

Un identificatore di esecuzione.

Campi

- `RunId`— UTF-8 stringa, lunga non meno di 1 o più di 255 byte, corrispondente a. Single-line string pattern

  L'ID di esecuzione.

- `JobRunId`— UTF-8 stringa, di lunghezza non inferiore a 1 o più di 255 byte, corrispondente a. Single-line string pattern

  L'ID del processo di esecuzione.

## StatisticModelResult struttura

Il risultato del modello statistico.

Campi

- `LowerBound`: numero (doppio).

  Il limite inferiore.

- `UpperBound`: numero (doppio).

  Il limite superiore.
- `PredictedValue`: numero (doppio).

  Il valore previsto.
- `ActualValue`: numero (doppio).

  Il valore effettivo.
- `Date`: timestamp.

  La data.
- `InclusionAnnotation`— UTF-8 stringa (valori validi: `INCLUDE` |`EXCLUDE`).

  L'annotazione di inclusione.

## DataQualityGlueTable struttura

Il database e la tabella utilizzati per AWS Glue Data Catalog i dati di input o output per le operazioni di qualità dei dati.

Campi

- `DatabaseName`— Obbligatorio: UTF-8 stringa di lunghezza non inferiore a 1 o superiore a 255 byte, corrispondente a. Single-line string pattern

  Un nome del database in AWS Glue Data Catalog.
- `TableName`— Obbligatorio: UTF-8 stringa, di lunghezza non inferiore a 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Un nome della tabella in AWS Glue Data Catalog.
- `CatalogId`— UTF-8 stringa, di lunghezza non inferiore a 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Un identificatore univoco per AWS Glue Data Catalog.
- `ConnectionName`— UTF-8 stringa, di lunghezza non inferiore a 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Nome della connessione a AWS Glue Data Catalog.

- `AdditionalOptions`: una matrice di mappe di coppie chiave-valore, non meno di 1 o più di 10 coppie.

  Ogni chiave è una UTF-8 stringa, lunga non meno di 1 o più di 255 byte, che corrisponde a. Single-line string pattern

  Ogni valore è una stringa Description, non superiore a 2.048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  Opzioni aggiuntive per la tabella. Al momento sono supportate due chiavi:

  - `pushDownPredicate`: filtra le partizioni senza dover elencare e leggere tutti i file nel set di dati.
  - `catalogPartitionPredicate`: per utilizzare l'eliminazione delle partizioni lato server utilizzando gli indici delle partizioni in AWS Glue Data Catalog.
- `PreProcessingQuery`— UTF-8 stringa, lunga non più di 51200 byte, corrispondente a. URI address multi-line string pattern

  Query SQL in formato SparkSQL che può essere utilizzata per preelaborare i dati per la tabella in AWS Glue Data Catalog, prima di eseguire la Data Quality Operation.

## Operazioni

- StartDataQualityRulesetEvaluationRun azione (Python: start_data_quality_ruleset_evaluation_run)
- CancelDataQualityRulesetEvaluationRun azione (Python: cancel_data_quality_ruleset_evaluation_run)
- GetDataQualityRulesetEvaluationRun azione (Python: get_data_quality_ruleset_evaluation_run)
- ListDataQualityRulesetEvaluationRuns azione (Python: list_data_quality_ruleset_evaluation_runs)
- StartDataQualityRuleRecommendationRun azione (Python: start_data_quality_rule_recommendation_run)
- CancelDataQualityRuleRecommendationRun azione (Python: cancel_data_quality_rule_recommendation_run)
- GetDataQualityRuleRecommendationRun azione (Python: get_data_quality_rule_recommendation_run)
- ListDataQualityRuleRecommendationRuns azione (Python: list_data_quality_rule_recommendation_runs)
- GetDataQualityResult azione (Python: get_data_quality_result)

- BatchGetDataQualityResult azione (Python: batch_get_data_quality_result)
- ListDataQualityResults azione (Python: list_data_quality_results)
- CreateDataQualityRuleset azione (Python: create_data_quality_ruleset)
- DeleteDataQualityRuleset azione (Python: delete_data_quality_ruleset)
- GetDataQualityRuleset azione (Python: get_data_quality_ruleset)
- ListDataQualityRulesets azione (Python: list_data_quality_rulesets)
- UpdateDataQualityRuleset azione (Python: update_data_quality_ruleset)
- ListDataQualityStatistics azione (Python: list_data_quality_statistics)
- TimestampFilter struttura
- CreateDataQualityRulesetRequest struttura
- GetDataQualityRulesetResponse struttura
- GetDataQualityResultResponse struttura
- StartDataQualityRuleRecommendationRunRequest struttura
- GetDataQualityRuleRecommendationRunResponse struttura
- BatchPutDataQualityStatisticAnnotation azione (Python: batch_put_data_quality_statistic_annotation)
- GetDataQualityModel azione (Python: get_data_quality_model)
- GetDataQualityModelResult azione (Python: get_data_quality_model_result)
- ListDataQualityStatisticAnnotations azione (Python: list_data_quality_statistic_annotations)
- PutDataQualityProfileAnnotation azione (Python: put_data_quality_profile_annotation)

## StartDataQualityRulesetEvaluationRun azione (Python: start_data_quality_ruleset_evaluation_run)

Una volta ottenuta una definizione del set di regole (consigliata o personalizzata), si chiama questa operazione per valutare il set di regole rispetto a una fonte di dati (tabella).AWS Glue La valutazione calcola i risultati che è possibile recuperare con l'API `GetDataQualityResult`.

Richiesta

- `DataSource` - obbligatorio: un oggetto DataSource.

L'origine dati (AWS Glue tabella) associata a questa esecuzione.

- `Role`— Obbligatorio: UTF-8 stringa.

  Un IAM ruolo fornito per crittografare i risultati dell'esecuzione.

- `NumberOfWorkers`: numero (intero).

  Il numero di worker `G.1X` da utilizzare nell'esecuzione. Il predefinito è 5.

- `Timeout`: numero (intero), almeno 1.

  Il timeout per una esecuzione in minuti. Questo è il tempo massimo durante il quale un'esecuzione può utilizzare le risorse prima di essere terminata e passare allo stato `TIMEOUT`. Il valore di default è 2.880 minuti (48 ore).

- `ClientToken`— UTF-8 stringa, lunga non meno di 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Utilizzato per l'idempotenza e consigliato per l'impostazione su un ID casuale (come un UUID) per evitare di creare o avviare più istanze della stessa risorsa.

- `AdditionalRunOptions`: un oggetto DataQualityEvaluationRunAdditionalRunOptions.

  Opzioni di esecuzione aggiuntive che è possibile specificare per l'esecuzione di una valutazione.

- `RulesetNames`— Obbligatorio: una matrice di UTF-8 stringhe, non meno di 1 o più di 10 stringhe.

  Un elenco di nomi di set di regole.

- `AdditionalDataSources`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una UTF-8 stringa, lunga non meno di 1 o più di 255 byte, che corrisponde a. Single-line string pattern

  Ogni valore è un oggetto DataSource.

  Una mappa di stringhe di riferimento a origini dati aggiuntive che è possibile specificare per l'esecuzione di una valutazione.

Risposta

- `RunId`— UTF-8 stringa, di lunghezza non inferiore a 1 o più di 255 byte, corrispondente a. Single-line string pattern

L'identificatore univoco dell'esecuzione associato a questa esecuzione di attività.

Errori

- `InvalidInputException`
- `EntityNotFoundException`
- `OperationTimeoutException`
- `InternalServiceException`
- `ConflictException`

## CancelDataQualityRulesetEvaluationRun azione (Python: cancel_data_quality_ruleset_evaluation_run)

Annulla un'esecuzione in cui un set di regole viene valutato rispetto a un'origine dati.

Richiesta

- `RunId`— Obbligatorio: UTF-8 stringa, lunga non meno di 1 o più di 255 byte, corrispondente a. Single-line string pattern

  L'identificatore univoco dell'esecuzione associato a questa esecuzione di attività.

Risposta

- Nessun parametro di risposta.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `OperationTimeoutException`
- `InternalServiceException`

# GetDataQualityRulesetEvaluationRun azione (Python: get_data_quality_ruleset_evaluation_run)

Richiama un'esecuzione in cui un set di regole viene valutato rispetto a un'origine dati.

Richiesta

- `RunId`— Obbligatorio: UTF-8 stringa, di lunghezza non inferiore a 1 o superiore a 255 byte, corrispondente a. Single-line string pattern

  L'identificatore univoco dell'esecuzione associato a questa esecuzione di attività.

Risposta

- `RunId`— UTF-8 stringa, di lunghezza non inferiore a 1 o più di 255 byte, corrispondente a. Single-line string pattern

  L'identificatore univoco dell'esecuzione associato a questa esecuzione di attività.
- `DataSource`: un oggetto DataSource.

  L'origine dati (una AWS Glue tabella) associata a questa esecuzione di valutazione.
- `Role`— UTF-8 stringa.

  Un IAM ruolo fornito per crittografare i risultati dell'esecuzione.
- `NumberOfWorkers`: numero (intero).

  Il numero di worker `G.1X` da utilizzare nell'esecuzione. Il predefinito è 5.
- `Timeout`: numero (intero), almeno 1.

  Il timeout per una esecuzione in minuti. Questo è il tempo massimo durante il quale un'esecuzione può utilizzare le risorse prima di essere terminata e passare allo stato `TIMEOUT`. Il valore di default è 2.880 minuti (48 ore).
- `AdditionalRunOptions`: un oggetto DataQualityEvaluationRunAdditionalRunOptions.

  Opzioni di esecuzione aggiuntive che è possibile specificare per l'esecuzione di una valutazione.
- `Status`— UTF-8 stringa (valori validi: `RUNNING` | `FINISHED` | `FAILED` | `PENDING_EXECUTION` | | `TIMED_OUT` | `CANCELING` `CANCELED` |`RECEIVED_BY_TASKRUNNER`).

Lo stato di questa esecuzione.

- `ErrorString`— UTF-8 stringa.

  Le stringhe di errore associate all'esecuzione.

- `StartedOn`: timestamp.

  La data e l'ora in cui è stata avviata questa esecuzione.

- `LastModifiedOn`: timestamp.

  Un Timestamp. L'ultimo momento in cui questa raccomandazione della regola di qualità dei dati è stata modificata.

- `CompletedOn`: timestamp.

  La data e l'ora in cui è stata completata questa esecuzione.

- `ExecutionTime`: numero (intero).

  La quantità di tempo (in secondi) durante la quale l'esecuzione ha utilizzato le risorse.

- `RulesetNames`— Una matrice di UTF-8 stringhe, non meno di 1 o più di 10 stringhe.

  Un elenco di nomi dei set di regole per l'esecuzione. Attualmente, questo parametro accetta un solo nome di set di regole.

- `ResultIds`— Una matrice di UTF-8 stringhe, non meno di 1 o più di 10 stringhe.

  Un elenco di ID dei risultati per i risultati della qualità dei dati per l'esecuzione.

- `AdditionalDataSources`: una matrice della mappa di coppie chiave-valore.

  Ogni chiave è una UTF-8 stringa, lunga non meno di 1 o più di 255 byte, che corrisponde a. Single-line string pattern

  Ogni valore è un oggetto DataSource.

  Una mappa di stringhe di riferimento a origini dati aggiuntive che è possibile specificare per l'esecuzione di una valutazione.

Errori

- `EntityNotFoundException`

- `InvalidInputException`
- `OperationTimeoutException`
- `InternalServiceException`

## ListDataQualityRulesetEvaluationRuns azione (Python: list_data_quality_ruleset_evaluation_runs)

Elenca tutte le esecuzioni che soddisfano i criteri di filtro, in cui un set di regole viene valutato rispetto a un'origine dati.

Richiesta

- `Filter`: un oggetto DataQualityRulesetEvaluationRunFilter.

  I criteri di filtro.
- `NextToken` UTF-8 — stringa.

  Un token di paginazione per partizionare i risultati.
- `MaxResults`: numero (intero), non inferiore a 1 o superiore a 1.000.

  Numero massimo di risultati da restituire.

Risposta

- `Runs`: una matrice di oggetti DataQualityRulesetEvaluationRunDescription.

  Un elenco di oggetti `DataQualityRulesetEvaluationRunDescription` che rappresentano le esecuzioni del set di regole della qualità dei dati.
- `NextToken`— UTF-8 corda.

  Un token di impaginazione, se sono disponibili altri risultati.

Errori

- `InvalidInputException`
- `OperationTimeoutException`
- `InternalServiceException`

# StartDataQualityRuleRecommendationRun azione (Python: start_data_quality_rule_recommendation_run)

Avvia un'esecuzione di raccomandazioni che viene utilizzata per generare regole quando non sai quali regole scrivere. AWS Glue Data Quality analizza i dati e fornisce consigli per un potenziale set di regole. È possibile quindi classificare il set di regole e modificare il set di regole generato a proprio piacimento.

Le esecuzioni di consigli vengono eliminate automaticamente dopo 90 giorni.

Richiesta

La richiesta di raccomandazione della regola di qualità dei dati.

- `DataSource` - obbligatorio: un oggetto DataSource.

  La fonte di dati (AWS Glue tabella) associata a questa esecuzione.
- `Role`— Obbligatorio: UTF-8 stringa.

  Un IAM ruolo fornito per crittografare i risultati dell'esecuzione.
- `NumberOfWorkers`: numero (intero).

  Il numero di worker `G.1X` da utilizzare nell'esecuzione. Il predefinito è 5.
- `Timeout`: numero (intero), almeno 1.

  Il timeout per una esecuzione in minuti. Questo è il tempo massimo durante il quale un'esecuzione può utilizzare le risorse prima di essere terminata e passare allo stato `TIMEOUT`. Il valore di default è 2.880 minuti (48 ore).
- `CreatedRulesetName`— UTF-8 stringa, lunga non meno di 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Un nome per il set di regole.
- `DataQualitySecurityConfiguration`— UTF-8 stringa, di lunghezza non inferiore a 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Il nome della configurazione di sicurezza creata con l'opzione di crittografia di qualità dei dati.
- `ClientToken`— UTF-8 stringa, di lunghezza non inferiore a 1 o più di 255 byte, corrispondente a. Single-line string pattern

Utilizzato per l'idempotenza e consigliato per l'impostazione su un ID casuale (come un UUID) per evitare di creare o avviare più istanze della stessa risorsa.

Risposta

- `RunId`— UTF-8 stringa, di lunghezza non inferiore a 1 o più di 255 byte, corrispondente a. Single-line string pattern

  L'identificatore univoco dell'esecuzione associato a questa esecuzione di attività.

Errori

- `InvalidInputException`
- `OperationTimeoutException`
- `InternalServiceException`
- `ConflictException`

## CancelDataQualityRuleRecommendationRun azione (Python: cancel_data_quality_rule_recommendation_run)

Annulla l'esecuzione della raccomandazione specificata utilizzata per generare le regole.

Richiesta

- `RunId`— Obbligatorio: UTF-8 stringa, lunga non meno di 1 o più di 255 byte, corrispondente a. Single-line string pattern

  L'identificatore univoco dell'esecuzione associato a questa esecuzione di attività.

Risposta

- Nessun parametro di risposta.

Errori

- `EntityNotFoundException`

- `InvalidInputException`
- `OperationTimeoutException`
- `InternalServiceException`

## GetDataQualityRuleRecommendationRun azione (Python: get_data_quality_rule_recommendation_run)

Ottiene l'esecuzione della raccomandazione specificata utilizzata per generare le regole.

Richiesta

- `RunId`— Obbligatorio: UTF-8 stringa, lunga non meno di 1 o più di 255 byte, corrispondente a. Single-line string pattern

  L'identificatore univoco dell'esecuzione associato a questa esecuzione di attività.

Risposta

La risposta dell'esecuzione della raccomandazione della regola di qualità dei dati.

- `RunId`— UTF-8 stringa, di lunghezza non inferiore a 1 o più di 255 byte, corrispondente a. Single-line string pattern

  L'identificatore univoco dell'esecuzione associato a questa esecuzione di attività.
- `DataSource`: un oggetto DataSource.

  L'origine dati (una AWS Glue tabella) associata a questa esecuzione.
- `Role`— UTF-8 stringa.

  Un IAM ruolo fornito per crittografare i risultati dell'esecuzione.
- `NumberOfWorkers`: numero (intero).

  Il numero di worker `G.1X` da utilizzare nell'esecuzione. Il predefinito è 5.
- `Timeout`: numero (intero), almeno 1.

  Il timeout per una esecuzione in minuti. Questo è il tempo massimo durante il quale un'esecuzione può utilizzare le risorse prima di essere terminata e passare allo stato `TIMEOUT`. Il valore di default è 2.880 minuti (48 ore).

- `Status`— UTF-8 stringa (valori validi: `RUNNING` | `FINISHED` | `FAILED` | `PENDING_EXECUTION` | | `TIMED_OUT` | `CANCELING` `CANCELED` |`RECEIVED_BY_TASKRUNNER`).

  Lo stato di questa esecuzione.
- `ErrorString`— UTF-8 stringa.

  Le stringhe di errore associate all'esecuzione.
- `StartedOn`: timestamp.

  La data e l'ora in cui è stata avviata questa esecuzione.
- `LastModifiedOn`: timestamp.

  Un Timestamp. L'ultimo momento in cui questa raccomandazione della regola di qualità dei dati è stata modificata.
- `CompletedOn`: timestamp.

  La data e l'ora in cui è stata completata questa esecuzione.
- `ExecutionTime`: numero (intero).

  La quantità di tempo (in secondi) durante la quale l'esecuzione ha utilizzato le risorse.
- `RecommendedRuleset`— UTF-8 stringa, lunga non meno di 1 o più di 65536 byte.

  Una volta completata l'esecuzione di una raccomandazione della regola di avvio, viene creato un set di regole consigliato (una serie di regole). Questo membro ha queste regole nel formato DQDL (Data Quality Definition Language).
- `CreatedRulesetName`— UTF-8 stringa, di lunghezza non inferiore a 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Il nome del set di regole che è stato creato dall'esecuzione.
- `DataQualitySecurityConfiguration`— UTF-8 stringa, di lunghezza non inferiore a 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Il nome della configurazione di sicurezza creata con l'opzione di crittografia di qualità dei dati.

Errori

- `EntityNotFoundException`
- `InvalidInputException`

- `OperationTimeoutException`
- `InternalServiceException`

## ListDataQualityRuleRecommendationRuns azione (Python: list_data_quality_rule_recommendation_runs)

Elenca le esecuzioni delle raccomandazioni che soddisfano i criteri di filtro.

Richiesta

- `Filter`: un oggetto DataQualityRuleRecommendationRunFilter.

  I criteri di filtro.
- `NextToken` UTF-8 — stringa.

  Un token di paginazione per partizionare i risultati.
- `MaxResults`: numero (intero), non inferiore a 1 o superiore a 1.000.

  Numero massimo di risultati da restituire.

Risposta

- `Runs`: una matrice di oggetti DataQualityRuleRecommendationRunDescription.

  Elenco di oggetti `DataQualityRuleRecommendationRunDescription`.
- `NextToken`— UTF-8 corda.

  Un token di impaginazione, se sono disponibili altri risultati.

Errori

- `InvalidInputException`
- `OperationTimeoutException`
- `InternalServiceException`

# GetDataQualityResult azione (Python: get_data_quality_result)

Recupera il risultato di una valutazione della regola della qualità dei dati.

Richiesta

- `ResultId`— Obbligatorio: UTF-8 stringa, lunga non meno di 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Un ID di risultato univoco per il risultato della qualità dei dati.

Risposta

La risposta del risultato di qualità dei dati.

- `ResultId`— UTF-8 stringa, di lunghezza non inferiore a 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Un ID di risultato univoco per il risultato della qualità dei dati.
- `ProfileId`— UTF-8 stringa, di lunghezza non inferiore a 1 o più di 255 byte, corrispondente a. Single-line string pattern

  L'ID profilo del risultato di qualità dei dati.
- `Score`: numero (doppio), non superiore a 1.

  Un punteggio aggregato della qualità dei dati. Rappresenta il rapporto tra le regole inviate e il numero totale di regole.
- `DataSource`: un oggetto DataSource.

  La tabella associata al risultato della qualità dei dati, se presente.
- `RulesetName`— UTF-8 stringa, di lunghezza non inferiore a 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Il nome del set di regole associato al risultato della qualità dei dati.
- `EvaluationContext`— UTF-8 stringa.

  Nel contesto di un lavoro in AWS Glue Studio, a ogni nodo dell'area di disegno viene in genere assegnato un nome e i nodi di qualità dei dati avranno dei nomi. Nel caso di più nodi, `evaluationContext` può differenziare i nodi.

- `StartedOn`: timestamp.

  La data e ora di inizio dell'esecuzione di questo risultato della qualità dei dati.

- `CompletedOn`: timestamp.

  La data e ora di completamento dell'esecuzione di questo risultato della qualità dei dati.

- `JobName`— UTF-8 stringa, lunga non meno di 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Il nome del processo associato al risultato della qualità dei dati, se presente.

- `JobRunId`— UTF-8 stringa, di lunghezza non inferiore a 1 o più di 255 byte, corrispondente a. Single-line string pattern

  L'ID di esecuzione del processo associato al risultato della qualità dei dati, se presente.

- `RulesetEvaluationRunId`— UTF-8 stringa, di lunghezza non inferiore a 1 o più di 255 byte, corrispondente a. Single-line string pattern

  L'ID di esecuzione univoco associato alla valutazione del set di regole.

- `RuleResults`: una matrice di oggetti DataQualityRuleResult, non superiore a 2.000 strutture.

  Un elenco di oggetti `DataQualityRuleResult` che rappresentano i risultati per ogni regola.

- `AnalyzerResults`: una matrice di oggetti DataQualityAnalyzerResult, non superiore a 2.000 strutture.

  Un elenco di oggetti `DataQualityAnalyzerResult` che rappresentano i risultati per ogni analizzatore.

- `Observations`: una matrice di oggetti DataQualityObservation, non superiore a 50 strutture.

  Un elenco di oggetti `DataQualityObservation` che rappresentano le osservazioni generate dopo la valutazione di regole e analizzatori.

- `AggregatedMetrics`: un oggetto DataQualityAggregatedMetrics.

  Un riepilogo degli `DataQualityAggregatedMetrics` oggetti che mostra il conteggio totale delle righe e delle regole elaborate, comprese le relative pass/fail statistiche basate sui risultati a livello di riga.

Errori

- `InvalidInputException`
- `OperationTimeoutException`
- `InternalServiceException`
- `EntityNotFoundException`

## BatchGetDataQualityResult azione (Python: batch_get_data_quality_result)

Recupera un elenco di risultati della qualità dei dati per gli ID dei risultati specificati.

Richiesta

- `ResultIds`— Obbligatorio: un array di stringhe, non meno di 1 o più di UTF-8 100 stringhe.

  Un elenco di ID dei risultati univoci per i risultati della qualità dei dati.

Risposta

- `Results` - obbligatorio: una matrice di oggetti DataQualityResult.

  Un elenco di oggetti `DataQualityResult` che rappresentano i risultati della qualità dei dati.
- `ResultsNotFound`— Una matrice di UTF-8 stringhe, non meno di 1 o più di 100 stringhe.

  Un elenco di ID dei risultati per i quali non sono stati trovati risultati.

Errori

- `InvalidInputException`
- `OperationTimeoutException`
- `InternalServiceException`

## ListDataQualityResults azione (Python: list_data_quality_results)

Restituisce tutti i risultati di esecuzione della qualità dei dati per il tuo account.

Richiesta

- `Filter`: un oggetto DataQualityResultFilterCriteria.

  I criteri di filtro.
- `NextToken` UTF-8 — stringa.

  Un token di paginazione per partizionare i risultati.
- `MaxResults`: numero (intero), non inferiore a 1 o superiore a 1.000.

  Numero massimo di risultati da restituire.

Risposta

- `Results` - obbligatorio: una matrice di oggetti DataQualityResultDescription.

  Elenco di oggetti `DataQualityResultDescription`.
- `NextToken`— UTF-8 corda.

  Un token di impaginazione, se sono disponibili altri risultati.

Errori

- `InvalidInputException`
- `OperationTimeoutException`
- `InternalServiceException`

## CreateDataQualityRuleset azione (Python: create_data_quality_ruleset)

Crea un set di regole di qualità dei dati con regole DQDL applicate a una tabella specificata. AWS Glue

Il set di regole viene creato utilizzando il Data Quality Definition Language (DQDL). Per ulteriori informazioni, consulta la guida per gli sviluppatori. AWS Glue

Richiesta

Una richiesta di creazione di un set di regole di qualità dei dati.

- `Name`— Obbligatorio: UTF-8 stringa, lunga non meno di 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Un nome univoco per il set di regole di qualità dei dati.

- `Description`: stringa di descrizione, non superiore a 2048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  Una descrizione del set di regole di qualità dei dati.

- `Ruleset`— Obbligatorio: UTF-8 stringa di lunghezza non inferiore a 1 o superiore a 65536 byte.

  Un set di regole Data Quality Definition Language (DQDL). Per ulteriori informazioni, consulta la guida per gli AWS Glue sviluppatori.

- `Tags`: una matrice di mappe con coppie chiave-valore, non superiore alle 50 coppie.

  Ogni chiave è una UTF-8 stringa di lunghezza non inferiore a 1 o superiore a 128 byte.

  Ogni valore è una UTF-8 stringa, lunga non più di 256 byte.

  Un elenco di tag applicati al set di regole di qualità dei dati.

- `TargetTable`: un oggetto DataQualityTargetTable.

  Una tabella di destinazione associata al set di regole di qualità dei dati.

- `RecommendationRunId`— UTF-8 stringa, lunga non meno di 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Un ID di esecuzione univoco per l'esecuzione della raccomandazione.

- `DataQualitySecurityConfiguration`— UTF-8 stringa, di lunghezza non inferiore a 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Il nome della configurazione di sicurezza creata con l'opzione di crittografia di qualità dei dati.

- `ClientToken`— UTF-8 stringa, di lunghezza non inferiore a 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Utilizzato per l'idempotenza e consigliato per l'impostazione su un ID casuale (come un UUID) per evitare di creare o avviare più istanze della stessa risorsa.

Risposta

- Name— UTF-8 stringa, di lunghezza non inferiore a 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Un nome univoco per il set di regole di qualità dei dati.

Errori

- `InvalidInputException`
- `AlreadyExistsException`
- `OperationTimeoutException`
- `InternalServiceException`
- `ResourceNumberLimitExceededException`

## DeleteDataQualityRuleset azione (Python: delete_data_quality_ruleset)

Elimina un set di regole di qualità dei dati.

Richiesta

- Name— Obbligatorio: UTF-8 stringa, lunga non meno di 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Un nome per il set di regole di qualità dei dati.

Risposta

- Nessun parametro di risposta.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `OperationTimeoutException`
- `InternalServiceException`

# GetDataQualityRuleset azione (Python: get_data_quality_ruleset)

Restituisce un set di regole esistente per identificatore o nome.

Richiesta

- `Name`— Obbligatorio: UTF-8 stringa, lunga non meno di 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Il nome del set di regole.

Risposta

Restituisce la risposta del set di regole di qualità dei dati.

- `Name`— UTF-8 stringa, di lunghezza non inferiore a 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Il nome del set di regole.
- `Description`: stringa di descrizione, non superiore a 2048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  Una descrizione del set di regole.
- `Ruleset`— UTF-8 stringa, di lunghezza non inferiore a 1 o superiore a 65536 byte.

  Un set di regole Data Quality Definition Language (DQDL). Per ulteriori informazioni, consulta la guida per gli AWS Glue sviluppatori.
- `TargetTable`: un oggetto DataQualityTargetTable.

  Il nome e il nome del database della tabella di destinazione.
- `CreatedOn`: timestamp.

  Un Timestamp. La data e l'ora di creazione del set di regole di qualità dei dati.
- `LastModifiedOn`: timestamp.

  Un Timestamp. L'ultimo momento in cui questo set di regole di qualità dei dati è stato modificato.
- `RecommendationRunId`— UTF-8 stringa, lunga non meno di 1 o più di 255 byte, corrispondente a. Single-line string pattern

Quando un set di regole è stato creato da un'esecuzione di raccomandazione, questo ID di esecuzione viene generato per collegare i due.

- `DataQualitySecurityConfiguration`— UTF-8 stringa, di lunghezza non inferiore a 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Il nome della configurazione di sicurezza creata con l'opzione di crittografia di qualità dei dati.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `OperationTimeoutException`
- `InternalServiceException`

## ListDataQualityRulesets azione (Python: list_data_quality_rulesets)

Restituisce un elenco impaginato di set di regole per l'elenco di tabelle specificato. AWS Glue

Richiesta

- `NextToken`— UTF-8 stringa.

  Un token di paginazione per partizionare i risultati.
- `MaxResults`: numero (intero), non inferiore a 1 o superiore a 1.000.

  Numero massimo di risultati da restituire.
- `Filter`: un oggetto DataQualityRulesetFilterCriteria.

  I criteri di filtro.
- `Tags`: una matrice di mappe con coppie chiave-valore, non superiore alle 50 coppie.

  Ogni chiave è una UTF-8 stringa di lunghezza non inferiore a 1 o superiore a 128 byte.

  Ogni valore è una UTF-8 stringa, lunga non più di 256 byte.

  Un elenco di tag di coppie chiave-valore.

Risposta

- `Rulesets`: una matrice di oggetti DataQualityRulesetListDetails.

  Un elenco impaginato di set di regole per l'elenco di tabelle specificato. AWS Glue

- `NextToken` UTF-8 — stringa.

  Un token di impaginazione, se sono disponibili altri risultati.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `OperationTimeoutException`
- `InternalServiceException`

## UpdateDataQualityRuleset azione (Python: update_data_quality_ruleset)

Aggiorna il set di regole di qualità dei dati specificato.

Richiesta

- `Name`— Obbligatorio: UTF-8 stringa, lunga non meno di 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Il nome del set di regole di qualità dei dati.

- `Description`: stringa di descrizione, non superiore a 2048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  Una descrizione del set di regole.

- `Ruleset`— UTF-8 stringa, di lunghezza non inferiore a 1 o superiore a 65536 byte.

  Un set di regole Data Quality Definition Language (DQDL). Per ulteriori informazioni, consulta la guida per gli AWS Glue sviluppatori.

Risposta

- `Name`— UTF-8 stringa, lunga non meno di 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Il nome del set di regole di qualità dei dati.
- `Description`: stringa di descrizione, non superiore a 2048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  Una descrizione del set di regole.
- `Ruleset`— UTF-8 stringa, di lunghezza non inferiore a 1 o superiore a 65536 byte.

  Un set di regole Data Quality Definition Language (DQDL). Per ulteriori informazioni, consulta la guida per gli AWS Glue sviluppatori.

Errori

- `EntityNotFoundException`
- `AlreadyExistsException`
- `IdempotentParameterMismatchException`
- `InvalidInputException`
- `OperationTimeoutException`
- `InternalServiceException`
- `ResourceNumberLimitExceededException`

## ListDataQualityStatistics azione (Python: list_data_quality_statistics)

Recupera un elenco di statistiche di qualità dei dati.

Richiesta

- `StatisticId`— UTF-8 stringa, lunga non meno di 1 o più di 255 byte, corrispondente a. Single-line string pattern

  L'ID statistica.
- `ProfileId`— UTF-8 stringa, di lunghezza non inferiore a 1 o più di 255 byte, corrispondente a. Single-line string pattern

L'ID profilo.

- `TimestampFilter`: un oggetto TimestampFilter.

  Un filtro di timestamp.

- `MaxResults`: numero (intero), non inferiore a 1 o superiore a 1.000.

  Numero massimo di risultati da restituire in questa richiesta.

- `NextToken`— UTF-8 stringa.

  Un token di impaginazione per richiedere la pagina successiva di risultati.

Risposta

- `Statistics`: una matrice di oggetti StatisticSummary.

  Un `StatisticSummaryList`.

- `NextToken`— UTF-8 corda.

  Un token di impaginazione per richiedere la pagina successiva di risultati.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `InternalServiceException`

## TimestampFilter struttura

Un filtro di timestamp.

Campi

- `RecordedBefore`: timestamp.

  Il timestamp prima del quale le statistiche devono essere incluse nei risultati.

- `RecordedAfter`: timestamp.

Il timestamp dopo il quale le statistiche devono essere incluse nei risultati.

## CreateDataQualityRulesetRequest struttura

Una richiesta di creazione di un set di regole di qualità dei dati.

Campi

- `Name`— Obbligatorio: UTF-8 stringa, lunga non meno di 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Un nome univoco per il set di regole di qualità dei dati.
- `Description`: stringa di descrizione, non superiore a 2048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  Una descrizione del set di regole di qualità dei dati.
- `Ruleset`— Obbligatorio: UTF-8 stringa di lunghezza non inferiore a 1 o superiore a 65536 byte.

  Un set di regole Data Quality Definition Language (DQDL). Per ulteriori informazioni, consulta la guida per gli AWS Glue sviluppatori.
- `Tags`: una matrice di mappe con coppie chiave-valore, non superiore alle 50 coppie.

  Ogni chiave è una UTF-8 stringa di lunghezza non inferiore a 1 o superiore a 128 byte.

  Ogni valore è una UTF-8 stringa, lunga non più di 256 byte.

  Un elenco di tag applicati al set di regole di qualità dei dati.
- `TargetTable`: un oggetto DataQualityTargetTable.

  Una tabella di destinazione associata al set di regole di qualità dei dati.
- `RecommendationRunId`— UTF-8 stringa, lunga non meno di 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Un ID di esecuzione univoco per l'esecuzione della raccomandazione.
- `DataQualitySecurityConfiguration`— UTF-8 stringa, di lunghezza non inferiore a 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Il nome della configurazione di sicurezza creata con l'opzione di crittografia di qualità dei dati.

- `ClientToken`— UTF-8 stringa, di lunghezza non inferiore a 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Utilizzato per l'idempotenza e consigliato per l'impostazione su un ID casuale (come un UUID) per evitare di creare o avviare più istanze della stessa risorsa.

## GetDataQualityRulesetResponse struttura

Restituisce la risposta del set di regole di qualità dei dati.

Campi

- `Name`— UTF-8 stringa, lunga non meno di 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Il nome del set di regole.

- `Description`: stringa di descrizione, non superiore a 2048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  Una descrizione del set di regole.

- `Ruleset`— UTF-8 stringa, di lunghezza non inferiore a 1 o superiore a 65536 byte.

  Un set di regole Data Quality Definition Language (DQDL). Per ulteriori informazioni, consulta la guida per gli AWS Glue sviluppatori.

- `TargetTable`: un oggetto DataQualityTargetTable.

  Il nome e il nome del database della tabella di destinazione.

- `CreatedOn`: timestamp.

  Un Timestamp. La data e l'ora di creazione del set di regole di qualità dei dati.

- `LastModifiedOn`: timestamp.

  Un Timestamp. L'ultimo momento in cui questo set di regole di qualità dei dati è stato modificato.

- `RecommendationRunId`— UTF-8 stringa, lunga non meno di 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Quando un set di regole è stato creato da un'esecuzione di raccomandazione, questo ID di esecuzione viene generato per collegare i due.

- `DataQualitySecurityConfiguration`— UTF-8 stringa, di lunghezza non inferiore a 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Il nome della configurazione di sicurezza creata con l'opzione di crittografia di qualità dei dati.

## GetDataQualityResultResponse struttura

La risposta del risultato di qualità dei dati.

Campi

- `ResultId`— UTF-8 stringa, lunga non meno di 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Un ID di risultato univoco per il risultato della qualità dei dati.
- `ProfileId`— UTF-8 stringa, di lunghezza non inferiore a 1 o più di 255 byte, corrispondente a. Single-line string pattern

  L'ID profilo del risultato di qualità dei dati.
- `Score`: numero (doppio), non superiore a 1.

  Un punteggio aggregato della qualità dei dati. Rappresenta il rapporto tra le regole inviate e il numero totale di regole.
- `DataSource`: un oggetto DataSource.

  La tabella associata al risultato della qualità dei dati, se presente.
- `RulesetName`— UTF-8 stringa, di lunghezza non inferiore a 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Il nome del set di regole associato al risultato della qualità dei dati.
- `EvaluationContext`— UTF-8 stringa.

  Nel contesto di un lavoro in AWS Glue Studio, a ogni nodo dell'area di disegno viene in genere assegnato un nome e i nodi di qualità dei dati avranno dei nomi. Nel caso di più nodi, `evaluationContext` può differenziare i nodi.
- `StartedOn`: timestamp.

  La data e ora di inizio dell'esecuzione di questo risultato della qualità dei dati.

- `CompletedOn`: timestamp.

  La data e ora di completamento dell'esecuzione di questo risultato della qualità dei dati.

- `JobName`— UTF-8 stringa, lunga non meno di 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Il nome del processo associato al risultato della qualità dei dati, se presente.

- `JobRunId`— UTF-8 stringa, di lunghezza non inferiore a 1 o più di 255 byte, corrispondente a. Single-line string pattern

  L'ID di esecuzione del processo associato al risultato della qualità dei dati, se presente.

- `RulesetEvaluationRunId`— UTF-8 stringa, di lunghezza non inferiore a 1 o più di 255 byte, corrispondente a. Single-line string pattern

  L'ID di esecuzione univoco associato alla valutazione del set di regole.

- `RuleResults`: una matrice di oggetti DataQualityRuleResult, non superiore a 2.000 strutture.

  Un elenco di oggetti `DataQualityRuleResult` che rappresentano i risultati per ogni regola.

- `AnalyzerResults`: una matrice di oggetti DataQualityAnalyzerResult, non superiore a 2.000 strutture.

  Un elenco di oggetti `DataQualityAnalyzerResult` che rappresentano i risultati per ogni analizzatore.

- `Observations`: una matrice di oggetti DataQualityObservation, non superiore a 50 strutture.

  Un elenco di oggetti `DataQualityObservation` che rappresentano le osservazioni generate dopo la valutazione di regole e analizzatori.

- `AggregatedMetrics`: un oggetto DataQualityAggregatedMetrics.

  Un riepilogo degli `DataQualityAggregatedMetrics` oggetti che mostra il conteggio totale delle righe e delle regole elaborate, comprese le relative pass/fail statistiche basate sui risultati a livello di riga.

## StartDataQualityRuleRecommendationRunRequest struttura

La richiesta di raccomandazione della regola di qualità dei dati.

Campi

- `DataSource` - obbligatorio: un oggetto DataSource.

  L'origine dati (AWS Glue tabella) associata a questa esecuzione.

- `Role`— Obbligatorio: UTF-8 stringa.

  Un IAM ruolo fornito per crittografare i risultati dell'esecuzione.

- `NumberOfWorkers`: numero (intero).

  Il numero di worker `G.1X` da utilizzare nell'esecuzione. Il predefinito è 5.

- `Timeout`: numero (intero), almeno 1.

  Il timeout per una esecuzione in minuti. Questo è il tempo massimo durante il quale un'esecuzione può utilizzare le risorse prima di essere terminata e passare allo stato `TIMEOUT`. Il valore di default è 2.880 minuti (48 ore).

- `CreatedRulesetName`— UTF-8 stringa, lunga non meno di 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Un nome per il set di regole.

- `DataQualitySecurityConfiguration`— UTF-8 stringa, di lunghezza non inferiore a 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Il nome della configurazione di sicurezza creata con l'opzione di crittografia di qualità dei dati.

- `ClientToken`— UTF-8 stringa, di lunghezza non inferiore a 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Utilizzato per l'idempotenza e consigliato per l'impostazione su un ID casuale (come un UUID) per evitare di creare o avviare più istanze della stessa risorsa.

## GetDataQualityRuleRecommendationRunResponse struttura

La risposta dell'esecuzione della raccomandazione della regola di qualità dei dati.

Campi

- `RunId`— UTF-8 stringa, lunga non meno di 1 o più di 255 byte, corrispondente a. Single-line string pattern

L'identificatore univoco dell'esecuzione associato a questa esecuzione di attività.

- `DataSource`: un oggetto DataSource.

  L'origine dati (una AWS Glue tabella) associata a questa esecuzione.

- `Role`— UTF-8 stringa.

  Un IAM ruolo fornito per crittografare i risultati dell'esecuzione.

- `NumberOfWorkers`: numero (intero).

  Il numero di worker `G.1X` da utilizzare nell'esecuzione. Il predefinito è 5.

- `Timeout`: numero (intero), almeno 1.

  Il timeout per una esecuzione in minuti. Questo è il tempo massimo durante il quale un'esecuzione può utilizzare le risorse prima di essere terminata e passare allo stato `TIMEOUT`. Il valore di default è 2.880 minuti (48 ore).

- `Status`— UTF-8 stringa (valori validi: `RUNNING` | `FINISHED` | `FAILED` | `PENDING_EXECUTION` | | `TIMED_OUT` | `CANCELING` `CANCELED` |`RECEIVED_BY_TASKRUNNER`).

  Lo stato di questa esecuzione.

- `ErrorString`— UTF-8 stringa.

  Le stringhe di errore associate all'esecuzione.

- `StartedOn`: timestamp.

  La data e l'ora in cui è stata avviata questa esecuzione.

- `LastModifiedOn`: timestamp.

  Un Timestamp. L'ultimo momento in cui questa raccomandazione della regola di qualità dei dati è stata modificata.

- `CompletedOn`: timestamp.

  La data e l'ora in cui è stata completata questa esecuzione.

- `ExecutionTime`: numero (intero).

  La quantità di tempo (in secondi) durante la quale l'esecuzione ha utilizzato le risorse.

- `RecommendedRuleset`— UTF-8 stringa, lunga non meno di 1 o più di 65536 byte.

Una volta completata l'esecuzione di una raccomandazione della regola di avvio, viene creato un set di regole consigliato (una serie di regole). Questo membro ha queste regole nel formato DQDL (Data Quality Definition Language).

- `CreatedRulesetName`— UTF-8 stringa, di lunghezza non inferiore a 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Il nome del set di regole che è stato creato dall'esecuzione.

- `DataQualitySecurityConfiguration`— UTF-8 stringa, di lunghezza non inferiore a 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Il nome della configurazione di sicurezza creata con l'opzione di crittografia di qualità dei dati.

## BatchPutDataQualityStatisticAnnotation azione (Python: batch_put_data_quality_statistic_annotation)

Annota i datapoint nel tempo per una statistica di qualità dei dati specifica. L'API richiede sia ProfileID che statisticID come parte dell'input. InclusionAnnotation L'API funziona solo per un singolo statisticId in più profili.

Richiesta

- `InclusionAnnotations` - obbligatorio: una matrice di oggetti DatapointInclusionAnnotation.

  Un elenco di `DatapointInclusionAnnotation`. InclusionAnnotationsDeve contenere un ProfileID e un StatisticId. Se ce ne sono più InclusionAnnotations, l'elenco deve fare riferimento a un singolo StatisticID su più ProfileID.

- `ClientToken`— UTF-8 stringa, lunga non meno di 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Token client.

Risposta

- `FailedInclusionAnnotations`: una matrice di oggetti AnnotationError.

  Un elenco di `AnnotationError`.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `InternalServiceException`
- `ResourceNumberLimitExceededException`

## GetDataQualityModel azione (Python: get_data_quality_model)

Recupera lo stato di addestramento del modello insieme a ulteriori informazioni (,,). CompletedOn StartedOn FailureReason

Richiesta

- `StatisticId`— UTF-8 stringa, lunga non meno di 1 o più di 255 byte, corrispondente a. Single-line string pattern

  L'ID statistica.
- `ProfileId`— Obbligatorio: UTF-8 stringa, di lunghezza non inferiore a 1 o più di 255 byte, corrispondente a. Single-line string pattern

  L'ID profilo.

Risposta

- `Status`— UTF-8 stringa (valori validi: `RUNNING` | `SUCCEEDED` |`FAILED`).

  Lo stato di addestramento del modello di qualità dei dati.
- `StartedOn`: timestamp.

  Il timestamp del momento di inizio dell'addestramento del modello.
- `CompletedOn`: timestamp.

  Il timestamp del momento di completamento dell'addestramento del modello.
- `FailureReason`— UTF-8 stringa, lunga non meno di 1 o più di 255 byte, corrispondente a. Single-line string pattern

  Il motivo per cui l'addestramento non è riuscito.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `OperationTimeoutException`
- `InternalServiceException`

## GetDataQualityModelResult azione (Python: get_data_quality_model_result)

Recupera le previsioni di una statistica per un determinato ID profilo.

Richiesta

- `StatisticId`— Obbligatorio: UTF-8 stringa, lunga non meno di 1 o più di 255 byte, corrispondente a. Single-line string pattern

  L'ID statistica.
- `ProfileId`— Obbligatorio: UTF-8 stringa, di lunghezza non inferiore a 1 o più di 255 byte, corrispondente a. Single-line string pattern

  L'ID profilo.

Risposta

- `CompletedOn`: timestamp.

  Il timestamp del momento di completamento dell'addestramento del modello.
- `Model`: una matrice di oggetti StatisticModelResult.

  Un elenco di `StatisticModelResult`.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `OperationTimeoutException`
- `InternalServiceException`

# ListDataQualityStatisticAnnotations azione (Python: list_data_quality_statistic_annotations)

Recupera le annotazioni di una statistica di qualità dei dati.

Richiesta

- `StatisticId`— UTF-8 stringa, lunga non meno di 1 o più di 255 byte, corrispondente a. Single-line string pattern

  L'ID statistica.
- `ProfileId`— UTF-8 stringa, di lunghezza non inferiore a 1 o più di 255 byte, corrispondente a. Single-line string pattern

  L'ID profilo.
- `TimestampFilter`: un oggetto TimestampFilter.

  Un filtro di timestamp.
- `MaxResults`: numero (intero), non inferiore a 1 o superiore a 1.000.

  Numero massimo di risultati da restituire in questa richiesta.
- `NextToken`— UTF-8 stringa.

  Un token di impaginazione per recuperare il set di risultati successivo.

Risposta

- `Annotations`: una matrice di oggetti StatisticAnnotation.

  Un elenco di `StatisticAnnotation` applicato alla statistica.
- `NextToken`— UTF-8 corda.

  Un token di impaginazione per recuperare il set di risultati successivo.

Errori

- `InvalidInputException`
- `InternalServiceException`

## PutDataQualityProfileAnnotation azione (Python: put_data_quality_profile_annotation)

Annota tutti i datapoint di un profilo.

Richiesta

- `ProfileId`— Obbligatorio: UTF-8 stringa, lunga non meno di 1 o più di 255 byte, corrispondente a. Single-line string pattern

  L'ID del profilo di monitoraggio della qualità dei dati da annotare.
- `InclusionAnnotation`— Obbligatorio: UTF-8 stringa (valori validi: `INCLUDE` |`EXCLUDE`).

  Il valore dell'annotazione di inclusione da applicare al profilo.

Risposta

- Nessun parametro di risposta.

Errori

- `EntityNotFoundException`
- `InvalidInputException`
- `InternalServiceException`

# API di rilevamento dati sensibili

L'API di rilevamento dei dati sensibili descrive le modalità APIs utilizzate per rilevare i dati sensibili nelle colonne e nelle righe dei dati strutturati.

## Tipi di dati

- CustomEntityType struttura

## CustomEntityType struttura

Un oggetto che rappresenta un modello personalizzato per il rilevamento di dati sensibili tra colonne e righe dei dati strutturati.

Campi

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Un nome per il pattern personalizzato che consente di recuperarlo o cancellarlo in un secondo momento. Questo nome deve essere unico per AWS account.
- `RegexString` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Stringa di espressione regolare utilizzata per rilevare dati sensibili in un modello personalizzato.
- `ContextWords`: una matrice di stringhe UTF-8, non inferiore a 1 o superiore a 20 stringhe.

  Un elenco di parole contestuali. Se nessuna di queste parole contestuali viene trovata nelle vicinanze dell'espressione regolare, i dati non verranno rilevati come dati sensibili.

  Se non vengono passate parole contestuali, viene controllata solo un'espressione regolare.

## Operazioni

- CreateCustomEntityType azione (Python: create_custom_entity_type)
- DeleteCustomEntityType azione (Python: delete_custom_entity_type)
- GetCustomEntityType azione (Python: get_custom_entity_type)
- BatchGetCustomEntityTypes azione (Python: batch_get_custom_entity_types)
- ListCustomEntityTypes azione (Python: list_custom_entity_types)

## CreateCustomEntityType azione (Python: create_custom_entity_type)

Crea un modello personalizzato utilizzato per rilevare dati sensibili tra le colonne e le righe dei dati strutturati.

Ogni modello personalizzato creato specifica un'espressione regolare e un elenco facoltativo di parole contestuali. Se non vengono passate parole contestuali, viene controllata solo un'espressione regolare.

Richiesta

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Un nome per il pattern personalizzato che consente di recuperarlo o cancellarlo in un secondo momento. Questo nome deve essere univoco per account. AWS
- `RegexString` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Stringa di espressione regolare utilizzata per rilevare dati sensibili in un modello personalizzato.
- `ContextWords`: una matrice di stringhe UTF-8, non inferiore a 1 o superiore a 20 stringhe.

  Un elenco di parole contestuali. Se nessuna di queste parole contestuali viene trovata nelle vicinanze dell'espressione regolare, i dati non verranno rilevati come dati sensibili.

  Se non vengono passate parole contestuali, viene controllata solo un'espressione regolare.
- `Tags`: una matrice di mappe con coppie chiave-valore, non superiore alle 50 coppie.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Ogni valore è una stringa UTF-8, lunga non più di 256 byte.

  Un elenco di tag applicati al tipo di entità personalizzato.

Risposta

- `Name`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del modello personalizzato che hai creato.

Errori

- `AccessDeniedException`

- `AlreadyExistsException`
- `IdempotentParameterMismatchException`
- `InternalServiceException`
- `InvalidInputException`
- `OperationTimeoutException`
- `ResourceNumberLimitExceededException`

## DeleteCustomEntityType azione (Python: delete_custom_entity_type)

Elimina un modello personalizzato specificandone il nome.

Richiesta

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del modello personalizzato da eliminare.

Risposta

- `Name`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del modello personalizzato che hai eliminato.

Errori

- `EntityNotFoundException`
- `AccessDeniedException`
- `InternalServiceException`
- `InvalidInputException`
- `OperationTimeoutException`

## GetCustomEntityType azione (Python: get_custom_entity_type)

Recupera i dettagli di un modello personalizzato specificandone il nome.

Richiesta

- `Name` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del modello personalizzato da recuperare.

Risposta

- `Name`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome del modello personalizzato recuperato.
- `RegexString`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Stringa di espressione regolare utilizzata per rilevare dati sensibili in un modello personalizzato.
- `ContextWords`: una matrice di stringhe UTF-8, non inferiore a 1 o superiore a 20 stringhe.

  Un elenco di parole contestuali, se specificato quando è stato creato il modello personalizzato. Se nessuna di queste parole contestuali viene trovata nelle vicinanze dell'espressione regolare, i dati non verranno rilevati come dati sensibili.

Errori

- `EntityNotFoundException`
- `AccessDeniedException`
- `InternalServiceException`
- `InvalidInputException`
- `OperationTimeoutException`

## BatchGetCustomEntityTypes azione (Python: batch_get_custom_entity_types)

Recupera i dettagli per i modelli personalizzati specificati da un elenco di nomi.

Richiesta

- `Names`: obbligatorio: una matrice di stringhe UTF-8, non inferiore a 1 o superiore a 50 stringhe.

  Un elenco di nomi dei modelli personalizzati da recuperare.

Risposta

- `CustomEntityTypes`: una matrice di oggetti CustomEntityType.

  Un elenco di oggetti `CustomEntityType` che rappresentano i modelli personalizzati creati.
- `CustomEntityTypesNotFound`: una matrice di stringhe UTF-8, non inferiore a 1 o superiore a 50 stringhe.

  Un elenco dei nomi dei modelli personalizzati che non sono stati trovati.

Errori

- `InvalidInputException`
- `InternalServiceException`
- `OperationTimeoutException`

## ListCustomEntityTypes azione (Python: list_custom_entity_types)

Elenca tutti i modelli personalizzati che sono stati creati.

Richiesta

- `NextToken`: stringa UTF-8.

  Un token di paginazione per partizionare i risultati.
- `MaxResults`: numero (intero), non inferiore a 1 o superiore a 1.000.

  Numero massimo di risultati da restituire.
- `Tags`: una matrice di mappe con coppie chiave-valore, non superiore alle 50 coppie.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Ogni valore è una stringa UTF-8, lunga non più di 256 byte.

Un elenco di tag di coppie chiave-valore.

Risposta

- `CustomEntityTypes`: una matrice di oggetti CustomEntityType.

  Un elenco di oggetti `CustomEntityType` che rappresentano modelli personalizzati.
- `NextToken`: stringa UTF-8.

  Un token di impaginazione, se sono disponibili altri risultati.

Errori

- `InvalidInputException`
- `OperationTimeoutException`
- `InternalServiceException`

# Taggare APIs AWS Glue

## Tipi di dati

- Struttura tag

## Struttura tag

L'`Tag`oggetto rappresenta un'etichetta che è possibile assegnare a una AWS risorsa. Ogni tag è composto da una chiave e da un valore opzionale, entrambi personalizzabili.

Per ulteriori informazioni sui tag e sul controllo dell'accesso alle risorse in AWS Glue, consulta AWS Tags in AWS Glue e Specifying AWS Glue Resource ARNs nella guida per sviluppatori.

Campi

- `key`: stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  La chiave di tag: La chiave è obbligatoria quando si crea un tag per un oggetto. La chiave rispetta la distinzione tra maiuscole e minuscole e non deve contenere il prefisso aws.

- `value`: stringa UTF-8, non superiore a 256 byte di lunghezza.

  Il valore del tag. Il valore è facoltativo quando si crea un tag per un oggetto. Il valore rispetta la distinzione tra maiuscole e minuscole e non deve contenere il prefisso aws.

## Operazioni

- TagResource azione (Python: tag_resource)
- UntagResource azione (Python: untag_resource)
- GetTags azione (Python: get_tags)

## TagResource azione (Python: tag_resource)

Aggiunge tag a una risorsa. Un tag è un'etichetta che puoi assegnare a una risorsa. AWS In AWS Glue, puoi taggare solo determinate risorse. Per informazioni sulle risorse cui è possibile applicare un tag, consulta Tag AWS in AWS Glue.

Oltre alle autorizzazioni di etichettatura relative ai tag APIs, è necessaria anche l'`glue:GetConnection`autorizzazione a richiamare il tagging APIs sulle connessioni e l'`glue:GetDatabase`autorizzazione a richiamare il APIs tagging sui database.

Richiesta

- `ResourceArn` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 10240 byte di lunghezza, corrispondente a Custom string pattern #50.

  L'ARN della AWS Glue risorsa a cui aggiungere i tag. Per ulteriori informazioni sulla AWS Glue risorsa ARNs, vedere lo schema di stringhe AWS Glue ARN.
- `TagsToAdd` - obbligatorio: una matrice di mappe di coppie chiave-valore, non superiore a 50 coppie.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Ogni valore è una stringa UTF-8, lunga non più di 256 byte.

  Tag da aggiungere a questa risorsa.

Risposta

- Nessun parametro di risposta.

Errori

- `ResourceNotFoundException`

## UntagResource azione (Python: untag_resource)

Rimuove i tag specificati da una risorsa di integrazione.

Richiesta

- `ResourceArn` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 10240 byte di lunghezza, corrispondente a Custom string pattern #50.

  Il nome della risorsa Amazon (ARN) della risorsa di integrazione.
- `TagsToRemove` - obbligatorio: una matrice di stringhe UTF-8, non superiore a 50 stringhe.

  Un elenco di tag di metadati da rimuovere dalla risorsa.

Risposta

- Nessun parametro di risposta.

Errori

- `ResourceNotFoundException`

## GetTags azione (Python: get_tags)

Recupera un elenco di tag associati a una risorsa.

Richiesta

- `ResourceArn` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 10240 byte di lunghezza, corrispondente a Custom string pattern #50.

Il nome della risorsa Amazon (ARN) della risorsa per cui recuperare i tag.

Risposta

- `Tags`: una matrice di mappe con coppie chiave-valore, non superiore alle 50 coppie.

  Ogni chiave è una stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  Ogni valore è una stringa UTF-8, lunga non più di 256 byte.

  I tag richiesti.

Errori

- `InvalidInputException`
- `InternalServiceException`
- `OperationTimeoutException`
- `EntityNotFoundException`

# Tipi di dati comuni

I tipi di dati comuni descrivono i vari tipi di dati comuni in AWS Glue.

## Struttura tag

L'`Tag`oggetto rappresenta un'etichetta che è possibile assegnare a una AWS risorsa. Ogni tag è composto da una chiave e da un valore opzionale, entrambi personalizzabili.

Per ulteriori informazioni sui tag e sul controllo dell'accesso alle risorse in AWS Glue, consulta AWS Tags in AWS Glue e Specifying AWS Glue Resource ARNs nella guida per sviluppatori.

Campi

- `key`: stringa UTF-8, non inferiore a 1 o superiore a 128 byte di lunghezza.

  La chiave di tag: La chiave è obbligatoria quando si crea un tag per un oggetto. La chiave rispetta la distinzione tra maiuscole e minuscole e non deve contenere il prefisso aws.
- `value`: stringa UTF-8, non superiore a 256 byte di lunghezza.

Il valore del tag. Il valore è facoltativo quando si crea un tag per un oggetto. Il valore rispetta la distinzione tra maiuscole e minuscole e non deve contenere il prefisso aws.

## DecimalNumber struttura

Contiene un valore numerico nel formato decimale.

Campi

- `UnscaledValue` - obbligatorio: blob.

  Il valore numerico non scalato.
- `Scale` - obbligatorio: numero (intero).

  La scala che determina la posizione del punto decimale nel valore non scalato.

## ErrorDetail struttura

Contiene dettagli su un errore.

Campi

- `ErrorCode`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il codice associato a questo errore.
- `ErrorMessage`: stringa di descrizione, non superiore a 2048 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  Messaggio che descrive l'errore.

## PropertyPredicate struttura

Definisce il predicato di una proprietà.

Campi

- `Key`: stringa Value, non inferiore a 1 o superiore a 1024 byte di lunghezza.

La chiave della proprietà.

- `Value`: stringa Value, non inferiore a 1 o superiore a 1024 byte di lunghezza.

  Valore della proprietà.

- `Comparator`: stringa UTF-8 (valori validi: `EQUALS` | `GREATER_THAN` | `LESS_THAN` | `GREATER_THAN_EQUALS` | `LESS_THAN_EQUALS`).

  Il comparatore utilizzato per confrontare questa proprietà con altre.

## ResourceUri struttura

Le risorse URIs per le funzioni.

Campi

- `ResourceType`: stringa UTF-8 (valori validi: `JAR` | `FILE` | `ARCHIVE`).

  Il tipo di risorsa.

- `Uri`: uniform resource identifier (uri), non inferiore a 1 e non superiore a 1024 byte di lunghezza, corrispondente a URI address multi-line string pattern.

  L'URI per l'accesso alla risorsa.

## ColumnStatistics struttura

Rappresenta le statistiche a livello di colonna generate per una tabella o una partizione.

Campi

- `ColumnName` - obbligatorio: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Nome della colonna a cui appartengono le statistiche.

- `ColumnType` - obbligatorio: il nome del tipo, non superiore a 20000 byte di lunghezza, corrispondente a Single-line string pattern.

  Il tipo di dati della colonna.

- `AnalyzedTime` - obbligatorio: timestamp.

Il timestamp dell'ora di generazione delle statistiche di colonna.

- `StatisticsData` - obbligatorio: un oggetto ColumnStatisticsData.

  Un oggetto `ColumnStatisticData` che contiene i valori dei dati delle statistiche.

## ColumnStatisticsError struttura

Incapsula un oggetto `ColumnStatistics` non riuscito e il motivo dell'errore.

Campi

- `ColumnStatistics`: un oggetto ColumnStatistics.

  `ColumnStatistics` della colonna.

- `Error`: un oggetto ErrorDetail.

  Un messaggio di errore con il motivo dell'errore di un'operazione.

## ColumnError struttura

Incapsula il nome di una colonna non riuscita e il motivo dell'errore.

Campi

- `ColumnName`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  Il nome della colonna non riuscita.

- `Error`: un oggetto ErrorDetail.

  Un messaggio di errore con il motivo dell'errore di un'operazione.

## ColumnStatisticsData struttura

Contiene i singoli tipi di dati delle statistiche delle colonne. Solo un oggetto dati deve essere impostato e indicato dall'attributo `Type`.

Campi

- `Type` - obbligatorio: stringa UTF-8 (valori validi: `BOOLEAN` | `DATE` | `DECIMAL` | `DOUBLE` | `LONG` | `STRING` | `BINARY`).

  Il tipo di dati delle statistiche delle colonne.
- `BooleanColumnStatisticsData`: un oggetto BooleanColumnStatisticsData.

  Dati statistici di colonna booleani.
- `DateColumnStatisticsData`: un oggetto DateColumnStatisticsData.

  Dati statistici di colonna date.
- `DecimalColumnStatisticsData`: un oggetto DecimalColumnStatisticsData.

  Dati statistici delle colonne decimali. UnscaledValues all'interno ci sono oggetti binari codificati in Base64 che memorizzano rappresentazioni big-endian, due come complemento, del valore non scalato del decimale.
- `DoubleColumnStatisticsData`: un oggetto DoubleColumnStatisticsData.

  Dati statistici di colonna doppi.
- `LongColumnStatisticsData`: un oggetto LongColumnStatisticsData.

  Dati statistici di colonna long.
- `StringColumnStatisticsData`: un oggetto StringColumnStatisticsData.

  Dati statistici di colonna stringa.
- `BinaryColumnStatisticsData`: un oggetto BinaryColumnStatisticsData.

  Dati statistici di colonna binari.

## BooleanColumnStatisticsData struttura

Definisce le statistiche di colonna supportate per le colonne di dati booleani.

Campi

- `NumberOfTrues` - obbligatorio: numero (long), non superiore a Nessuno.

  Il numero di valori true nella colonna.

- `NumberOfFalses` - obbligatorio: numero (long), non superiore a Nessuno.

  Il numero di valori false nella colonna.
- `NumberOfNulls` - obbligatorio: numero (long), non superiore a Nessuno.

  Il numero di valori null nella colonna.

## DateColumnStatisticsData struttura

Definisce le statistiche di colonna supportate per le colonne di dati timestamp.

Campi

- `MinimumValue`: timestamp.

  Il valore più basso nella colonna.
- `MaximumValue`: timestamp.

  Il valore più alto nella colonna.
- `NumberOfNulls` - obbligatorio: numero (long), non superiore a Nessuno.

  Il numero di valori null nella colonna.
- `NumberOfDistinctValues` - obbligatorio: numero (long), non superiore a Nessuno.

  Il numero di valori distinti in una colonna.

## DecimalColumnStatisticsData struttura

Definisce le statistiche di colonna supportate per le colonne di dati con numeri a virgola fissa.

Campi

- `MinimumValue`: un oggetto DecimalNumber.

  Il valore più basso nella colonna.
- `MaximumValue`: un oggetto DecimalNumber.

  Il valore più alto nella colonna.
- `NumberOfNulls` - obbligatorio: numero (long), non superiore a Nessuno.

Il numero di valori null nella colonna.

- `NumberOfDistinctValues` - obbligatorio: numero (long), non superiore a Nessuno.

  Il numero di valori distinti in una colonna.

## DoubleColumnStatisticsData struttura

Definisce le statistiche di colonna supportate per le colonne di dati con numeri a virgola mobile.

Campi

- `MinimumValue`: numero (doppio).

  Il valore più basso nella colonna.
- `MaximumValue`: numero (doppio).

  Il valore più alto nella colonna.
- `NumberOfNulls` - obbligatorio: numero (long), non superiore a Nessuno.

  Il numero di valori null nella colonna.
- `NumberOfDistinctValues` - obbligatorio: numero (long), non superiore a Nessuno.

  Il numero di valori distinti in una colonna.

## LongColumnStatisticsData struttura

Definisce le statistiche di colonna supportate per le colonne di dati interi.

Campi

- `MinimumValue`: numero (lungo).

  Il valore più basso nella colonna.
- `MaximumValue`: numero (lungo).

  Il valore più alto nella colonna.
- `NumberOfNulls` - obbligatorio: numero (long), non superiore a Nessuno.

Il numero di valori null nella colonna.

- `NumberOfDistinctValues` - obbligatorio: numero (long), non superiore a Nessuno.

  Il numero di valori distinti in una colonna.

## StringColumnStatisticsData struttura

Definisce le statistiche di colonna supportate per i valori dei dati di sequenza.

Campi

- `MaximumLength` - obbligatorio: numero (long), non superiore a Nessuno.

  La dimensione della stringa più lunga nella colonna.

- `AverageLength`: obbligatorio: numero (long), non superiore a Nessuno.

  La lunghezza media della stringa nella colonna.

- `NumberOfNulls` - obbligatorio: numero (long), non superiore a Nessuno.

  Il numero di valori null nella colonna.

- `NumberOfDistinctValues` - obbligatorio: numero (long), non superiore a Nessuno.

  Il numero di valori distinti in una colonna.

## BinaryColumnStatisticsData struttura

Definisce le statistiche di colonna supportate per i valori dei dati di sequenza di bit.

Campi

- `MaximumLength` - obbligatorio: numero (long), non superiore a Nessuno.

  La dimensione della sequenza di bit più lunga nella colonna.

- `AverageLength` - obbligatorio: numero (long), non superiore a Nessuno.

  La lunghezza media della sequenza di bit nella colonna.

- `NumberOfNulls` - obbligatorio: numero (long), non superiore a Nessuno.

Il numero di valori null nella colonna.

## Modelli di stringa

L'API usa le seguenti espressioni regolari per definire i contenuti validi per vari membri e parametri di stringa:

- Pattern di stringa a una riga: "`[\u0020-\uD7FF\uE000-\uFFFD\uD800\uDC00-\uDBFF\uDFFF\t]*`"
- Pattern di stringa a più righe per indirizzo URI: "`[\u0020-\uD7FF\uE000-\uFFFD\uD800\uDC00-\uDBFF\uDFFF\r\n\t]*`"
- Pattern di stringa Logstash Grok: "`[\u0020-\uD7FF\uE000-\uFFFD\uD800\uDC00-\uDBFF\uDFFF\r\t]*`"
- Pattern di stringa identificatore: "`[A-Za-z_][A-Za-z0-9_]*`"
- Pattern di stringa ARN AWS IAM: "`arn:aws:iam::\d{12}:role/.*`"
- Pattern di stringa di versione: "`^[a-zA-Z0-9-_]+$`"
- Pattern di stringa gruppo di log: "`[\.\-_/#A-Za-z0-9]+`"
- Pattern di stringa flusso di log: "`[^:*]*`"
- Pattern di stringa personalizzato n. 10: "`[a-zA-Z0-9-_]+`"
- Pattern di stringa personalizzato n. 11: "`[-a-zA-Z0-9+=/:_]*`"
- Pattern di stringa personalizzato n. 12: "`[\S\s]*`"
- Pattern di stringa personalizzato n. 13: "`.*\S.*`"
- Pattern di stringa personalizzato n. 14: "`[a-zA-Z0-9-=._/@]+`"
- Pattern di stringa personalizzato n. 15: "`[1-9][0-9]*|[1-9][0-9]*-[1-9][0-9]*`"
- Pattern di stringa personalizzato n. 16: "`[A-Z][A-Za-z\.]+`"
- Pattern di stringa personalizzato n. 17: "`[\S]*`"
- Pattern di stringa personalizzato n. 18: "`[\w]*`"
- Pattern di stringa personalizzato n. 19: "`arn:aws[a-z\-]*:iam::\d{12}:role/?[a-zA-Z_0-9+=,.@\-_/]+`"
- Pattern di stringa personalizzato n. 20: "`subnet-[a-z0-9]+`"
- Pattern di stringa personalizzato n. 21: "`\d{12}`"

- Pattern di stringa personalizzato n. 22: "([a-z]+)-([a-z]+-)?([a-z]+)-[0-9]+[a-z]+"
- Pattern di stringa personalizzato n. 23: "[a-zA-Z0-9.-]*"
- Pattern di stringa personalizzato n. 24: "arn:aws[a-z0-9\-]*:lambda:[a-z0-9\-]+:\d{12}:function:([\w\-]{1,64})"
- Pattern di stringa personalizzato n. 25: "^(?!(.*[.\/\\]|aws:)).*$"
- Pattern di stringa personalizzato n. 26: "[^\r\n]"
- Pattern di stringa personalizzato n. 27: "^\w+\.\w+\.\w+$"
- Pattern di stringa personalizzato n. 28: "^\w+\.\w+$"
- Pattern di stringa personalizzato n. 29: "^$|arn:aws[a-z0-9-]*:kms:.*"
- Pattern di stringa personalizzato n. 30: "arn:aws[^:]*:iam::[0-9]*:role/.+"
- Pattern di stringa personalizzato n. 31: "[\.\-_A-Za-z0-9]+"
- Pattern di stringa personalizzato n. 32: "^s3://([^/]+)/([^/]+/)*([^/]+)$"
- Pattern di stringa personalizzato n. 33: ".*"
- Pattern di stringa personalizzato n 34: "^(Sun|Mon|Tue|Wed|Thu|Fri|Sat):([01]?[0-9]|2[0-3])$"
- Pattern di stringa personalizzato n. 35: "[a-zA-Z0-9_.-]+"
- Pattern di stringa personalizzato n. 36: "^arn:aws(-(cn|us-gov|eusc|iso(-[bef])?))?:secretsmanager:.*$"
- Pattern di stringa personalizzato n. 37: "\S+"
- Pattern di stringa personalizzato n. 38: "^[\x20-\x7E]*$"
- Pattern di stringa personalizzato n. 39: "^([a-zA-Z0-9_=]+)\.([a-zA-Z0-9_=]+)\.([a-zA-Z0-9_\-\+\/=]*)"
- Pattern di stringa personalizzato n. 40: "^(https?://[-a-zA-Z0-9+&@#/%?=~_|!:,.;]*[-a-zA-Z0-9+&@#/%=~_|]"
- Pattern di stringa personalizzato n. 41: "^(https?):\/\/[^\s/$.?#].[^\s]*$"
- Pattern di stringa personalizzato n. 42: "arn:aws:kms:.*"
- Pattern di stringa personalizzato n. 43: "^subnet-[a-z0-9]+$"
- Pattern di stringa personalizzato n. 44: "[\p{L}\p{N}\p{P}]*"
- Pattern di stringa personalizzato n. 45: "[a-f0-9]{8}-[a-f0-9]{4}-[a-f0-9]{4}-[a-f0-9]{4}-[a-f0-9]{12}"

- Pattern di stringa personalizzato n. 46: "[a-zA-Z0-9-_$#.]+"
- Pattern di stringa personalizzato n. 47: "^\d{12}$"
- Pattern di stringa personalizzato n. 48: "^(\w+\.)+\w+$"
- Pattern di stringa personalizzato n. 49: "^([2-3]|3[.]9)$"
- Pattern di stringa personalizzato n. 50: "arn:aws(-(cn|us-gov|eusc|iso(-[bef])?))?:glue:.*"
- Pattern di stringa personalizzato n. 51: "[0-9a-fA-F]{8}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{4}-[0-9a-fA-F]{12}"
- Pattern di stringa personalizzato n. 52: "(^arn:aws(-(cn|us-gov|eusc|iso(-[bef])?))?:iam::\w{12}:root)"
- Pattern di stringa personalizzato n. 53: "^arn:aws(-(cn|us-gov|eusc|iso(-[bef])?))?:iam::[0-9]{12}:role/.+"
- Pattern di stringa personalizzato n. 54: "[\s\S]*"
- Pattern di stringa personalizzato n. 55: "([\u0020-\uD7FF\uE000-\uFFFD\uD800\uDC00-\uDBFF\uDFFF]|[^\S\r\n"'= ;])*"
- Pattern di stringa personalizzato n. 56: "^[A-Z_]+$"
- Pattern di stringa personalizzato n. 57: "^[A-Za-z0-9]+$"
- Pattern di stringa personalizzato n. 58: "[*A-Za-z0-9_-]*"
- Pattern di stringa personalizzato n. 59: "([\u0020-\u007E\r\s\n])*"
- Pattern di stringa personalizzato n. 60: "[A-Za-z0-9_-]*"
- Pattern di stringa personalizzato n. 61: "([\u0009\u000B\u000C\u0020-\uD7FF\uE000-\uFFFD\uD800\uDC00-\uDBFF\uDFFF])*"
- Schema di stringhe personalizzato #62 — "([\u0020-\uD7FF\uE000-\uFFFD\uD800\uDC00-\uDBFF\uDFFF\s])*»
- Schema di stringhe personalizzato #63 — "([^\r\n])*»

# Eccezioni

Questa sezione descrive AWS Glue le eccezioni che è possibile utilizzare per individuare l'origine dei problemi e risolverli. Per ulteriori informazioni sui codici di errore HTTP e sulle stringhe per le eccezioni correlate al machine learning, consultare the section called “Eccezioni di machine learning AWS Glue”.

## AccessDeniedException struttura

L'accesso a una risorsa è stato rifiutato.

Campi

- `Message`: stringa UTF-8.

  Messaggio che descrive il problema.

## AlreadyExistsException struttura

Una risorsa da creare o aggiungere esiste già.

Campi

- `Message`: stringa UTF-8.

  Messaggio che descrive il problema.

## ConcurrentModificationException struttura

Due processi stanno tentando di modificare una risorsa contemporaneamente.

Campi

- `Message`: stringa UTF-8.

  Messaggio che descrive il problema.

## ConcurrentRunsExceededException struttura

Troppi processi vengono eseguiti simultaneamente.

Campi

- `Message`: stringa UTF-8.

  Messaggio che descrive il problema.

## CrawlerNotRunningException struttura

Il crawler specificato non è in esecuzione.

Campi

- `Message`: stringa UTF-8.

  Messaggio che descrive il problema.

## CrawlerRunningException struttura

L'operazione non è stata eseguita perché il crawler è già in esecuzione.

Campi

- `Message`: stringa UTF-8.

  Messaggio che descrive il problema.

## CrawlerStoppingException struttura

Il crawler specificato è in fase di arresto.

Campi

- `Message`: stringa UTF-8.

  Messaggio che descrive il problema.

## EntityNotFoundException struttura

Un'entità specificata non esiste.

Campi

- `Message`: stringa UTF-8.

  Messaggio che descrive il problema.
- `FromFederationSource`: booleano.

Indica se l'eccezione si riferisce o meno a un'origine federata.

## FederationSourceException struttura

Un'origine di federazione ha riscontrato un errore.

Campi

- `FederationSourceErrorCode`: stringa UTF-8 (valori validi: `AccessDeniedException` | `EntityNotFoundException` | `InvalidCredentialsException` | `InvalidInputException` | `InvalidResponseException` | `OperationTimeoutException` | `OperationNotSupportedException` | `InternalServiceException` | `PartialFailureException` | `ThrottlingException`).

  Il codice di errore del problema.
- `Message`: stringa UTF-8.

  Il messaggio che descrive il problema.

## FederationSourceRetryableException struttura

Un'origine di federazione ha riscontrato un errore, ma l'operazione può essere ritentata.

Campi

- `Message`: stringa UTF-8.

  Messaggio che descrive il problema.

## GlueEncryptionException struttura

Un'operazione di crittografia non è riuscita.

Campi

- `Message`: stringa UTF-8.

  Il messaggio che descrive il problema.

## IdempotentParameterMismatchException struttura

Lo stesso identificatore univoco è stato associato a due record diversi.

Campi

- `Message`: stringa UTF-8.

  Messaggio che descrive il problema.

## IllegalWorkflowStateException struttura

Il flusso di lavoro non è valido per eseguire un'operazione richiesta.

Campi

- `Message`: stringa UTF-8.

  Messaggio che descrive il problema.

## InternalServiceException struttura

Si è verificato un errore di servizio interno.

Campi

- `Message`: stringa UTF-8.

  Messaggio che descrive il problema.

## InvalidExecutionEngineException struttura

È stato specificato un motore di esecuzione sconosciuto o non valido.

Campi

- `message`: stringa UTF-8.

  Messaggio che descrive il problema.

## InvalidInputException struttura

L'input fornito non è valido.

Campi

- `Message`: stringa UTF-8.

  Messaggio che descrive il problema.
- `FromFederationSource`: booleano.

  Indica se l'eccezione si riferisce o meno a un'origine federata.

## InvalidStateException struttura

Un errore che indica che i dati sono in uno stato non valido.

Campi

- `Message`: stringa UTF-8.

  Messaggio che descrive il problema.

## InvalidTaskStatusTransitionException struttura

La corretta transizione da un'attività a quella successiva non è riuscita.

Campi

- `message`: stringa UTF-8.

  Messaggio che descrive il problema.

## JobDefinitionErrorException struttura

Una definizione di processo non è valida.

Campi

- `message`: stringa UTF-8.

Messaggio che descrive il problema.

## JobRunInTerminalStateException struttura

Lo stato terminale dell'esecuzione di un processo segnala un errore.

Campi

- `message`: stringa UTF-8.

  Messaggio che descrive il problema.

## JobRunInvalidStateTransitionException struttura

L'esecuzione di un processo ha riscontrato una transizione non valida dallo stato di origine allo stato di destinazione.

Campi

- `jobRunId`: stringa UTF-8, non inferiore a 1 o superiore a 255 byte di lunghezza, corrispondente a Single-line string pattern.

  L'ID dell'esecuzione del processo in questione.
- `message`: stringa UTF-8.

  Messaggio che descrive il problema.
- `sourceState`: stringa UTF-8 (valori validi: `STARTING` | `RUNNING` | `STOPPING` | `STOPPED` | `SUCCEEDED` | `FAILED` | `TIMEOUT` | `ERROR` | `WAITING` | `EXPIRED`).

  Lo stato di origine.
- `targetState`: stringa UTF-8 (valori validi: `STARTING` | `RUNNING` | `STOPPING` | `STOPPED` | `SUCCEEDED` | `FAILED` | `TIMEOUT` | `ERROR` | `WAITING` | `EXPIRED`).

  Lo stato di destinazione.

## JobRunNotInTerminalStateException struttura

Un'esecuzione di processo non è in uno stato terminale.

Campi

- `message`: stringa UTF-8.

  Messaggio che descrive il problema.

## LateRunnerException struttura

L'esecuzione di un processo è in ritardo.

Campi

- `Message`: stringa UTF-8.

  Messaggio che descrive il problema.

## NoScheduleException struttura

Non è presente una pianificazione applicabile.

Campi

- `Message`: stringa UTF-8.

  Messaggio che descrive il problema.

## OperationTimeoutException struttura

Timeout dell'operazione.

Campi

- `Message`: stringa UTF-8.

  Messaggio che descrive il problema.

## ResourceNotReadyException struttura

Una risorsa non era pronta per una transazione.

Campi

- `Message`: stringa UTF-8.

  Messaggio che descrive il problema.

## ResourceNumberLimitExceededException struttura

Il limite numerico di una risorsa è stato superato.

Campi

- `Message`: stringa UTF-8.

  Messaggio che descrive il problema.

## SchedulerNotRunningException struttura

Il pianificatore specificato non è in esecuzione.

Campi

- `Message`: stringa UTF-8.

  Messaggio che descrive il problema.

## SchedulerRunningException struttura

Il pianificatore specificato è già in esecuzione.

Campi

- `Message`: stringa UTF-8.

  Messaggio che descrive il problema.

## SchedulerTransitioningException struttura

Il pianificatore specificato è in transizione.

Campi

- `Message`: stringa UTF-8.

  Messaggio che descrive il problema.

## UnrecognizedRunnerException struttura

L'esecuzione del processo non è stata riconosciuta.

Campi

- `Message`: stringa UTF-8.

  Messaggio che descrive il problema.

## ValidationException struttura

Un valore non può essere convalidato.

Campi

- `Message`: stringa UTF-8.

  Messaggio che descrive il problema.

## VersionMismatchException struttura

Si è verificato un conflitto di versione.

Campi

- `message`: stringa UTF-8.

  Messaggio che descrive il problema.

# AWS Glue Esempi di codice API con AWS SDK

I seguenti esempi di codice mostrano come utilizzare un kit AWS Glue di sviluppo AWS software (SDK).

Nozioni di base: esempi di codice che mostrano come eseguire le operazioni essenziali all'interno di un servizio.

Le azioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le azioni mostrino come richiamare le singole funzioni del servizio, è possibile visualizzarle contestualizzate negli scenari correlati.

Scenari: esempi di codice che mostrano come eseguire un'attività specifica chiamando più funzioni all'interno dello stesso servizio o combinate con altri Servizi AWS.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

AWS Glue Esempi di codice API

- Esempi di base per l' AWS Glue utilizzo degli AWS SDK
  - ciao AWS Glue
  - Scopri le nozioni di base di AWS Glue un SDK AWS
  - Azioni per l' AWS Glue utilizzo degli AWS SDK
    - Utilizzo CreateCrawler con un AWS SDK
    - Utilizzo CreateJob con un AWS SDK o una CLI
    - Utilizzo DeleteCrawler con un AWS SDK
    - Utilizzo DeleteDatabase con un AWS SDK
    - Utilizzo DeleteJob con un AWS SDK o una CLI
    - Utilizzo DeleteTable con un AWS SDK
    - Utilizzo GetCrawler con un AWS SDK
    - Utilizzo GetDatabase con un AWS SDK
    - Utilizzo GetDatabases con un AWS SDK o una CLI
    - Utilizzo GetJob con un AWS SDK o una CLI
    - Utilizzo GetJobRun con un AWS SDK o una CLI

    - Utilizzo GetJobRuns con un AWS SDK o una CLI
    - Utilizzo GetTables con un AWS SDK o una CLI
    - Utilizzo ListJobs con un AWS SDK
    - Utilizzo StartCrawler con un AWS SDK o una CLI
    - Utilizzo StartJobRun con un AWS SDK o una CLI
- Scenari per l' AWS Glue utilizzo degli AWS SDK
    - Guida introduttiva al Glue Data Catalog

# Esempi di base per l' AWS Glue utilizzo degli AWS SDK

I seguenti esempi di codice mostrano come utilizzare le nozioni di base di AWS Glue with SDK. AWS

Esempi

- ciao AWS Glue
- Scopri le nozioni di base di AWS Glue un SDK AWS
- Azioni per l' AWS Glue utilizzo degli AWS SDK
    - Utilizzo CreateCrawler con un AWS SDK
    - Utilizzo CreateJob con un AWS SDK o una CLI
    - Utilizzo DeleteCrawler con un AWS SDK
    - Utilizzo DeleteDatabase con un AWS SDK
    - Utilizzo DeleteJob con un AWS SDK o una CLI
    - Utilizzo DeleteTable con un AWS SDK
    - Utilizzo GetCrawler con un AWS SDK
    - Utilizzo GetDatabase con un AWS SDK
    - Utilizzo GetDatabases con un AWS SDK o una CLI
    - Utilizzo GetJob con un AWS SDK o una CLI
    - Utilizzo GetJobRun con un AWS SDK o una CLI
    - Utilizzo GetJobRuns con un AWS SDK o una CLI
    - Utilizzo GetTables con un AWS SDK o una CLI
    - Utilizzo ListJobs con un AWS SDK
    - Utilizzo StartCrawler con un AWS SDK o una CLI

- Utilizzo StartJobRun con un AWS SDK o una CLI

## ciao AWS Glue

L'esempio di codice seguente mostra come iniziare a utilizzare AWS Glue.

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
namespace GlueActions;

public class HelloGlue
{
    private static ILogger logger = null!;

    static async Task Main(string[] args)
    {
        // Set up dependency injection for AWS Glue.
        using var host = Host.CreateDefaultBuilder(args)
            .ConfigureLogging(logging =>
                logging.AddFilter("System", LogLevel.Debug)
                    .AddFilter<DebugLoggerProvider>("Microsoft",
 LogLevel.Information)
                    .AddFilter<ConsoleLoggerProvider>("Microsoft",
 LogLevel.Trace))
            .ConfigureServices((_, services) =>
                services.AddAWSService<IAmazonGlue>()
                .AddTransient<GlueWrapper>()
            )
            .Build();

        logger = LoggerFactory.Create(builder => { builder.AddConsole(); })
            .CreateLogger<HelloGlue>();
        var glueClient = host.Services.GetRequiredService<IAmazonGlue>();
```

```
        var request = new ListJobsRequest();

        var jobNames = new List<string>();

        do
        {
            var response = await glueClient.ListJobsAsync(request);
            jobNames.AddRange(response.JobNames);
            request.NextToken = response.NextToken;
        }
        while (request.NextToken is not null);

        Console.Clear();
        Console.WriteLine("Hello, Glue. Let's list your existing Glue Jobs:");
        if (jobNames.Count == 0)
        {
            Console.WriteLine("You don't have any AWS Glue jobs.");
        }
        else
        {
            jobNames.ForEach(Console.WriteLine);
        }
    }
}
```

- Per i dettagli sull'API, consulta la ListJobssezione AWS SDK per .NET API Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Codice per il file CMakeLists.txt CMake.

```
# Set the minimum required version of CMake for this project.
cmake_minimum_required(VERSION 3.13)

# Set the AWS service components used by this project.
set(SERVICE_COMPONENTS glue)

# Set this project's name.
project("hello_glue")

# Set the C++ standard to use to build this target.
# At least C++ 11 is required for the AWS SDK for C++.
set(CMAKE_CXX_STANDARD 11)

# Use the MSVC variable to determine if this is a Windows build.
set(WINDOWS_BUILD ${MSVC})

if (WINDOWS_BUILD) # Set the location where CMake can find the installed
 libraries for the AWS SDK.
    string(REPLACE ";" "/aws-cpp-sdk-all;" SYSTEM_MODULE_PATH
 "${CMAKE_SYSTEM_PREFIX_PATH}/aws-cpp-sdk-all")
    list(APPEND CMAKE_PREFIX_PATH ${SYSTEM_MODULE_PATH})
endif ()

# Find the AWS SDK for C++ package.
find_package(AWSSDK REQUIRED COMPONENTS ${SERVICE_COMPONENTS})

if (WINDOWS_BUILD AND AWSSDK_INSTALL_AS_SHARED_LIBS)
     # Copy relevant AWS SDK for C++ libraries into the current binary directory
 for running and debugging.

     # set(BIN_SUB_DIR "/Debug") # if you are building from the command line you
 may need to uncomment this
                                     # and set the proper subdirectory to the
 executables' location.

     AWSSDK_CPY_DYN_LIBS(SERVICE_COMPONENTS ""
 ${CMAKE_CURRENT_BINARY_DIR}${BIN_SUB_DIR})
endif ()

add_executable(${PROJECT_NAME}
        hello_glue.cpp)

target_link_libraries(${PROJECT_NAME}
```

```
        ${AWSSDK_LINK_LIBRARIES})
```

Codice per il file di origine hello_glue.cpp.

```
#include <aws/core/Aws.h>
#include <aws/glue/GlueClient.h>
#include <aws/glue/model/ListJobsRequest.h>
#include <iostream>

/*
 *  A "Hello Glue" starter application which initializes an AWS Glue client and
 lists the
 *  AWS Glue job definitions.
 *
 *  main function
 *
 *  Usage: 'hello_glue'
 *
 */

int main(int argc, char **argv) {
    Aws::SDKOptions options;
    // Optionally change the log level for debugging.
//   options.loggingOptions.logLevel = Utils::Logging::LogLevel::Debug;
    Aws::InitAPI(options); // Should only be called once.
    int result = 0;
    {
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region (overrides config file).
        // clientConfig.region = "us-east-1";

        Aws::Glue::GlueClient glueClient(clientConfig);

        std::vector<Aws::String> jobs;

        Aws::String nextToken;  // Used for pagination.
        do {
            Aws::Glue::Model::ListJobsRequest listJobsRequest;
            if (!nextToken.empty()) {
                listJobsRequest.SetNextToken(nextToken);
            }
```

```
            Aws::Glue::Model::ListJobsOutcome listRunsOutcome =
 glueClient.ListJobs(
                    listJobsRequest);

            if (listRunsOutcome.IsSuccess()) {
                const std::vector<Aws::String> &jobNames =
 listRunsOutcome.GetResult().GetJobNames();
                jobs.insert(jobs.end(), jobNames.begin(), jobNames.end());

                nextToken = listRunsOutcome.GetResult().GetNextToken();
            } else {
                std::cerr << "Error listing jobs. "
                          << listRunsOutcome.GetError().GetMessage()
                          << std::endl;
                result = 1;
                break;
            }
        } while (!nextToken.empty());

        std::cout << "Your account has " << jobs.size() << " jobs."
                  << std::endl;
        for (size_t i = 0; i < jobs.size(); ++i) {
            std::cout << "   " << i + 1 << ". " << jobs[i] << std::endl;
        }
    }
    Aws::ShutdownAPI(options); // Should only be called once.
    return result;
}
```

- Per i dettagli sull'API, vedi ListJobsin AWS SDK per C++ API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
package com.example.glue;

import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.glue.GlueClient;
import software.amazon.awssdk.services.glue.model.ListJobsRequest;
import software.amazon.awssdk.services.glue.model.ListJobsResponse;
import java.util.List;

public class HelloGlue {
    public static void main(String[] args) {
        GlueClient glueClient = GlueClient.builder()
                .region(Region.US_EAST_1)
                .build();

        listJobs(glueClient);
    }

    public static void listJobs(GlueClient glueClient) {
        ListJobsRequest request = ListJobsRequest.builder()
                .maxResults(10)
                .build();
        ListJobsResponse response = glueClient.listJobs(request);
        List<String> jobList = response.jobNames();
        jobList.forEach(job -> {
            System.out.println("Job Name: " + job);
        });
    }
}
```

- Per i dettagli sull'API, consulta la ListJobssezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import { ListJobsCommand, GlueClient } from "@aws-sdk/client-glue";

const client = new GlueClient({});

export const main = async () => {
  const command = new ListJobsCommand({});

  const { JobNames } = await client.send(command);
  const formattedJobNames = JobNames.join("\n");
  console.log("Job names: ");
  console.log(formattedJobNames);
  return JobNames;
};
```

- Per i dettagli sull'API, consulta la ListJobssezione AWS SDK per JavaScript API Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import boto3
from botocore.exceptions import ClientError


def hello_glue():
    """
    Lists the job definitions in your AWS Glue account, using the AWS SDK for
 Python (Boto3).
    """
    try:
        # Create the Glue client
        glue = boto3.client("glue")
```

```
        # List the jobs, limiting the results to 10 per page
        paginator = glue.get_paginator("get_jobs")
        response_iterator = paginator.paginate(
            PaginationConfig={"MaxItems": 10, "PageSize": 10}
        )

        # Print the job names
        print("Here are the jobs in your account:")
        for page in response_iterator:
            for job in page["Jobs"]:
                print(f"\t{job['Name']}")

    except ClientError as e:
        print(f"Error: {e}")


if __name__ == "__main__":
    hello_glue()
```

- Per i dettagli sull'API, consulta ListJobs AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
require 'aws-sdk-glue'
require 'logger'

# GlueManager is a class responsible for managing AWS Glue operations
# such as listing all Glue jobs in the current AWS account.
class GlueManager
  def initialize(client)
    @client = client
```

```
    @logger = Logger.new($stdout)
  end

  # Lists and prints all Glue jobs in the current AWS account.
  def list_jobs
    @logger.info('Here are the Glue jobs in your account:')

    paginator = @client.get_jobs(max_results: 10)
    jobs = []

    paginator.each_page do |page|
      jobs.concat(page.jobs)
    end

    if jobs.empty?
      @logger.info("You don't have any Glue jobs.")
    else
      jobs.each do |job|
        @logger.info("- #{job.name}")
      end
    end
  end
end

if $PROGRAM_NAME == __FILE__
  glue_client = Aws::Glue::Client.new
  manager = GlueManager.new(glue_client)
  manager.list_jobs
end
```

- Per i dettagli sull'API, consulta la ListJobssezione AWS SDK per Ruby API Reference.

## Rust

### SDK per Rust

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
        let mut list_jobs = glue.list_jobs().into_paginator().send();
        while let Some(list_jobs_output) = list_jobs.next().await {
            match list_jobs_output {
                Ok(list_jobs) => {
                    let names = list_jobs.job_names();
                    info!(?names, "Found these jobs")
                }
                Err(err) => return Err(GlueMvpError::from_glue_sdk(err)),
            }
        }
```

- Per i dettagli sulle API, consulta la ListJobsguida di riferimento all'API AWS SDK for Rust.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

## Scopri le nozioni di base di AWS Glue un SDK AWS

Gli esempi di codice seguenti mostrano come:

- Crea un crawler che esegua la scansione di un bucket Amazon S3 pubblico e generi un database di metadati. CSV-formatted
- Elenca le informazioni su database e tabelle nel tuo. AWS Glue Data Catalog
- Crea un processo per estrarre i dati CSV dal bucket S3, trasformare i dati e caricare l' JSON-formatted output in un altro bucket S3.
- Elenca le informazioni sulle esecuzioni dei processi, visualizza i dati trasformati e pulisci le risorse.

Per ulteriori informazioni, consulta Tutorial: Guida introduttiva a AWS Glue Studio.

.NET

SDK per .NET

> **Note**
>
> C'è altro da sapere GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Crea una classe che racchiuda le AWS Glue funzioni utilizzate nello scenario.

```
using System.Net;

namespace GlueActions;

public class GlueWrapper
{
    private readonly IAmazonGlue _amazonGlue;

    /// <summary>
    /// Constructor for the AWS Glue actions wrapper.
    /// </summary>
    /// <param name="amazonGlue"></param>
    public GlueWrapper(IAmazonGlue amazonGlue)
    {
        _amazonGlue = amazonGlue;
    }

    /// <summary>
    /// Create an AWS Glue crawler.
    /// </summary>
    /// <param name="crawlerName">The name for the crawler.</param>
    /// <param name="crawlerDescription">A description of the crawler.</param>
    /// <param name="role">The AWS Identity and Access Management (IAM) role to
    /// be assumed by the crawler.</param>
    /// <param name="schedule">The schedule on which the crawler will be
 executed.</param>
    /// <param name="s3Path">The path to the Amazon Simple Storage Service
 (Amazon S3)
    /// bucket where the Python script has been stored.</param>
    /// <param name="dbName">The name to use for the database that will be
```

```
    /// created by the crawler.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> CreateCrawlerAsync(
        string crawlerName,
        string crawlerDescription,
        string role,
        string schedule,
        string s3Path,
        string dbName)
    {
        var s3Target = new S3Target
        {
            Path = s3Path,
        };

        var targetList = new List<S3Target>
        {
            s3Target,
        };

        var targets = new CrawlerTargets
        {
            S3Targets = targetList,
        };

        var crawlerRequest = new CreateCrawlerRequest
        {
            DatabaseName = dbName,
            Name = crawlerName,
            Description = crawlerDescription,
            Targets = targets,
            Role = role,
            Schedule = schedule,
        };

        var response = await _amazonGlue.CreateCrawlerAsync(crawlerRequest);
        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }


    /// <summary>
    /// Create an AWS Glue job.
    /// </summary>
    /// <param name="jobName">The name of the job.</param>
```

```
    /// <param name="roleName">The name of the IAM role to be assumed by
    /// the job.</param>
    /// <param name="description">A description of the job.</param>
    /// <param name="scriptUrl">The URL to the script.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> CreateJobAsync(string dbName, string tableName,
string bucketUrl, string jobName, string roleName, string description, string
scriptUrl)
    {
        var command = new JobCommand
        {
            PythonVersion = "3",
            Name = "glueetl",
            ScriptLocation = scriptUrl,
        };

        var arguments = new Dictionary<string, string>
        {
            { "--input_database", dbName },
            { "--input_table", tableName },
            { "--output_bucket_url", bucketUrl }
        };

        var request = new CreateJobRequest
        {
            Command = command,
            DefaultArguments = arguments,
            Description = description,
            GlueVersion = "3.0",
            Name = jobName,
            NumberOfWorkers = 10,
            Role = roleName,
            WorkerType = "G.1X"
        };

        var response = await _amazonGlue.CreateJobAsync(request);
        return response.HttpStatusCode == HttpStatusCode.OK;
    }


    /// <summary>
    /// Delete an AWS Glue crawler.
    /// </summary>
    /// <param name="crawlerName">The name of the crawler.</param>
```

```
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeleteCrawlerAsync(string crawlerName)
    {
        var response = await _amazonGlue.DeleteCrawlerAsync(new
DeleteCrawlerRequest { Name = crawlerName });
        return response.HttpStatusCode == HttpStatusCode.OK;
    }


    /// <summary>
    /// Delete the AWS Glue database.
    /// </summary>
    /// <param name="dbName">The name of the database.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeleteDatabaseAsync(string dbName)
    {
        var response = await _amazonGlue.DeleteDatabaseAsync(new
DeleteDatabaseRequest { Name = dbName });
        return response.HttpStatusCode == HttpStatusCode.OK;
    }


    /// <summary>
    /// Delete an AWS Glue job.
    /// </summary>
    /// <param name="jobName">The name of the job.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeleteJobAsync(string jobName)
    {
        var response = await _amazonGlue.DeleteJobAsync(new DeleteJobRequest
{ JobName = jobName });
        return response.HttpStatusCode == HttpStatusCode.OK;
    }


    /// <summary>
    /// Delete a table from an AWS Glue database.
    /// </summary>
    /// <param name="tableName">The table to delete.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeleteTableAsync(string dbName, string tableName)
    {
        var response = await _amazonGlue.DeleteTableAsync(new DeleteTableRequest
{ Name = tableName, DatabaseName = dbName });
```

```
        return response.HttpStatusCode == HttpStatusCode.OK;
    }


    /// <summary>
    /// Get information about an AWS Glue crawler.
    /// </summary>
    /// <param name="crawlerName">The name of the crawler.</param>
    /// <returns>A Crawler object describing the crawler.</returns>
    public async Task<Crawler?> GetCrawlerAsync(string crawlerName)
    {
        var crawlerRequest = new GetCrawlerRequest
        {
            Name = crawlerName,
        };

        var response = await _amazonGlue.GetCrawlerAsync(crawlerRequest);
        if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
        {
            var databaseName = response.Crawler.DatabaseName;
            Console.WriteLine($"{crawlerName} has the database {databaseName}");
            return response.Crawler;
        }

        Console.WriteLine($"No information regarding {crawlerName} could be
found.");
        return null;
    }


    /// <summary>
    /// Get information about the state of an AWS Glue crawler.
    /// </summary>
    /// <param name="crawlerName">The name of the crawler.</param>
    /// <returns>A value describing the state of the crawler.</returns>
    public async Task<CrawlerState> GetCrawlerStateAsync(string crawlerName)
    {
        var response = await _amazonGlue.GetCrawlerAsync(
            new GetCrawlerRequest { Name = crawlerName });
        return response.Crawler.State;
    }


    /// <summary>
```

```
    /// Get information about an AWS Glue database.
    /// </summary>
    /// <param name="dbName">The name of the database.</param>
    /// <returns>A Database object containing information about the database.</
returns>
    public async Task<Database> GetDatabaseAsync(string dbName)
    {
        var databasesRequest = new GetDatabaseRequest
        {
            Name = dbName,
        };

        var response = await _amazonGlue.GetDatabaseAsync(databasesRequest);
        return response.Database;
    }


    /// <summary>
    /// Get information about a specific AWS Glue job run.
    /// </summary>
    /// <param name="jobName">The name of the job.</param>
    /// <param name="jobRunId">The Id of the job run.</param>
    /// <returns>A JobRun object with information about the job run.</returns>
    public async Task<JobRun> GetJobRunAsync(string jobName, string jobRunId)
    {
        var response = await _amazonGlue.GetJobRunAsync(new GetJobRunRequest
 { JobName = jobName, RunId = jobRunId });
        return response.JobRun;
    }


    /// <summary>
    /// Get information about all AWS Glue runs of a specific job.
    /// </summary>
    /// <param name="jobName">The name of the job.</param>
    /// <returns>A list of JobRun objects.</returns>
    public async Task<List<JobRun>> GetJobRunsAsync(string jobName)
    {
        var jobRuns = new List<JobRun>();

        var request = new GetJobRunsRequest
        {
            JobName = jobName,
        };
```

```
        // No need to loop to get all the log groups--the SDK does it for us
behind the scenes
        var paginatorForJobRuns =
            _amazonGlue.Paginators.GetJobRuns(request);

        await foreach (var response in paginatorForJobRuns.Responses)
        {
            response.JobRuns.ForEach(jobRun =>
            {
                jobRuns.Add(jobRun);
            });
        }

        return jobRuns;
    }


    /// <summary>
    /// Get a list of tables for an AWS Glue database.
    /// </summary>
    /// <param name="dbName">The name of the database.</param>
    /// <returns>A list of Table objects.</returns>
    public async Task<List<Table>> GetTablesAsync(string dbName)
    {
        var request = new GetTablesRequest { DatabaseName = dbName };
        var tables = new List<Table>();

        // Get a paginator for listing the tables.
        var tablePaginator = _amazonGlue.Paginators.GetTables(request);

        await foreach (var response in tablePaginator.Responses)
        {
            tables.AddRange(response.TableList);
        }

        return tables;
    }


    /// <summary>
    /// List AWS Glue jobs using a paginator.
    /// </summary>
    /// <returns>A list of AWS Glue job names.</returns>
```

```
    public async Task<List<string>> ListJobsAsync()
    {
        var jobNames = new List<string>();

        var listJobsPaginator = _amazonGlue.Paginators.ListJobs(new
ListJobsRequest { MaxResults = 10 });
        await foreach (var response in listJobsPaginator.Responses)
        {
            jobNames.AddRange(response.JobNames);
        }

        return jobNames;
    }


    /// <summary>
    /// Start an AWS Glue crawler.
    /// </summary>
    /// <param name="crawlerName">The name of the crawler.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> StartCrawlerAsync(string crawlerName)
    {
        var crawlerRequest = new StartCrawlerRequest
        {
            Name = crawlerName,
        };

        var response = await _amazonGlue.StartCrawlerAsync(crawlerRequest);

        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }


    /// <summary>
    /// Start an AWS Glue job run.
    /// </summary>
    /// <param name="jobName">The name of the job.</param>
    /// <returns>A string representing the job run Id.</returns>
    public async Task<string> StartJobRunAsync(
        string jobName,
        string inputDatabase,
        string inputTable,
        string bucketName)
    {
```

```
        var request = new StartJobRunRequest
        {
            JobName = jobName,
            Arguments = new Dictionary<string, string>
            {
                {"--input_database", inputDatabase},
                {"--input_table", inputTable},
                {"--output_bucket_url", $"s3://{bucketName}/"}
            }
        };

        var response = await _amazonGlue.StartJobRunAsync(request);
        return response.JobRunId;
    }

}
```

Creazione di una classe che esegue lo scenario.

```
global using Amazon.Glue;
global using GlueActions;
global using Microsoft.Extensions.Configuration;
global using Microsoft.Extensions.DependencyInjection;
global using Microsoft.Extensions.Hosting;
global using Microsoft.Extensions.Logging;
global using Microsoft.Extensions.Logging.Console;
global using Microsoft.Extensions.Logging.Debug;



using Amazon.Glue.Model;
using Amazon.S3;
using Amazon.S3.Model;

namespace GlueBasics;

public class GlueBasics
{
    private static ILogger logger = null!;
    private static IConfiguration _configuration = null!;
```

```
   static async Task Main(string[] args)
   {
       // Set up dependency injection for AWS Glue.
       using var host = Host.CreateDefaultBuilder(args)
           .ConfigureLogging(logging =>
               logging.AddFilter("System", LogLevel.Debug)
                   .AddFilter<DebugLoggerProvider>("Microsoft",
LogLevel.Information)
                   .AddFilter<ConsoleLoggerProvider>("Microsoft",
LogLevel.Trace))
           .ConfigureServices((_, services) =>
           services.AddAWSService<IAmazonGlue>()
           .AddTransient<GlueWrapper>()
           .AddTransient<UiWrapper>()
           )
           .Build();

       logger = LoggerFactory.Create(builder => { builder.AddConsole(); })
       .CreateLogger<GlueBasics>();

       _configuration = new ConfigurationBuilder()
           .SetBasePath(Directory.GetCurrentDirectory())
           .AddJsonFile("settings.json") // Load settings from .json file.
           .AddJsonFile("settings.local.json",
               true) // Optionally load local settings.
           .Build();

       // These values are stored in settings.json
       // Once you have run the CDK script to deploy the resources,
       // edit the file to set "BucketName", "RoleName", and "ScriptURL"
       // to the appropriate values. Also set "CrawlerName" to the name
       // you want to give the crawler when it is created.
       string bucketName = _configuration["BucketName"]!;
       string bucketUrl = _configuration["BucketUrl"]!;
       string crawlerName = _configuration["CrawlerName"]!;
       string roleName = _configuration["RoleName"]!;
       string sourceData = _configuration["SourceData"]!;
       string dbName = _configuration["DbName"]!;
       string cron = _configuration["Cron"]!;
       string scriptUrl = _configuration["ScriptURL"]!;
       string jobName = _configuration["JobName"]!;

       var wrapper = host.Services.GetRequiredService<GlueWrapper>();
       var uiWrapper = host.Services.GetRequiredService<UiWrapper>();
```

```
        uiWrapper.DisplayOverview();
        uiWrapper.PressEnter();

        // Create the crawler and wait for it to be ready.
        uiWrapper.DisplayTitle("Create AWS Glue crawler");
        Console.WriteLine("Let's begin by creating the AWS Glue crawler.");

        var crawlerDescription = "Crawler created for the AWS Glue Basics
scenario.";
        var crawlerCreated = await wrapper.CreateCrawlerAsync(crawlerName,
crawlerDescription, roleName, cron, sourceData, dbName);
        if (crawlerCreated)
        {
            Console.WriteLine($"The crawler: {crawlerName} has been created. Now
let's wait until it's ready.");
            CrawlerState crawlerState;
            do
            {
                crawlerState = await wrapper.GetCrawlerStateAsync(crawlerName);
            }
            while (crawlerState != "READY");
            Console.WriteLine($"The crawler {crawlerName} is now ready for
use.");
        }
        else
        {
            Console.WriteLine($"Couldn't create crawler {crawlerName}.");
            return; // Exit the application.
        }

        uiWrapper.DisplayTitle("Start AWS Glue crawler");
        Console.WriteLine("Now let's wait until the crawler has successfully
started.");
        var crawlerStarted = await wrapper.StartCrawlerAsync(crawlerName);
        if (crawlerStarted)
        {
            CrawlerState crawlerState;
            do
            {
                crawlerState = await wrapper.GetCrawlerStateAsync(crawlerName);
            }
            while (crawlerState != "READY");
```

```
            Console.WriteLine($"The crawler {crawlerName} is now ready for
use.");
        }
        else
        {
            Console.WriteLine($"Couldn't start the crawler {crawlerName}.");
            return; // Exit the application.
        }

        uiWrapper.PressEnter();

        Console.WriteLine($"\nLet's take a look at the database: {dbName}");
        var database = await wrapper.GetDatabaseAsync(dbName);

        if (database != null)
        {
            uiWrapper.DisplayTitle($"{database.Name} Details");
            Console.WriteLine($"{database.Name} created on
{database.CreateTime}");
            Console.WriteLine(database.Description);
        }

        uiWrapper.PressEnter();

        var tables = await wrapper.GetTablesAsync(dbName);
        if (tables.Count > 0)
        {
            tables.ForEach(table =>
            {
                Console.WriteLine($"{table.Name}\tCreated:
{table.CreateTime}\tUpdated: {table.UpdateTime}");
            });
        }

        uiWrapper.PressEnter();

        uiWrapper.DisplayTitle("Create AWS Glue job");
        Console.WriteLine("Creating a new AWS Glue job.");
        var description = "An AWS Glue job created using the AWS SDK for .NET";
        await wrapper.CreateJobAsync(dbName, tables[0].Name, bucketUrl, jobName,
roleName, description, scriptUrl);

        uiWrapper.PressEnter();
```

```
        uiWrapper.DisplayTitle("Starting AWS Glue job");
        Console.WriteLine("Starting the new AWS Glue job...");
        var jobRunId = await wrapper.StartJobRunAsync(jobName, dbName,
tables[0].Name, bucketName);
        var jobRunComplete = false;
        var jobRun = new JobRun();
        do
        {
            jobRun = await wrapper.GetJobRunAsync(jobName, jobRunId);
            if (jobRun.JobRunState == "SUCCEEDED" || jobRun.JobRunState ==
"STOPPED" ||
                jobRun.JobRunState == "FAILED" || jobRun.JobRunState ==
"TIMEOUT")
            {
                jobRunComplete = true;
            }
        } while (!jobRunComplete);

        uiWrapper.DisplayTitle($"Data in {bucketName}");

        // Get the list of data stored in the S3 bucket.
        var s3Client = new AmazonS3Client();

        var response = await s3Client.ListObjectsAsync(new ListObjectsRequest
{ BucketName = bucketName });
        response.S3Objects.ForEach(s3Object =>
        {
            Console.WriteLine(s3Object.Key);
        });

        uiWrapper.DisplayTitle("AWS Glue jobs");
        var jobNames = await wrapper.ListJobsAsync();
        jobNames.ForEach(jobName =>
        {
            Console.WriteLine(jobName);
        });

        uiWrapper.PressEnter();

        uiWrapper.DisplayTitle("Get AWS Glue job run information");
        Console.WriteLine("Getting information about the AWS Glue job.");
        var jobRuns = await wrapper.GetJobRunsAsync(jobName);

        jobRuns.ForEach(jobRun =>
```

```
        {

 Console.WriteLine($"{jobRun.JobName}\t{jobRun.JobRunState}\t{jobRun.CompletedOn}");
        });

        uiWrapper.PressEnter();

        uiWrapper.DisplayTitle("Deleting resources");
        Console.WriteLine("Deleting the AWS Glue job used by the example.");
        await wrapper.DeleteJobAsync(jobName);

        Console.WriteLine("Deleting the tables from the database.");
        tables.ForEach(async table =>
        {
            await wrapper.DeleteTableAsync(dbName, table.Name);
        });

        Console.WriteLine("Deleting the database.");
        await wrapper.DeleteDatabaseAsync(dbName);

        Console.WriteLine("Deleting the AWS Glue crawler.");
        await wrapper.DeleteCrawlerAsync(crawlerName);

        Console.WriteLine("The AWS Glue scenario has completed.");
        uiWrapper.PressEnter();
    }
}


namespace GlueBasics;

public class UiWrapper
{
    public readonly string SepBar = new string('-', Console.WindowWidth);

    /// <summary>
    /// Show information about the scenario.
    /// </summary>
    public void DisplayOverview()
    {
        Console.Clear();
        DisplayTitle("Amazon Glue: get started with crawlers and jobs");

        Console.WriteLine("This example application does the following:");
```

```
        Console.WriteLine("\t 1. Create a crawler, pass it the IAM role and the
URL to the public S3 bucket that contains the source data");
        Console.WriteLine("\t 2. Start the crawler.");
        Console.WriteLine("\t 3. Get the database created by the crawler and the
tables in the database.");
        Console.WriteLine("\t 4. Create a job.");
        Console.WriteLine("\t 5. Start a job run.");
        Console.WriteLine("\t 6. Wait for the job run to complete.");
        Console.WriteLine("\t 7. Show the data stored in the bucket.");
        Console.WriteLine("\t 8. List jobs for the account.");
        Console.WriteLine("\t 9. Get job run details for the job that was run.");
        Console.WriteLine("\t10. Delete the demo job.");
        Console.WriteLine("\t11. Delete the database and tables created for the
demo.");
        Console.WriteLine("\t12. Delete the crawler.");
    }

    /// <summary>
    /// Display a message and wait until the user presses enter.
    /// </summary>
    public void PressEnter()
    {
        Console.Write("\nPlease press <Enter> to continue. ");
        _ = Console.ReadLine();
    }

    /// <summary>
    /// Pad a string with spaces to center it on the console display.
    /// </summary>
    /// <param name="strToCenter">The string to center on the screen.</param>
    /// <returns>The string padded to make it center on the screen.</returns>
    public string CenterString(string strToCenter)
    {
        var padAmount = (Console.WindowWidth - strToCenter.Length) / 2;
        var leftPad = new string(' ', padAmount);
        return $"{leftPad}{strToCenter}";
    }

    /// <summary>
    /// Display a line of hyphens, the centered text of the title and another
    /// line of hyphens.
    /// </summary>
    /// <param name="strTitle">The string to be displayed.</param>
    public void DisplayTitle(string strTitle)
```

```
    {
        Console.WriteLine(SepBar);
        Console.WriteLine(CenterString(strTitle));
        Console.WriteLine(SepBar);
    }
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API AWS SDK per .NET .
  - CreateCrawler
  - CreateJob
  - DeleteCrawler
  - DeleteDatabase
  - DeleteJob
  - DeleteTable
  - GetCrawler
  - GetDatabase
  - GetDatabases
  - GetJob
  - GetJobRun
  - GetJobRuns
  - GetTables
  - ListJobs
  - StartCrawler
  - StartJobRun

C++

SDK per C++

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
//! Scenario which demonstrates using AWS Glue to add a crawler and run a job.
/*!
 \\sa runGettingStartedWithGlueScenario()
 \param bucketName: An S3 bucket created in the setup.
 \param roleName: An AWS Identity and Access Management (IAM) role created in the
 setup.
 \param clientConfig: AWS client configuration.
 \return bool: Successful completion.
 */

bool AwsDoc::Glue::runGettingStartedWithGlueScenario(const Aws::String
 &bucketName,
                                                      const Aws::String &roleName,
                                                      const
 Aws::Client::ClientConfiguration &clientConfig) {
    Aws::Glue::GlueClient client(clientConfig);

    Aws::String roleArn;
    if (!getRoleArn(roleName, roleArn, clientConfig)) {
        std::cerr << "Error getting role ARN for role." << std::endl;
        return false;
    }

    // 1. Upload the job script to the S3 bucket.
    {
        std::cout << "Uploading the job script '"
                  << AwsDoc::Glue::PYTHON_SCRIPT
                  << "'." << std::endl;

        if (!AwsDoc::Glue::uploadFile(bucketName,
                                      AwsDoc::Glue::PYTHON_SCRIPT_PATH,
                                      AwsDoc::Glue::PYTHON_SCRIPT,
```

```
                                        clientConfig)) {
            std::cerr << "Error uploading the job file." << std::endl;
            return false;
        }
    }

    // 2. Create a crawler.
    {
        Aws::Glue::Model::S3Target s3Target;
        s3Target.SetPath("s3://crawler-public-us-east-1/flight/2016/csv");
        Aws::Glue::Model::CrawlerTargets crawlerTargets;
        crawlerTargets.AddS3Targets(s3Target);

        Aws::Glue::Model::CreateCrawlerRequest request;
        request.SetTargets(crawlerTargets);
        request.SetName(CRAWLER_NAME);
        request.SetDatabaseName(CRAWLER_DATABASE_NAME);
        request.SetTablePrefix(CRAWLER_DATABASE_PREFIX);
        request.SetRole(roleArn);

        Aws::Glue::Model::CreateCrawlerOutcome outcome =
client.CreateCrawler(request);

        if (outcome.IsSuccess()) {
            std::cout << "Successfully created the crawler." << std::endl;
        }
        else {
            std::cerr << "Error creating a crawler. " <<
outcome.GetError().GetMessage()
                      << std::endl;
            deleteAssets("", CRAWLER_DATABASE_NAME, "", bucketName,
clientConfig);
            return false;
        }
    }

    // 3. Get a crawler.
    {
        Aws::Glue::Model::GetCrawlerRequest request;
        request.SetName(CRAWLER_NAME);

        Aws::Glue::Model::GetCrawlerOutcome outcome = client.GetCrawler(request);

        if (outcome.IsSuccess()) {
```

```
            Aws::Glue::Model::CrawlerState crawlerState =
outcome.GetResult().GetCrawler().GetState();
            std::cout << "Retrieved crawler with state " <<

Aws::Glue::Model::CrawlerStateMapper::GetNameForCrawlerState(
                              crawlerState)
                      << "." << std::endl;
        }
        else {
            std::cerr << "Error retrieving a crawler.  "
                      << outcome.GetError().GetMessage() << std::endl;
            deleteAssets(CRAWLER_NAME, CRAWLER_DATABASE_NAME, "", bucketName,
                         clientConfig);
            return false;
        }
    }

    // 4. Start a crawler.
    {
        Aws::Glue::Model::StartCrawlerRequest request;
        request.SetName(CRAWLER_NAME);

        Aws::Glue::Model::StartCrawlerOutcome outcome =
client.StartCrawler(request);


        if (outcome.IsSuccess() || (Aws::Glue::GlueErrors::CRAWLER_RUNNING ==
                                    outcome.GetError().GetErrorType())) {
            if (!outcome.IsSuccess()) {
                std::cout << "Crawler was already started." << std::endl;
            }
            else {
                std::cout << "Successfully started crawler." << std::endl;
            }

            std::cout << "This may take a while to run." << std::endl;

            Aws::Glue::Model::CrawlerState crawlerState =
Aws::Glue::Model::CrawlerState::NOT_SET;
            int iterations = 0;
            while (Aws::Glue::Model::CrawlerState::READY != crawlerState) {
                std::this_thread::sleep_for(std::chrono::seconds(1));
                ++iterations;
                if ((iterations % 10) == 0) { // Log status every 10 seconds.
```

```
                    std::cout << "Crawler status " <<

Aws::Glue::Model::CrawlerStateMapper::GetNameForCrawlerState(
                                       crawlerState)
                              << ". After " << iterations
                              << " seconds elapsed."
                              << std::endl;
                }
                Aws::Glue::Model::GetCrawlerRequest getCrawlerRequest;
                getCrawlerRequest.SetName(CRAWLER_NAME);

                Aws::Glue::Model::GetCrawlerOutcome getCrawlerOutcome =
client.GetCrawler(
                        getCrawlerRequest);

                if (getCrawlerOutcome.IsSuccess()) {
                    crawlerState =
getCrawlerOutcome.GetResult().GetCrawler().GetState();
                }
                else {
                    std::cerr << "Error getting crawler.  "
                              << getCrawlerOutcome.GetError().GetMessage() <<
std::endl;
                    break;
                }
            }

            if (Aws::Glue::Model::CrawlerState::READY == crawlerState) {
                std::cout << "Crawler finished running after " << iterations
                          << " seconds."
                          << std::endl;
            }
        }
        else {
            std::cerr << "Error starting a crawler.  "
                      << outcome.GetError().GetMessage()
                      << std::endl;

            deleteAssets(CRAWLER_NAME, CRAWLER_DATABASE_NAME, "", bucketName,
                         clientConfig);
            return false;
        }
    }
```

```
   // 5. Get a database.
   {
       Aws::Glue::Model::GetDatabaseRequest request;
       request.SetName(CRAWLER_DATABASE_NAME);

       Aws::Glue::Model::GetDatabaseOutcome outcome =
client.GetDatabase(request);

       if (outcome.IsSuccess()) {
           const Aws::Glue::Model::Database &database =
outcome.GetResult().GetDatabase();

           std::cout << "Successfully retrieve the database\n" <<
                     database.Jsonize().View().WriteReadable() << "'." <<
std::endl;
       }
       else {
           std::cerr << "Error getting the database.  "
                     << outcome.GetError().GetMessage() << std::endl;
           deleteAssets(CRAWLER_NAME, CRAWLER_DATABASE_NAME, "", bucketName,
                        clientConfig);
           return false;
       }
   }

   // 6. Get tables.
   Aws::String tableName;
   {
       Aws::Glue::Model::GetTablesRequest request;
       request.SetDatabaseName(CRAWLER_DATABASE_NAME);
       std::vector<Aws::Glue::Model::Table> all_tables;
       Aws::String nextToken; // Used for pagination.
       do {
           Aws::Glue::Model::GetTablesOutcome outcome =
client.GetTables(request);

           if (outcome.IsSuccess()) {
               const std::vector<Aws::Glue::Model::Table> &tables =
outcome.GetResult().GetTableList();
               all_tables.insert(all_tables.end(), tables.begin(),
tables.end());
               nextToken = outcome.GetResult().GetNextToken();
           }
           else {
```

```
                std::cerr << "Error getting the tables. "
                          << outcome.GetError().GetMessage()
                          << std::endl;
                deleteAssets(CRAWLER_NAME, CRAWLER_DATABASE_NAME, "", bucketName,
                             clientConfig);
                return false;
            }
        } while (!nextToken.empty());

        std::cout << "The database contains " << all_tables.size()
                  << (all_tables.size() == 1 ?
                      " table." : "tables.") << std::endl;
        std::cout << "Here is a list of the tables in the database.";
        for (size_t index = 0; index < all_tables.size(); ++index) {
            std::cout << "    " << index + 1 << ":  " <<
all_tables[index].GetName()
                      << std::endl;
        }

        if (!all_tables.empty()) {
            int tableIndex = askQuestionForIntRange(
                    "Enter an index to display the database detail ",
                    1, static_cast<int>(all_tables.size()));
            std::cout << all_tables[tableIndex -
1].Jsonize().View().WriteReadable()
                      << std::endl;

            tableName = all_tables[tableIndex - 1].GetName();
        }
    }

    // 7. Create a job.
    {
        Aws::Glue::Model::CreateJobRequest request;
        request.SetName(JOB_NAME);
        request.SetRole(roleArn);
        request.SetGlueVersion(GLUE_VERSION);

        Aws::Glue::Model::JobCommand command;
        command.SetName(JOB_COMMAND_NAME);
        command.SetPythonVersion(JOB_PYTHON_VERSION);
        command.SetScriptLocation(
                Aws::String("s3://") + bucketName + "/" + PYTHON_SCRIPT);
        request.SetCommand(command);
```

```
        Aws::Glue::Model::CreateJobOutcome outcome = client.CreateJob(request);

        if (outcome.IsSuccess()) {
            std::cout << "Successfully created the job." << std::endl;
        }
        else {
            std::cerr << "Error creating the job. " <<
outcome.GetError().GetMessage()
                      << std::endl;
            deleteAssets(CRAWLER_NAME, CRAWLER_DATABASE_NAME, "", bucketName,
                         clientConfig);
            return false;
        }
    }

    // 8. Start a job run.
    {
        Aws::Glue::Model::StartJobRunRequest request;
        request.SetJobName(JOB_NAME);

        Aws::Map<Aws::String, Aws::String> arguments;
        arguments["--input_database"] = CRAWLER_DATABASE_NAME;
        arguments["--input_table"] = tableName;
        arguments["--output_bucket_url"] = Aws::String("s3://") + bucketName +
"/";
        request.SetArguments(arguments);

        Aws::Glue::Model::StartJobRunOutcome outcome =
client.StartJobRun(request);

        if (outcome.IsSuccess()) {
            std::cout << "Successfully started the job." << std::endl;

            Aws::String jobRunId = outcome.GetResult().GetJobRunId();

            int iterator = 0;
            bool done = false;
            while (!done) {
                ++iterator;
                std::this_thread::sleep_for(std::chrono::seconds(1));
                Aws::Glue::Model::GetJobRunRequest jobRunRequest;
                jobRunRequest.SetJobName(JOB_NAME);
                jobRunRequest.SetRunId(jobRunId);
```

```
                Aws::Glue::Model::GetJobRunOutcome jobRunOutcome =
client.GetJobRun(
                        jobRunRequest);

                if (jobRunOutcome.IsSuccess()) {
                    const Aws::Glue::Model::JobRun &jobRun =
jobRunOutcome.GetResult().GetJobRun();
                    Aws::Glue::Model::JobRunState jobRunState =
jobRun.GetJobRunState();

                    if ((jobRunState == Aws::Glue::Model::JobRunState::STOPPED)
||
                        (jobRunState == Aws::Glue::Model::JobRunState::FAILED) ||
                        (jobRunState == Aws::Glue::Model::JobRunState::TIMEOUT))
{
                        std::cerr << "Error running job. "
                                  << jobRun.GetErrorMessage()
                                  << std::endl;
                        deleteAssets(CRAWLER_NAME, CRAWLER_DATABASE_NAME,
JOB_NAME,
                                     bucketName,
                                     clientConfig);
                        return false;
                    }
                    else if (jobRunState ==
                             Aws::Glue::Model::JobRunState::SUCCEEDED) {
                        std::cout << "Job run succeeded after  " << iterator <<
                                  " seconds elapsed." << std::endl;
                        done = true;
                    }
                    else if ((iterator % 10) == 0) { // Log status every 10
seconds.
                        std::cout << "Job run status " <<

Aws::Glue::Model::JobRunStateMapper::GetNameForJobRunState(
                                          jobRunState) <<
                                  ". " << iterator <<
                                  " seconds elapsed." << std::endl;
                    }
                }
                else {
                    std::cerr << "Error retrieving job run state. "
                              << jobRunOutcome.GetError().GetMessage()
```

```
                            << std::endl;
                    deleteAssets(CRAWLER_NAME, CRAWLER_DATABASE_NAME, JOB_NAME,
                                 bucketName, clientConfig);
                    return false;
                }
            }
        }
        else {
            std::cerr << "Error starting a job. " <<
outcome.GetError().GetMessage()
                      << std::endl;
            deleteAssets(CRAWLER_NAME, CRAWLER_DATABASE_NAME, JOB_NAME,
bucketName,
                         clientConfig);
            return false;
        }
    }

    // 9. List the output data stored in the S3 bucket.
    {
        Aws::S3::S3Client s3Client;
        Aws::S3::Model::ListObjectsV2Request request;
        request.SetBucket(bucketName);
        request.SetPrefix(OUTPUT_FILE_PREFIX);

        Aws::String continuationToken; // Used for pagination.
        std::vector<Aws::S3::Model::Object> allObjects;
        do {
            if (!continuationToken.empty()) {
                request.SetContinuationToken(continuationToken);
            }
            Aws::S3::Model::ListObjectsV2Outcome outcome =
s3Client.ListObjectsV2(
                    request);

            if (outcome.IsSuccess()) {
                const std::vector<Aws::S3::Model::Object> &objects =
                        outcome.GetResult().GetContents();
                allObjects.insert(allObjects.end(), objects.begin(),
objects.end());
                continuationToken =
outcome.GetResult().GetNextContinuationToken();
            }
            else {
```

```
                std::cerr << "Error listing objects. "
                          << outcome.GetError().GetMessage()
                          << std::endl;
                break;
            }
        } while (!continuationToken.empty());

        std::cout << "Data from your job is in " << allObjects.size() <<
                  " files in the S3 bucket, " << bucketName << "." << std::endl;

        for (size_t i = 0; i < allObjects.size(); ++i) {
            std::cout << "    " << i + 1 << ". " << allObjects[i].GetKey()
                      << std::endl;
        }

        int objectIndex = askQuestionForIntRange(
                std::string(
                        "Enter the number of a block to download it and see the
 first ") +
                std::to_string(LINES_OF_RUN_FILE_TO_DISPLAY) +
                " lines of JSON output in the block: ", 1,
                static_cast<int>(allObjects.size()));

        Aws::String objectKey = allObjects[objectIndex - 1].GetKey();

        std::stringstream stringStream;
        if (getObjectFromBucket(bucketName, objectKey, stringStream,
                                clientConfig)) {
            for (int i = 0; i < LINES_OF_RUN_FILE_TO_DISPLAY && stringStream; +
+i) {
                std::string line;
                std::getline(stringStream, line);
                std::cout << "    " << line << std::endl;
            }
        }
        else {
            deleteAssets(CRAWLER_NAME, CRAWLER_DATABASE_NAME, JOB_NAME,
 bucketName,
                         clientConfig);
            return false;
        }
    }

    // 10. List all the jobs.
```

```
    Aws::String jobName;
    {
        Aws::Glue::Model::ListJobsRequest listJobsRequest;

        Aws::String nextToken;
        std::vector<Aws::String> allJobNames;

        do {
            if (!nextToken.empty()) {
                listJobsRequest.SetNextToken(nextToken);
            }
            Aws::Glue::Model::ListJobsOutcome listRunsOutcome = client.ListJobs(
                    listJobsRequest);

            if (listRunsOutcome.IsSuccess()) {
                const std::vector<Aws::String> &jobNames =
listRunsOutcome.GetResult().GetJobNames();
                allJobNames.insert(allJobNames.end(), jobNames.begin(),
jobNames.end());
                nextToken = listRunsOutcome.GetResult().GetNextToken();
            }
            else {
                std::cerr << "Error listing jobs. "
                          << listRunsOutcome.GetError().GetMessage()
                          << std::endl;
            }
        } while (!nextToken.empty());
        std::cout << "Your account has " << allJobNames.size() << " jobs."
                  << std::endl;
        for (size_t i = 0; i < allJobNames.size(); ++i) {
            std::cout << "   " << i + 1 << ". " << allJobNames[i] << std::endl;
        }
        int jobIndex = askQuestionForIntRange(
                Aws::String("Enter a number between 1 and ") +
                std::to_string(allJobNames.size()) +
                " to see the list of runs for a job: ",
                1, static_cast<int>(allJobNames.size()));

        jobName = allJobNames[jobIndex - 1];
    }

    // 11. Get the job runs for a job.
    Aws::String jobRunID;
    if (!jobName.empty()) {
```

```
        Aws::Glue::Model::GetJobRunsRequest getJobRunsRequest;
        getJobRunsRequest.SetJobName(jobName);

        Aws::String nextToken; // Used for pagination.
        std::vector<Aws::Glue::Model::JobRun> allJobRuns;
        do {
            if (!nextToken.empty()) {
                getJobRunsRequest.SetNextToken(nextToken);
            }
            Aws::Glue::Model::GetJobRunsOutcome jobRunsOutcome =
client.GetJobRuns(
                    getJobRunsRequest);

            if (jobRunsOutcome.IsSuccess()) {
                const std::vector<Aws::Glue::Model::JobRun> &jobRuns =
jobRunsOutcome.GetResult().GetJobRuns();
                allJobRuns.insert(allJobRuns.end(), jobRuns.begin(),
jobRuns.end());

                nextToken = jobRunsOutcome.GetResult().GetNextToken();
            }
            else {
                std::cerr << "Error getting job runs. "
                          << jobRunsOutcome.GetError().GetMessage()
                          << std::endl;
                break;
            }
        } while (!nextToken.empty());

        std::cout << "There are " << allJobRuns.size() << " runs in the job '"
                  <<
                  jobName << "'." << std::endl;

        for (size_t i = 0; i < allJobRuns.size(); ++i) {
            std::cout << "   " << i + 1 << ". " << allJobRuns[i].GetJobName()
                      << std::endl;
        }

        int runIndex = askQuestionForIntRange(
                Aws::String("Enter a number between 1 and ") +
                std::to_string(allJobRuns.size()) +
                " to see details for a run: ",
                1, static_cast<int>(allJobRuns.size()));
        jobRunID = allJobRuns[runIndex - 1].GetId();
```

```
    }

    // 12. Get a single job run.
    if (!jobRunID.empty()) {
        Aws::Glue::Model::GetJobRunRequest jobRunRequest;
        jobRunRequest.SetJobName(jobName);
        jobRunRequest.SetRunId(jobRunID);

        Aws::Glue::Model::GetJobRunOutcome jobRunOutcome = client.GetJobRun(
                jobRunRequest);

        if (jobRunOutcome.IsSuccess()) {
            std::cout << "Displaying the job run JSON description." << std::endl;
            std::cout
                    <<
 jobRunOutcome.GetResult().GetJobRun().Jsonize().View().WriteReadable()
                    << std::endl;
        }
        else {
            std::cerr << "Error get a job run. "
                      << jobRunOutcome.GetError().GetMessage()
                      << std::endl;
        }
    }

    return deleteAssets(CRAWLER_NAME, CRAWLER_DATABASE_NAME, JOB_NAME,
 bucketName,
                        clientConfig);
}

//! Cleanup routine to delete created assets.
/*!
 \\sa deleteAssets()
 \param crawler: Name of an AWS Glue crawler.
 \param database: The name of an AWS Glue database.
 \param job: The name of an AWS Glue job.
 \param bucketName: The name of an S3 bucket.
 \param clientConfig: AWS client configuration.
 \return bool: Successful completion.
 */
bool AwsDoc::Glue::deleteAssets(const Aws::String &crawler, const Aws::String
 &database,
                                const Aws::String &job, const Aws::String
 &bucketName,
```

```
                                const Aws::Client::ClientConfiguration
&clientConfig) {
   const Aws::Glue::GlueClient client(clientConfig);
   bool result = true;

   // 13. Delete a job.
   if (!job.empty()) {
       Aws::Glue::Model::DeleteJobRequest request;
       request.SetJobName(job);

       Aws::Glue::Model::DeleteJobOutcome outcome = client.DeleteJob(request);


       if (outcome.IsSuccess()) {
           std::cout << "Successfully deleted the job." << std::endl;
       }
       else {
           std::cerr << "Error deleting the job. " <<
outcome.GetError().GetMessage()
                     << std::endl;
           result = false;
       }
   }

   // 14. Delete a database.
   if (!database.empty()) {
       Aws::Glue::Model::DeleteDatabaseRequest request;
       request.SetName(database);

       Aws::Glue::Model::DeleteDatabaseOutcome outcome = client.DeleteDatabase(
               request);

       if (outcome.IsSuccess()) {
           std::cout << "Successfully deleted the database." << std::endl;
       }
       else {
           std::cerr << "Error deleting database. " <<
outcome.GetError().GetMessage()
                     << std::endl;
           result = false;
       }
   }

   // 15. Delete a crawler.
```

```
    if (!crawler.empty()) {
        Aws::Glue::Model::DeleteCrawlerRequest request;
        request.SetName(crawler);

        Aws::Glue::Model::DeleteCrawlerOutcome outcome =
 client.DeleteCrawler(request);

        if (outcome.IsSuccess()) {
            std::cout << "Successfully deleted the crawler." << std::endl;
        }
        else {
            std::cerr << "Error deleting the crawler. "
                      << outcome.GetError().GetMessage() << std::endl;
            result = false;
        }
    }

    // 16. Delete the job script and run data from the S3 bucket.
    result &= AwsDoc::Glue::deleteAllObjectsInS3Bucket(bucketName,
                                                       clientConfig);
    return result;
}

//! Routine which uploads a file to an S3 bucket.
/*!
 \\sa uploadFile()
 \param bucketName: An S3 bucket created in the setup.
 \param filePath: The path of the file to upload.
 \param fileName The name for the uploaded file.
 \param clientConfig: AWS client configuration.
 \return bool: Successful completion.
 */
bool
AwsDoc::Glue::uploadFile(const Aws::String &bucketName,
                         const Aws::String &filePath,
                         const Aws::String &fileName,
                         const Aws::Client::ClientConfiguration &clientConfig) {
    Aws::S3::S3Client s3_client(clientConfig);

    Aws::S3::Model::PutObjectRequest request;
    request.SetBucket(bucketName);
    request.SetKey(fileName);

    std::shared_ptr<Aws::IOStream> inputData =
```

```
            Aws::MakeShared<Aws::FStream>("SampleAllocationTag",
                                          filePath.c_str(),
                                          std::ios_base::in |
 std::ios_base::binary);

    if (!*inputData) {
        std::cerr << "Error unable to read file " << filePath << std::endl;
        return false;
    }

    request.SetBody(inputData);

    Aws::S3::Model::PutObjectOutcome outcome =
            s3_client.PutObject(request);

    if (!outcome.IsSuccess()) {
        std::cerr << "Error: PutObject: " <<
                  outcome.GetError().GetMessage() << std::endl;
    }
    else {
        std::cout << "Added object '" << filePath << "' to bucket '"
                  << bucketName << "'." << std::endl;
    }

    return outcome.IsSuccess();
}

//! Routine which deletes all objects in an S3 bucket.
/*!
 \\sa deleteAllObjectsInS3Bucket()
 \param bucketName: The S3 bucket name.
 \param clientConfig: AWS client configuration.
 \return bool: Successful completion.
 */
bool AwsDoc::Glue::deleteAllObjectsInS3Bucket(const Aws::String &bucketName,
                                              const
 Aws::Client::ClientConfiguration &clientConfig) {
    Aws::S3::S3Client client(clientConfig);
    Aws::S3::Model::ListObjectsV2Request listObjectsRequest;
    listObjectsRequest.SetBucket(bucketName);

    Aws::String continuationToken; // Used for pagination.
    bool result = true;
    do {
```

```
        if (!continuationToken.empty()) {
            listObjectsRequest.SetContinuationToken(continuationToken);
        }

        Aws::S3::Model::ListObjectsV2Outcome listObjectsOutcome =
client.ListObjectsV2(
                listObjectsRequest);

        if (listObjectsOutcome.IsSuccess()) {
            const std::vector<Aws::S3::Model::Object> &objects =
listObjectsOutcome.GetResult().GetContents();
            if (!objects.empty()) {
                Aws::S3::Model::DeleteObjectsRequest deleteObjectsRequest;
                deleteObjectsRequest.SetBucket(bucketName);

                std::vector<Aws::S3::Model::ObjectIdentifier> objectIdentifiers;
                for (const Aws::S3::Model::Object &object: objects) {
                    objectIdentifiers.push_back(
                            Aws::S3::Model::ObjectIdentifier().WithKey(
                                    object.GetKey()));
                }
                Aws::S3::Model::Delete objectsDelete;
                objectsDelete.SetObjects(objectIdentifiers);
                objectsDelete.SetQuiet(true);
                deleteObjectsRequest.SetDelete(objectsDelete);

                Aws::S3::Model::DeleteObjectsOutcome deleteObjectsOutcome =
                        client.DeleteObjects(deleteObjectsRequest);

                if (!deleteObjectsOutcome.IsSuccess()) {
                    std::cerr << "Error deleting objects. " <<
                              deleteObjectsOutcome.GetError().GetMessage() <<
std::endl;
                    result = false;
                    break;
                }
                else {
                    std::cout << "Successfully deleted the objects." <<
std::endl;

                }
            }
            else {
                std::cout << "No objects to delete in '" << bucketName << "'."
```

```
                          << std::endl;
            }

            continuationToken =
 listObjectsOutcome.GetResult().GetNextContinuationToken();
        }
        else {
            std::cerr << "Error listing objects. "
                      << listObjectsOutcome.GetError().GetMessage() << std::endl;
            result = false;
            break;
        }
    } while (!continuationToken.empty());

    return result;
}

//! Routine which retrieves an object from an S3 bucket.
/*!
 \\sa getObjectFromBucket()
 \param bucketName: The S3 bucket name.
 \param objectKey: The object's name.
 \param objectStream: A stream to receive the retrieved data.
 \param clientConfig: AWS client configuration.
 \return bool: Successful completion.
 */
bool AwsDoc::Glue::getObjectFromBucket(const Aws::String &bucketName,
                                       const Aws::String &objectKey,
                                       std::ostream &objectStream,
                                       const Aws::Client::ClientConfiguration
 &clientConfig) {
    Aws::S3::S3Client client(clientConfig);
    Aws::S3::Model::GetObjectRequest request;
    request.SetBucket(bucketName);
    request.SetKey(objectKey);

    Aws::S3::Model::GetObjectOutcome outcome = client.GetObject(request);


    if (outcome.IsSuccess()) {
        std::cout << "Successfully retrieved '" << objectKey << "'." <<
 std::endl;
        auto &body = outcome.GetResult().GetBody();
        objectStream << body.rdbuf();
```

```
    }
    else {
        std::cerr << "Error retrieving object. " <<
 outcome.GetError().GetMessage()
                  << std::endl;
    }

    return outcome.IsSuccess();
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API AWS SDK per C++ .
  - CreateCrawler
  - CreateJob
  - DeleteCrawler
  - DeleteDatabase
  - DeleteJob
  - DeleteTable
  - GetCrawler
  - GetDatabase
  - GetDatabases
  - GetJob
  - GetJobRun
  - GetJobRuns
  - GetTables
  - ListJobs
  - StartCrawler
  - StartJobRun

Java

SDK per Java 2.x

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 * <p>
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 *
 * To set up the resources, see this documentation topic:
 *
 * https://docs.aws.amazon.com/glue/latest/ug/tutorial-add-crawler.html
 *
 * This example performs the following tasks:
 *
 * 1. Create a database.
 * 2. Create a crawler.
 * 3. Get a crawler.
 * 4. Start a crawler.
 * 5. Get a database.
 * 6. Get tables.
 * 7. Create a job.
 * 8. Start a job run.
 * 9. List all jobs.
 * 10. Get job runs.
 * 11. Delete a job.
 * 12. Delete a database.
 * 13. Delete a crawler.
 */

public class GlueScenario {
```

```
   public static final String DASHES = new String(new char[80]).replace("\0",
"-");

   public static void main(String[] args) throws InterruptedException {
       final String usage = """

           Usage:
               <iam> <s3Path> <cron> <dbName> <crawlerName> <jobName>
<scriptLocation> <locationUri> <bucketNameSc>\s

           Where:
               iam - The ARN of the IAM role that has AWS Glue and S3
permissions.\s
               s3Path - The Amazon Simple Storage Service (Amazon S3) target
that contains data (for example, s3://<bucket name>/read).
               cron - A cron expression used to specify the schedule  (i.e.,
cron(15 12 * * ? *).
               dbName - The database name.\s
               crawlerName - The name of the crawler.\s
               jobName - The name you assign to this job definition.
               scriptLocation - The Amazon S3 path to a script that runs a job.
               locationUri - The location of the database (you can find this
file in resources folder).
               bucketNameSc - The Amazon S3 bucket name used when creating a job
               """;

       if (args.length != 9) {
           System.out.println(usage);
           return;
       }
       Scanner scanner = new Scanner(System.in);
       String iam = args[0];
       String s3Path = args[1];
       String cron = args[2];
       String dbName = args[3];
       String crawlerName = args[4];
       String jobName = args[5];
       String scriptLocation = args[6];
       String locationUri = args[7];
       String bucketNameSc = args[8];

       Region region = Region.US_EAST_1;
       GlueClient glueClient = GlueClient.builder()
           .region(region)
```

```
            .build();
        System.out.println(DASHES);
        System.out.println("Welcome to the AWS Glue scenario.");
        System.out.println("""
            AWS Glue is a fully managed extract, transform, and load (ETL)
service provided by Amazon
            Web Services (AWS). It is designed to simplify the process of
building, running, and maintaining
            ETL pipelines, which are essential for data integration and data
warehousing tasks.

            One of the key features of AWS Glue is its ability to automatically
discover and catalog data
            stored in various sources, such as Amazon S3, Amazon RDS, Amazon
Redshift, and other databases.
            This cataloging process creates a central metadata repository, known
as the AWS Glue Data Catalog,
            which provides a unified view of an organization's data assets. This
metadata can then be used to
            create ETL jobs, which can be scheduled and run on-demand or on a
regular basis.

            Lets get started.

            """);
        waitForInputToContinue(scanner);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("1. Create a database.");
        try {
            createDatabase(glueClient, dbName, locationUri);
        } catch (GlueException e) {
            if (e.awsErrorDetails().errorMessage().equals("Database already
exists.")) {
                System.out.println("Database " + dbName + " already exists.
Skipping creation.");
            } else {
                System.err.println(e.awsErrorDetails().errorMessage());
                return;
            }
        }

        waitForInputToContinue(scanner);
```

```
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("2. Create a crawler.");
        try {
            createGlueCrawler(glueClient, iam, s3Path, cron, dbName,
crawlerName);
        } catch (GlueException e) {
            if (e.awsErrorDetails().errorMessage().contains("already exists")) {
                System.out.println("Crawler " + crawlerName + " already exists.
Skipping creation.");
            } else {
                System.err.println(e.awsErrorDetails().errorMessage());
                System.exit(1);
            }
        }
        waitForInputToContinue(scanner);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("3. Get a crawler.");
        try {
            getSpecificCrawler(glueClient, crawlerName);
        } catch (GlueException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            return;
        }
        waitForInputToContinue(scanner);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("4. Start a crawler.");
        try {
            startSpecificCrawler(glueClient, crawlerName);
        } catch (GlueException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            return;
        }
        waitForInputToContinue(scanner);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("5. Get a database.");
        try {
```

```
            getSpecificDatabase(glueClient, dbName);
        } catch (GlueException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            return;
        }
        waitForInputToContinue(scanner);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("*** Wait 5 min for the tables to become available");
        TimeUnit.MINUTES.sleep(5);
        System.out.println("6. Get tables.");
        String myTableName;
        try {
            myTableName = getGlueTables(glueClient, dbName);
        } catch (GlueException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            return;
        }
        waitForInputToContinue(scanner);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("7. Create a job.");
        try {
            createJob(glueClient, jobName, iam, scriptLocation);
        } catch (GlueException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            return;
        }
        waitForInputToContinue(scanner);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("8. Start a Job run.");
        try {
            startJob(glueClient, jobName, dbName, myTableName, bucketNameSc);
        } catch (GlueException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            return;
        }
        waitForInputToContinue(scanner);
        System.out.println(DASHES);
```

```
        System.out.println(DASHES);
        System.out.println("9. List all jobs.");
        try {
            getAllJobs(glueClient);
        } catch (GlueException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            return;
        }
        waitForInputToContinue(scanner);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("10. Get job runs.");
        try {
            getJobRuns(glueClient, jobName);
        } catch (GlueException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            return;
        }
        waitForInputToContinue(scanner);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("11. Delete a job.");
        try {
            deleteJob(glueClient, jobName);
        } catch (GlueException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            return;
        }
        System.out.println("*** Wait 5 MIN for the " + crawlerName + " to stop");
        TimeUnit.MINUTES.sleep(5);
        waitForInputToContinue(scanner);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("12. Delete a database.");
        try {
            deleteDatabase(glueClient, dbName);
        } catch (GlueException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            return;
        }
        waitForInputToContinue(scanner);
```

```
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("Delete a crawler.");
        try {
            deleteSpecificCrawler(glueClient, crawlerName);
        } catch (GlueException e) {
            System.err.println(e.awsErrorDetails().errorMessage());
            return;
        }
        waitForInputToContinue(scanner);
        System.out.println(DASHES);

        System.out.println(DASHES);
        System.out.println("Successfully completed the AWS Glue Scenario");
        System.out.println(DASHES);
    }


    /**
     * Creates a Glue database with the specified name and location URI.
     *
     * @param glueClient  The Glue client to use for the database creation.
     * @param dbName      The name of the database to create.
     * @param locationUri The location URI for the database.
     */
    public static void createDatabase(GlueClient glueClient, String dbName,
String locationUri) {
        try {
            DatabaseInput input = DatabaseInput.builder()
                .description("Built with the AWS SDK for Java V2")
                .name(dbName)
                .locationUri(locationUri)
                .build();

            CreateDatabaseRequest request = CreateDatabaseRequest.builder()
                .databaseInput(input)
                .build();

            glueClient.createDatabase(request);
            System.out.println(dbName + " was successfully created");

        } catch (GlueException e) {
            throw e;
```

```
        }
    }


    /**
     * Creates a new AWS Glue crawler using the AWS Glue Java API.
     *
     * @param glueClient  the AWS Glue client used to interact with the AWS Glue
service
     * @param iam         the IAM role that the crawler will use to access the
data source
     * @param s3Path      the S3 path that the crawler will scan for data
     * @param cron        the cron expression that defines the crawler's schedule
     * @param dbName      the name of the AWS Glue database where the crawler
will store the metadata
     * @param crawlerName the name of the crawler to be created
     */
    public static void createGlueCrawler(GlueClient glueClient,
                                         String iam,
                                         String s3Path,
                                         String cron,
                                         String dbName,
                                         String crawlerName) {

        try {
            S3Target s3Target = S3Target.builder()
                .path(s3Path)
                .build();

            List<S3Target> targetList = new ArrayList<>();
            targetList.add(s3Target);
            CrawlerTargets targets = CrawlerTargets.builder()
                .s3Targets(targetList)
                .build();

            CreateCrawlerRequest crawlerRequest = CreateCrawlerRequest.builder()
                .databaseName(dbName)
                .name(crawlerName)
                .description("Created by the AWS Glue Java API")
                .targets(targets)
                .role(iam)
                .schedule(cron)
                .build();
```

```
            glueClient.createCrawler(crawlerRequest);
            System.out.println(crawlerName + " was successfully created");

        } catch (GlueException e) {
            throw e;
        }
    }

    /**
     * Retrieves a specific crawler from the AWS Glue service and waits for it to
be in the "READY" state.
     *
     * @param glueClient  the AWS Glue client used to interact with the Glue
service
     * @param crawlerName the name of the crawler to be retrieved
     */
    public static void getSpecificCrawler(GlueClient glueClient, String
crawlerName) throws InterruptedException {
        try {
            GetCrawlerRequest crawlerRequest = GetCrawlerRequest.builder()
                .name(crawlerName)
                .build();

            boolean ready = false;
            while (!ready) {
                GetCrawlerResponse response =
glueClient.getCrawler(crawlerRequest);
                String status = response.crawler().stateAsString();
                if (status.compareTo("READY") == 0) {
                    ready = true;
                }
                Thread.sleep(3000);
            }

            System.out.println("The crawler is now ready");

        } catch (GlueException | InterruptedException e) {
            throw e;
        }
    }

    /**
     * Starts a specific AWS Glue crawler.
     *
```

```
    * @param glueClient  the AWS Glue client to use for the crawler operation
    * @param crawlerName the name of the crawler to start
    * @throws GlueException if there is an error starting the crawler
    */
   public static void startSpecificCrawler(GlueClient glueClient, String
crawlerName) {
       try {
           StartCrawlerRequest crawlerRequest = StartCrawlerRequest.builder()
               .name(crawlerName)
               .build();

           glueClient.startCrawler(crawlerRequest);
           System.out.println(crawlerName + " was successfully started!");

       } catch (GlueException e) {
           throw e;
       }
   }

   /**
    * Retrieves the specific database from the AWS Glue service.
    *
    * @param glueClient   an instance of the AWS Glue client used to interact
with the service
    * @param databaseName the name of the database to retrieve
    * @throws GlueException if there is an error retrieving the database from
the AWS Glue service
    */
   public static void getSpecificDatabase(GlueClient glueClient, String
databaseName) {
       try {
           GetDatabaseRequest databasesRequest = GetDatabaseRequest.builder()
               .name(databaseName)
               .build();

           GetDatabaseResponse response =
glueClient.getDatabase(databasesRequest);
           Instant createDate = response.database().createTime();

           // Convert the Instant to readable date.
           DateTimeFormatter formatter =
DateTimeFormatter.ofLocalizedDateTime(FormatStyle.SHORT)
               .withLocale(Locale.US)
               .withZone(ZoneId.systemDefault());
```

```
            formatter.format(createDate);
            System.out.println("The create date of the database is " +
createDate);

        } catch (GlueException e) {
            throw e;
        }
    }


    /**
     * Retrieves the names of the tables in the specified Glue database.
     *
     * @param glueClient the Glue client to use for the operation
     * @param dbName     the name of the Glue database to retrieve the table
names from
     * @return the name of the first table retrieved, or an empty string if no
tables were found
     */
    public static String getGlueTables(GlueClient glueClient, String dbName) {
        String myTableName = "";
        try {
            GetTablesRequest tableRequest = GetTablesRequest.builder()
                .databaseName(dbName)
                .build();

            GetTablesResponse response = glueClient.getTables(tableRequest);
            List<Table> tables = response.tableList();
            if (tables.isEmpty()) {
                System.out.println("No tables were returned");
            } else {
                for (Table table : tables) {
                    myTableName = table.name();
                    System.out.println("Table name is: " + myTableName);
                }
            }

        } catch (GlueException e) {
            throw e;
        }
        return myTableName;
    }
```

```
   /**
    * Starts a job run in AWS Glue.
    *
    * @param glueClient    the AWS Glue client to use for the job run
    * @param jobName       the name of the Glue job to run
    * @param inputDatabase the name of the input database
    * @param inputTable    the name of the input table
    * @param outBucket     the URL of the output S3 bucket
    * @throws GlueException if there is an error starting the job run
    */
   public static void startJob(GlueClient glueClient, String jobName, String
inputDatabase, String inputTable,
                               String outBucket) {
       try {
           Map<String, String> myMap = new HashMap<>();
           myMap.put("--input_database", inputDatabase);
           myMap.put("--input_table", inputTable);
           myMap.put("--output_bucket_url", outBucket);

           StartJobRunRequest runRequest = StartJobRunRequest.builder()
               .workerType(WorkerType.G_1_X)
               .numberOfWorkers(10)
               .arguments(myMap)
               .jobName(jobName)
               .build();

           StartJobRunResponse response = glueClient.startJobRun(runRequest);
           System.out.println("The request Id of the job is " +
response.responseMetadata().requestId());

       } catch (GlueException e) {
           throw e;
       }
   }


   /**
    * Creates a new AWS Glue job.
    *
    * @param glueClient     the AWS Glue client to use for the operation
    * @param jobName        the name of the job to create
    * @param iam            the IAM role to associate with the job
    * @param scriptLocation the location of the script to be used by the job
```

```
     * @throws GlueException if there is an error creating the job
     */
    public static void createJob(GlueClient glueClient, String jobName, String
iam, String scriptLocation) {
        try {
            JobCommand command = JobCommand.builder()
                .pythonVersion("3")
                .name("glueetl")
                .scriptLocation(scriptLocation)
                .build();

            CreateJobRequest jobRequest = CreateJobRequest.builder()
                .description("A Job created by using the AWS SDK for Java V2")
                .glueVersion("2.0")
                .workerType(WorkerType.G_1_X)
                .numberOfWorkers(10)
                .name(jobName)
                .role(iam)
                .command(command)
                .build();

            glueClient.createJob(jobRequest);
            System.out.println(jobName + " was successfully created.");

        } catch (GlueException e) {
            throw e;
        }
    }


    /**
     * Retrieves and prints information about all the jobs in the Glue data
catalog.
     *
     * @param glueClient the Glue client used to interact with the AWS Glue
service
     */
    public static void getAllJobs(GlueClient glueClient) {
        try {
            GetJobsRequest jobsRequest = GetJobsRequest.builder()
                .maxResults(10)
                .build();

            GetJobsResponse jobsResponse = glueClient.getJobs(jobsRequest);
```

```
            List<Job> jobs = jobsResponse.jobs();
            for (Job job : jobs) {
                System.out.println("Job name is : " + job.name());
                System.out.println("The job worker type is : " +
job.workerType().name());
            }

        } catch (GlueException e) {
            throw e;
        }
    }

    /**
     * Retrieves the job runs for a given Glue job and prints the status of the
job runs.
     *
     * @param glueClient the Glue client used to make API calls
     * @param jobName    the name of the Glue job to retrieve the job runs for
     */
    public static void getJobRuns(GlueClient glueClient, String jobName) {
        try {
            GetJobRunsRequest runsRequest = GetJobRunsRequest.builder()
                .jobName(jobName)
                .maxResults(20)
                .build();

            boolean jobDone = false;
            while (!jobDone) {
                GetJobRunsResponse response = glueClient.getJobRuns(runsRequest);
                List<JobRun> jobRuns = response.jobRuns();
                for (JobRun jobRun : jobRuns) {
                    String jobState = jobRun.jobRunState().name();
                    if (jobState.compareTo("SUCCEEDED") == 0) {
                        System.out.println(jobName + " has succeeded");
                        jobDone = true;

                    } else if (jobState.compareTo("STOPPED") == 0) {
                        System.out.println("Job run has stopped");
                        jobDone = true;

                    } else if (jobState.compareTo("FAILED") == 0) {
                        System.out.println("Job run has failed");
                        jobDone = true;
```

```
                    } else if (jobState.compareTo("TIMEOUT") == 0) {
                        System.out.println("Job run has timed out");
                        jobDone = true;

                    } else {
                        System.out.println("*** Job run state is " +
jobRun.jobRunState().name());
                        System.out.println("Job run Id is " + jobRun.id());
                        System.out.println("The Glue version is " +
jobRun.glueVersion());
                    }
                    TimeUnit.SECONDS.sleep(5);
                }
            }

        } catch (GlueException e) {
            throw e;
        } catch (InterruptedException e) {
            throw new RuntimeException(e);
        }
    }


    /**
     * Deletes a Glue job.
     *
     * @param glueClient the Glue client to use for the operation
     * @param jobName    the name of the job to be deleted
     * @throws GlueException if there is an error deleting the job
     */
    public static void deleteJob(GlueClient glueClient, String jobName) {
        try {
            DeleteJobRequest jobRequest = DeleteJobRequest.builder()
                .jobName(jobName)
                .build();

            glueClient.deleteJob(jobRequest);
            System.out.println(jobName + " was successfully deleted");

        } catch (GlueException e) {
            throw e;
        }
    }
```

```
   /**
    * Deletes a AWS Glue Database.
    *
    * @param glueClient   An instance of the AWS Glue client used to interact
with the AWS Glue service.
    * @param databaseName The name of the database to be deleted.
    * @throws GlueException If an error occurs while deleting the database.
    */
   public static void deleteDatabase(GlueClient glueClient, String databaseName)
{
       try {
           DeleteDatabaseRequest request = DeleteDatabaseRequest.builder()
               .name(databaseName)
               .build();

           glueClient.deleteDatabase(request);
           System.out.println(databaseName + " was successfully deleted");

       } catch (GlueException e) {
           throw e;
       }
   }


   /**
    * Deletes a specific AWS Glue crawler.
    *
    * @param glueClient  the AWS Glue client object
    * @param crawlerName the name of the crawler to be deleted
    * @throws GlueException if an error occurs during the deletion process
    */
   public static void deleteSpecificCrawler(GlueClient glueClient, String
crawlerName) {
       try {
           DeleteCrawlerRequest deleteCrawlerRequest =
DeleteCrawlerRequest.builder()
               .name(crawlerName)
               .build();

           glueClient.deleteCrawler(deleteCrawlerRequest);
           System.out.println(crawlerName + " was deleted");

       } catch (GlueException e) {
           throw e;
```

```
        }
    }

    private static void waitForInputToContinue(Scanner scanner) {
        while (true) {
            System.out.println("");
            System.out.println("Enter 'c' followed by <ENTER> to continue:");
            String input = scanner.nextLine();

            if (input.trim().equalsIgnoreCase("c")) {
                System.out.println("Continuing with the program...");
                System.out.println("");
                break;
            } else {
                // Handle invalid input.
                System.out.println("Invalid input. Please try again.");
            }
        }
    }
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API AWS SDK for Java 2.x .
  - CreateCrawler
  - CreateJob
  - DeleteCrawler
  - DeleteDatabase
  - DeleteJob
  - DeleteTable
  - GetCrawler
  - GetDatabase
  - GetDatabases
  - GetJob
  - GetJobRun
  - GetJobRuns
  - GetTables

- ListJobs
- StartCrawler
- StartJobRun

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Crea ed esegui un crawler che esegua la scansione di un bucket Amazon Simple Storage Service (Amazon S3) pubblico e generi un database di metadati che descrive i dati trovati. CSV-formatted

```
const createCrawler = (name, role, dbName, tablePrefix, s3TargetPath) => {
  const client = new GlueClient({});

  const command = new CreateCrawlerCommand({
    Name: name,
    Role: role,
    DatabaseName: dbName,
    TablePrefix: tablePrefix,
    Targets: {
      S3Targets: [{ Path: s3TargetPath }],
    },
  });

  return client.send(command);
};

const getCrawler = (name) => {
  const client = new GlueClient({});

  const command = new GetCrawlerCommand({
    Name: name,
  });
```

```
  return client.send(command);
};

const startCrawler = (name) => {
  const client = new GlueClient({});

  const command = new StartCrawlerCommand({
    Name: name,
  });

  return client.send(command);
};

const crawlerExists = async ({ getCrawler }, crawlerName) => {
  try {
    await getCrawler(crawlerName);
    return true;
  } catch {
    return false;
  }
};

/**
 * @param {{ createCrawler: import('../../../actions/create-
crawler.js').createCrawler}} actions
 */
const makeCreateCrawlerStep = (actions) => async (context) => {
  if (await crawlerExists(actions, process.env.CRAWLER_NAME)) {
    log("Crawler already exists. Skipping creation.");
  } else {
    await actions.createCrawler(
      process.env.CRAWLER_NAME,
      process.env.ROLE_NAME,
      process.env.DATABASE_NAME,
      process.env.TABLE_PREFIX,
      process.env.S3_TARGET_PATH,
    );

    log("Crawler created successfully.", { type: "success" });
  }

  return { ...context };
};
```

```
/**
 * @param {(name: string) => Promise<import('@aws-sdk/client-
glue').GetCrawlerCommandOutput>} getCrawler
 * @param {string} crawlerName
 */
const waitForCrawler = async (getCrawler, crawlerName) => {
  const waitTimeInSeconds = 30;
  const { Crawler } = await getCrawler(crawlerName);

  if (!Crawler) {
    throw new Error(`Crawler with name ${crawlerName} not found.`);
  }

  if (Crawler.State === "READY") {
    return;
  }

  log(`Crawler is ${Crawler.State}. Waiting ${waitTimeInSeconds} seconds...`);
  await wait(waitTimeInSeconds);
  return waitForCrawler(getCrawler, crawlerName);
};

const makeStartCrawlerStep =
  ({ startCrawler, getCrawler }) =>
  async (context) => {
    log("Starting crawler.");
    await startCrawler(process.env.CRAWLER_NAME);
    log("Crawler started.", { type: "success" });

    log("Waiting for crawler to finish running. This can take a while.");
    await waitForCrawler(getCrawler, process.env.CRAWLER_NAME);
    log("Crawler ready.", { type: "success" });

    return { ...context };
  };
```

Elenca le informazioni su database e tabelle nel tuo. AWS Glue Data Catalog

```
const getDatabase = (name) => {
  const client = new GlueClient({});
```

```
  const command = new GetDatabaseCommand({
    Name: name,
  });

  return client.send(command);
};

const getTables = (databaseName) => {
  const client = new GlueClient({});

  const command = new GetTablesCommand({
    DatabaseName: databaseName,
  });

  return client.send(command);
};

const makeGetDatabaseStep =
  ({ getDatabase }) =>
  async (context) => {
    const {
      Database: { Name },
    } = await getDatabase(process.env.DATABASE_NAME);
    log(`Database: ${Name}`);
    return { ...context };
  };

/**
 * @param {{ getTables: () => Promise<import('@aws-sdk/client-
glue').GetTablesCommandOutput}} config
 */
const makeGetTablesStep =
  ({ getTables }) =>
  async (context) => {
    const { TableList } = await getTables(process.env.DATABASE_NAME);
    log("Tables:");
    log(TableList.map((table) => `  • ${table.Name}\n`));
    return { ...context };
  };
```

Crea ed esegui un processo che estrae i dati CSV dal bucket Amazon S3 di origine, li trasforma rimuovendo e rinominando i campi e carica l'output JSON-formatted in un altro bucket Amazon S3.

```
const createJob = (name, role, scriptBucketName, scriptKey) => {
  const client = new GlueClient({});

  const command = new CreateJobCommand({
    Name: name,
    Role: role,
    Command: {
      Name: "glueetl",
      PythonVersion: "3",
      ScriptLocation: `s3://${scriptBucketName}/${scriptKey}`,
    },
    GlueVersion: "3.0",
  });

  return client.send(command);
};

const startJobRun = (jobName, dbName, tableName, bucketName) => {
  const client = new GlueClient({});

  const command = new StartJobRunCommand({
    JobName: jobName,
    Arguments: {
      "--input_database": dbName,
      "--input_table": tableName,
      "--output_bucket_url": `s3://${bucketName}/`,
    },
  });

  return client.send(command);
};

const makeCreateJobStep =
  ({ createJob }) =>
  async (context) => {
    log("Creating Job.");
    await createJob(
      process.env.JOB_NAME,
      process.env.ROLE_NAME,
```

```
      process.env.BUCKET_NAME,
      process.env.PYTHON_SCRIPT_KEY,
    );
    log("Job created.", { type: "success" });

    return { ...context };
  };

/**
 * @param {(name: string, runId: string) => Promise<import('@aws-sdk/client-
glue').GetJobRunCommandOutput> }  getJobRun
 * @param {string} jobName
 * @param {string} jobRunId
 */
const waitForJobRun = async (getJobRun, jobName, jobRunId) => {
  const waitTimeInSeconds = 30;
  const { JobRun } = await getJobRun(jobName, jobRunId);

  if (!JobRun) {
    throw new Error(`Job run with id ${jobRunId} not found.`);
  }

  switch (JobRun.JobRunState) {
    case "FAILED":
    case "TIMEOUT":
    case "STOPPED":
    case "ERROR":
      throw new Error(
        `Job ${JobRun.JobRunState}. Error: ${JobRun.ErrorMessage}`,
      );
    case "SUCCEEDED":
      return;
    default:
      break;
  }

  log(
    `Job ${JobRun.JobRunState}. Waiting ${waitTimeInSeconds} more seconds...`,
  );
  await wait(waitTimeInSeconds);
  return waitForJobRun(getJobRun, jobName, jobRunId);
};

/**
```

```
 * @param {{ prompter: { prompt: () => Promise<{ shouldOpen: boolean }>} }}
 context
 */
const promptToOpen = async (context) => {
  const { shouldOpen } = await context.prompter.prompt({
    name: "shouldOpen",
    type: "confirm",
    message: "Open the output bucket in your browser?",
  });

  if (shouldOpen) {
    return open(
      `https://s3.console.aws.amazon.com/s3/buckets/${process.env.BUCKET_NAME} to
 view the output.`,
    );
  }
};

const makeStartJobRunStep =
  ({ startJobRun, getJobRun }) =>
  async (context) => {
    log("Starting job.");
    const { JobRunId } = await startJobRun(
      process.env.JOB_NAME,
      process.env.DATABASE_NAME,
      process.env.TABLE_NAME,
      process.env.BUCKET_NAME,
    );
    log("Job started.", { type: "success" });

    log("Waiting for job to finish running. This can take a while.");
    await waitForJobRun(getJobRun, process.env.JOB_NAME, JobRunId);
    log("Job run succeeded.", { type: "success" });

    await promptToOpen(context);

    return { ...context };
  };
```

Elencare le informazioni sulle esecuzioni dei processi e visualizzare alcuni dei dati trasformati.

```
const getJobRuns = (jobName) => {
```

```
  const client = new GlueClient({});
  const command = new GetJobRunsCommand({
    JobName: jobName,
  });

  return client.send(command);
};

const getJobRun = (jobName, jobRunId) => {
  const client = new GlueClient({});
  const command = new GetJobRunCommand({
    JobName: jobName,
    RunId: jobRunId,
  });

  return client.send(command);
};

/**
 * @typedef {{ prompter: { prompt: () => Promise<{jobName: string}> } }} Context
 */

/**
 * @typedef {() => Promise<import('@aws-sdk/client-
glue').GetJobRunCommandOutput>} getJobRun
 */

/**
 * @typedef {() => Promise<import('@aws-sdk/client-
glue').GetJobRunsCommandOutput} getJobRuns
 */

/**
 *
 * @param {getJobRun} getJobRun
 * @param {string} jobName
 * @param {string} jobRunId
 */
const logJobRunDetails = async (getJobRun, jobName, jobRunId) => {
  const { JobRun } = await getJobRun(jobName, jobRunId);
  log(JobRun, { type: "object" });
};

/**
```

```
 *
 * @param {{getJobRuns: getJobRuns, getJobRun: getJobRun }} funcs
 */
const makePickJobRunStep =
  ({ getJobRuns, getJobRun }) =>
  async (/** @type { Context } */ context) => {
    if (context.selectedJobName) {
      const { JobRuns } = await getJobRuns(context.selectedJobName);

      const { jobRunId } = await context.prompter.prompt({
        name: "jobRunId",
        type: "list",
        message: "Select a job run to see details.",
        choices: JobRuns.map((run) => run.Id),
      });

      logJobRunDetails(getJobRun, context.selectedJobName, jobRunId);
    }

    return { ...context };
  };
```

Eliminare tutte le risorse create dalla demo.

```
const deleteJob = (jobName) => {
  const client = new GlueClient({});

  const command = new DeleteJobCommand({
    JobName: jobName,
  });

  return client.send(command);
};

const deleteTable = (databaseName, tableName) => {
  const client = new GlueClient({});

  const command = new DeleteTableCommand({
    DatabaseName: databaseName,
    Name: tableName,
  });
```

```
  return client.send(command);
};

const deleteDatabase = (databaseName) => {
  const client = new GlueClient({});

  const command = new DeleteDatabaseCommand({
    Name: databaseName,
  });

  return client.send(command);
};

const deleteCrawler = (crawlerName) => {
  const client = new GlueClient({});

  const command = new DeleteCrawlerCommand({
    Name: crawlerName,
  });

  return client.send(command);
};

/**
 *
 * @param {import('../../../actions/delete-job.js').deleteJob} deleteJobFn
 * @param {string[]} jobNames
 * @param {{ prompter: { prompt: () => Promise<any> }}} context
 */
const handleDeleteJobs = async (deleteJobFn, jobNames, context) => {
  /**
   * @type {{ selectedJobNames: string[] }}
   */
  const { selectedJobNames } = await context.prompter.prompt({
    name: "selectedJobNames",
    type: "checkbox",
    message: "Let's clean up jobs. Select jobs to delete.",
    choices: jobNames,
  });

  if (selectedJobNames.length === 0) {
    log("No jobs selected.");
  } else {
    log("Deleting jobs.");
```

```
    await Promise.all(
      selectedJobNames.map((n) => deleteJobFn(n).catch(console.error)),
    );
    log("Jobs deleted.", { type: "success" });
  }
};

/**
 * @param {{
 *   listJobs: import('../../../actions/list-jobs.js').listJobs,
 *   deleteJob: import('../../../actions/delete-job.js').deleteJob
 * }} config
 */
const makeCleanUpJobsStep =
  ({ listJobs, deleteJob }) =>
  async (context) => {
    const { JobNames } = await listJobs();
    if (JobNames.length > 0) {
      await handleDeleteJobs(deleteJob, JobNames, context);
    }

    return { ...context };
  };

/**
 * @param {import('../../../actions/delete-table.js').deleteTable} deleteTable
 * @param {string} databaseName
 * @param {string[]} tableNames
 */
const deleteTables = (deleteTable, databaseName, tableNames) =>
  Promise.all(
    tableNames.map((tableName) =>
      deleteTable(databaseName, tableName).catch(console.error),
    ),
  );

/**
 * @param {{
 *  getTables: import('../../../actions/get-tables.js').getTables,
 *  deleteTable: import('../../../actions/delete-table.js').deleteTable
 * }} config
 */
const makeCleanUpTablesStep =
  ({ getTables, deleteTable }) =>
```

```
  /**
   * @param {{ prompter: { prompt: () => Promise<any>}}} context
   */
  async (context) => {
    const { TableList } = await getTables(process.env.DATABASE_NAME).catch(
      () => ({ TableList: null }),
    );

    if (TableList && TableList.length > 0) {
      /**
       * @type {{ tableNames: string[] }}
       */
      const { tableNames } = await context.prompter.prompt({
        name: "tableNames",
        type: "checkbox",
        message: "Let's clean up tables. Select tables to delete.",
        choices: TableList.map((t) => t.Name),
      });

      if (tableNames.length === 0) {
        log("No tables selected.");
      } else {
        log("Deleting tables.");
        await deleteTables(deleteTable, process.env.DATABASE_NAME, tableNames);
        log("Tables deleted.", { type: "success" });
      }
    }

    return { ...context };
  };

/**
 * @param {import('../../../actions/delete-database.js').deleteDatabase}
 deleteDatabase
 * @param {string[]} databaseNames
 */
const deleteDatabases = (deleteDatabase, databaseNames) =>
  Promise.all(
    databaseNames.map((dbName) => deleteDatabase(dbName).catch(console.error)),
  );

/**
 * @param {{
 *   getDatabases: import('../../../actions/get-databases.js').getDatabases
```

```
 *   deleteDatabase: import('../../../actions/delete-database.js').deleteDatabase
 * }} config
 */
const makeCleanUpDatabasesStep =
  ({ getDatabases, deleteDatabase }) =>
  /**
   * @param {{ prompter: { prompt: () => Promise<any>}} context
   */
  async (context) => {
    const { DatabaseList } = await getDatabases();

    if (DatabaseList.length > 0) {
      /** @type {{ dbNames: string[] }} */
      const { dbNames } = await context.prompter.prompt({
        name: "dbNames",
        type: "checkbox",
        message: "Let's clean up databases. Select databases to delete.",
        choices: DatabaseList.map((db) => db.Name),
      });

      if (dbNames.length === 0) {
        log("No databases selected.");
      } else {
        log("Deleting databases.");
        await deleteDatabases(deleteDatabase, dbNames);
        log("Databases deleted.", { type: "success" });
      }
    }

    return { ...context };
  };

const cleanUpCrawlerStep = async (context) => {
  log("Deleting crawler.");

  try {
    await deleteCrawler(process.env.CRAWLER_NAME);
    log("Crawler deleted.", { type: "success" });
  } catch (err) {
    if (err.name === "EntityNotFoundException") {
      log("Crawler is already deleted.");
    } else {
      throw err;
    }
```

```
  }

  return { ...context };
};
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API AWS SDK per JavaScript .
  - CreateCrawler
  - CreateJob
  - DeleteCrawler
  - DeleteDatabase
  - DeleteJob
  - DeleteTable
  - GetCrawler
  - GetDatabase
  - GetDatabases
  - GetJob
  - GetJobRun
  - GetJobRuns
  - GetTables
  - ListJobs
  - StartCrawler
  - StartJobRun

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro da GitHub fare. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun main(args: Array<String>) {
    val usage = """
        Usage:
            <iam> <s3Path> <cron> <dbName> <crawlerName> <jobName>
 <scriptLocation> <locationUri>

        Where:
            iam - The Amazon Resource Name (ARN) of the AWS Identity and Access
 Management (IAM) role that has AWS Glue and Amazon Simple Storage Service
 (Amazon S3) permissions.
            s3Path - The Amazon Simple Storage Service (Amazon S3) target that
 contains data (for example, CSV data).
            cron - A cron expression used to specify the schedule (for example,
 cron(15 12 * * ? *).
            dbName - The database name.
            crawlerName - The name of the crawler.
            jobName - The name you assign to this job definition.
            scriptLocation - Specifies the Amazon S3 path to a script that runs a
 job.
            locationUri - Specifies the location of the database
        """

    if (args.size != 8) {
        println(usage)
        exitProcess(1)
    }

    val iam = args[0]
    val s3Path = args[1]
    val cron = args[2]
    val dbName = args[3]
    val crawlerName = args[4]
    val jobName = args[5]
    val scriptLocation = args[6]
    val locationUri = args[7]

    println("About to start the AWS Glue Scenario")
    createDatabase(dbName, locationUri)
    createCrawler(iam, s3Path, cron, dbName, crawlerName)
    getCrawler(crawlerName)
    startCrawler(crawlerName)
    getDatabase(dbName)
    getGlueTables(dbName)
```

```
    createJob(jobName, iam, scriptLocation)
    startJob(jobName)
    getJobs()
    getJobRuns(jobName)
    deleteJob(jobName)
    println("*** Wait for 5 MIN so the $crawlerName is ready to be deleted")
    TimeUnit.MINUTES.sleep(5)
    deleteMyDatabase(dbName)
    deleteCrawler(crawlerName)
}

suspend fun createDatabase(
    dbName: String?,
    locationUriVal: String?,
) {
    val input =
        DatabaseInput {
            description = "Built with the AWS SDK for Kotlin"
            name = dbName
            locationUri = locationUriVal
        }

    val request =
        CreateDatabaseRequest {
            databaseInput = input
        }

    GlueClient.fromEnvironment { region = "us-east-1" }.use { glueClient ->
        glueClient.createDatabase(request)
        println("The database was successfully created")
    }
}

suspend fun createCrawler(
    iam: String?,
    s3Path: String?,
    cron: String?,
    dbName: String?,
    crawlerName: String,
) {
    val s3Target =
        S3Target {
            path = s3Path
        }
```

```
    val targetList = ArrayList<S3Target>()
    targetList.add(s3Target)

    val targetOb =
        CrawlerTargets {
            s3Targets = targetList
        }

    val crawlerRequest =
        CreateCrawlerRequest {
            databaseName = dbName
            name = crawlerName
            description = "Created by the AWS Glue Java API"
            targets = targetOb
            role = iam
            schedule = cron
        }

    GlueClient.fromEnvironment { region = "us-east-1" }.use { glueClient ->
        glueClient.createCrawler(crawlerRequest)
        println("$crawlerName was successfully created")
    }
}

suspend fun getCrawler(crawlerName: String?) {
    val request =
        GetCrawlerRequest {
            name = crawlerName
        }

    GlueClient.fromEnvironment { region = "us-east-1" }.use { glueClient ->
        val response = glueClient.getCrawler(request)
        val role = response.crawler?.role
        println("The role associated with this crawler is $role")
    }
}

suspend fun startCrawler(crawlerName: String) {
    val crawlerRequest =
        StartCrawlerRequest {
            name = crawlerName
        }
```

```
    GlueClient.fromEnvironment { region = "us-east-1" }.use { glueClient ->
        glueClient.startCrawler(crawlerRequest)
        println("$crawlerName was successfully started.")
    }
}

suspend fun getDatabase(databaseName: String?) {
    val request =
        GetDatabaseRequest {
            name = databaseName
        }

    GlueClient.fromEnvironment { region = "us-east-1" }.use { glueClient ->
        val response = glueClient.getDatabase(request)
        val dbDesc = response.database?.description
        println("The database description is $dbDesc")
    }
}

suspend fun getGlueTables(dbName: String?) {
    val tableRequest =
        GetTablesRequest {
            databaseName = dbName
        }

    GlueClient.fromEnvironment { region = "us-east-1" }.use { glueClient ->
        val response = glueClient.getTables(tableRequest)
        response.tableList?.forEach { tableName ->
            println("Table name is ${tableName.name}")
        }
    }
}

suspend fun startJob(jobNameVal: String?) {
    val runRequest =
        StartJobRunRequest {
            workerType = WorkerType.G1X
            numberOfWorkers = 10
            jobName = jobNameVal
        }

    GlueClient.fromEnvironment { region = "us-east-1" }.use { glueClient ->
        val response = glueClient.startJobRun(runRequest)
        println("The job run Id is ${response.jobRunId}")
```

```
    }
}

suspend fun createJob(
    jobName: String,
    iam: String?,
    scriptLocationVal: String?,
) {
    val commandOb =
        JobCommand {
            pythonVersion = "3"
            name = "MyJob1"
            scriptLocation = scriptLocationVal
        }

    val jobRequest =
        CreateJobRequest {
            description = "A Job created by using the AWS SDK for Java V2"
            glueVersion = "2.0"
            workerType = WorkerType.G1X
            numberOfWorkers = 10
            name = jobName
            role = iam
            command = commandOb
        }

    GlueClient.fromEnvironment { region = "us-east-1" }.use { glueClient ->
        glueClient.createJob(jobRequest)
        println("$jobName was successfully created.")
    }
}

suspend fun getJobs() {
    val request =
        GetJobsRequest {
            maxResults = 10
        }

    GlueClient.fromEnvironment { region = "us-east-1" }.use { glueClient ->
        val response = glueClient.getJobs(request)
        response.jobs?.forEach { job ->
            println("Job name is ${job.name}")
        }
    }
```

```
}

suspend fun getJobRuns(jobNameVal: String?) {
    val request =
        GetJobRunsRequest {
            jobName = jobNameVal
        }

    GlueClient.fromEnvironment { region = "us-east-1" }.use { glueClient ->
        val response = glueClient.getJobRuns(request)
        response.jobRuns?.forEach { job ->
            println("Job name is ${job.jobName}")
        }
    }
}

suspend fun deleteJob(jobNameVal: String) {
    val jobRequest =
        DeleteJobRequest {
            jobName = jobNameVal
        }

    GlueClient.fromEnvironment { region = "us-east-1" }.use { glueClient ->
        glueClient.deleteJob(jobRequest)
        println("$jobNameVal was successfully deleted")
    }
}

suspend fun deleteMyDatabase(databaseName: String) {
    val request =
        DeleteDatabaseRequest {
            name = databaseName
        }

    GlueClient.fromEnvironment { region = "us-east-1" }.use { glueClient ->
        glueClient.deleteDatabase(request)
        println("$databaseName was successfully deleted")
    }
}

suspend fun deleteCrawler(crawlerName: String) {
    val request =
        DeleteCrawlerRequest {
            name = crawlerName
```

```
        }
    GlueClient.fromEnvironment { region = "us-east-1" }.use { glueClient ->
        glueClient.deleteCrawler(request)
        println("$crawlerName was deleted")
    }
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API AWS SDK per Kotlin.
  - CreateCrawler
  - CreateJob
  - DeleteCrawler
  - DeleteDatabase
  - DeleteJob
  - DeleteTable
  - GetCrawler
  - GetDatabase
  - GetDatabases
  - GetJob
  - GetJobRun
  - GetJobRuns
  - GetTables
  - ListJobs
  - StartCrawler
  - StartJobRun

PHP

SDK per PHP

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
namespace Glue;

use Aws\Glue\GlueClient;
use Aws\S3\S3Client;
use AwsUtilities\AWSServiceClass;
use GuzzleHttp\Psr7\Stream;
use Iam\IAMService;

class GettingStartedWithGlue
{
    public function run()
    {
        echo("\n");
        echo("--------------------------------------\n");
        print("Welcome to the AWS Glue getting started demo using PHP!\n");
        echo("--------------------------------------\n");

        $clientArgs = [
            'region' => 'us-west-2',
            'version' => 'latest',
            'profile' => 'default',
        ];
        $uniqid = uniqid();

        $glueClient = new GlueClient($clientArgs);
        $glueService = new GlueService($glueClient);
        $iamService = new IAMService();
        $crawlerName = "example-crawler-test-" . $uniqid;

        AWSServiceClass::$waitTime = 5;
        AWSServiceClass::$maxWaitAttempts = 20;
```

```
        $role = $iamService->getRole("AWSGlueServiceRole-DocExample");

        $databaseName = "doc-example-database-$uniqid";
        $path = 's3://crawler-public-us-east-1/flight/2016/csv';
        $glueService->createCrawler($crawlerName, $role['Role']['Arn'],
$databaseName, $path);
        $glueService->startCrawler($crawlerName);

        echo "Waiting for crawler";
        do {
            $crawler = $glueService->getCrawler($crawlerName);
            echo ".";
            sleep(10);
        } while ($crawler['Crawler']['State'] != "READY");
        echo "\n";

        $database = $glueService->getDatabase($databaseName);
        echo "Found a database named " . $database['Database']['Name'] . "\n";

        //Upload job script
        $s3client = new S3Client($clientArgs);
        $bucketName = "test-glue-bucket-" . $uniqid;
        $s3client->createBucket([
            'Bucket' => $bucketName,
            'CreateBucketConfiguration' => ['LocationConstraint' => 'us-west-2'],
        ]);

        $s3client->putObject([
            'Bucket' => $bucketName,
            'Key' => 'run_job.py',
            'SourceFile' => __DIR__ . '/flight_etl_job_script.py'
        ]);
        $s3client->putObject([
            'Bucket' => $bucketName,
            'Key' => 'setup_scenario_getting_started.yaml',
            'SourceFile' => __DIR__ . '/setup_scenario_getting_started.yaml'
        ]);

        $tables = $glueService->getTables($databaseName);

        $jobName = 'test-job-' . $uniqid;
        $scriptLocation = "s3://$bucketName/run_job.py";
        $job = $glueService->createJob($jobName, $role['Role']['Arn'],
$scriptLocation);
```

```
        $outputBucketUrl = "s3://$bucketName";
        $runId = $glueService->startJobRun($jobName, $databaseName, $tables,
$outputBucketUrl)['JobRunId'];

        echo "waiting for job";
        do {
            $jobRun = $glueService->getJobRun($jobName, $runId);
            echo ".";
            sleep(10);
        } while (!array_intersect([$jobRun['JobRun']['JobRunState']],
['SUCCEEDED', 'STOPPED', 'FAILED', 'TIMEOUT']));
        echo "\n";

        $jobRuns = $glueService->getJobRuns($jobName);

        $objects = $s3client->listObjects([
            'Bucket' => $bucketName,
        ])['Contents'];

        foreach ($objects as $object) {
            echo $object['Key'] . "\n";
        }

        echo "Downloading " . $objects[1]['Key'] . "\n";
        /** @var Stream $downloadObject */
        $downloadObject = $s3client->getObject([
            'Bucket' => $bucketName,
            'Key' => $objects[1]['Key'],
        ])['Body']->getContents();
        echo "Here is the first 1000 characters in the object.";
        echo substr($downloadObject, 0, 1000);

        $jobs = $glueService->listJobs();
        echo "Current jobs:\n";
        foreach ($jobs['JobNames'] as $jobsName) {
            echo "{$jobsName}\n";
        }

        echo "Delete the job.\n";
        $glueClient->deleteJob([
            'JobName' => $job['Name'],
        ]);
```

```
        echo "Delete the tables.\n";
        foreach ($tables['TableList'] as $table) {
            $glueService->deleteTable($table['Name'], $databaseName);
        }

        echo "Delete the databases.\n";
        $glueClient->deleteDatabase([
            'Name' => $databaseName,
        ]);

        echo "Delete the crawler.\n";
        $glueClient->deleteCrawler([
            'Name' => $crawlerName,
        ]);

        $deleteObjects = $s3client->listObjectsV2([
            'Bucket' => $bucketName,
        ]);
        echo "Delete all objects in the bucket.\n";
        $deleteObjects = $s3client->deleteObjects([
            'Bucket' => $bucketName,
            'Delete' => [
                'Objects' => $deleteObjects['Contents'],
            ]
        ]);
        echo "Delete the bucket.\n";
        $s3client->deleteBucket(['Bucket' => $bucketName]);

        echo "This job was brought to you by the number $uniqid\n";
    }
}

namespace Glue;

use Aws\Glue\GlueClient;
use Aws\Result;

use function PHPUnit\Framework\isEmpty;

class GlueService extends \AwsUtilities\AWSServiceClass
{
    protected GlueClient $glueClient;

    public function __construct($glueClient)
```

```
    {
        $this->glueClient = $glueClient;
    }

    public function getCrawler($crawlerName)
    {
        return $this->customWaiter(function () use ($crawlerName) {
            return $this->glueClient->getCrawler([
                'Name' => $crawlerName,
            ]);
        });
    }

    public function createCrawler($crawlerName, $role, $databaseName, $path):
Result
    {
        return $this->customWaiter(function () use ($crawlerName, $role,
$databaseName, $path) {
            return $this->glueClient->createCrawler([
                'Name' => $crawlerName,
                'Role' => $role,
                'DatabaseName' => $databaseName,
                'Targets' => [
                    'S3Targets' =>
                        [[
                            'Path' => $path,
                        ]]
                ],
            ]);
        });
    }

    public function startCrawler($crawlerName): Result
    {
        return $this->glueClient->startCrawler([
            'Name' => $crawlerName,
        ]);
    }

    public function getDatabase(string $databaseName): Result
    {
        return $this->customWaiter(function () use ($databaseName) {
            return $this->glueClient->getDatabase([
                'Name' => $databaseName,
```

```
            ]);
        });
    }

    public function getTables($databaseName): Result
    {
        return $this->glueClient->getTables([
            'DatabaseName' => $databaseName,
        ]);
    }

    public function createJob($jobName, $role, $scriptLocation, $pythonVersion =
'3', $glueVersion = '3.0'): Result
    {
        return $this->glueClient->createJob([
            'Name' => $jobName,
            'Role' => $role,
            'Command' => [
                'Name' => 'glueetl',
                'ScriptLocation' => $scriptLocation,
                'PythonVersion' => $pythonVersion,
            ],
            'GlueVersion' => $glueVersion,
        ]);
    }

    public function startJobRun($jobName, $databaseName, $tables,
$outputBucketUrl): Result
    {
        return $this->glueClient->startJobRun([
            'JobName' => $jobName,
            'Arguments' => [
                'input_database' => $databaseName,
                'input_table' => $tables['TableList'][0]['Name'],
                'output_bucket_url' => $outputBucketUrl,
                '--input_database' => $databaseName,
                '--input_table' => $tables['TableList'][0]['Name'],
                '--output_bucket_url' => $outputBucketUrl,
            ],
        ]);
    }

    public function listJobs($maxResults = null, $nextToken = null, $tags = []):
Result
```

```
    {
        $arguments = [];
        if ($maxResults) {
            $arguments['MaxResults'] = $maxResults;
        }
        if ($nextToken) {
            $arguments['NextToken'] = $nextToken;
        }
        if (!empty($tags)) {
            $arguments['Tags'] = $tags;
        }
        return $this->glueClient->listJobs($arguments);
    }

    public function getJobRuns($jobName, $maxResults = 0, $nextToken = ''):
Result
    {
        $arguments = ['JobName' => $jobName];
        if ($maxResults) {
            $arguments['MaxResults'] = $maxResults;
        }
        if ($nextToken) {
            $arguments['NextToken'] = $nextToken;
        }
        return $this->glueClient->getJobRuns($arguments);
    }

    public function getJobRun($jobName, $runId, $predecessorsIncluded = false):
Result
    {
        return $this->glueClient->getJobRun([
            'JobName' => $jobName,
            'RunId' => $runId,
            'PredecessorsIncluded' => $predecessorsIncluded,
        ]);
    }

    public function deleteJob($jobName)
    {
        return $this->glueClient->deleteJob([
            'JobName' => $jobName,
        ]);
    }
```

```
    public function deleteTable($tableName, $databaseName)
    {
        return $this->glueClient->deleteTable([
            'DatabaseName' => $databaseName,
            'Name' => $tableName,
        ]);
    }

    public function deleteDatabase($databaseName)
    {
        return $this->glueClient->deleteDatabase([
            'Name' => $databaseName,
        ]);
    }

    public function deleteCrawler($crawlerName)
    {
        return $this->glueClient->deleteCrawler([
            'Name' => $crawlerName,
        ]);
    }
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API AWS SDK per PHP .
  - CreateCrawler
  - CreateJob
  - DeleteCrawler
  - DeleteDatabase
  - DeleteJob
  - DeleteTable
  - GetCrawler
  - GetDatabase
  - GetDatabases
  - GetJob
  - GetJobRun
  - GetJobRuns

- GetTables
- ListJobs
- StartCrawler
- StartJobRun

Python

SDK per Python (Boto3)

> **Note**
>
> C'è dell'altro GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Crea una classe che racchiuda le AWS Glue funzioni utilizzate nello scenario.

```
class GlueWrapper:
    """Encapsulates AWS Glue actions."""

    def __init__(self, glue_client):
        """
        :param glue_client: A Boto3 Glue client.
        """
        self.glue_client = glue_client


    def get_crawler(self, name):
        """
        Gets information about a crawler.

        :param name: The name of the crawler to look up.
        :return: Data about the crawler.
        """
        crawler = None
        try:
            response = self.glue_client.get_crawler(Name=name)
            crawler = response["Crawler"]
        except ClientError as err:
            if err.response["Error"]["Code"] == "EntityNotFoundException":
```

```
                logger.info("Crawler %s doesn't exist.", name)
            else:
                logger.error(
                    "Couldn't get crawler %s. Here's why: %s: %s",
                    name,
                    err.response["Error"]["Code"],
                    err.response["Error"]["Message"],
                )
                raise
        return crawler


    def create_crawler(self, name, role_arn, db_name, db_prefix, s3_target):
        """
        Creates a crawler that can crawl the specified target and populate a
        database in your AWS Glue Data Catalog with metadata that describes the
data
        in the target.

        :param name: The name of the crawler.
        :param role_arn: The Amazon Resource Name (ARN) of an AWS Identity and
Access
                         Management (IAM) role that grants permission to let AWS
Glue
                         access the resources it needs.
        :param db_name: The name to give the database that is created by the
crawler.
        :param db_prefix: The prefix to give any database tables that are created
by
                          the crawler.
        :param s3_target: The URL to an S3 bucket that contains data that is
                          the target of the crawler.
        """
        try:
            self.glue_client.create_crawler(
                Name=name,
                Role=role_arn,
                DatabaseName=db_name,
                TablePrefix=db_prefix,
                Targets={"S3Targets": [{"Path": s3_target}]},
            )
        except ClientError as err:
            logger.error(
                "Couldn't create crawler. Here's why: %s: %s",
```

```
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise


    def start_crawler(self, name):
        """
        Starts a crawler. The crawler crawls its configured target and creates
        metadata that describes the data it finds in the target data source.

        :param name: The name of the crawler to start.
        """
        try:
            self.glue_client.start_crawler(Name=name)
        except ClientError as err:
            logger.error(
                "Couldn't start crawler %s. Here's why: %s: %s",
                name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise


    def get_database(self, name):
        """
        Gets information about a database in your Data Catalog.

        :param name: The name of the database to look up.
        :return: Information about the database.
        """
        try:
            response = self.glue_client.get_database(Name=name)
        except ClientError as err:
            logger.error(
                "Couldn't get database %s. Here's why: %s: %s",
                name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return response["Database"]
```

```
    def get_tables(self, db_name):
        """
        Gets a list of tables in a Data Catalog database.

        :param db_name: The name of the database to query.
        :return: The list of tables in the database.
        """
        try:
            response = self.glue_client.get_tables(DatabaseName=db_name)
        except ClientError as err:
            logger.error(
                "Couldn't get tables %s. Here's why: %s: %s",
                db_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return response["TableList"]


    def create_job(self, name, description, role_arn, script_location):
        """
        Creates a job definition for an extract, transform, and load (ETL) job
that can
        be run by AWS Glue.

        :param name: The name of the job definition.
        :param description: The description of the job definition.
        :param role_arn: The ARN of an IAM role that grants AWS Glue the
permissions
                         it requires to run the job.
        :param script_location: The Amazon S3 URL of a Python ETL script that is
run as
                                part of the job. The script defines how the data
is
                                transformed.
        """
        try:
            self.glue_client.create_job(
                Name=name,
                Description=description,
```

```
                Role=role_arn,
                Command={
                    "Name": "glueetl",
                    "ScriptLocation": script_location,
                    "PythonVersion": "3",
                },
                GlueVersion="3.0",
            )
        except ClientError as err:
            logger.error(
                "Couldn't create job %s. Here's why: %s: %s",
                name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise


    def start_job_run(self, name, input_database, input_table,
output_bucket_name):
        """
        Starts a job run. A job run extracts data from the source, transforms it,
        and loads it to the output bucket.

        :param name: The name of the job definition.
        :param input_database: The name of the metadata database that contains
tables
                               that describe the source data. This is typically
created
                               by a crawler.
        :param input_table: The name of the table in the metadata database that
                            describes the source data.
        :param output_bucket_name: The S3 bucket where the output is written.
        :return: The ID of the job run.
        """
        try:
            # The custom Arguments that are passed to this function are used by
the
            # Python ETL script to determine the location of input and output
data.
            response = self.glue_client.start_job_run(
                JobName=name,
                Arguments={
                    "--input_database": input_database,
```

```
                "--input_table": input_table,
                "--output_bucket_url": f"s3://{output_bucket_name}/",
            },
        )
    except ClientError as err:
        logger.error(
            "Couldn't start job run %s. Here's why: %s: %s",
            name,
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise
    else:
        return response["JobRunId"]


def list_jobs(self):
    """
    Lists the names of job definitions in your account.

    :return: The list of job definition names.
    """
    try:
        response = self.glue_client.list_jobs()
    except ClientError as err:
        logger.error(
            "Couldn't list jobs. Here's why: %s: %s",
            err.response["Error"]["Code"],
            err.response["Error"]["Message"],
        )
        raise
    else:
        return response["JobNames"]


def get_job_runs(self, job_name):
    """
    Gets information about runs that have been performed for a specific job
    definition.

    :param job_name: The name of the job definition to look up.
    :return: The list of job runs.
    """
    try:
```

```
            response = self.glue_client.get_job_runs(JobName=job_name)
        except ClientError as err:
            logger.error(
                "Couldn't get job runs for %s. Here's why: %s: %s",
                job_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return response["JobRuns"]


    def get_job_run(self, name, run_id):
        """
        Gets information about a single job run.

        :param name: The name of the job definition for the run.
        :param run_id: The ID of the run.
        :return: Information about the run.
        """
        try:
            response = self.glue_client.get_job_run(JobName=name, RunId=run_id)
        except ClientError as err:
            logger.error(
                "Couldn't get job run %s/%s. Here's why: %s: %s",
                name,
                run_id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return response["JobRun"]


    def delete_job(self, job_name):
        """
        Deletes a job definition. This also deletes data about all runs that are
        associated with this job definition.

        :param job_name: The name of the job definition to delete.
        """
        try:
```

```
            self.glue_client.delete_job(JobName=job_name)
        except ClientError as err:
            logger.error(
                "Couldn't delete job %s. Here's why: %s: %s",
                job_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise


    def delete_table(self, db_name, table_name):
        """
        Deletes a table from a metadata database.

        :param db_name: The name of the database that contains the table.
        :param table_name: The name of the table to delete.
        """
        try:
            self.glue_client.delete_table(DatabaseName=db_name, Name=table_name)
        except ClientError as err:
            logger.error(
                "Couldn't delete table %s. Here's why: %s: %s",
                table_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise


    def delete_database(self, name):
        """
        Deletes a metadata database from your Data Catalog.

        :param name: The name of the database to delete.
        """
        try:
            self.glue_client.delete_database(Name=name)
        except ClientError as err:
            logger.error(
                "Couldn't delete database %s. Here's why: %s: %s",
                name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
```

```
            )
            raise


    def delete_crawler(self, name):
        """
        Deletes a crawler.

        :param name: The name of the crawler to delete.
        """
        try:
            self.glue_client.delete_crawler(Name=name)
        except ClientError as err:
            logger.error(
                "Couldn't delete crawler %s. Here's why: %s: %s",
                name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
```

Creazione di una classe che esegue lo scenario.

```
class GlueCrawlerJobScenario:
    """
    Encapsulates a scenario that shows how to create an AWS Glue crawler and job
 and use
    them to transform data from CSV to JSON format.
    """

    def __init__(self, glue_client, glue_service_role, glue_bucket):
        """
        :param glue_client: A Boto3 AWS Glue client.
        :param glue_service_role: An AWS Identity and Access Management (IAM)
 role
                                  that AWS Glue can assume to gain access to the
                                  resources it requires.
        :param glue_bucket: An S3 bucket that can hold a job script and output
 data
```

```
                            from AWS Glue job runs.
        """
        self.glue_client = glue_client
        self.glue_service_role = glue_service_role
        self.glue_bucket = glue_bucket

    @staticmethod
    def wait(seconds, tick=12):
        """
        Waits for a specified number of seconds, while also displaying an
animated
        spinner.

        :param seconds: The number of seconds to wait.
        :param tick: The number of frames per second used to animate the spinner.
        """
        progress = "|/-\\"
        waited = 0
        while waited < seconds:
            for frame in range(tick):
                sys.stdout.write(f"\r{progress[frame % len(progress)]}")
                sys.stdout.flush()
                time.sleep(1 / tick)
            waited += 1

    def upload_job_script(self, job_script):
        """
        Uploads a Python ETL script to an S3 bucket. The script is used by the
AWS Glue
        job to transform data.

        :param job_script: The relative path to the job script.
        """
        try:
            self.glue_bucket.upload_file(Filename=job_script, Key=job_script)
            print(f"Uploaded job script '{job_script}' to the example bucket.")
        except S3UploadFailedError as err:
            logger.error("Couldn't upload job script. Here's why: %s", err)
            raise

    def run(self, crawler_name, db_name, db_prefix, data_source, job_script,
job_name):
        """
```

```
        Runs the scenario. This is an interactive experience that runs at a
command
        prompt and asks you for input throughout.

        :param crawler_name: The name of the crawler used in the scenario. If the
                             crawler does not exist, it is created.
        :param db_name: The name to give the metadata database created by the
crawler.
        :param db_prefix: The prefix to give tables added to the database by the
                          crawler.
        :param data_source: The location of the data source that is targeted by
the
                            crawler and extracted during job runs.
        :param job_script: The job script that is used to transform data during
job
                           runs.
        :param job_name: The name to give the job definition that is created
during the
                         scenario.
        """
        wrapper = GlueWrapper(self.glue_client)
        print(f"Checking for crawler {crawler_name}.")
        crawler = wrapper.get_crawler(crawler_name)
        if crawler is None:
            print(f"Creating crawler {crawler_name}.")
            wrapper.create_crawler(
                crawler_name,
                self.glue_service_role.arn,
                db_name,
                db_prefix,
                data_source,
            )
            print(f"Created crawler {crawler_name}.")
            crawler = wrapper.get_crawler(crawler_name)
        pprint(crawler)
        print("-" * 88)

        print(
            f"When you run the crawler, it crawls data stored in {data_source}
and "
            f"creates a metadata database in the AWS Glue Data Catalog that
describes "
            f"the data in the data source."
        )
```

```
        print("In this example, the source data is in CSV format.")
        ready = False
        while not ready:
            ready = Question.ask_question(
                "Ready to start the crawler? (y/n) ", Question.is_yesno
            )
        wrapper.start_crawler(crawler_name)
        print("Let's wait for the crawler to run. This typically takes a few
minutes.")
        crawler_state = None
        while crawler_state != "READY":
            self.wait(10)
            crawler = wrapper.get_crawler(crawler_name)
            crawler_state = crawler["State"]
            print(f"Crawler is {crawler['State']}.")
        print("-" * 88)

        database = wrapper.get_database(db_name)
        print(f"The crawler created database {db_name}:")
        pprint(database)
        print(f"The database contains these tables:")
        tables = wrapper.get_tables(db_name)
        for index, table in enumerate(tables):
            print(f"\t{index + 1}. {table['Name']}")
        table_index = Question.ask_question(
            f"Enter the number of a table to see more detail: ",
            Question.is_int,
            Question.in_range(1, len(tables)),
        )
        pprint(tables[table_index - 1])
        print("-" * 88)

        print(f"Creating job definition {job_name}.")
        wrapper.create_job(
            job_name,
            "Getting started example job.",
            self.glue_service_role.arn,
            f"s3://{self.glue_bucket.name}/{job_script}",
        )
        print("Created job definition.")
        print(
            f"When you run the job, it extracts data from {data_source},
transforms it "
            f"by using the {job_script} script, and loads the output into "
```

```
            f"S3 bucket {self.glue_bucket.name}."
        )
        print(
            "In this example, the data is transformed from CSV to JSON, and only
a few "
            "fields are included in the output."
        )
        job_run_status = None
        if Question.ask_question(f"Ready to run? (y/n) ", Question.is_yesno):
            job_run_id = wrapper.start_job_run(
                job_name, db_name, tables[0]["Name"], self.glue_bucket.name
            )
            print(f"Job {job_name} started. Let's wait for it to run.")
            while job_run_status not in ["SUCCEEDED", "STOPPED", "FAILED",
"TIMEOUT"]:
                self.wait(10)
                job_run = wrapper.get_job_run(job_name, job_run_id)
                job_run_status = job_run["JobRunState"]
                print(f"Job {job_name}/{job_run_id} is {job_run_status}.")
        print("-" * 88)

        if job_run_status == "SUCCEEDED":
            print(
                f"Data from your job run is stored in your S3 bucket
'{self.glue_bucket.name}':"
            )
            try:
                keys = [
                    obj.key for obj in
self.glue_bucket.objects.filter(Prefix="run-")
                ]
                for index, key in enumerate(keys):
                    print(f"\t{index + 1}: {key}")
                lines = 4
                key_index = Question.ask_question(
                    f"Enter the number of a block to download it and see the
first {lines} "
                    f"lines of JSON output in the block: ",
                    Question.is_int,
                    Question.in_range(1, len(keys)),
                )
                job_data = io.BytesIO()
                self.glue_bucket.download_fileobj(keys[key_index - 1], job_data)
                job_data.seek(0)
```

```
                for _ in range(lines):
                    print(job_data.readline().decode("utf-8"))
            except ClientError as err:
                logger.error(
                    "Couldn't get job run data. Here's why: %s: %s",
                    err.response["Error"]["Code"],
                    err.response["Error"]["Message"],
                )
                raise
            print("-" * 88)

        job_names = wrapper.list_jobs()
        if job_names:
            print(f"Your account has {len(job_names)} jobs defined:")
            for index, job_name in enumerate(job_names):
                print(f"\t{index + 1}. {job_name}")
            job_index = Question.ask_question(
                f"Enter a number between 1 and {len(job_names)} to see the list
of runs for "
                f"a job: ",
                Question.is_int,
                Question.in_range(1, len(job_names)),
            )
            job_runs = wrapper.get_job_runs(job_names[job_index - 1])
            if job_runs:
                print(f"Found {len(job_runs)} runs for job {job_names[job_index -
1]}:")
                for index, job_run in enumerate(job_runs):
                    print(
                        f"\t{index + 1}. {job_run['JobRunState']} on "
                        f"{job_run['CompletedOn']:%Y-%m-%d %H:%M:%S}"
                    )
                run_index = Question.ask_question(
                    f"Enter a number between 1 and {len(job_runs)} to see details
for a run: ",
                    Question.is_int,
                    Question.in_range(1, len(job_runs)),
                )
                pprint(job_runs[run_index - 1])
            else:
                print(f"No runs found for job {job_names[job_index - 1]}")
        else:
            print("Your account doesn't have any jobs defined.")
        print("-" * 88)
```

```
        print(
            f"Let's clean up. During this example we created job definition
 '{job_name}'."
        )
        if Question.ask_question(
            "Do you want to delete the definition and all runs? (y/n) ",
            Question.is_yesno,
        ):
            wrapper.delete_job(job_name)
            print(f"Job definition '{job_name}' deleted.")
        tables = wrapper.get_tables(db_name)
        print(f"We also created database '{db_name}' that contains these
 tables:")
        for table in tables:
            print(f"\t{table['Name']}")
        if Question.ask_question(
            "Do you want to delete the tables and the database? (y/n) ",
            Question.is_yesno,
        ):
            for table in tables:
                wrapper.delete_table(db_name, table["Name"])
                print(f"Deleted table {table['Name']}.")
            wrapper.delete_database(db_name)
            print(f"Deleted database {db_name}.")
        print(f"We also created crawler '{crawler_name}'.")
        if Question.ask_question(
            "Do you want to delete the crawler? (y/n) ", Question.is_yesno
        ):
            wrapper.delete_crawler(crawler_name)
            print(f"Deleted crawler {crawler_name}.")
        print("-" * 88)


def parse_args(args):
    """
    Parse command line arguments.

    :param args: The command line arguments.
    :return: The parsed arguments.
    """
    parser = argparse.ArgumentParser(
        description="Runs the AWS Glue getting started with crawlers and jobs
 scenario. "
```

```
        "Before you run this scenario, set up scaffold resources by running "
        "'python scaffold.py deploy'."
    )
    parser.add_argument(
        "role_name",
        help="The name of an IAM role that AWS Glue can assume. This role must
 grant access "
        "to Amazon S3 and to the permissions granted by the AWSGlueServiceRole "
        "managed policy.",
    )
    parser.add_argument(
        "bucket_name",
        help="The name of an S3 bucket that AWS Glue can access to get the job
 script and "
        "put job results.",
    )
    parser.add_argument(
        "--job_script",
        default="flight_etl_job_script.py",
        help="The name of the job script file that is used in the scenario.",
    )
    return parser.parse_args(args)


def main():
    args = parse_args(sys.argv[1:])
    try:
        print("-" * 88)
        print(
            "Welcome to the AWS Glue getting started with crawlers and jobs
 scenario."
        )
        print("-" * 88)
        scenario = GlueCrawlerJobScenario(
            boto3.client("glue"),
            boto3.resource("iam").Role(args.role_name),
            boto3.resource("s3").Bucket(args.bucket_name),
        )
        scenario.upload_job_script(args.job_script)
        scenario.run(
            "doc-example-crawler",
            "doc-example-database",
            "doc-example-",
            "s3://crawler-public-us-east-1/flight/2016/csv",
```

```
            args.job_script,
            "doc-example-job",
        )
        print("-" * 88)
        print(
            "To destroy scaffold resources, including the IAM role and S3 bucket
 "
            "used in this scenario, run 'python scaffold.py destroy'."
        )
        print("\nThanks for watching!")
        print("-" * 88)
    except Exception:
        logging.exception("Something went wrong with the example.")
```

Create uno script ETL utilizzato da AWS Glue per estrarre, trasformare e caricare i dati durante l'esecuzione dei job.

```
import sys
from awsglue.transforms import *
from awsglue.utils import getResolvedOptions
from pyspark.context import SparkContext
from awsglue.context import GlueContext
from awsglue.job import Job

"""
These custom arguments must be passed as Arguments to the StartJobRun request.
    --input_database    The name of a metadata database that is contained in
 your
                        AWS Glue Data Catalog and that contains tables that
 describe
                        the data to be processed.
    --input_table       The name of a table in the database that describes the
 data to
                        be processed.
    --output_bucket_url An S3 bucket that receives the transformed output data.
"""
args = getResolvedOptions(
    sys.argv, ["JOB_NAME", "input_database", "input_table", "output_bucket_url"]
)
sc = SparkContext()
```

```
glueContext = GlueContext(sc)
spark = glueContext.spark_session
job = Job(glueContext)
job.init(args["JOB_NAME"], args)

# Script generated for node S3 Flight Data.
S3FlightData_node1 = glueContext.create_dynamic_frame.from_catalog(
    database=args["input_database"],
    table_name=args["input_table"],
    transformation_ctx="S3FlightData_node1",
)

# This mapping performs two main functions:
# 1. It simplifies the output by removing most of the fields from the data.
# 2. It renames some fields. For example, `fl_date` is renamed to `flight_date`.
ApplyMapping_node2 = ApplyMapping.apply(
    frame=S3FlightData_node1,
    mappings=[
        ("year", "long", "year", "long"),
        ("month", "long", "month", "tinyint"),
        ("day_of_month", "long", "day", "tinyint"),
        ("fl_date", "string", "flight_date", "string"),
        ("carrier", "string", "carrier", "string"),
        ("fl_num", "long", "flight_num", "long"),
        ("origin_city_name", "string", "origin_city_name", "string"),
        ("origin_state_abr", "string", "origin_state_abr", "string"),
        ("dest_city_name", "string", "dest_city_name", "string"),
        ("dest_state_abr", "string", "dest_state_abr", "string"),
        ("dep_time", "long", "departure_time", "long"),
        ("wheels_off", "long", "wheels_off", "long"),
        ("wheels_on", "long", "wheels_on", "long"),
        ("arr_time", "long", "arrival_time", "long"),
        ("mon", "string", "mon", "string"),
    ],
    transformation_ctx="ApplyMapping_node2",
)

# Script generated for node Revised Flight Data.
RevisedFlightData_node3 = glueContext.write_dynamic_frame.from_options(
    frame=ApplyMapping_node2,
    connection_type="s3",
    format="json",
    connection_options={"path": args["output_bucket_url"], "partitionKeys": []},
    transformation_ctx="RevisedFlightData_node3",
```

```
)

job.commit()
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API AWS SDK per Python (Boto3).
  - CreateCrawler
  - CreateJob
  - DeleteCrawler
  - DeleteDatabase
  - DeleteJob
  - DeleteTable
  - GetCrawler
  - GetDatabase
  - GetDatabases
  - GetJob
  - GetJobRun
  - GetJobRuns
  - GetTables
  - ListJobs
  - StartCrawler
  - StartJobRun

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Crea una classe che racchiuda le AWS Glue funzioni utilizzate nello scenario.

```
# The `GlueWrapper` class serves as a wrapper around the AWS Glue API, providing
 a simplified interface for common operations.
# It encapsulates the functionality of the AWS SDK for Glue and provides methods
 for interacting with Glue crawlers, databases, tables, jobs, and S3 resources.
# The class initializes with a Glue client and a logger, allowing it to make API
 calls and log any errors or informational messages.
class GlueWrapper
  def initialize(glue_client, logger)
    @glue_client = glue_client
    @logger = logger
  end

  # Retrieves information about a specific crawler.
  #
  # @param name [String] The name of the crawler to retrieve information about.
  # @return [Aws::Glue::Types::Crawler, nil] The crawler object if found, or nil
 if not found.
  def get_crawler(name)
    @glue_client.get_crawler(name: name)
  rescue Aws::Glue::Errors::EntityNotFoundException
    @logger.info("Crawler #{name} doesn't exist.")
    false
  rescue Aws::Glue::Errors::GlueException => e
    @logger.error("Glue could not get crawler #{name}: \n#{e.message}")
    raise
  end

  # Creates a new crawler with the specified configuration.
  #
  # @param name [String] The name of the crawler.
  # @param role_arn [String] The ARN of the IAM role to be used by the crawler.
  # @param db_name [String] The name of the database where the crawler stores its
 metadata.
  # @param db_prefix [String] The prefix to be added to the names of tables that
 the crawler creates.
  # @param s3_target [String] The S3 path that the crawler will crawl.
  # @return [void]
  def create_crawler(name, role_arn, db_name, _db_prefix, s3_target)
    @glue_client.create_crawler(
      name: name,
      role: role_arn,
      database_name: db_name,
```

```
      targets: {
        s3_targets: [
          {
            path: s3_target
          }
        ]
      }
    )
  rescue Aws::Glue::Errors::GlueException => e
    @logger.error("Glue could not create crawler: \n#{e.message}")
    raise
  end

  # Starts a crawler with the specified name.
  #
  # @param name [String] The name of the crawler to start.
  # @return [void]
  def start_crawler(name)
    @glue_client.start_crawler(name: name)
  rescue Aws::Glue::Errors::ServiceError => e
    @logger.error("Glue could not start crawler #{name}: \n#{e.message}")
    raise
  end

  # Deletes a crawler with the specified name.
  #
  # @param name [String] The name of the crawler to delete.
  # @return [void]
  def delete_crawler(name)
    @glue_client.delete_crawler(name: name)
  rescue Aws::Glue::Errors::ServiceError => e
    @logger.error("Glue could not delete crawler #{name}: \n#{e.message}")
    raise
  end

  # Retrieves information about a specific database.
  #
  # @param name [String] The name of the database to retrieve information about.
  # @return [Aws::Glue::Types::Database, nil] The database object if found, or
 nil if not found.
  def get_database(name)
    response = @glue_client.get_database(name: name)
    response.database
  rescue Aws::Glue::Errors::GlueException => e
```

```
    @logger.error("Glue could not get database #{name}: \n#{e.message}")
    raise
  end

  # Retrieves a list of tables in the specified database.
  #
  # @param db_name [String] The name of the database to retrieve tables from.
  # @return [Array<Aws::Glue::Types::Table>]
  def get_tables(db_name)
    response = @glue_client.get_tables(database_name: db_name)
    response.table_list
  rescue Aws::Glue::Errors::GlueException => e
    @logger.error("Glue could not get tables #{db_name}: \n#{e.message}")
    raise
  end

  # Creates a new job with the specified configuration.
  #
  # @param name [String] The name of the job.
  # @param description [String] The description of the job.
  # @param role_arn [String] The ARN of the IAM role to be used by the job.
  # @param script_location [String] The location of the ETL script for the job.
  # @return [void]
  def create_job(name, description, role_arn, script_location)
    @glue_client.create_job(
      name: name,
      description: description,
      role: role_arn,
      command: {
        name: 'glueetl',
        script_location: script_location,
        python_version: '3'
      },
      glue_version: '3.0'
    )
  rescue Aws::Glue::Errors::GlueException => e
    @logger.error("Glue could not create job #{name}: \n#{e.message}")
    raise
  end

  # Starts a job run for the specified job.
  #
  # @param name [String] The name of the job to start the run for.
  # @param input_database [String] The name of the input database for the job.
```

```
  # @param input_table [String] The name of the input table for the job.
  # @param output_bucket_name [String] The name of the output S3 bucket for the
job.
  # @return [String] The ID of the started job run.
  def start_job_run(name, input_database, input_table, output_bucket_name)
    response = @glue_client.start_job_run(
      job_name: name,
      arguments: {
        '--input_database': input_database,
        '--input_table': input_table,
        '--output_bucket_url': "s3://#{output_bucket_name}/"
      }
    )
    response.job_run_id
  rescue Aws::Glue::Errors::GlueException => e
    @logger.error("Glue could not start job run #{name}: \n#{e.message}")
    raise
  end

  # Retrieves a list of jobs in AWS Glue.
  #
  # @return [Aws::Glue::Types::ListJobsResponse]
  def list_jobs
    @glue_client.list_jobs
  rescue Aws::Glue::Errors::GlueException => e
    @logger.error("Glue could not list jobs: \n#{e.message}")
    raise
  end

  # Retrieves a list of job runs for the specified job.
  #
  # @param job_name [String] The name of the job to retrieve job runs for.
  # @return [Array<Aws::Glue::Types::JobRun>]
  def get_job_runs(job_name)
    response = @glue_client.get_job_runs(job_name: job_name)
    response.job_runs
  rescue Aws::Glue::Errors::GlueException => e
    @logger.error("Glue could not get job runs: \n#{e.message}")
  end

  # Retrieves data for a specific job run.
  #
  # @param job_name [String] The name of the job run to retrieve data for.
  # @return [Glue::Types::GetJobRunResponse]
```

```
  def get_job_run(job_name, run_id)
    @glue_client.get_job_run(job_name: job_name, run_id: run_id)
  rescue Aws::Glue::Errors::GlueException => e
    @logger.error("Glue could not get job runs: \n#{e.message}")
  end

  # Deletes a job with the specified name.
  #
  # @param job_name [String] The name of the job to delete.
  # @return [void]
  def delete_job(job_name)
    @glue_client.delete_job(job_name: job_name)
  rescue Aws::Glue::Errors::ServiceError => e
    @logger.error("Glue could not delete job: \n#{e.message}")
  end

  # Deletes a table with the specified name.
  #
  # @param database_name [String] The name of the catalog database in which the
table resides.
  # @param table_name [String] The name of the table to be deleted.
  # @return [void]
  def delete_table(database_name, table_name)
    @glue_client.delete_table(database_name: database_name, name: table_name)
  rescue Aws::Glue::Errors::ServiceError => e
    @logger.error("Glue could not delete job: \n#{e.message}")
  end

  # Removes a specified database from a Data Catalog.
  #
  # @param database_name [String] The name of the database to delete.
  # @return [void]
  def delete_database(database_name)
    @glue_client.delete_database(name: database_name)
  rescue Aws::Glue::Errors::ServiceError => e
    @logger.error("Glue could not delete database: \n#{e.message}")
  end

  # Uploads a job script file to an S3 bucket.
  #
  # @param file_path [String] The local path of the job script file.
  # @param bucket_resource [Aws::S3::Bucket] The S3 bucket resource to upload the
file to.
  # @return [void]
```

```
  def upload_job_script(file_path, bucket_resource)
    File.open(file_path) do |file|
      bucket_resource.client.put_object({
                                          body: file,
                                          bucket: bucket_resource.name,
                                          key: file_path
                                        })
    end
  rescue Aws::S3::Errors::S3UploadFailedError => e
    @logger.error("S3 could not upload job script: \n#{e.message}")
    raise
  end
end
```

Creazione di una classe che esegue lo scenario.

```
class GlueCrawlerJobScenario
  def initialize(glue_client, glue_service_role, glue_bucket, logger)
    @glue_client = glue_client
    @glue_service_role = glue_service_role
    @glue_bucket = glue_bucket
    @logger = logger
  end

  def run(crawler_name, db_name, db_prefix, data_source, job_script, job_name)
    wrapper = GlueWrapper.new(@glue_client, @logger)
    setup_crawler(wrapper, crawler_name, db_name, db_prefix, data_source)
    query_database(wrapper, crawler_name, db_name)
    create_and_run_job(wrapper, job_script, job_name, db_name)
  end

  private

  def setup_crawler(wrapper, crawler_name, db_name, db_prefix, data_source)
    new_step(1, 'Create a crawler')
    crawler = wrapper.get_crawler(crawler_name)
    unless crawler
      puts "Creating crawler #{crawler_name}."
      wrapper.create_crawler(crawler_name, @glue_service_role.arn, db_name,
 db_prefix, data_source)
      puts "Successfully created #{crawler_name}."
    end
```

```
    wrapper.start_crawler(crawler_name)
    monitor_crawler(wrapper, crawler_name)
  end

  def monitor_crawler(wrapper, crawler_name)
    new_step(2, 'Monitor Crawler')
    crawler_state = nil
    until crawler_state == 'READY'
      custom_wait(15)
      crawler = wrapper.get_crawler(crawler_name)
      crawler_state = crawler[0]['state']
      print "Crawler status: #{crawler_state}".yellow
    end
  end

  def query_database(wrapper, _crawler_name, db_name)
    new_step(3, 'Query the database.')
    wrapper.get_database(db_name)
    puts "The crawler created database #{db_name}:"
    puts "Database contains tables: #{wrapper.get_tables(db_name).map { |t|
 t['name'] }}"
  end

  def create_and_run_job(wrapper, job_script, job_name, db_name)
    new_step(4, 'Create and run job.')
    wrapper.upload_job_script(job_script, @glue_bucket)
    wrapper.create_job(job_name, 'ETL Job', @glue_service_role.arn, "s3://
#{@glue_bucket.name}/#{job_script}")
    run_job(wrapper, job_name, db_name)
  end

  def run_job(wrapper, job_name, db_name)
    new_step(5, 'Run the job.')
    wrapper.start_job_run(job_name, db_name, wrapper.get_tables(db_name)[0]
['name'], @glue_bucket.name)
    job_run_status = nil
    until %w[SUCCEEDED FAILED STOPPED].include?(job_run_status)
      custom_wait(10)
      job_run = wrapper.get_job_runs(job_name)
      job_run_status = job_run[0]['job_run_state']
      print "Job #{job_name} status: #{job_run_status}".yellow
    end
  end
end
```

```
def main
  banner('../../helpers/banner.txt')
  puts 'Starting AWS Glue demo...'

  # Load resource names from YAML.
  resource_names = YAML.load_file('resource_names.yaml')

  # Setup services and resources.
  iam_role = Aws::IAM::Resource.new(region: 'us-
east-1').role(resource_names['glue_service_role'])
  s3_bucket = Aws::S3::Resource.new(region: 'us-
east-1').bucket(resource_names['glue_bucket'])

  # Instantiate scenario and run.
  scenario = GlueCrawlerJobScenario.new(Aws::Glue::Client.new(region: 'us-
east-1'), iam_role, s3_bucket, @logger)
  random_suffix = rand(10**4)
  scenario.run("crawler-#{random_suffix}", "db-#{random_suffix}", "prefix-
#{random_suffix}-", 's3://data_source',
               'job_script.py', "job-#{random_suffix}")

  puts 'Demo complete.'
end
```

Create uno script ETL utilizzato da AWS Glue per estrarre, trasformare e caricare i dati durante l'esecuzione dei job.

```
import sys
from awsglue.transforms import *
from awsglue.utils import getResolvedOptions
from pyspark.context import SparkContext
from awsglue.context import GlueContext
from awsglue.job import Job

"""
These custom arguments must be passed as Arguments to the StartJobRun request.
    --input_database    The name of a metadata database that is contained in
 your
                        AWS Glue Data Catalog and that contains tables that
 describe
                        the data to be processed.
```

```
    --input_table       The name of a table in the database that describes the
 data to
                        be processed.
    --output_bucket_url An S3 bucket that receives the transformed output data.
"""
args = getResolvedOptions(
    sys.argv, ["JOB_NAME", "input_database", "input_table", "output_bucket_url"]
)
sc = SparkContext()
glueContext = GlueContext(sc)
spark = glueContext.spark_session
job = Job(glueContext)
job.init(args["JOB_NAME"], args)

# Script generated for node S3 Flight Data.
S3FlightData_node1 = glueContext.create_dynamic_frame.from_catalog(
    database=args["input_database"],
    table_name=args["input_table"],
    transformation_ctx="S3FlightData_node1",
)

# This mapping performs two main functions:
# 1. It simplifies the output by removing most of the fields from the data.
# 2. It renames some fields. For example, `fl_date` is renamed to `flight_date`.
ApplyMapping_node2 = ApplyMapping.apply(
    frame=S3FlightData_node1,
    mappings=[
        ("year", "long", "year", "long"),
        ("month", "long", "month", "tinyint"),
        ("day_of_month", "long", "day", "tinyint"),
        ("fl_date", "string", "flight_date", "string"),
        ("carrier", "string", "carrier", "string"),
        ("fl_num", "long", "flight_num", "long"),
        ("origin_city_name", "string", "origin_city_name", "string"),
        ("origin_state_abr", "string", "origin_state_abr", "string"),
        ("dest_city_name", "string", "dest_city_name", "string"),
        ("dest_state_abr", "string", "dest_state_abr", "string"),
        ("dep_time", "long", "departure_time", "long"),
        ("wheels_off", "long", "wheels_off", "long"),
        ("wheels_on", "long", "wheels_on", "long"),
        ("arr_time", "long", "arrival_time", "long"),
        ("mon", "string", "mon", "string"),
    ],
    transformation_ctx="ApplyMapping_node2",
```

```
)

# Script generated for node Revised Flight Data.
RevisedFlightData_node3 = glueContext.write_dynamic_frame.from_options(
    frame=ApplyMapping_node2,
    connection_type="s3",
    format="json",
    connection_options={"path": args["output_bucket_url"], "partitionKeys": []},
    transformation_ctx="RevisedFlightData_node3",
)

job.commit()
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API AWS SDK per Ruby .
  - CreateCrawler
  - CreateJob
  - DeleteCrawler
  - DeleteDatabase
  - DeleteJob
  - DeleteTable
  - GetCrawler
  - GetDatabase
  - GetDatabases
  - GetJob
  - GetJobRun
  - GetJobRuns
  - GetTables
  - ListJobs
  - StartCrawler
  - StartJobRun

Rust

SDK per Rust

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Crea ed esegui un crawler che esegua la scansione di un bucket Amazon Simple Storage Service (Amazon S3) pubblico e generi un database di metadati che descrive i dati trovati. CSV-formatted

```
        let create_crawler = glue
            .create_crawler()
            .name(self.crawler())
            .database_name(self.database())
            .role(self.iam_role.expose_secret())
            .targets(
                CrawlerTargets::builder()
                    .s3_targets(S3Target::builder().path(CRAWLER_TARGET).build())
                    .build(),
            )
            .send()
            .await;

        match create_crawler {
            Err(err) => {
                let glue_err: aws_sdk_glue::Error = err.into();
                match glue_err {
                    aws_sdk_glue::Error::AlreadyExistsException(_) => {
                        info!("Using existing crawler");
                        Ok(())
                    }
                    _ => Err(GlueMvpError::GlueSdk(glue_err)),
                }
            }
            Ok(_) => Ok(()),
        }?;
```

```
        let start_crawler =
glue.start_crawler().name(self.crawler()).send().await;

        match start_crawler {
            Ok(_) => Ok(()),
            Err(err) => {
                let glue_err: aws_sdk_glue::Error = err.into();
                match glue_err {
                    aws_sdk_glue::Error::CrawlerRunningException(_) => Ok(()),
                    _ => Err(GlueMvpError::GlueSdk(glue_err)),
                }
            }
        }?;
```

Elenca le informazioni su database e tabelle nel tuo. AWS Glue Data Catalog

```
        let database = glue
            .get_database()
            .name(self.database())
            .send()
            .await
            .map_err(GlueMvpError::from_glue_sdk)?
            .to_owned();
        let database = database
            .database()
            .ok_or_else(|| GlueMvpError::Unknown("Could not find
database".into()))?;

        let tables = glue
            .get_tables()
            .database_name(self.database())
            .send()
            .await
            .map_err(GlueMvpError::from_glue_sdk)?;

        let tables = tables.table_list();
```

Crea ed esegui un processo che estrae i dati CSV dal bucket Amazon S3 di origine, li trasforma rimuovendo e rinominando i campi e carica l'output JSON-formatted in un altro bucket Amazon S3.

```
        let create_job = glue
            .create_job()
            .name(self.job())
            .role(self.iam_role.expose_secret())
            .command(
                JobCommand::builder()
                    .name("glueetl")
                    .python_version("3")
                    .script_location(format!("s3://{}/job.py", self.bucket()))
                    .build(),
            )
            .glue_version("3.0")
            .send()
            .await
            .map_err(GlueMvpError::from_glue_sdk)?;

        let job_name = create_job.name().ok_or_else(|| {
            GlueMvpError::Unknown("Did not get job name after creating
job".into())
        })?;

        let job_run_output = glue
            .start_job_run()
            .job_name(self.job())
            .arguments("--input_database", self.database())
            .arguments(
                "--input_table",
                self.tables
                    .first()
                    .ok_or_else(|| GlueMvpError::Unknown("Missing crawler
table".into()))?
                    .name(),
            )
            .arguments("--output_bucket_url", self.bucket())
            .send()
            .await
            .map_err(GlueMvpError::from_glue_sdk)?;

        let job = job_run_output
            .job_run_id()
            .ok_or_else(|| GlueMvpError::Unknown("Missing run id from just
started job".into()))?
            .to_string();
```

Eliminare tutte le risorse create dalla demo.

```
        glue.delete_job()
            .job_name(self.job())
            .send()
            .await
            .map_err(GlueMvpError::from_glue_sdk)?;

        for t in &self.tables {
            glue.delete_table()
                .name(t.name())
                .database_name(self.database())
                .send()
                .await
                .map_err(GlueMvpError::from_glue_sdk)?;
        }

        glue.delete_database()
            .name(self.database())
            .send()
            .await
            .map_err(GlueMvpError::from_glue_sdk)?;

        glue.delete_crawler()
            .name(self.crawler())
            .send()
            .await
            .map_err(GlueMvpError::from_glue_sdk)?;
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API SDK AWS per Rust.
  - CreateCrawler
  - CreateJob
  - DeleteCrawler
  - DeleteDatabase
  - DeleteJob
  - DeleteTable

- GetCrawler
- GetDatabase
- GetDatabases
- GetJob
- GetJobRun
- GetJobRuns
- GetTables
- ListJobs
- StartCrawler
- StartJobRun

Swift

SDK per Swift

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

Il file `Package.swift`.

```
// swift-tools-version: 5.9
//
// The swift-tools-version declares the minimum version of Swift required to
// build this package.

import PackageDescription

let package = Package(
    name: "glue-scenario",
    // Let Xcode know the minimum Apple platforms supported.
    platforms: [
        .macOS(.v13),
        .iOS(.v15)
    ],
    dependencies: [
```

```
        // Dependencies declare other packages that this package depends on.
        .package(
            url: "https://github.com/awslabs/aws-sdk-swift",
            from: "1.0.0"),
        .package(
            url: "https://github.com/apple/swift-argument-parser.git",
            branch: "main"
        )
    ],
    targets: [
        // Targets are the basic building blocks of a package, defining a module
 or a test suite.
        // Targets can depend on other targets in this package and products
        // from dependencies.
        .executableTarget(
            name: "glue-scenario",
            dependencies: [
                .product(name: "AWSGlue", package: "aws-sdk-swift"),
                .product(name: "AWSS3", package: "aws-sdk-swift"),
                .product(name: "ArgumentParser", package: "swift-argument-
parser")
            ],
            path: "Sources")

    ]
)
```

File del codice Swift: `entry.swift`.

```
// An example that shows how to use the AWS SDK for Swift to demonstrate
// creating and using crawlers and jobs using AWS Glue.
//
// 0. Upload the Python job script to Amazon S3 so it can be used when
//    calling `startJobRun()` later.
// 1. Create a crawler, pass it the IAM role and the URL of the public Amazon
//    S3 bucket that contains the source data:
//    s3://crawler-public-us-east-1/flight/2016/csv.
// 2. Start the crawler. This takes time, so after starting it, use a loop
//    that calls `getCrawler()` until the state is "READY".
// 3. Get the database created by the crawler, and the tables in the
//    database. Display them to the user.
// 4. Create a job. Pass it the IAM role and the URL to a Python ETL script
```

```
//    previously uploaded to the user's S3 bucket.
// 5. Start a job run, passing the following custom arguments. These are
//    expected by the ETL script, so must exactly match.
//    * `--input_database: <name of the database created by the crawler>`
//    * `--input_table: <name of the table created by the crawler>`
//    * `--output_bucket_url: <URL to the scaffold bucket created for the
//      user>`
// 6. Loop and get the job run until it returns one of the following states:
//    "SUCCEEDED", "STOPPED", "FAILED", or "TIMEOUT".
// 7. Output data is stored in a group of files in the user's S3 bucket.
//    Either direct the user to their location or download a file and display
//    the results inline.
// 8. List the jobs for the user's account.
// 9. Get job run details for a job run.
// 10. Delete the demo job.
// 11. Delete the database and tables created by the example.
// 12. Delete the crawler created by the example.

import ArgumentParser
import AWSS3
import Foundation
import Smithy

import AWSClientRuntime
import AWSGlue

struct ExampleCommand: ParsableCommand {
    @Option(help: "The AWS IAM role to use for AWS Glue calls.")
    var role: String

    @Option(help: "The Amazon S3 bucket to use for this example.")
    var bucket: String

    @Option(help: "The Amazon S3 URL of the data to crawl.")
    var s3url: String = "s3://crawler-public-us-east-1/flight/2016/csv"

    @Option(help: "The Python script to run as a job with AWS Glue.")
    var script: String = "./flight_etl_job_script.py"

    @Option(help: "The AWS Region to run AWS API calls in.")
    var awsRegion = "us-east-1"

    @Option(help: "A prefix string to use when naming tables.")
    var tablePrefix = "swift-glue-basics-table"
```

```
    @Option(
        help: ArgumentHelp("The level of logging for the Swift SDK to perform."),
        completion: .list([
            "critical",
            "debug",
            "error",
            "info",
            "notice",
            "trace",
            "warning"
        ])
    )
    var logLevel: String = "error"

    static var configuration = CommandConfiguration(
        commandName: "glue-scenario",
        abstract: """
        Demonstrates various features of AWS Glue.
        """,
        discussion: """
        An example showing how to use AWS Glue to create, run, and monitor
        crawlers and jobs.
        """
    )

    /// Generate and return a unique file name that begins with the specified
    /// string.
    ///
    /// - Parameters:
    ///   - prefix: Text to use at the beginning of the returned name.
    ///
    /// - Returns: A string containing a unique filename that begins with the
    ///   specified `prefix`.
    ///
    /// The returned name uses a random number between 1 million and 1 billion to
    /// provide reasonable certainty of uniqueness for the purposes of this
    /// example.
    func tempName(prefix: String) -> String {
        return "\(prefix)-\(Int.random(in: 1000000..<1000000000))"
    }

    /// Upload a file to an Amazon S3 bucket.
    ///
```

```
    /// - Parameters:
    ///   - s3Client: The S3 client to use when uploading the file.
    ///   - path: The local path of the source file to upload.
    ///   - toBucket: The name of the S3 bucket into which to upload the file.
    ///   - key: The key (name) to give the file in the S3 bucket.
    ///
    /// - Returns: `true` if the file is uploaded successfully, otherwise
`false`.
    func uploadFile(s3Client: S3Client, path: String, toBucket: String, key:
String) async -> Bool {
        do {
            let fileData: Data = try Data(contentsOf: URL(fileURLWithPath: path))
            let dataStream = ByteStream.data(fileData)
            _ = try await s3Client.putObject(
                input: PutObjectInput(
                    body: dataStream,
                    bucket: toBucket,
                    key: key
                )
            )
        } catch {
            print("*** An unexpected error occurred uploading the script to the
Amazon S3 bucket \"\(bucket)\".")
            return false
        }

        return true
    }

    /// Create a new AWS Glue crawler.
    ///
    /// - Parameters:
    ///   - glueClient: An AWS Glue client to use for the crawler.
    ///   - crawlerName: A name for the new crawler.
    ///   - iamRole: The name of an Amazon IAM role for the crawler to use.
    ///   - s3Path: The path of an Amazon S3 folder to use as a target location.
    ///   - cronSchedule: A `cron` schedule indicating when to run the crawler.
    ///   - databaseName: The name of an AWS Glue database to operate on.
    ///
    /// - Returns: `true` if the crawler is created successfully, otherwise
`false`.
    func createCrawler(glueClient: GlueClient, crawlerName: String, iamRole:
String,
```

```
                        s3Path: String, cronSchedule: String, databaseName:
String) async -> Bool {
        let s3Target = GlueClientTypes.S3Target(path: s3url)
        let targetList = GlueClientTypes.CrawlerTargets(s3Targets: [s3Target])

        do {
            _ = try await glueClient.createCrawler(
                input: CreateCrawlerInput(
                    databaseName: databaseName,
                    description: "Created by the AWS SDK for Swift Scenario
Example for AWS Glue.",
                    name: crawlerName,
                    role: iamRole,
                    schedule: cronSchedule,
                    tablePrefix: tablePrefix,
                    targets: targetList
                )
            )
        } catch _ as AlreadyExistsException {
            print("*** A crawler named \"\(crawlerName)\" already exists.")
            return false
        } catch _ as OperationTimeoutException {
            print("*** The attempt to create the AWS Glue crawler timed out.")
            return false
        } catch {
            print("*** An unexpected error occurred creating the AWS Glue
crawler: \(error.localizedDescription)")
            return false
        }

        return true
    }

    /// Delete an AWS Glue crawler.
    ///
    /// - Parameters:
    ///   - glueClient: The AWS Glue client to use.
    ///   - name: The name of the crawler to delete.
    ///
    /// - Returns: `true` if successful, otherwise `false`.
    func deleteCrawler(glueClient: GlueClient, name: String) async -> Bool {
        do {
            _ = try await glueClient.deleteCrawler(
                input: DeleteCrawlerInput(name: name)
```

```
            )
        } catch {
            return false
        }
        return true
    }

    /// Start running an AWS Glue crawler.
    ///
    /// - Parameters:
    ///   - glueClient: The AWS Glue client to use when starting the crawler.
    ///   - name: The name of the crawler to start running.
    ///
    /// - Returns: `true` if the crawler is started successfully, otherwise
`false`.
    func startCrawler(glueClient: GlueClient, name: String) async -> Bool {
        do {
            _ = try await glueClient.startCrawler(
                input: StartCrawlerInput(name: name)
            )
        } catch {
            print("*** An unexpected error occurred starting the crawler.")
            return false
        }

        return true
    }

    /// Get the state of the specified AWS Glue crawler.
    ///
    /// - Parameters:
    ///   - glueClient: The AWS Glue client to use.
    ///   - name: The name of the crawler whose state should be returned.
    ///
    /// - Returns: A `GlueClientTypes.CrawlerState` value describing the
    ///   state of the crawler.
    func getCrawlerState(glueClient: GlueClient, name: String) async ->
GlueClientTypes.CrawlerState {
        do {
            let output = try await glueClient.getCrawler(
                input: GetCrawlerInput(name: name)
            )

            // If the crawler or its state is `nil`, report that the crawler
```

```
            // is stopping. This may not be what you want for your
            // application but it works for this one!

            guard let crawler = output.crawler else {
                return GlueClientTypes.CrawlerState.stopping
            }
            guard let state = crawler.state else {
                return GlueClientTypes.CrawlerState.stopping
            }
            return state
        } catch {
            return GlueClientTypes.CrawlerState.stopping
        }
    }

    /// Wait until the specified crawler is ready to run.
    ///
    /// - Parameters:
    ///   - glueClient: The AWS Glue client to use.
    ///   - name: The name of the crawler to wait for.
    ///
    /// - Returns: `true` if the crawler is ready, `false` if the client is
    ///   stopping (and will therefore never be ready).
    func waitUntilCrawlerReady(glueClient: GlueClient, name: String) async ->
Bool {
        while true {
            let state = await getCrawlerState(glueClient: glueClient, name: name)

            if state == .ready {
                return true
            } else if state == .stopping {
                return false
            }

            // Wait four seconds before trying again.

            do {
                try await Task.sleep(for: .seconds(4))
            } catch {
                print("*** Error pausing the task.")
            }
        }
    }
```

```
    /// Create a new AWS Glue job.
    ///
    /// - Parameters:
    ///   - glueClient: The AWS Glue client to use.
    ///   - jobName: The name to give the new job.
    ///   - role: The IAM role for the job to use when accessing AWS services.
    ///   - scriptLocation: The AWS S3 URI of the script to be run by the job.
    ///
    /// - Returns: `true` if the job is created successfully, otherwise `false`.
    func createJob(glueClient: GlueClient, name jobName: String, role: String,
                   scriptLocation: String) async -> Bool {
        let command = GlueClientTypes.JobCommand(
            name: "glueetl",
            pythonVersion: "3",
            scriptLocation: scriptLocation
        )

        do {
            _ = try await glueClient.createJob(
                input: CreateJobInput(
                    command: command,
                    description: "Created by the AWS SDK for Swift Glue basic
scenario example.",
                    glueVersion: "3.0",
                    name: jobName,
                    numberOfWorkers: 10,
                    role: role,
                    workerType: .g1x
                )
            )
        } catch {
            return false
        }
        return true
    }

    /// Return a list of the AWS Glue jobs listed on the user's account.
    ///
    /// - Parameters:
    ///   - glueClient: The AWS Glue client to use.
    ///   - maxJobs: The maximum number of jobs to return (default: 100).
    ///
    /// - Returns: An array of strings listing the names of all available AWS
    ///   Glue jobs.
```

```
func listJobs(glueClient: GlueClient, maxJobs: Int = 100) async -> [String] {
    var jobList: [String] = []
    var nextToken: String?

    repeat {
        do {
            let output = try await glueClient.listJobs(
                input: ListJobsInput(
                    maxResults: maxJobs,
                    nextToken: nextToken
                )
            )

            guard let jobs = output.jobNames else {
                return jobList
            }

            jobList = jobList + jobs
            nextToken = output.nextToken
        } catch {
            return jobList
        }
    } while (nextToken != nil)

    return jobList
}

/// Delete an AWS Glue job.
///
/// - Parameters:
///   - glueClient: The AWS Glue client to use.
///   - jobName: The name of the job to delete.
///
/// - Returns: `true` if the job is successfully deleted, otherwise `false`.
func deleteJob(glueClient: GlueClient, name jobName: String) async -> Bool {
    do {
        _ = try await glueClient.deleteJob(
            input: DeleteJobInput(jobName: jobName)
        )
    } catch {
        return false
    }
    return true
}
```

```
   /// Create an AWS Glue database.
   ///
   /// - Parameters:
   ///   - glueClient: The AWS Glue client to use.
   ///   - databaseName: The name to give the new database.
   ///   - location: The URL of the source data to use with AWS Glue.
   ///
   /// - Returns: `true` if the database is created successfully, otherwise
`false`.
   func createDatabase(glueClient: GlueClient, name databaseName: String,
location: String) async -> Bool {
       let databaseInput = GlueClientTypes.DatabaseInput(
           description: "Created by the AWS SDK for Swift Glue basic scenario
example.",
           locationUri: location,
           name: databaseName
       )

       do {
           _ = try await glueClient.createDatabase(
               input: CreateDatabaseInput(
                   databaseInput: databaseInput
               )
           )
       } catch {
           return false
       }

       return true
   }

   /// Get the AWS Glue database with the specified name.
   ///
   /// - Parameters:
   ///   - glueClient: The AWS Glue client to use.
   ///   - name: The name of the database to return.
   ///
   /// - Returns: The `GlueClientTypes.Database` object describing the
   ///   specified database, or `nil` if an error occurs or the database
   ///   isn't found.
   func getDatabase(glueClient: GlueClient, name: String) async ->
GlueClientTypes.Database? {
       do {
```

```
            let output = try await glueClient.getDatabase(
                input: GetDatabaseInput(name: name)
            )

            return output.database
        } catch {
            return nil
        }
    }

    /// Returns a list of the tables in the specified database.
    ///
    /// - Parameters:
    ///   - glueClient: The AWS Glue client to use.
    ///   - databaseName: The name of the database whose tables are to be
    ///     returned.
    ///
    /// - Returns: An array of `GlueClientTypes.Table` objects, each
    ///   describing one table in the named database. An empty array indicates
    ///   that there are either no tables in the database, or an error
    ///   occurred before any tables could be found.
    func getTablesInDatabase(glueClient: GlueClient, databaseName: String) async
-> [GlueClientTypes.Table] {
        var tables: [GlueClientTypes.Table] = []
        var nextToken: String?

        repeat {
            do {
                let output = try await glueClient.getTables(
                    input: GetTablesInput(
                        databaseName: databaseName,
                        nextToken: nextToken
                    )
                )

                guard let tableList = output.tableList else {
                    return tables
                }

                tables = tables + tableList
                nextToken = output.nextToken
            } catch {
                return tables
            }
```

```
        } while nextToken != nil

        return tables
    }

    /// Delete the specified database.
    ///
    /// - Parameters:
    ///   - glueClient: The AWS Glue client to use.
    ///   - databaseName: The name of the database to delete.
    ///   - deleteTables: A Bool indicating whether or not to delete the
    ///     tables in the database before attempting to delete the database.
    ///
    /// - Returns: `true` if the database (and optionally its tables) are
    ///   deleted, otherwise `false`.
    func deleteDatabase(glueClient: GlueClient, name databaseName: String,
                        withTables deleteTables: Bool = false) async -> Bool {
        if deleteTables {
            var tableNames: [String] = []

            // Get a list of the names of all of the tables in the database.

            let tableList = await self.getTablesInDatabase(glueClient:
glueClient, databaseName: databaseName)
            for table in tableList {
                guard let name = table.name else {
                    continue
                }
                tableNames.append(name)
            }

            // Delete the tables. If there's only one table, use
            // `deleteTable()`, otherwise, use `batchDeleteTable()`. You can
            // use `batchDeleteTable()` for a single table, but this
            // demonstrates the use of `deleteTable()`.

            if tableNames.count == 1 {
                do {
                    print("    Deleting table...")
                    _ = try await glueClient.deleteTable(
                        input: DeleteTableInput(
                            databaseName: databaseName,
                            name: tableNames[0]
                        )
```

```
                    )
                } catch {
                    print("*** Unable to delete the table.")
                }
            } else {
                do {
                    print("    Deleting tables...")
                    _ = try await glueClient.batchDeleteTable(
                        input: BatchDeleteTableInput(
                            databaseName: databaseName,
                            tablesToDelete: tableNames
                        )
                    )
                } catch {
                    print("*** Unable to delete the tables.")
                }
            }
        }

        // Delete the database itself.

        do {
            print("    Deleting the database itself...")
            _ = try await glueClient.deleteDatabase(
                input: DeleteDatabaseInput(name: databaseName)
            )
        } catch {
            print("*** Unable to delete the database.")
            return false
        }
        return true
    }

    /// Start an AWS Glue job run.
    ///
    /// - Parameters:
    ///   - glueClient: The AWS Glue client to use.
    ///   - jobName: The name of the job to run.
    ///   - databaseName: The name of the AWS Glue database to run the job
against.
    ///   - tableName: The name of the table in the database to run the job
against.
    ///   - outputURL: The AWS S3 URI of the bucket location into which to
    ///     write the resulting output.
```

```
    ///
    /// - Returns: `true` if the job run is started successfully, otherwise
`false`.
    func startJobRun(glueClient: GlueClient, name jobName: String, databaseName:
String,
                     tableName: String, outputURL: String) async -> String? {
        do {
            let output = try await glueClient.startJobRun(
                input: StartJobRunInput(
                    arguments: [
                        "--input_database": databaseName,
                        "--input_table": tableName,
                        "--output_bucket_url": outputURL
                    ],
                    jobName: jobName,
                    numberOfWorkers: 10,
                    workerType: .g1x
                )
            )

            guard let id = output.jobRunId else {
                return nil
            }

            return id
        } catch {
            return nil
        }
    }

    /// Return a list of the job runs for the specified job.
    ///
    /// - Parameters:
    ///   - glueClient: The AWS Glue client to use.
    ///   - jobName: The name of the job for which to return its job runs.
    ///   - maxResults: The maximum number of job runs to return (default:
    ///     1000).
    ///
    /// - Returns: An array of `GlueClientTypes.JobRun` objects describing
    ///   each job run.
    func getJobRuns(glueClient: GlueClient, name jobName: String, maxResults:
Int? = nil) async -> [GlueClientTypes.JobRun] {
        do {
            let output = try await glueClient.getJobRuns(
```

```
                input: GetJobRunsInput(
                    jobName: jobName,
                    maxResults: maxResults
                )
            )

            guard let jobRuns = output.jobRuns else {
                print("*** No job runs found.")
                return []
            }

            return jobRuns
        } catch is EntityNotFoundException {
            print("*** The specified job name, \(jobName), doesn't exist.")
            return []
        } catch {
            print("*** Unexpected error getting job runs:")
            dump(error)
            return []
        }
    }

    /// Get information about a specific AWS Glue job run.
    ///
    /// - Parameters:
    ///   - glueClient: The AWS Glue client to use.
    ///   - jobName: The name of the job to return job run data for.
    ///   - id: The run ID of the specific job run to return.
    ///
    /// - Returns: A `GlueClientTypes.JobRun` object describing the state of
    ///   the job run, or `nil` if an error occurs.
    func getJobRun(glueClient: GlueClient, name jobName: String, id: String)
async -> GlueClientTypes.JobRun? {
        do {
            let output = try await glueClient.getJobRun(
                input: GetJobRunInput(
                    jobName: jobName,
                    runId: id
                )
            )

            return output.jobRun
        } catch {
            return nil
```

```
        }
    }

    /// Called by ``main()`` to run the bulk of the example.
    func runAsync() async throws {
        // A name to give the Python script upon upload to the Amazon S3
        // bucket.
        let scriptName = "jobscript.py"

        // Schedule string in `cron` format, as described here:
        // https://docs.aws.amazon.com/glue/latest/dg/monitor-data-warehouse-
schedule.html
        let cron = "cron(15 12 * * ? *)"

        let glueConfig = try await GlueClient.GlueClientConfiguration(region:
 awsRegion)
        let glueClient = GlueClient(config: glueConfig)

        let s3Config = try await S3Client.S3ClientConfiguration(region:
 awsRegion)
        let s3Client = S3Client(config: s3Config)

        // Create random names for things that need them.

        let crawlerName = tempName(prefix: "swift-glue-basics-crawler")
        let databaseName = tempName(prefix: "swift-glue-basics-db")

        // Create a name for the AWS Glue job.

        let jobName = tempName(prefix: "scenario-job")

        // The URL of the Python script on S3.

        let scriptURL = "s3://\(bucket)/\(scriptName)"

        print("Welcome to the AWS SDK for Swift basic scenario for AWS Glue!")

        //===========================================================================
        // 0. Upload the Python script to the target bucket so it's available
        //    for use by the Amazon Glue service.
        //===========================================================================

        print("Uploading the Python script: \(script) as key \(scriptName)")
        print("Destination bucket: \(bucket)")
```

```
        if !(await uploadFile(s3Client: s3Client, path: script, toBucket: bucket,
key: scriptName)) {
            return
        }

        //====================================================================
        // 1. Create the database and crawler using the randomized names
        //    generated previously.
        //====================================================================

        print("Creating database \"\(databaseName)\"...")
        if !(await createDatabase(glueClient: glueClient, name: databaseName,
location: s3url)) {
            print("*** Unable to create the database.")
            return
        }

        print("Creating crawler \"\(crawlerName)\"...")
        if !(await createCrawler(glueClient: glueClient, crawlerName:
crawlerName,
                                 iamRole: role, s3Path: s3url, cronSchedule:
cron,
                                 databaseName: databaseName)) {
            return
        }

        //====================================================================
        // 2. Start the crawler, then wait for it to be ready.
        //====================================================================

        print("Starting the crawler and waiting until it's ready...")
        if !(await startCrawler(glueClient: glueClient, name: crawlerName)) {
            _ = await deleteCrawler(glueClient: glueClient, name: crawlerName)
            return
        }

        if !(await waitUntilCrawlerReady(glueClient: glueClient, name:
crawlerName)) {
            _ = await deleteCrawler(glueClient: glueClient, name: crawlerName)
        }

        //====================================================================
        // 3. Get the database and table created by the crawler.
        //====================================================================
```

```
        print("Getting the crawler's database...")
        let database = await getDatabase(glueClient: glueClient, name:
databaseName)

        guard let database else {
            print("*** Unable to get the database.")
            return
        }
        print("Database URI: \(database.locationUri ?? "<unknown>")")

        let tableList = await getTablesInDatabase(glueClient: glueClient,
databaseName: databaseName)

        print("Found \(tableList.count) table(s):")
        for table in tableList {
            print("  \(table.name ?? "<unnamed>")")
        }

        if tableList.count != 1 {
            print("*** Incorrect number of tables found. There should only be
one.")
            _ = await deleteDatabase(glueClient: glueClient, name: databaseName,
withTables: true)
            _ = await deleteCrawler(glueClient: glueClient, name: crawlerName)
            return
        }

        guard let tableName = tableList[0].name else {
            print("*** Table is unnamed.")
            _ = await deleteDatabase(glueClient: glueClient, name: databaseName,
withTables: true)
            _ = await deleteCrawler(glueClient: glueClient, name: crawlerName)
            return
        }

        //====================================================================
        // 4. Create a job.
        //====================================================================

        print("Creating a job...")
        if !(await createJob(glueClient: glueClient, name: jobName, role: role,
                             scriptLocation: scriptURL)) {
```

```
            _ = await deleteDatabase(glueClient: glueClient, name: databaseName,
withTables: true)
            _ = await deleteCrawler(glueClient: glueClient, name: crawlerName)
            return
        }

        //====================================================================
        // 5. Start a job run.
        //====================================================================

        print("Starting the job...")

        // Construct the Amazon S3 URL for the job run's output. This is in
        // the bucket specified on the command line, with a folder name that's
        // unique for this job run.

        let timeStamp = Date().timeIntervalSince1970
        let jobPath = "\(jobName)-\(Int(timeStamp))"
        let outputURL = "s3://\(bucket)/\(jobPath)"

        // Start the job run.

        let jobRunID = await startJobRun(glueClient: glueClient, name: jobName,
                                         databaseName: databaseName,
                                         tableName: tableName,
                                         outputURL: outputURL)

        guard let jobRunID else {
            print("*** Job run ID is invalid.")
            _ = await deleteJob(glueClient: glueClient, name: jobName)
            _ = await deleteDatabase(glueClient: glueClient, name: databaseName,
withTables: true)
            _ = await deleteCrawler(glueClient: glueClient, name: crawlerName)
            return
        }

        //====================================================================
        // 6. Wait for the job run to indicate that the run is complete.
        //====================================================================

        print("Waiting for job run to end...")

        var jobRunFinished = false
        var jobRunState: GlueClientTypes.JobRunState
```

```
        repeat {
            let jobRun = await getJobRun(glueClient: glueClient, name: jobName,
 id: jobRunID)
            guard let jobRun else {
                print("*** Unable to get the job run.")
                _ = await deleteJob(glueClient: glueClient, name: jobName)
                _ = await deleteDatabase(glueClient: glueClient, name:
 databaseName, withTables: true)
                _ = await deleteCrawler(glueClient: glueClient, name:
 crawlerName)
                return
            }
            jobRunState = jobRun.jobRunState ?? .failed

            //
=====================================================================
            // 7. Output where to find the data if the job run was successful.
            //    If the job run failed for any reason, output an appropriate
            //    error message.
            //
=====================================================================

            switch jobRunState {
                case .succeeded:
                    print("Job run succeeded. JSON files are in the Amazon S3
 path:")
                    print("    \(outputURL)")
                    jobRunFinished = true
                case .stopped:
                    jobRunFinished = true
                case .error:
                    print("*** Error: Job run ended in an error.
 \(jobRun.errorMessage ?? "")")
                    jobRunFinished = true
                case .failed:
                    print("*** Error: Job run failed. \(jobRun.errorMessage ??
 "")")
                    jobRunFinished = true
                case .timeout:
                    print("*** Warning: Job run timed out.")
                    jobRunFinished = true
                default:
                    do {
```

```
                    try await Task.sleep(for: .milliseconds(250))
                } catch {
                    print("*** Error pausing the task.")
                }
            }
        } while jobRunFinished != true

        //=========================================================================
        // 7.5. List the job runs for this job, showing each job run's ID and
        // its execution time.
        //=========================================================================

        print("Getting all job runs for the job \(jobName):")
        let jobRuns = await getJobRuns(glueClient: glueClient, name: jobName)

        if jobRuns.count == 0 {
            print("    <no job runs found>")
        } else {
            print("Found \(jobRuns.count) job runs... listing execution times:")
            for jobRun in jobRuns {
                print("    \(jobRun.id ?? "<unnamed>"): \(jobRun.executionTime)
seconds")
            }
        }

        //=========================================================================
        // 8. List the jobs for the user's account.
        //=========================================================================

        print("\nThe account has the following jobs:")
        let jobs = await listJobs(glueClient: glueClient)

        if jobs.count == 0 {
            print("    <no jobs found>")
        } else {
            for job in jobs {
                print("    \(job)")
            }
        }

        //=========================================================================
        // 9. Get the job run details for a job run.
        //=========================================================================
```

```
        print("Information about the job run:")
        let jobRun = await getJobRun(glueClient: glueClient, name: jobName, id:
jobRunID)

        guard let jobRun else {
            print("*** Unable to retrieve the job run.")
            _ = await deleteJob(glueClient: glueClient, name: jobName)
            _ = await deleteDatabase(glueClient: glueClient, name: databaseName,
withTables: true)
            _ = await deleteCrawler(glueClient: glueClient, name: crawlerName)
            return
        }

        let startDate = jobRun.startedOn ?? Date(timeIntervalSince1970: 0)
        let endDate = jobRun.completedOn ?? Date(timeIntervalSince1970: 0)
        let dateFormatter: DateFormatter = DateFormatter()
        dateFormatter.dateStyle = .long
        dateFormatter.timeStyle = .long

        print("    Started at: \(dateFormatter.string(from: startDate))")
        print("  Completed at: \(dateFormatter.string(from: endDate))")

        //===================================================================
        // 10. Delete the job.
        //===================================================================

        print("\nDeleting the job...")
        _ = await deleteJob(glueClient: glueClient, name: jobName)

        //===================================================================
        // 11. Delete the database and tables created by this example.
        //===================================================================

        print("Deleting the database...")
        _ = await deleteDatabase(glueClient: glueClient, name: databaseName,
withTables: true)

        //===================================================================
        // 12. Delete the crawler.
        //===================================================================

        print("Deleting the crawler...")
        if !(await deleteCrawler(glueClient: glueClient, name: crawlerName)) {
            return
```

```
        }
    }
}

/// The program's asynchronous entry point.
@main
struct Main {
    static func main() async {
        let args = Array(CommandLine.arguments.dropFirst())

        do {
            let command = try ExampleCommand.parse(args)
            try await command.runAsync()
        } catch {
            ExampleCommand.exit(withError: error)
        }
    }
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dell'API SDK AWS per Swift.
  - CreateCrawler
  - CreateJob
  - DeleteCrawler
  - DeleteDatabase
  - DeleteJob
  - DeleteTable
  - GetCrawler
  - GetDatabase
  - GetDatabases
  - GetJob
  - GetJobRun
  - GetJobRuns
  - GetTables
  - ListJobs
  - StartCrawler

- StartJobRun

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

## Azioni per l' AWS Glue utilizzo degli AWS SDK

I seguenti esempi di codice mostrano come eseguire AWS Glue azioni individuali con gli AWS SDK. Ogni esempio include un collegamento a GitHub, dove è possibile trovare le istruzioni per la configurazione e l'esecuzione del codice.

Questi estratti richiamano l' AWS Glue API e sono estratti di codice di programmi più grandi che devono essere eseguiti nel contesto. È possibile visualizzare le azioni nel contesto in Scenari per l' AWS Glue utilizzo degli AWS SDK.

Gli esempi seguenti includono solo le azioni più comunemente utilizzate. Per un elenco completo, consulta la documentazione di riferimento dell'API AWS Glue.

Esempi

- Utilizzo CreateCrawler con un AWS SDK
- Utilizzo CreateJob con un AWS SDK o una CLI
- Utilizzo DeleteCrawler con un AWS SDK
- Utilizzo DeleteDatabase con un AWS SDK
- Utilizzo DeleteJob con un AWS SDK o una CLI
- Utilizzo DeleteTable con un AWS SDK
- Utilizzo GetCrawler con un AWS SDK
- Utilizzo GetDatabase con un AWS SDK
- Utilizzo GetDatabases con un AWS SDK o una CLI
- Utilizzo GetJob con un AWS SDK o una CLI
- Utilizzo GetJobRun con un AWS SDK o una CLI
- Utilizzo GetJobRuns con un AWS SDK o una CLI
- Utilizzo GetTables con un AWS SDK o una CLI

- Utilizzo ListJobs con un AWS SDK
- Utilizzo StartCrawler con un AWS SDK o una CLI
- Utilizzo StartJobRun con un AWS SDK o una CLI

## Utilizzo **`CreateCrawler`** con un AWS SDK

Gli esempi di codice seguenti mostrano come utilizzare `CreateCrawler`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Informazioni di base

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// Create an AWS Glue crawler.
    /// </summary>
    /// <param name="crawlerName">The name for the crawler.</param>
    /// <param name="crawlerDescription">A description of the crawler.</param>
    /// <param name="role">The AWS Identity and Access Management (IAM) role to
    /// be assumed by the crawler.</param>
    /// <param name="schedule">The schedule on which the crawler will be
executed.</param>
    /// <param name="s3Path">The path to the Amazon Simple Storage Service
(Amazon S3)
    /// bucket where the Python script has been stored.</param>
    /// <param name="dbName">The name to use for the database that will be
    /// created by the crawler.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> CreateCrawlerAsync(
```

```
        string crawlerName,
        string crawlerDescription,
        string role,
        string schedule,
        string s3Path,
        string dbName)
    {
        var s3Target = new S3Target
        {
            Path = s3Path,
        };

        var targetList = new List<S3Target>
        {
            s3Target,
        };

        var targets = new CrawlerTargets
        {
            S3Targets = targetList,
        };

        var crawlerRequest = new CreateCrawlerRequest
        {
            DatabaseName = dbName,
            Name = crawlerName,
            Description = crawlerDescription,
            Targets = targets,
            Role = role,
            Schedule = schedule,
        };

        var response = await _amazonGlue.CreateCrawlerAsync(crawlerRequest);
        return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
    }
```

- Per i dettagli sull'API, consulta la CreateCrawlersezione AWS SDK per .NET API Reference.

C++

SDK per C++

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
       Aws::Client::ClientConfiguration clientConfig;
       // Optional: Set to the AWS Region in which the bucket was created
(overrides config file).
       // clientConfig.region = "us-east-1";

   Aws::Glue::GlueClient client(clientConfig);

       Aws::Glue::Model::S3Target s3Target;
       s3Target.SetPath("s3://crawler-public-us-east-1/flight/2016/csv");
       Aws::Glue::Model::CrawlerTargets crawlerTargets;
       crawlerTargets.AddS3Targets(s3Target);

       Aws::Glue::Model::CreateCrawlerRequest request;
       request.SetTargets(crawlerTargets);
       request.SetName(CRAWLER_NAME);
       request.SetDatabaseName(CRAWLER_DATABASE_NAME);
       request.SetTablePrefix(CRAWLER_DATABASE_PREFIX);
       request.SetRole(roleArn);

       Aws::Glue::Model::CreateCrawlerOutcome outcome =
client.CreateCrawler(request);

       if (outcome.IsSuccess()) {
           std::cout << "Successfully created the crawler." << std::endl;
       }
       else {
           std::cerr << "Error creating a crawler. " <<
outcome.GetError().GetMessage()
                     << std::endl;
           deleteAssets("", CRAWLER_DATABASE_NAME, "", bucketName,
clientConfig);
           return false;
```

```
        }
```

- Per i dettagli sull'API, consulta la CreateCrawlersezione AWS SDK per C++ API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
   /**
    * Creates a new AWS Glue crawler using the AWS Glue Java API.
    *
    * @param glueClient  the AWS Glue client used to interact with the AWS Glue
service
    * @param iam         the IAM role that the crawler will use to access the
data source
    * @param s3Path      the S3 path that the crawler will scan for data
    * @param cron        the cron expression that defines the crawler's schedule
    * @param dbName      the name of the AWS Glue database where the crawler
will store the metadata
    * @param crawlerName the name of the crawler to be created
    */
   public static void createGlueCrawler(GlueClient glueClient,
                                         String iam,
                                         String s3Path,
                                         String cron,
                                         String dbName,
                                         String crawlerName) {

       try {
           S3Target s3Target = S3Target.builder()
               .path(s3Path)
               .build();

           List<S3Target> targetList = new ArrayList<>();
```

```
            targetList.add(s3Target);
            CrawlerTargets targets = CrawlerTargets.builder()
                .s3Targets(targetList)
                .build();

            CreateCrawlerRequest crawlerRequest = CreateCrawlerRequest.builder()
                .databaseName(dbName)
                .name(crawlerName)
                .description("Created by the AWS Glue Java API")
                .targets(targets)
                .role(iam)
                .schedule(cron)
                .build();

            glueClient.createCrawler(crawlerRequest);
            System.out.println(crawlerName + " was successfully created");

        } catch (GlueException e) {
            throw e;
        }
    }
```

- Per i dettagli sull'API, consulta la CreateCrawlersezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
const createCrawler = (name, role, dbName, tablePrefix, s3TargetPath) => {
  const client = new GlueClient({});

  const command = new CreateCrawlerCommand({
    Name: name,
```

```
    Role: role,
    DatabaseName: dbName,
    TablePrefix: tablePrefix,
    Targets: {
      S3Targets: [{ Path: s3TargetPath }],
    },
  });

  return client.send(command);
};
```

- Per i dettagli sull'API, consulta la CreateCrawlersezione AWS SDK per JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun createGlueCrawler(
    iam: String?,
    s3Path: String?,
    cron: String?,
    dbName: String?,
    crawlerName: String,
) {
    val s3Target =
        S3Target {
            path = s3Path
        }

    // Add the S3Target to a list.
    val targetList = mutableListOf<S3Target>()
    targetList.add(s3Target)
```

```
    val targetOb =
        CrawlerTargets {
            s3Targets = targetList
        }

    val request =
        CreateCrawlerRequest {
            databaseName = dbName
            name = crawlerName
            description = "Created by the AWS Glue Kotlin API"
            targets = targetOb
            role = iam
            schedule = cron
        }

    GlueClient.fromEnvironment { region = "us-west-2" }.use { glueClient ->
        glueClient.createCrawler(request)
        println("$crawlerName was successfully created")
    }
}
```

- Per i dettagli sull'API, CreateCrawlerconsulta AWS SDK for Kotlin API reference.

PHP

SDK per PHP

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
        $crawlerName = "example-crawler-test-" . $uniqid;

        $role = $iamService->getRole("AWSGlueServiceRole-DocExample");

        $path = 's3://crawler-public-us-east-1/flight/2016/csv';
        $glueService->createCrawler($crawlerName, $role['Role']['Arn'],
$databaseName, $path);
```

```
    public function createCrawler($crawlerName, $role, $databaseName, $path):
Result
    {
        return $this->customWaiter(function () use ($crawlerName, $role,
$databaseName, $path) {
            return $this->glueClient->createCrawler([
                'Name' => $crawlerName,
                'Role' => $role,
                'DatabaseName' => $databaseName,
                'Targets' => [
                    'S3Targets' =>
                        [[
                            'Path' => $path,
                        ]]
                ],
            ]);
        });
    }
```

- Per i dettagli sull'API, consulta la CreateCrawlersezione AWS SDK per PHP API Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
class GlueWrapper:
    """Encapsulates AWS Glue actions."""

    def __init__(self, glue_client):
        """
        :param glue_client: A Boto3 Glue client.
        """
        self.glue_client = glue_client
```

```
    def create_crawler(self, name, role_arn, db_name, db_prefix, s3_target):
        """
        Creates a crawler that can crawl the specified target and populate a
        database in your AWS Glue Data Catalog with metadata that describes the
data
        in the target.

        :param name: The name of the crawler.
        :param role_arn: The Amazon Resource Name (ARN) of an AWS Identity and
Access
                         Management (IAM) role that grants permission to let AWS
Glue
                         access the resources it needs.
        :param db_name: The name to give the database that is created by the
crawler.
        :param db_prefix: The prefix to give any database tables that are created
by
                          the crawler.
        :param s3_target: The URL to an S3 bucket that contains data that is
                          the target of the crawler.
        """
        try:
            self.glue_client.create_crawler(
                Name=name,
                Role=role_arn,
                DatabaseName=db_name,
                TablePrefix=db_prefix,
                Targets={"S3Targets": [{"Path": s3_target}]},
            )
        except ClientError as err:
            logger.error(
                "Couldn't create crawler. Here's why: %s: %s",
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
```

- Per i dettagli sull'API, consulta CreateCrawler AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
# The `GlueWrapper` class serves as a wrapper around the AWS Glue API, providing
 a simplified interface for common operations.
# It encapsulates the functionality of the AWS SDK for Glue and provides methods
 for interacting with Glue crawlers, databases, tables, jobs, and S3 resources.
# The class initializes with a Glue client and a logger, allowing it to make API
 calls and log any errors or informational messages.
class GlueWrapper
  def initialize(glue_client, logger)
    @glue_client = glue_client
    @logger = logger
  end

  # Creates a new crawler with the specified configuration.
  #
  # @param name [String] The name of the crawler.
  # @param role_arn [String] The ARN of the IAM role to be used by the crawler.
  # @param db_name [String] The name of the database where the crawler stores its
 metadata.
  # @param db_prefix [String] The prefix to be added to the names of tables that
 the crawler creates.
  # @param s3_target [String] The S3 path that the crawler will crawl.
  # @return [void]
  def create_crawler(name, role_arn, db_name, _db_prefix, s3_target)
    @glue_client.create_crawler(
      name: name,
      role: role_arn,
      database_name: db_name,
      targets: {
        s3_targets: [
          {
            path: s3_target
```

```
        }
      ]
    }
  )
rescue Aws::Glue::Errors::GlueException => e
  @logger.error("Glue could not create crawler: \n#{e.message}")
  raise
end
```

- Per i dettagli sull'API, consulta la CreateCrawlersezione AWS SDK per Ruby API Reference.

## Rust

### SDK per Rust

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
        let create_crawler = glue
            .create_crawler()
            .name(self.crawler())
            .database_name(self.database())
            .role(self.iam_role.expose_secret())
            .targets(
                CrawlerTargets::builder()
                    .s3_targets(S3Target::builder().path(CRAWLER_TARGET).build())
                    .build(),
            )
            .send()
            .await;

        match create_crawler {
            Err(err) => {
                let glue_err: aws_sdk_glue::Error = err.into();
                match glue_err {
                    aws_sdk_glue::Error::AlreadyExistsException(_) => {
                        info!("Using existing crawler");
                        Ok(())
```

```
                }
                _ => Err(GlueMvpError::GlueSdk(glue_err)),
            }
        }
        Ok(_) => Ok(()),
    }?;
```

- Per i dettagli sulle API, consulta la CreateCrawlerguida di riferimento all'API AWS SDK for Rust.

## SAP ABAP

### SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
TRY.
    " iv_crawler_name = 'my-crawler'
    " iv_role_arn = 'arn:aws:iam::123456789012:role/AWSGlueServiceRole-Test'
    " iv_database_name = 'my-database'
    " iv_table_prefix = 'test_'
    " iv_s3_target = 's3://example-bucket/data/'

    DATA(lt_s3_targets) = VALUE /aws1/cl_glus3target=>tt_s3targetlist(
      ( NEW /aws1/cl_glus3target( iv_path = iv_s3_target ) ) ).

    DATA(lo_targets) = NEW /aws1/cl_glucrawlertargets(
      it_s3targets = lt_s3_targets ).

    lo_glu->createcrawler(
      iv_name = iv_crawler_name
      iv_role = iv_role_arn
      iv_databasename = iv_database_name
      iv_tableprefix = iv_table_prefix
      io_targets = lo_targets ).
    MESSAGE 'Crawler created successfully.' TYPE 'I'.
```

```
      CATCH /aws1/cx_glualreadyexistsex.
        MESSAGE 'Crawler already exists.' TYPE 'E'.
      CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).
        DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).
        MESSAGE lv_invalid_error TYPE 'E'.
      CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
        DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
        MESSAGE lv_timeout_error TYPE 'E'.
      CATCH /aws1/cx_gluresrcnumlmtexcdex INTO DATA(lo_limit_ex).
        DATA(lv_limit_error) = lo_limit_ex->if_message~get_longtext( ).
        MESSAGE lv_limit_error TYPE 'E'.
    ENDTRY.
```

- Per i dettagli sulle API, CreateCrawlerconsulta AWS SDK for SAP ABAP API reference.

## Swift

### SDK per Swift

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import AWSClientRuntime
import AWSGlue

    /// Create a new AWS Glue crawler.
    ///
    /// - Parameters:
    ///   - glueClient: An AWS Glue client to use for the crawler.
    ///   - crawlerName: A name for the new crawler.
    ///   - iamRole: The name of an Amazon IAM role for the crawler to use.
    ///   - s3Path: The path of an Amazon S3 folder to use as a target location.
    ///   - cronSchedule: A `cron` schedule indicating when to run the crawler.
    ///   - databaseName: The name of an AWS Glue database to operate on.
    ///
    /// - Returns: `true` if the crawler is created successfully, otherwise
 `false`.
```

```
   func createCrawler(glueClient: GlueClient, crawlerName: String, iamRole:
String,
                      s3Path: String, cronSchedule: String, databaseName:
String) async -> Bool {
       let s3Target = GlueClientTypes.S3Target(path: s3url)
       let targetList = GlueClientTypes.CrawlerTargets(s3Targets: [s3Target])

       do {
           _ = try await glueClient.createCrawler(
               input: CreateCrawlerInput(
                   databaseName: databaseName,
                   description: "Created by the AWS SDK for Swift Scenario
Example for AWS Glue.",
                   name: crawlerName,
                   role: iamRole,
                   schedule: cronSchedule,
                   tablePrefix: tablePrefix,
                   targets: targetList
               )
           )
       } catch _ as AlreadyExistsException {
           print("*** A crawler named \"\(crawlerName)\" already exists.")
           return false
       } catch _ as OperationTimeoutException {
           print("*** The attempt to create the AWS Glue crawler timed out.")
           return false
       } catch {
           print("*** An unexpected error occurred creating the AWS Glue
crawler: \(error.localizedDescription)")
           return false
       }

       return true
   }
```

- Per i dettagli sull'API, consulta la CreateCrawlerguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

## Utilizzo **`CreateJob`** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare `CreateJob`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Informazioni di base

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
   /// <summary>
   /// Create an AWS Glue job.
   /// </summary>
   /// <param name="jobName">The name of the job.</param>
   /// <param name="roleName">The name of the IAM role to be assumed by
   /// the job.</param>
   /// <param name="description">A description of the job.</param>
   /// <param name="scriptUrl">The URL to the script.</param>
   /// <returns>A Boolean value indicating the success of the action.</returns>
   public async Task<bool> CreateJobAsync(string dbName, string tableName,
string bucketUrl, string jobName, string roleName, string description, string
scriptUrl)
   {
       var command = new JobCommand
       {
           PythonVersion = "3",
           Name = "glueetl",
           ScriptLocation = scriptUrl,
       };

       var arguments = new Dictionary<string, string>
       {
           { "--input_database", dbName },
```

```
            { "--input_table", tableName },
            { "--output_bucket_url", bucketUrl }
        };

        var request = new CreateJobRequest
        {
            Command = command,
            DefaultArguments = arguments,
            Description = description,
            GlueVersion = "3.0",
            Name = jobName,
            NumberOfWorkers = 10,
            Role = roleName,
            WorkerType = "G.1X"
        };

        var response = await _amazonGlue.CreateJobAsync(request);
        return response.HttpStatusCode == HttpStatusCode.OK;
    }
```

- Per i dettagli sull'API, consulta la CreateJobsezione AWS SDK per .NET API Reference.

C++

SDK per C++

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region in which the bucket was created
(overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::Glue::GlueClient client(clientConfig);

        Aws::Glue::Model::CreateJobRequest request;
```

```
        request.SetName(JOB_NAME);
        request.SetRole(roleArn);
        request.SetGlueVersion(GLUE_VERSION);

        Aws::Glue::Model::JobCommand command;
        command.SetName(JOB_COMMAND_NAME);
        command.SetPythonVersion(JOB_PYTHON_VERSION);
        command.SetScriptLocation(
                Aws::String("s3://") + bucketName + "/" + PYTHON_SCRIPT);
        request.SetCommand(command);

        Aws::Glue::Model::CreateJobOutcome outcome = client.CreateJob(request);

        if (outcome.IsSuccess()) {
            std::cout << "Successfully created the job." << std::endl;
        }
        else {
            std::cerr << "Error creating the job. " <<
 outcome.GetError().GetMessage()
                      << std::endl;
            deleteAssets(CRAWLER_NAME, CRAWLER_DATABASE_NAME, "", bucketName,
                         clientConfig);
            return false;
        }
```

- Per i dettagli sull'API, consulta la CreateJobsezione AWS SDK per C++ API Reference.

## CLI

### AWS CLI

#### Come creare un processo di trasformazione dei dati

L'esempio `create-job` seguente crea un processo di streaming che esegue uno script archiviato in S3.

```
aws glue create-job \
    --name my-testing-job \
    --role AWSGlueServiceRoleDefault \
    --command '{ \
        "Name": "gluestreaming", \
        "ScriptLocation": "s3://amzn-s3-demo-bucket/folder/" \
```

```
    }' \
    --region us-east-1 \
    --output json \
    --default-arguments '{ \
        "--job-language":"scala", \
        "--class":"GlueApp" \
    }' \
    --profile my-profile \
    --endpoint https://glue.us-east-1.amazonaws.com
```

Contenuto di `test_script.scala`:

```
import com.amazonaws.services.glue.ChoiceOption
import com.amazonaws.services.glue.GlueContext
import com.amazonaws.services.glue.MappingSpec
import com.amazonaws.services.glue.ResolveSpec
import com.amazonaws.services.glue.errors.CallSite
import com.amazonaws.services.glue.util.GlueArgParser
import com.amazonaws.services.glue.util.Job
import com.amazonaws.services.glue.util.JsonOptions
import org.apache.spark.SparkContext
import scala.collection.JavaConverters._

object GlueApp {
    def main(sysArgs: Array[String]) {
        val spark: SparkContext = new SparkContext()
        val glueContext: GlueContext = new GlueContext(spark)
        // @params: [JOB_NAME]
        val args = GlueArgParser.getResolvedOptions(sysArgs,
 Seq("JOB_NAME").toArray)
       Job.init(args("JOB_NAME"), glueContext, args.asJava)
       // @type: DataSource
       // @args: [database = "tempdb", table_name = "s3-source",
 transformation_ctx = "datasource0"]
       // @return: datasource0
       // @inputs: []
       val datasource0 = glueContext.getCatalogSource(database = "tempdb",
 tableName = "s3-source", redshiftTmpDir = "", transformationContext =
 "datasource0").getDynamicFrame()
       // @type: ApplyMapping
       // @args: [mapping = [("sensorid", "int", "sensorid", "int"),
 ("currenttemperature", "int", "currenttemperature", "int"), ("status", "string",
 "status", "string")], transformation_ctx = "applymapping1"]
```

```
        // @return: applymapping1
        // @inputs: [frame = datasource0]
        val applymapping1 = datasource0.applyMapping(mappings = Seq(("sensorid",
 "int", "sensorid", "int"), ("currenttemperature", "int", "currenttemperature",
 "int"), ("status", "string", "status", "string")), caseSensitive = false,
 transformationContext = "applymapping1")
        // @type: SelectFields
        // @args: [paths = ["sensorid", "currenttemperature", "status"],
 transformation_ctx = "selectfields2"]
        // @return: selectfields2
        // @inputs: [frame = applymapping1]
        val selectfields2 = applymapping1.selectFields(paths = Seq("sensorid",
 "currenttemperature", "status"), transformationContext = "selectfields2")
        // @type: ResolveChoice
        // @args: [choice = "MATCH_CATALOG", database = "tempdb", table_name =
 "my-s3-sink", transformation_ctx = "resolvechoice3"]
        // @return: resolvechoice3
        // @inputs: [frame = selectfields2]
        val resolvechoice3 = selectfields2.resolveChoice(choiceOption =
 Some(ChoiceOption("MATCH_CATALOG")), database = Some("tempdb"), tableName =
 Some("my-s3-sink"), transformationContext = "resolvechoice3")
        // @type: DataSink
        // @args: [database = "tempdb", table_name = "my-s3-sink",
 transformation_ctx = "datasink4"]
        // @return: datasink4
        // @inputs: [frame = resolvechoice3]
        val datasink4 = glueContext.getCatalogSink(database = "tempdb",
 tableName = "my-s3-sink", redshiftTmpDir = "", transformationContext =
 "datasink4").writeDynamicFrame(resolvechoice3)
        Job.commit()
    }
}
```

Output:

```
{
    "Name": "my-testing-job"
}
```

Per ulteriori informazioni, consulta Authoring Jobs in AWS Glue nella Glue Developer Guide.AWS

- Per i dettagli sull'API, consulta CreateJob AWS CLICommand Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
   /**
    * Creates a new AWS Glue job.
    *
    * @param glueClient     the AWS Glue client to use for the operation
    * @param jobName        the name of the job to create
    * @param iam            the IAM role to associate with the job
    * @param scriptLocation the location of the script to be used by the job
    * @throws GlueException if there is an error creating the job
    */
   public static void createJob(GlueClient glueClient, String jobName, String
iam, String scriptLocation) {
       try {
           JobCommand command = JobCommand.builder()
               .pythonVersion("3")
               .name("glueetl")
               .scriptLocation(scriptLocation)
               .build();

           CreateJobRequest jobRequest = CreateJobRequest.builder()
               .description("A Job created by using the AWS SDK for Java V2")
               .glueVersion("2.0")
               .workerType(WorkerType.G_1_X)
               .numberOfWorkers(10)
               .name(jobName)
               .role(iam)
               .command(command)
               .build();

           glueClient.createJob(jobRequest);
           System.out.println(jobName + " was successfully created.");
```

```
        } catch (GlueException e) {
            throw e;
        }
    }
```

- Per i dettagli sull'API, consulta la CreateJobsezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
const createJob = (name, role, scriptBucketName, scriptKey) => {
  const client = new GlueClient({});

  const command = new CreateJobCommand({
    Name: name,
    Role: role,
    Command: {
      Name: "glueetl",
      PythonVersion: "3",
      ScriptLocation: `s3://${scriptBucketName}/${scriptKey}`,
    },
    GlueVersion: "3.0",
  });

  return client.send(command);
};
```

- Per i dettagli sull'API, consulta la CreateJobsezione AWS SDK per JavaScript API Reference.

PHP

SDK per PHP

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
        $role = $iamService->getRole("AWSGlueServiceRole-DocExample");

        $jobName = 'test-job-' . $uniqid;

        $scriptLocation = "s3://$bucketName/run_job.py";
        $job = $glueService->createJob($jobName, $role['Role']['Arn'],
$scriptLocation);

    public function createJob($jobName, $role, $scriptLocation, $pythonVersion =
'3', $glueVersion = '3.0'): Result
    {
        return $this->glueClient->createJob([
            'Name' => $jobName,
            'Role' => $role,
            'Command' => [
                'Name' => 'glueetl',
                'ScriptLocation' => $scriptLocation,
                'PythonVersion' => $pythonVersion,
            ],
            'GlueVersion' => $glueVersion,
        ]);
    }
```

- Per i dettagli sull'API, consulta la CreateJobsezione AWS SDK per PHP API Reference.

PowerShell

Strumenti per PowerShell V4

Esempio 1: Questo esempio crea un nuovo lavoro in AWS Glue. Il valore del nome del comando è sempre**`glueetl`**. AWS Glue supporta l'esecuzione di script di lavoro scritti in Python o Scala. In questo esempio, il job script (MyTestGlueJob.py) è scritto in Python. I parametri Python vengono specificati nella **`$DefArgs`** variabile e quindi passati al PowerShell comando nel **`DefaultArguments`** parametro, che accetta una tabella hash. I parametri nella **`$JobParams`** variabile provengono dall' CreateJob API, documentata nell'argomento Jobs (https://docs.aws.amazon.com/glue/latest/dg/aws-glue-api-jobs-job.html) del riferimento all'API AWS Glue.

```
$Command = New-Object Amazon.Glue.Model.JobCommand
$Command.Name = 'glueetl'
$Command.ScriptLocation = 's3://amzn-s3-demo-source-bucket/admin/
MyTestGlueJob.py'
$Command

$Source = "source_test_table"
$Target = "target_test_table"
$Connections = $Source, $Target

$DefArgs = @{
     '--TempDir' = 's3://amzn-s3-demo-bucket/admin'
     '--job-bookmark-option' = 'job-bookmark-disable'
     '--job-language' = 'python'
     }
$DefArgs

$ExecutionProp = New-Object Amazon.Glue.Model.ExecutionProperty
$ExecutionProp.MaxConcurrentRuns = 1
$ExecutionProp

$JobParams = @{
    "AllocatedCapacity"    = "5"
    "Command"              = $Command
    "Connections_Connection" = $Connections
    "DefaultArguments"  = $DefArgs
    "Description"       = "This is a test"
    "ExecutionProperty" = $ExecutionProp
    "MaxRetries"        = "1"
```

```
    "Name"               = "MyOregonTestGlueJob"
    "Role"               = "Amazon-GlueServiceRoleForSSM"
    "Timeout"            = "20"
     }

New-GlueJob @JobParams
```

- Per i dettagli sull'API, vedere CreateJobin AWS Strumenti per PowerShell Cmdlet Reference (V4).

Strumenti per V5 PowerShell

Esempio 1: Questo esempio crea un nuovo lavoro in AWS Glue. Il valore del nome del comando è sempre**`glueetl`**. AWS Glue supporta l'esecuzione di script di lavoro scritti in Python o Scala. In questo esempio, il job script (MyTestGlueJob.py) è scritto in Python. I parametri Python vengono specificati nella **`$DefArgs`** variabile e quindi passati al PowerShell comando nel **`DefaultArguments`** parametro, che accetta una tabella hash. I parametri nella **`$JobParams`** variabile provengono dall' CreateJob API, documentata nell'argomento Jobs (https://docs.aws.amazon.com/glue/latest/dg/aws-glue-api-jobs-job.html) del riferimento all'API AWS Glue.

```
$Command = New-Object Amazon.Glue.Model.JobCommand
$Command.Name = 'glueetl'
$Command.ScriptLocation = 's3://amzn-s3-demo-source-bucket/admin/
MyTestGlueJob.py'
$Command

$Source = "source_test_table"
$Target = "target_test_table"
$Connections = $Source, $Target

$DefArgs = @{
     '--TempDir' = 's3://amzn-s3-demo-bucket/admin'
     '--job-bookmark-option' = 'job-bookmark-disable'
     '--job-language' = 'python'
     }
$DefArgs

$ExecutionProp = New-Object Amazon.Glue.Model.ExecutionProperty
$ExecutionProp.MaxConcurrentRuns = 1
$ExecutionProp

$JobParams = @{
```

```
    "AllocatedCapacity"    = "5"
    "Command"              = $Command
    "Connections_Connection" = $Connections
    "DefaultArguments"  = $DefArgs
    "Description"       = "This is a test"
    "ExecutionProperty" = $ExecutionProp
    "MaxRetries"        = "1"
    "Name"              = "MyOregonTestGlueJob"
    "Role"              = "Amazon-GlueServiceRoleForSSM"
    "Timeout"           = "20"
     }

New-GlueJob @JobParams
```

- Per i dettagli sull'API, vedere CreateJobin AWS Strumenti per PowerShell Cmdlet Reference (V5).

Python

SDK per Python (Boto3)

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
class GlueWrapper:
    """Encapsulates AWS Glue actions."""

    def __init__(self, glue_client):
        """
        :param glue_client: A Boto3 Glue client.
        """
        self.glue_client = glue_client


    def create_job(self, name, description, role_arn, script_location):
        """
        Creates a job definition for an extract, transform, and load (ETL) job
 that can
```

```
        be run by AWS Glue.

        :param name: The name of the job definition.
        :param description: The description of the job definition.
        :param role_arn: The ARN of an IAM role that grants AWS Glue the
permissions
                         it requires to run the job.
        :param script_location: The Amazon S3 URL of a Python ETL script that is
run as
                                part of the job. The script defines how the data
is
                                transformed.
        """
        try:
            self.glue_client.create_job(
                Name=name,
                Description=description,
                Role=role_arn,
                Command={
                    "Name": "glueetl",
                    "ScriptLocation": script_location,
                    "PythonVersion": "3",
                },
                GlueVersion="3.0",
            )
        except ClientError as err:
            logger.error(
                "Couldn't create job %s. Here's why: %s: %s",
                name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
```

- Per i dettagli sull'API, consulta CreateJob AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
# The `GlueWrapper` class serves as a wrapper around the AWS Glue API, providing
 a simplified interface for common operations.
# It encapsulates the functionality of the AWS SDK for Glue and provides methods
 for interacting with Glue crawlers, databases, tables, jobs, and S3 resources.
# The class initializes with a Glue client and a logger, allowing it to make API
 calls and log any errors or informational messages.
class GlueWrapper
  def initialize(glue_client, logger)
    @glue_client = glue_client
    @logger = logger
  end

  # Creates a new job with the specified configuration.
  #
  # @param name [String] The name of the job.
  # @param description [String] The description of the job.
  # @param role_arn [String] The ARN of the IAM role to be used by the job.
  # @param script_location [String] The location of the ETL script for the job.
  # @return [void]
  def create_job(name, description, role_arn, script_location)
    @glue_client.create_job(
      name: name,
      description: description,
      role: role_arn,
      command: {
        name: 'glueetl',
        script_location: script_location,
        python_version: '3'
      },
      glue_version: '3.0'
    )
```

```
  rescue Aws::Glue::Errors::GlueException => e
    @logger.error("Glue could not create job #{name}: \n#{e.message}")
    raise
  end
```

- Per i dettagli sull'API, consulta la CreateJobsezione AWS SDK per Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
        let create_job = glue
            .create_job()
            .name(self.job())
            .role(self.iam_role.expose_secret())
            .command(
                JobCommand::builder()
                    .name("glueetl")
                    .python_version("3")
                    .script_location(format!("s3://{}/job.py", self.bucket()))
                    .build(),
            )
            .glue_version("3.0")
            .send()
            .await
            .map_err(GlueMvpError::from_glue_sdk)?;

        let job_name = create_job.name().ok_or_else(|| {
            GlueMvpError::Unknown("Did not get job name after creating
job".into())
        })?;
```

- Per i dettagli sulle API, consulta la CreateJobguida di riferimento all'API AWS SDK for Rust.

## SAP ABAP

### SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
TRY.
    " iv_job_name = 'my-etl-job'
    " iv_description = 'ETL job for data transformation'
    " iv_role_arn = 'arn:aws:iam::123456789012:role/AWSGlueServiceRole-Test'
    " iv_script_location = 's3://example-bucket/scripts/my-script.py'

    DATA(lo_command) = NEW /aws1/cl_glujobcommand(
      iv_name = 'glueetl'
      iv_scriptlocation = iv_script_location
      iv_pythonversion = '3' ).

    lo_glu->createjob(
      iv_name = iv_job_name
      iv_description = iv_description
      iv_role = iv_role_arn
      io_command = lo_command
      iv_glueversion = '3.0' ).
    MESSAGE 'Job created successfully.' TYPE 'I'.
  CATCH /aws1/cx_glualreadyexistsex.
    MESSAGE 'Job already exists.' TYPE 'E'.
  CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).
    DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).
    MESSAGE lv_invalid_error TYPE 'E'.
  CATCH /aws1/cx_gluinternalserviceex INTO DATA(lo_internal_ex).
    DATA(lv_internal_error) = lo_internal_ex->if_message~get_longtext( ).
    MESSAGE lv_internal_error TYPE 'E'.
  CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
    DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
    MESSAGE lv_timeout_error TYPE 'E'.
  CATCH /aws1/cx_gluresrcnumlmtexcdex INTO DATA(lo_limit_ex).
    DATA(lv_limit_error) = lo_limit_ex->if_message~get_longtext( ).
    MESSAGE lv_limit_error TYPE 'E'.
```

```
    ENDTRY.
```

- Per i dettagli sulle API, CreateJobconsulta AWS SDK for SAP ABAP API reference.

## Swift

### SDK per Swift

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import AWSClientRuntime
import AWSGlue

    /// Create a new AWS Glue job.
    ///
    /// - Parameters:
    ///   - glueClient: The AWS Glue client to use.
    ///   - jobName: The name to give the new job.
    ///   - role: The IAM role for the job to use when accessing AWS services.
    ///   - scriptLocation: The AWS S3 URI of the script to be run by the job.
    ///
    /// - Returns: `true` if the job is created successfully, otherwise `false`.
    func createJob(glueClient: GlueClient, name jobName: String, role: String,
                   scriptLocation: String) async -> Bool {
        let command = GlueClientTypes.JobCommand(
            name: "glueetl",
            pythonVersion: "3",
            scriptLocation: scriptLocation
        )

        do {
            _ = try await glueClient.createJob(
                input: CreateJobInput(
                    command: command,
                    description: "Created by the AWS SDK for Swift Glue basic
 scenario example.",
                    glueVersion: "3.0",
```

```
                    name: jobName,
                    numberOfWorkers: 10,
                    role: role,
                    workerType: .g1x
                )
            )
        } catch {
            return false
        }
        return true
    }
```

- Per i dettagli sull'API, consulta la CreateJobguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

## Utilizzo **`DeleteCrawler`** con un AWS SDK

Gli esempi di codice seguenti mostrano come utilizzare `DeleteCrawler`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Informazioni di base

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// Delete an AWS Glue crawler.
```

```
    /// </summary>
    /// <param name="crawlerName">The name of the crawler.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeleteCrawlerAsync(string crawlerName)
    {
        var response = await _amazonGlue.DeleteCrawlerAsync(new
DeleteCrawlerRequest { Name = crawlerName });
        return response.HttpStatusCode == HttpStatusCode.OK;
    }
```

- Per i dettagli sull'API, DeleteCrawlerconsulta AWS SDK per .NET API Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region in which the bucket was created
(overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::Glue::GlueClient client(clientConfig);

        Aws::Glue::Model::DeleteCrawlerRequest request;
        request.SetName(crawler);

        Aws::Glue::Model::DeleteCrawlerOutcome outcome =
client.DeleteCrawler(request);

        if (outcome.IsSuccess()) {
            std::cout << "Successfully deleted the crawler." << std::endl;
        }
        else {
            std::cerr << "Error deleting the crawler. "
```

```
                    << outcome.GetError().GetMessage() << std::endl;
            result = false;
        }
```

- Per i dettagli sull'API, DeleteCrawlerconsulta AWS SDK per C++ API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /**
     * Deletes a specific AWS Glue crawler.
     *
     * @param glueClient  the AWS Glue client object
     * @param crawlerName the name of the crawler to be deleted
     * @throws GlueException if an error occurs during the deletion process
     */
    public static void deleteSpecificCrawler(GlueClient glueClient, String
crawlerName) {
        try {
            DeleteCrawlerRequest deleteCrawlerRequest =
DeleteCrawlerRequest.builder()
                .name(crawlerName)
                .build();

            glueClient.deleteCrawler(deleteCrawlerRequest);
            System.out.println(crawlerName + " was deleted");

        } catch (GlueException e) {
            throw e;
        }
    }
```

- Per i dettagli sull'API, DeleteCrawlerconsulta AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
const deleteCrawler = (crawlerName) => {
  const client = new GlueClient({});

  const command = new DeleteCrawlerCommand({
    Name: crawlerName,
  });

  return client.send(command);
};
```

- Per i dettagli sull'API, DeleteCrawlerconsulta AWS SDK per JavaScript API Reference.

PHP

SDK per PHP

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
        echo "Delete the crawler.\n";
        $glueClient->deleteCrawler([
            'Name' => $crawlerName,
        ]);
```

```
    public function deleteCrawler($crawlerName)
    {
        return $this->glueClient->deleteCrawler([
            'Name' => $crawlerName,
        ]);
    }
```

- Per i dettagli sull'API, DeleteCrawlerconsulta AWS SDK per PHP API Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
class GlueWrapper:
    """Encapsulates AWS Glue actions."""

    def __init__(self, glue_client):
        """
        :param glue_client: A Boto3 Glue client.
        """
        self.glue_client = glue_client


    def delete_crawler(self, name):
        """
        Deletes a crawler.

        :param name: The name of the crawler to delete.
        """
        try:
            self.glue_client.delete_crawler(Name=name)
        except ClientError as err:
            logger.error(
                "Couldn't delete crawler %s. Here's why: %s: %s",
```

```
                name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
```

- Per i dettagli sull'API, consulta DeleteCrawler AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
# The `GlueWrapper` class serves as a wrapper around the AWS Glue API, providing
 a simplified interface for common operations.
# It encapsulates the functionality of the AWS SDK for Glue and provides methods
 for interacting with Glue crawlers, databases, tables, jobs, and S3 resources.
# The class initializes with a Glue client and a logger, allowing it to make API
 calls and log any errors or informational messages.
class GlueWrapper
  def initialize(glue_client, logger)
    @glue_client = glue_client
    @logger = logger
  end

  # Deletes a crawler with the specified name.
  #
  # @param name [String] The name of the crawler to delete.
  # @return [void]
  def delete_crawler(name)
    @glue_client.delete_crawler(name: name)
  rescue Aws::Glue::Errors::ServiceError => e
    @logger.error("Glue could not delete crawler #{name}: \n#{e.message}")
    raise
```

```
end
```

- Per i dettagli sull'API, DeleteCrawlerconsulta AWS SDK per Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
        glue.delete_crawler()
            .name(self.crawler())
            .send()
            .await
            .map_err(GlueMvpError::from_glue_sdk)?;
```

- Per i dettagli sulle API, consulta il riferimento DeleteCrawlerall'API AWS SDK for Rust.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    TRY.
        " iv_crawler_name = 'my-crawler'
        lo_glu->deletecrawler( iv_name = iv_crawler_name ).
        MESSAGE 'Crawler deleted successfully.' TYPE 'I'.
      CATCH /aws1/cx_glucrawlerrunningex.
```

```
        MESSAGE 'Crawler is currently running.' TYPE 'E'.
      CATCH /aws1/cx_gluentitynotfoundex.
        MESSAGE 'Crawler does not exist.' TYPE 'E'.
      CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
        DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
        MESSAGE lv_timeout_error TYPE 'E'.
      CATCH /aws1/cx_gluschdrtransingex.
        MESSAGE 'Scheduler is transitioning.' TYPE 'E'.
    ENDTRY.
```

- Per i dettagli sulle API, DeleteCrawlerconsulta AWS SDK for SAP ABAP API reference.

## Swift

### SDK per Swift

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import AWSClientRuntime
import AWSGlue

    /// Delete an AWS Glue crawler.
    ///
    /// - Parameters:
    ///   - glueClient: The AWS Glue client to use.
    ///   - name: The name of the crawler to delete.
    ///
    /// - Returns: `true` if successful, otherwise `false`.
    func deleteCrawler(glueClient: GlueClient, name: String) async -> Bool {
        do {
            _ = try await glueClient.deleteCrawler(
                input: DeleteCrawlerInput(name: name)
            )
        } catch {
            return false
        }
        return true
```

```
    }
```

- Per i dettagli sull'API, consulta la DeleteCrawlerguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

## Utilizzo **`DeleteDatabase`** con un AWS SDK

Gli esempi di codice seguenti mostrano come utilizzare `DeleteDatabase`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Informazioni di base
- Guida introduttiva al Glue Data Catalog

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// Delete the AWS Glue database.
    /// </summary>
    /// <param name="dbName">The name of the database.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeleteDatabaseAsync(string dbName)
    {
        var response = await _amazonGlue.DeleteDatabaseAsync(new
DeleteDatabaseRequest { Name = dbName });
        return response.HttpStatusCode == HttpStatusCode.OK;
```

```
    }
```

- Per i dettagli sull'API, DeleteDatabaseconsulta AWS SDK per .NET API Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
       Aws::Client::ClientConfiguration clientConfig;
       // Optional: Set to the AWS Region in which the bucket was created
(overrides config file).
       // clientConfig.region = "us-east-1";

   Aws::Glue::GlueClient client(clientConfig);

       Aws::Glue::Model::DeleteDatabaseRequest request;
       request.SetName(database);

       Aws::Glue::Model::DeleteDatabaseOutcome outcome = client.DeleteDatabase(
               request);

       if (outcome.IsSuccess()) {
           std::cout << "Successfully deleted the database." << std::endl;
       }
       else {
           std::cerr << "Error deleting database. " <<
outcome.GetError().GetMessage()
                     << std::endl;
           result = false;
       }
```

- Per i dettagli sull'API, DeleteDatabaseconsulta AWS SDK per C++ API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
   /**
    * Deletes a AWS Glue Database.
    *
    * @param glueClient   An instance of the AWS Glue client used to interact
with the AWS Glue service.
    * @param databaseName The name of the database to be deleted.
    * @throws GlueException If an error occurs while deleting the database.
    */
   public static void deleteDatabase(GlueClient glueClient, String databaseName)
{
       try {
           DeleteDatabaseRequest request = DeleteDatabaseRequest.builder()
               .name(databaseName)
               .build();

           glueClient.deleteDatabase(request);
           System.out.println(databaseName + " was successfully deleted");

       } catch (GlueException e) {
           throw e;
       }
   }
```

- Per i dettagli sull'API, DeleteDatabaseconsulta AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
const deleteDatabase = (databaseName) => {
  const client = new GlueClient({});

  const command = new DeleteDatabaseCommand({
    Name: databaseName,
  });

  return client.send(command);
};
```

- Per i dettagli sull'API, DeleteDatabaseconsulta AWS SDK per JavaScript API Reference.

PHP

SDK per PHP

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
        echo "Delete the databases.\n";
        $glueClient->deleteDatabase([
            'Name' => $databaseName,
        ]);

    public function deleteDatabase($databaseName)
    {
```

```
        return $this->glueClient->deleteDatabase([
            'Name' => $databaseName,
        ]);
    }
```

- Per i dettagli sull'API, DeleteDatabaseconsulta AWS SDK per PHP API Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
class GlueWrapper:
    """Encapsulates AWS Glue actions."""

    def __init__(self, glue_client):
        """
        :param glue_client: A Boto3 Glue client.
        """
        self.glue_client = glue_client


    def delete_database(self, name):
        """
        Deletes a metadata database from your Data Catalog.

        :param name: The name of the database to delete.
        """
        try:
            self.glue_client.delete_database(Name=name)
        except ClientError as err:
            logger.error(
                "Couldn't delete database %s. Here's why: %s: %s",
                name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
```

```
            )
            raise
```

- Per i dettagli sull'API, consulta DeleteDatabase AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
# The `GlueWrapper` class serves as a wrapper around the AWS Glue API, providing
 a simplified interface for common operations.
# It encapsulates the functionality of the AWS SDK for Glue and provides methods
 for interacting with Glue crawlers, databases, tables, jobs, and S3 resources.
# The class initializes with a Glue client and a logger, allowing it to make API
 calls and log any errors or informational messages.
class GlueWrapper
  def initialize(glue_client, logger)
    @glue_client = glue_client
    @logger = logger
  end

  # Removes a specified database from a Data Catalog.
  #
  # @param database_name [String] The name of the database to delete.
  # @return [void]
  def delete_database(database_name)
    @glue_client.delete_database(name: database_name)
  rescue Aws::Glue::Errors::ServiceError => e
    @logger.error("Glue could not delete database: \n#{e.message}")
  end
```

- Per i dettagli sull'API, DeleteDatabaseconsulta AWS SDK per Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
        glue.delete_database()
            .name(self.database())
            .send()
            .await
            .map_err(GlueMvpError::from_glue_sdk)?;
```

- Per i dettagli sulle API, consulta il riferimento DeleteDatabaseall'API AWS SDK for Rust.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    TRY.
        " iv_database_name = 'my-database'
        lo_glu->deletedatabase( iv_name = iv_database_name ).
        MESSAGE 'Database deleted successfully.' TYPE 'I'.
      CATCH /aws1/cx_gluentitynotfoundex.
        MESSAGE 'Database does not exist.' TYPE 'E'.
      CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).
        DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).
```

```
        MESSAGE lv_invalid_error TYPE 'E'.
      CATCH /aws1/cx_gluinternalserviceex INTO DATA(lo_internal_ex).
        DATA(lv_internal_error) = lo_internal_ex->if_message~get_longtext( ).
        MESSAGE lv_internal_error TYPE 'E'.
      CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
        DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
        MESSAGE lv_timeout_error TYPE 'E'.
    ENDTRY.
```

- Per i dettagli sulle API, DeleteDatabaseconsulta AWS SDK for SAP ABAP API reference.

## Swift

### SDK per Swift

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import AWSClientRuntime
import AWSGlue

    /// Delete the specified database.
    ///
    /// - Parameters:
    ///   - glueClient: The AWS Glue client to use.
    ///   - databaseName: The name of the database to delete.
    ///   - deleteTables: A Bool indicating whether or not to delete the
    ///     tables in the database before attempting to delete the database.
    ///
    /// - Returns: `true` if the database (and optionally its tables) are
    ///   deleted, otherwise `false`.
    func deleteDatabase(glueClient: GlueClient, name databaseName: String,
                        withTables deleteTables: Bool = false) async -> Bool {
        if deleteTables {
            var tableNames: [String] = []

            // Get a list of the names of all of the tables in the database.
```

```
            let tableList = await self.getTablesInDatabase(glueClient:
glueClient, databaseName: databaseName)
            for table in tableList {
                guard let name = table.name else {
                    continue
                }
                tableNames.append(name)
            }

            // Delete the tables. If there's only one table, use
            // `deleteTable()`, otherwise, use `batchDeleteTable()`. You can
            // use `batchDeleteTable()` for a single table, but this
            // demonstrates the use of `deleteTable()`.

            if tableNames.count == 1 {
                do {
                    print("    Deleting table...")
                    _ = try await glueClient.deleteTable(
                        input: DeleteTableInput(
                            databaseName: databaseName,
                            name: tableNames[0]
                        )
                    )
                } catch {
                    print("*** Unable to delete the table.")
                }
            } else {
                do {
                    print("    Deleting tables...")
                    _ = try await glueClient.batchDeleteTable(
                        input: BatchDeleteTableInput(
                            databaseName: databaseName,
                            tablesToDelete: tableNames
                        )
                    )
                } catch {
                    print("*** Unable to delete the tables.")
                }
            }
        }

        // Delete the database itself.

        do {
```

```
            print("    Deleting the database itself...")
            _ = try await glueClient.deleteDatabase(
                input: DeleteDatabaseInput(name: databaseName)
            )
        } catch {
            print("*** Unable to delete the database.")
            return false
        }
        return true
    }
```

- Per i dettagli sull'API, consulta la DeleteDatabaseguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

## Utilizzo **`DeleteJob`** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare `DeleteJob`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Informazioni di base

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
```

```
    /// Delete an AWS Glue job.
    /// </summary>
    /// <param name="jobName">The name of the job.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeleteJobAsync(string jobName)
    {
        var response = await _amazonGlue.DeleteJobAsync(new DeleteJobRequest
{ JobName = jobName });
        return response.HttpStatusCode == HttpStatusCode.OK;
    }
```

- Per i dettagli sull'API, consulta la DeleteJobsezione AWS SDK per .NET API Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region in which the bucket was created
(overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::Glue::GlueClient client(clientConfig);

        Aws::Glue::Model::DeleteJobRequest request;
        request.SetJobName(job);

        Aws::Glue::Model::DeleteJobOutcome outcome = client.DeleteJob(request);


        if (outcome.IsSuccess()) {
            std::cout << "Successfully deleted the job." << std::endl;
        }
        else {
```

```
            std::cerr << "Error deleting the job. " <<
outcome.GetError().GetMessage()
                      << std::endl;
            result = false;
        }
```

- Per i dettagli sull'API, consulta la DeleteJobsezione AWS SDK per C++ API Reference.

CLI

AWS CLI

Come eliminare un processo

L'esempio `delete-job` seguente elimina un processo non più necessario.

```
aws glue delete-job \
    --job-name my-testing-job
```

Output:

```
{
    "JobName": "my-testing-job"
}
```

Per ulteriori informazioni, consulta Working with Jobs on the AWS Glue Console nella AWS Glue Developer Guide.

- Per i dettagli sull'API, consulta DeleteJob AWS CLICommand Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /**
     * Deletes a Glue job.
     *
     * @param glueClient the Glue client to use for the operation
     * @param jobName    the name of the job to be deleted
     * @throws GlueException if there is an error deleting the job
     */
    public static void deleteJob(GlueClient glueClient, String jobName) {
        try {
            DeleteJobRequest jobRequest = DeleteJobRequest.builder()
                .jobName(jobName)
                .build();

            glueClient.deleteJob(jobRequest);
            System.out.println(jobName + " was successfully deleted");

        } catch (GlueException e) {
            throw e;
        }
    }
```

- Per i dettagli sull'API, consulta la DeleteJobsezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
const deleteJob = (jobName) => {
  const client = new GlueClient({});

  const command = new DeleteJobCommand({
    JobName: jobName,
  });
```

```
  return client.send(command);
};
```

- Per i dettagli sull'API, consulta la DeleteJobsezione AWS SDK per JavaScript API Reference.

PHP

SDK per PHP

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
        echo "Delete the job.\n";
        $glueClient->deleteJob([
            'JobName' => $job['Name'],
        ]);

    public function deleteJob($jobName)
    {
        return $this->glueClient->deleteJob([
            'JobName' => $jobName,
        ]);
    }
```

- Per i dettagli sull'API, consulta la DeleteJobsezione AWS SDK per PHP API Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
class GlueWrapper:
    """Encapsulates AWS Glue actions."""

    def __init__(self, glue_client):
        """
        :param glue_client: A Boto3 Glue client.
        """
        self.glue_client = glue_client


    def delete_job(self, job_name):
        """
        Deletes a job definition. This also deletes data about all runs that are
        associated with this job definition.

        :param job_name: The name of the job definition to delete.
        """
        try:
            self.glue_client.delete_job(JobName=job_name)
        except ClientError as err:
            logger.error(
                "Couldn't delete job %s. Here's why: %s: %s",
                job_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
```

- Per i dettagli sull'API, consulta DeleteJob AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
# The `GlueWrapper` class serves as a wrapper around the AWS Glue API, providing
 a simplified interface for common operations.
# It encapsulates the functionality of the AWS SDK for Glue and provides methods
 for interacting with Glue crawlers, databases, tables, jobs, and S3 resources.
# The class initializes with a Glue client and a logger, allowing it to make API
 calls and log any errors or informational messages.
class GlueWrapper
  def initialize(glue_client, logger)
    @glue_client = glue_client
    @logger = logger
  end

  # Deletes a job with the specified name.
  #
  # @param job_name [String] The name of the job to delete.
  # @return [void]
  def delete_job(job_name)
    @glue_client.delete_job(job_name: job_name)
  rescue Aws::Glue::Errors::ServiceError => e
    @logger.error("Glue could not delete job: \n#{e.message}")
  end
```

- Per i dettagli sull'API, consulta la DeleteJobsezione AWS SDK per Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
        glue.delete_job()
            .job_name(self.job())
            .send()
            .await
            .map_err(GlueMvpError::from_glue_sdk)?;
```

- Per i dettagli sulle API, consulta la DeleteJobguida di riferimento all'API AWS SDK for Rust.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    TRY.
        " iv_job_name = 'my-etl-job'
        lo_glu->deletejob( iv_jobname = iv_job_name ).
        MESSAGE 'Job deleted successfully.' TYPE 'I'.
      CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).
        DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).
        MESSAGE lv_invalid_error TYPE 'E'.
      CATCH /aws1/cx_gluinternalserviceex INTO DATA(lo_internal_ex).
        DATA(lv_internal_error) = lo_internal_ex->if_message~get_longtext( ).
        MESSAGE lv_internal_error TYPE 'E'.
      CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
```

```
        DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
        MESSAGE lv_timeout_error TYPE 'E'.
    ENDTRY.
```

- Per i dettagli sulle API, DeleteJobconsulta AWS SDK for SAP ABAP API reference.

Swift

SDK per Swift

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import AWSClientRuntime
import AWSGlue

    /// Delete an AWS Glue job.
    ///
    /// - Parameters:
    ///   - glueClient: The AWS Glue client to use.
    ///   - jobName: The name of the job to delete.
    ///
    /// - Returns: `true` if the job is successfully deleted, otherwise `false`.
    func deleteJob(glueClient: GlueClient, name jobName: String) async -> Bool {
        do {
            _ = try await glueClient.deleteJob(
                input: DeleteJobInput(jobName: jobName)
            )
        } catch {
            return false
        }
        return true
    }
```

- Per i dettagli sull'API, consulta la DeleteJobguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

## Utilizzo **`DeleteTable`** con un AWS SDK

Gli esempi di codice seguenti mostrano come utilizzare `DeleteTable`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Informazioni di base
- Guida introduttiva al Glue Data Catalog

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// Delete a table from an AWS Glue database.
    /// </summary>
    /// <param name="tableName">The table to delete.</param>
    /// <returns>A Boolean value indicating the success of the action.</returns>
    public async Task<bool> DeleteTableAsync(string dbName, string tableName)
    {
        var response = await _amazonGlue.DeleteTableAsync(new DeleteTableRequest
{ Name = tableName, DatabaseName = dbName });
        return response.HttpStatusCode == HttpStatusCode.OK;
    }
```

- Per i dettagli sull'API, DeleteTableconsulta AWS SDK per .NET API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
const deleteTable = (databaseName, tableName) => {
  const client = new GlueClient({});

  const command = new DeleteTableCommand({
    DatabaseName: databaseName,
    Name: tableName,
  });

  return client.send(command);
};
```

- Per i dettagli sull'API, DeleteTableconsulta AWS SDK per JavaScript API Reference.

PHP

SDK per PHP

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
        echo "Delete the tables.\n";
        foreach ($tables['TableList'] as $table) {
            $glueService->deleteTable($table['Name'], $databaseName);
        }

    public function deleteTable($tableName, $databaseName)
```

```
    {
        return $this->glueClient->deleteTable([
            'DatabaseName' => $databaseName,
            'Name' => $tableName,
        ]);
    }
```

- Per i dettagli sull'API, DeleteTableconsulta AWS SDK per PHP API Reference.

## Python

### SDK per Python (Boto3)

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
class GlueWrapper:
    """Encapsulates AWS Glue actions."""

    def __init__(self, glue_client):
        """
        :param glue_client: A Boto3 Glue client.
        """
        self.glue_client = glue_client


    def delete_table(self, db_name, table_name):
        """
        Deletes a table from a metadata database.

        :param db_name: The name of the database that contains the table.
        :param table_name: The name of the table to delete.
        """
        try:
            self.glue_client.delete_table(DatabaseName=db_name, Name=table_name)
        except ClientError as err:
            logger.error(
                "Couldn't delete table %s. Here's why: %s: %s",
```

```
                table_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
```

- Per i dettagli sull'API, consulta DeleteTable AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
# The `GlueWrapper` class serves as a wrapper around the AWS Glue API, providing
 a simplified interface for common operations.
# It encapsulates the functionality of the AWS SDK for Glue and provides methods
 for interacting with Glue crawlers, databases, tables, jobs, and S3 resources.
# The class initializes with a Glue client and a logger, allowing it to make API
 calls and log any errors or informational messages.
class GlueWrapper
  def initialize(glue_client, logger)
    @glue_client = glue_client
    @logger = logger
  end

  # Deletes a table with the specified name.
  #
  # @param database_name [String] The name of the catalog database in which the
 table resides.
  # @param table_name [String] The name of the table to be deleted.
  # @return [void]
  def delete_table(database_name, table_name)
    @glue_client.delete_table(database_name: database_name, name: table_name)
  rescue Aws::Glue::Errors::ServiceError => e
```

```
    @logger.error("Glue could not delete job: \n#{e.message}")
  end
```

- Per i dettagli sull'API, DeleteTableconsulta AWS SDK per Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
        for t in &self.tables {
            glue.delete_table()
                .name(t.name())
                .database_name(self.database())
                .send()
                .await
                .map_err(GlueMvpError::from_glue_sdk)?;
        }
```

- Per i dettagli sulle API, consulta il riferimento DeleteTableall'API AWS SDK for Rust.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    TRY.
```

```
        " iv_database_name = 'my-database'
        " iv_table_name = 'my-table'
        lo_glu->deletetable(
          iv_databasename = iv_database_name
          iv_name = iv_table_name ).
        MESSAGE 'Table deleted successfully.' TYPE 'I'.
      CATCH /aws1/cx_gluentitynotfoundex.
        MESSAGE 'Table or database does not exist.' TYPE 'E'.
      CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).
        DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).
        MESSAGE lv_invalid_error TYPE 'E'.
      CATCH /aws1/cx_gluinternalserviceex INTO DATA(lo_internal_ex).
        DATA(lv_internal_error) = lo_internal_ex->if_message~get_longtext( ).
        MESSAGE lv_internal_error TYPE 'E'.
      CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
        DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
        MESSAGE lv_timeout_error TYPE 'E'.
    ENDTRY.
```

- Per i dettagli sulle API, DeleteTableconsulta AWS SDK for SAP ABAP API reference.

## Swift

### SDK per Swift

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import AWSClientRuntime
import AWSGlue

                do {
                    print("    Deleting table...")
                    _ = try await glueClient.deleteTable(
                        input: DeleteTableInput(
                            databaseName: databaseName,
                            name: tableNames[0]
                        )
```

```
            )
        } catch {
            print("*** Unable to delete the table.")
        }
```

- Per i dettagli sull'API, consulta la DeleteTableguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

## Utilizzo **`GetCrawler`** con un AWS SDK

Gli esempi di codice seguenti mostrano come utilizzare `GetCrawler`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Informazioni di base

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
/// <summary>
/// Get information about an AWS Glue crawler.
/// </summary>
/// <param name="crawlerName">The name of the crawler.</param>
/// <returns>A Crawler object describing the crawler.</returns>
public async Task<Crawler?> GetCrawlerAsync(string crawlerName)
{
    var crawlerRequest = new GetCrawlerRequest
    {
```

```
            Name = crawlerName,
        };

        var response = await _amazonGlue.GetCrawlerAsync(crawlerRequest);
        if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
        {
            var databaseName = response.Crawler.DatabaseName;
            Console.WriteLine($"{crawlerName} has the database {databaseName}");
            return response.Crawler;
        }

        Console.WriteLine($"No information regarding {crawlerName} could be
found.");
        return null;
    }
```

- Per i dettagli sull'API, consulta la GetCrawlersezione AWS SDK per .NET API Reference.

C++

SDK per C++

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region in which the bucket was created
(overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::Glue::GlueClient client(clientConfig);

        Aws::Glue::Model::GetCrawlerRequest request;
        request.SetName(CRAWLER_NAME);

        Aws::Glue::Model::GetCrawlerOutcome outcome = client.GetCrawler(request);
```

```
       if (outcome.IsSuccess()) {
           Aws::Glue::Model::CrawlerState crawlerState =
outcome.GetResult().GetCrawler().GetState();
           std::cout << "Retrieved crawler with state " <<

Aws::Glue::Model::CrawlerStateMapper::GetNameForCrawlerState(
                            crawlerState)
                     << "." << std::endl;
       }
       else {
           std::cerr << "Error retrieving a crawler.  "
                     << outcome.GetError().GetMessage() << std::endl;
           deleteAssets(CRAWLER_NAME, CRAWLER_DATABASE_NAME, "", bucketName,
                        clientConfig);
           return false;
       }
```

- Per i dettagli sull'API, consulta la GetCrawlersezione AWS SDK per C++ API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
   /**
    * Retrieves a specific crawler from the AWS Glue service and waits for it to
be in the "READY" state.
    *
    * @param glueClient  the AWS Glue client used to interact with the Glue
service
    * @param crawlerName the name of the crawler to be retrieved
    */
   public static void getSpecificCrawler(GlueClient glueClient, String
crawlerName) throws InterruptedException {
       try {
           GetCrawlerRequest crawlerRequest = GetCrawlerRequest.builder()
```

```
                .name(crawlerName)
                .build();

            boolean ready = false;
            while (!ready) {
                GetCrawlerResponse response =
 glueClient.getCrawler(crawlerRequest);
                String status = response.crawler().stateAsString();
                if (status.compareTo("READY") == 0) {
                    ready = true;
                }
                Thread.sleep(3000);
            }

            System.out.println("The crawler is now ready");

        } catch (GlueException | InterruptedException e) {
            throw e;
        }
    }
```

- Per i dettagli sull'API, consulta la GetCrawlersezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
const getCrawler = (name) => {
  const client = new GlueClient({});

  const command = new GetCrawlerCommand({
    Name: name,
  });

  return client.send(command);
```

```
};
```

- Per i dettagli sull'API, consulta la GetCrawlersezione AWS SDK per JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun getSpecificCrawler(crawlerName: String?) {
    val request =
        GetCrawlerRequest {
            name = crawlerName
        }
    GlueClient.fromEnvironment { region = "us-east-1" }.use { glueClient ->
        val response = glueClient.getCrawler(request)
        val role = response.crawler?.role
        println("The role associated with this crawler is $role")
    }
}
```

- Per i dettagli sull'API, GetCrawlerconsulta AWS SDK for Kotlin API reference.

PHP

SDK per PHP

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
        echo "Waiting for crawler";
        do {
            $crawler = $glueService->getCrawler($crawlerName);
            echo ".";
            sleep(10);
        } while ($crawler['Crawler']['State'] != "READY");
        echo "\n";

    public function getCrawler($crawlerName)
    {
        return $this->customWaiter(function () use ($crawlerName) {
            return $this->glueClient->getCrawler([
                'Name' => $crawlerName,
            ]);
        });
    }
```

- Per i dettagli sull'API, consulta la GetCrawlersezione AWS SDK per PHP API Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
class GlueWrapper:
    """Encapsulates AWS Glue actions."""

    def __init__(self, glue_client):
        """
        :param glue_client: A Boto3 Glue client.
        """
        self.glue_client = glue_client


    def get_crawler(self, name):
```

```
        """
        Gets information about a crawler.

        :param name: The name of the crawler to look up.
        :return: Data about the crawler.
        """
        crawler = None
        try:
            response = self.glue_client.get_crawler(Name=name)
            crawler = response["Crawler"]
        except ClientError as err:
            if err.response["Error"]["Code"] == "EntityNotFoundException":
                logger.info("Crawler %s doesn't exist.", name)
            else:
                logger.error(
                    "Couldn't get crawler %s. Here's why: %s: %s",
                    name,
                    err.response["Error"]["Code"],
                    err.response["Error"]["Message"],
                )
                raise
        return crawler
```

- Per i dettagli sull'API, consulta GetCrawler AWSSDK for Python (Boto3) API Reference.

## Ruby

### SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
# The `GlueWrapper` class serves as a wrapper around the AWS Glue API, providing
 a simplified interface for common operations.
# It encapsulates the functionality of the AWS SDK for Glue and provides methods
 for interacting with Glue crawlers, databases, tables, jobs, and S3 resources.
```

```
# The class initializes with a Glue client and a logger, allowing it to make API
 calls and log any errors or informational messages.
class GlueWrapper
  def initialize(glue_client, logger)
    @glue_client = glue_client
    @logger = logger
  end

  # Retrieves information about a specific crawler.
  #
  # @param name [String] The name of the crawler to retrieve information about.
  # @return [Aws::Glue::Types::Crawler, nil] The crawler object if found, or nil
 if not found.
  def get_crawler(name)
    @glue_client.get_crawler(name: name)
  rescue Aws::Glue::Errors::EntityNotFoundException
    @logger.info("Crawler #{name} doesn't exist.")
    false
  rescue Aws::Glue::Errors::GlueException => e
    @logger.error("Glue could not get crawler #{name}: \n#{e.message}")
    raise
  end
```

- Per i dettagli sull'API, consulta la GetCrawlersezione AWS SDK per Ruby API Reference.

## Rust

### SDK per Rust

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
            let tmp_crawler = glue
                .get_crawler()
                .name(self.crawler())
                .send()
                .await
                .map_err(GlueMvpError::from_glue_sdk)?;
```

- Per i dettagli sulle API, consulta la GetCrawlerguida di riferimento all'API AWS SDK for Rust.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
TRY.
    " iv_crawler_name = 'my-crawler'
    oo_result = lo_glu->getcrawler( iv_name = iv_crawler_name ).
    DATA(lo_crawler) = oo_result->get_crawler( ).
    MESSAGE 'Crawler information retrieved.' TYPE 'I'.
  CATCH /aws1/cx_gluentitynotfoundex.
    MESSAGE 'Crawler does not exist.' TYPE 'I'.
  CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
    DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
    MESSAGE lv_timeout_error TYPE 'E'.
ENDTRY.
```

- Per i dettagli sulle API, GetCrawlerconsulta AWS SDK for SAP ABAP API reference.

Swift

SDK per Swift

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import AWSClientRuntime
import AWSGlue

    /// Get the state of the specified AWS Glue crawler.
    ///
    /// - Parameters:
    ///   - glueClient: The AWS Glue client to use.
    ///   - name: The name of the crawler whose state should be returned.
    ///
    /// - Returns: A `GlueClientTypes.CrawlerState` value describing the
    ///   state of the crawler.
    func getCrawlerState(glueClient: GlueClient, name: String) async ->
 GlueClientTypes.CrawlerState {
        do {
            let output = try await glueClient.getCrawler(
                input: GetCrawlerInput(name: name)
            )

            // If the crawler or its state is `nil`, report that the crawler
            // is stopping. This may not be what you want for your
            // application but it works for this one!

            guard let crawler = output.crawler else {
                return GlueClientTypes.CrawlerState.stopping
            }
            guard let state = crawler.state else {
                return GlueClientTypes.CrawlerState.stopping
            }
            return state
        } catch {
            return GlueClientTypes.CrawlerState.stopping
        }
    }
```

- Per i dettagli sull'API, consulta la GetCrawlerguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

## Utilizzo **`GetDatabase`** con un AWS SDK

Gli esempi di codice seguenti mostrano come utilizzare `GetDatabase`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nei seguenti esempi di codice:

- Informazioni di base
- Guida introduttiva al Glue Data Catalog

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// Get information about an AWS Glue database.
    /// </summary>
    /// <param name="dbName">The name of the database.</param>
    /// <returns>A Database object containing information about the database.</
returns>
    public async Task<Database> GetDatabaseAsync(string dbName)
    {
        var databasesRequest = new GetDatabaseRequest
        {
            Name = dbName,
        };

        var response = await _amazonGlue.GetDatabaseAsync(databasesRequest);
        return response.Database;
    }
```

- Per i dettagli sull'API, consulta la GetDatabasesezione AWS SDK per .NET API Reference.

C++

SDK per C++

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
       Aws::Client::ClientConfiguration clientConfig;
       // Optional: Set to the AWS Region in which the bucket was created
(overrides config file).
       // clientConfig.region = "us-east-1";

   Aws::Glue::GlueClient client(clientConfig);

       Aws::Glue::Model::GetDatabaseRequest request;
       request.SetName(CRAWLER_DATABASE_NAME);

       Aws::Glue::Model::GetDatabaseOutcome outcome =
client.GetDatabase(request);

       if (outcome.IsSuccess()) {
           const Aws::Glue::Model::Database &database =
outcome.GetResult().GetDatabase();

           std::cout << "Successfully retrieve the database\n" <<
                     database.Jsonize().View().WriteReadable() << "'." <<
std::endl;
       }
       else {
           std::cerr << "Error getting the database.  "
                     << outcome.GetError().GetMessage() << std::endl;
           deleteAssets(CRAWLER_NAME, CRAWLER_DATABASE_NAME, "", bucketName,
                        clientConfig);
           return false;
       }
```

- Per i dettagli sull'API, consulta la GetDatabasesezione AWS SDK per C++ API Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
   /**
    * Retrieves the specific database from the AWS Glue service.
    *
    * @param glueClient   an instance of the AWS Glue client used to interact
with the service
    * @param databaseName the name of the database to retrieve
    * @throws GlueException if there is an error retrieving the database from
the AWS Glue service
    */
   public static void getSpecificDatabase(GlueClient glueClient, String
databaseName) {
       try {
           GetDatabaseRequest databasesRequest = GetDatabaseRequest.builder()
               .name(databaseName)
               .build();

           GetDatabaseResponse response =
glueClient.getDatabase(databasesRequest);
           Instant createDate = response.database().createTime();

           // Convert the Instant to readable date.
           DateTimeFormatter formatter =
DateTimeFormatter.ofLocalizedDateTime(FormatStyle.SHORT)
               .withLocale(Locale.US)
               .withZone(ZoneId.systemDefault());

           formatter.format(createDate);
           System.out.println("The create date of the database is " +
createDate);

       } catch (GlueException e) {
           throw e;
```

```
        }
    }
```

- Per i dettagli sull'API, consulta la GetDatabasesezione AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
const getDatabase = (name) => {
  const client = new GlueClient({});

  const command = new GetDatabaseCommand({
    Name: name,
  });

  return client.send(command);
};
```

- Per i dettagli sull'API, consulta la GetDatabasesezione AWS SDK per JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun getSpecificDatabase(databaseName: String?) {
    val request =
        GetDatabaseRequest {
            name = databaseName
        }

    GlueClient.fromEnvironment { region = "us-east-1" }.use { glueClient ->
        val response = glueClient.getDatabase(request)
        val dbDesc = response.database?.description
        println("The database description is $dbDesc")
    }
}
```

- Per i dettagli sull'API, GetDatabaseconsulta AWS SDK for Kotlin API reference.

PHP

SDK per PHP

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
        $databaseName = "doc-example-database-$uniqid";

        $database = $glueService->getDatabase($databaseName);
        echo "Found a database named " . $database['Database']['Name'] . "\n";

    public function getDatabase(string $databaseName): Result
    {
        return $this->customWaiter(function () use ($databaseName) {
            return $this->glueClient->getDatabase([
                'Name' => $databaseName,
            ]);
        });
    }
```

- Per i dettagli sull'API, consulta la GetDatabasesezione AWS SDK per PHP API Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è di più su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
class GlueWrapper:
    """Encapsulates AWS Glue actions."""

    def __init__(self, glue_client):
        """
        :param glue_client: A Boto3 Glue client.
        """
        self.glue_client = glue_client


    def get_database(self, name):
        """
        Gets information about a database in your Data Catalog.

        :param name: The name of the database to look up.
        :return: Information about the database.
        """
        try:
            response = self.glue_client.get_database(Name=name)
        except ClientError as err:
            logger.error(
                "Couldn't get database %s. Here's why: %s: %s",
                name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return response["Database"]
```

- Per i dettagli sull'API, consulta GetDatabase AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
# The `GlueWrapper` class serves as a wrapper around the AWS Glue API, providing
 a simplified interface for common operations.
# It encapsulates the functionality of the AWS SDK for Glue and provides methods
 for interacting with Glue crawlers, databases, tables, jobs, and S3 resources.
# The class initializes with a Glue client and a logger, allowing it to make API
 calls and log any errors or informational messages.
class GlueWrapper
  def initialize(glue_client, logger)
    @glue_client = glue_client
    @logger = logger
  end

  # Retrieves information about a specific database.
  #
  # @param name [String] The name of the database to retrieve information about.
  # @return [Aws::Glue::Types::Database, nil] The database object if found, or
 nil if not found.
  def get_database(name)
    response = @glue_client.get_database(name: name)
    response.database
  rescue Aws::Glue::Errors::GlueException => e
    @logger.error("Glue could not get database #{name}: \n#{e.message}")
    raise
  end
```

- Per i dettagli sull'API, consulta la GetDatabasesezione AWS SDK per Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
        let database = glue
            .get_database()
            .name(self.database())
            .send()
            .await
            .map_err(GlueMvpError::from_glue_sdk)?
            .to_owned();
        let database = database
            .database()
            .ok_or_else(|| GlueMvpError::Unknown("Could not find
database".into()))?;
```

- Per i dettagli sulle API, consulta la GetDatabaseguida di riferimento all'API AWS SDK for Rust.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    TRY.
```

```
        " iv_database_name = 'my-database'
        oo_result = lo_glu->getdatabase( iv_name = iv_database_name ).
        DATA(lo_database) = oo_result->get_database( ).
        MESSAGE 'Database information retrieved.' TYPE 'I'.
      CATCH /aws1/cx_gluentitynotfoundex.
        MESSAGE 'Database does not exist.' TYPE 'E'.
      CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).
        DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).
        MESSAGE lv_invalid_error TYPE 'E'.
      CATCH /aws1/cx_gluinternalserviceex INTO DATA(lo_internal_ex).
        DATA(lv_internal_error) = lo_internal_ex->if_message~get_longtext( ).
        MESSAGE lv_internal_error TYPE 'E'.
      CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
        DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
        MESSAGE lv_timeout_error TYPE 'E'.
    ENDTRY.
```

- Per i dettagli sulle API, GetDatabaseconsulta AWS SDK for SAP ABAP API reference.

## Swift

### SDK per Swift

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import AWSClientRuntime
import AWSGlue

    /// Get the AWS Glue database with the specified name.
    ///
    /// - Parameters:
    ///   - glueClient: The AWS Glue client to use.
    ///   - name: The name of the database to return.
    ///
    /// - Returns: The `GlueClientTypes.Database` object describing the
    ///   specified database, or `nil` if an error occurs or the database
    ///   isn't found.
```

```
    func getDatabase(glueClient: GlueClient, name: String) async ->
GlueClientTypes.Database? {
        do {
            let output = try await glueClient.getDatabase(
                input: GetDatabaseInput(name: name)
            )

            return output.database
        } catch {
            return nil
        }
    }
```

- Per i dettagli sull'API, consulta la GetDatabaseguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

## Utilizzo **`GetDatabases`** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare `GetDatabases`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Informazioni di base

CLI

AWS CLI

Per elencare le definizioni di alcuni o tutti i database del AWS Glue Data Catalog

L'esempio `get-databases` seguente restituisce informazioni sui database del Catalogo dati.

```
aws glue get-databases
```

Output:

```
{
    "DatabaseList": [
        {
            "Name": "default",
            "Description": "Default Hive database",
            "LocationUri": "file:/spark-warehouse",
            "CreateTime": 1602084052.0,
            "CreateTableDefaultPermissions": [
                {
                    "Principal": {
                        "DataLakePrincipalIdentifier": "IAM_ALLOWED_PRINCIPALS"
                    },
                    "Permissions": [
                        "ALL"
                    ]
                }
            ],
            "CatalogId": "111122223333"
        },
        {
            "Name": "flights-db",
            "CreateTime": 1587072847.0,
            "CreateTableDefaultPermissions": [
                {
                    "Principal": {
                        "DataLakePrincipalIdentifier": "IAM_ALLOWED_PRINCIPALS"
                    },
                    "Permissions": [
                        "ALL"
                    ]
                }
            ],
            "CatalogId": "111122223333"
        },
        {
            "Name": "legislators",
            "CreateTime": 1601415625.0,
            "CreateTableDefaultPermissions": [
                {
                    "Principal": {
                        "DataLakePrincipalIdentifier": "IAM_ALLOWED_PRINCIPALS"
                    },
                    "Permissions": [
```

```
                        "ALL"
                    ]
                }
            ],
            "CatalogId": "111122223333"
        },
        {
            "Name": "tempdb",
            "CreateTime": 1601498566.0,
            "CreateTableDefaultPermissions": [
                {
                    "Principal": {
                        "DataLakePrincipalIdentifier": "IAM_ALLOWED_PRINCIPALS"
                    },
                    "Permissions": [
                        "ALL"
                    ]
                }
            ],
            "CatalogId": "111122223333"
        }
    ]
}
```

Per ulteriori informazioni, consulta Definizione di un database nel catalogo dati nella Guida per gli sviluppatori di AWS Glue.

- Per i dettagli sulle API, consultate GetDatabases AWS CLICommand Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
const getDatabases = () => {
  const client = new GlueClient({});
```

```
  const command = new GetDatabasesCommand({});

  return client.send(command);
};
```

- Per i dettagli sull'API, consulta la GetDatabasessezione AWS SDK per JavaScript API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

## Utilizzo **`GetJob`** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare `GetJob`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Informazioni di base

CLI

AWS CLI

Come recuperare le informazioni relative a un processo

L'esempio `get-job` seguente recupera le informazioni relative a un processo.

```
aws glue get-job \
    --job-name my-testing-job
```

Output:

```
{
    "Job": {
        "Name": "my-testing-job",
        "Role": "Glue_DefaultRole",
        "CreatedOn": 1602805698.167,
        "LastModifiedOn": 1602805698.167,
```

```
        "ExecutionProperty": {
            "MaxConcurrentRuns": 1
        },
        "Command": {
            "Name": "gluestreaming",
            "ScriptLocation": "s3://janetst-bucket-01/Scripts/test_script.scala",
            "PythonVersion": "2"
        },
        "DefaultArguments": {
            "--class": "GlueApp",
            "--job-language": "scala"
        },
        "MaxRetries": 0,
        "AllocatedCapacity": 10,
        "MaxCapacity": 10.0,
        "GlueVersion": "1.0"
    }
}
```

Per ulteriori informazioni, consulta Processi nella Guida per gli sviluppatori di AWS Glue.

- Per i dettagli sull'API, consulta GetJob AWS CLICommand Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
const getJob = (jobName) => {
  const client = new GlueClient({});

  const command = new GetJobCommand({
    JobName: jobName,
  });

  return client.send(command);
};
```

- Per i dettagli sull'API, consulta la GetJobsezione AWS SDK per JavaScript API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

## Utilizzo **`GetJobRun`** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare `GetJobRun`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Informazioni di base

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// Get information about a specific AWS Glue job run.
    /// </summary>
    /// <param name="jobName">The name of the job.</param>
    /// <param name="jobRunId">The Id of the job run.</param>
    /// <returns>A JobRun object with information about the job run.</returns>
    public async Task<JobRun> GetJobRunAsync(string jobName, string jobRunId)
    {
        var response = await _amazonGlue.GetJobRunAsync(new GetJobRunRequest
{ JobName = jobName, RunId = jobRunId });
        return response.JobRun;
    }
```

- Per i dettagli sull'API, GetJobRunconsulta AWS SDK per .NET API Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
       Aws::Client::ClientConfiguration clientConfig;
       // Optional: Set to the AWS Region in which the bucket was created
(overrides config file).
       // clientConfig.region = "us-east-1";

   Aws::Glue::GlueClient client(clientConfig);

       Aws::Glue::Model::GetJobRunRequest jobRunRequest;
       jobRunRequest.SetJobName(jobName);
       jobRunRequest.SetRunId(jobRunID);

       Aws::Glue::Model::GetJobRunOutcome jobRunOutcome = client.GetJobRun(
               jobRunRequest);

       if (jobRunOutcome.IsSuccess()) {
           std::cout << "Displaying the job run JSON description." << std::endl;
           std::cout
                   <<
jobRunOutcome.GetResult().GetJobRun().Jsonize().View().WriteReadable()
                   << std::endl;
       }
       else {
           std::cerr << "Error get a job run. "
                     << jobRunOutcome.GetError().GetMessage()
                     << std::endl;
       }
```

- Per i dettagli sull'API, GetJobRunconsulta AWS SDK per C++ API Reference.

CLI

AWS CLI

Come ottenere informazioni relative all'esecuzione di un processo

L'esempio `get-job-run` seguente recupera le informazioni relative all'esecuzione di un processo.

```
aws glue get-job-run \
    --job-name "Combine legistators data" \
    --run-id jr_012e176506505074d94d761755e5c62538ee1aad6f17d39f527e9140cf0c9a5e
```

Output:

```
{
    "JobRun": {
        "Id":
 "jr_012e176506505074d94d761755e5c62538ee1aad6f17d39f527e9140cf0c9a5e",
        "Attempt": 0,
        "JobName": "Combine legistators data",
        "StartedOn": 1602873931.255,
        "LastModifiedOn": 1602874075.985,
        "CompletedOn": 1602874075.985,
        "JobRunState": "SUCCEEDED",
        "Arguments": {
            "--enable-continuous-cloudwatch-log": "true",
            "--enable-metrics": "",
            "--enable-spark-ui": "true",
            "--job-bookmark-option": "job-bookmark-enable",
            "--spark-event-logs-path": "s3://aws-glue-assets-111122223333-us-
east-1/sparkHistoryLogs/"
        },
        "PredecessorRuns": [],
        "AllocatedCapacity": 10,
        "ExecutionTime": 117,
        "Timeout": 2880,
        "MaxCapacity": 10.0,
        "WorkerType": "G.1X",
        "NumberOfWorkers": 10,
```

```
        "LogGroupName": "/aws-glue/jobs",
        "GlueVersion": "2.0"
    }
}
```

Per ulteriori informazioni, consulta Esecuzioni di processi nella Guida per gli sviluppatori di AWS Glue.

- Per i dettagli sull'API, consulta GetJobRun AWS CLICommand Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
const getJobRun = (jobName, jobRunId) => {
  const client = new GlueClient({});
  const command = new GetJobRunCommand({
    JobName: jobName,
    RunId: jobRunId,
  });

  return client.send(command);
};
```

- Per i dettagli sull'API, GetJobRunconsulta AWS SDK per JavaScript API Reference.

PHP

SDK per PHP

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
        $jobName = 'test-job-' . $uniqid;

        $outputBucketUrl = "s3://$bucketName";
        $runId = $glueService->startJobRun($jobName, $databaseName, $tables,
$outputBucketUrl)['JobRunId'];

        echo "waiting for job";
        do {
            $jobRun = $glueService->getJobRun($jobName, $runId);
            echo ".";
            sleep(10);
        } while (!array_intersect([$jobRun['JobRun']['JobRunState']],
['SUCCEEDED', 'STOPPED', 'FAILED', 'TIMEOUT']));
        echo "\n";

    public function getJobRun($jobName, $runId, $predecessorsIncluded = false):
Result
    {
        return $this->glueClient->getJobRun([
            'JobName' => $jobName,
            'RunId' => $runId,
            'PredecessorsIncluded' => $predecessorsIncluded,
        ]);
    }
```

- Per i dettagli sull'API, GetJobRunconsulta AWS SDK per PHP API Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
class GlueWrapper:
    """Encapsulates AWS Glue actions."""

    def __init__(self, glue_client):
        """
        :param glue_client: A Boto3 Glue client.
        """
        self.glue_client = glue_client


    def get_job_run(self, name, run_id):
        """
        Gets information about a single job run.

        :param name: The name of the job definition for the run.
        :param run_id: The ID of the run.
        :return: Information about the run.
        """
        try:
            response = self.glue_client.get_job_run(JobName=name, RunId=run_id)
        except ClientError as err:
            logger.error(
                "Couldn't get job run %s/%s. Here's why: %s: %s",
                name,
                run_id,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return response["JobRun"]
```

- Per i dettagli sull'API, consulta GetJobRun AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
# The `GlueWrapper` class serves as a wrapper around the AWS Glue API, providing
 a simplified interface for common operations.
# It encapsulates the functionality of the AWS SDK for Glue and provides methods
 for interacting with Glue crawlers, databases, tables, jobs, and S3 resources.
# The class initializes with a Glue client and a logger, allowing it to make API
 calls and log any errors or informational messages.
class GlueWrapper
  def initialize(glue_client, logger)
    @glue_client = glue_client
    @logger = logger
  end

  # Retrieves data for a specific job run.
  #
  # @param job_name [String] The name of the job run to retrieve data for.
  # @return [Glue::Types::GetJobRunResponse]
  def get_job_run(job_name, run_id)
    @glue_client.get_job_run(job_name: job_name, run_id: run_id)
  rescue Aws::Glue::Errors::GlueException => e
    @logger.error("Glue could not get job runs: \n#{e.message}")
  end
```

- Per i dettagli sull'API, GetJobRunconsulta AWS SDK per Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
        let get_job_run = || async {
            Ok::<JobRun, GlueMvpError>(
                glue.get_job_run()
                    .job_name(self.job())
                    .run_id(job_run_id.to_string())
                    .send()
                    .await
                    .map_err(GlueMvpError::from_glue_sdk)?
                    .job_run()
                    .ok_or_else(|| GlueMvpError::Unknown("Failed to get
job_run".into()))?
                    .to_owned(),
            )
        };

        let mut job_run = get_job_run().await?;
        let mut state =
job_run.job_run_state().unwrap_or(&unknown_state).to_owned();

        while matches!(
            state,
            JobRunState::Starting | JobRunState::Stopping | JobRunState::Running
        ) {
            info!(?state, "Waiting for job to finish");
            tokio::time::sleep(self.wait_delay).await;

            job_run = get_job_run().await?;
            state = job_run.job_run_state().unwrap_or(&unknown_state).to_owned();
        }
```

- Per i dettagli sulle API, consulta il riferimento GetJobRunall'API AWS SDK for Rust.

## SAP ABAP

### SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
TRY.
    " iv_job_name = 'my-etl-job'
    " iv_run_id = 'jr_abcd1234567890abcdef1234567890abcdef12345678'
    oo_result = lo_glu->getjobrun(
      iv_jobname = iv_job_name
      iv_runid = iv_run_id ).
    DATA(lo_job_run) = oo_result->get_jobrun( ).
    MESSAGE 'Job run information retrieved.' TYPE 'I'.
  CATCH /aws1/cx_gluentitynotfoundex.
    MESSAGE 'Job or job run does not exist.' TYPE 'E'.
  CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).
    DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).
    MESSAGE lv_invalid_error TYPE 'E'.
  CATCH /aws1/cx_gluinternalserviceex INTO DATA(lo_internal_ex).
    DATA(lv_internal_error) = lo_internal_ex->if_message~get_longtext( ).
    MESSAGE lv_internal_error TYPE 'E'.
  CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
    DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
    MESSAGE lv_timeout_error TYPE 'E'.
ENDTRY.
```

- Per i dettagli sulle API, GetJobRunconsulta AWS SDK for SAP ABAP API reference.

Swift

SDK per Swift

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import AWSClientRuntime
import AWSGlue

    /// Get information about a specific AWS Glue job run.
    ///
    /// - Parameters:
    ///   - glueClient: The AWS Glue client to use.
    ///   - jobName: The name of the job to return job run data for.
    ///   - id: The run ID of the specific job run to return.
    ///
    /// - Returns: A `GlueClientTypes.JobRun` object describing the state of
    ///   the job run, or `nil` if an error occurs.
    func getJobRun(glueClient: GlueClient, name jobName: String, id: String)
 async -> GlueClientTypes.JobRun? {
        do {
            let output = try await glueClient.getJobRun(
                input: GetJobRunInput(
                    jobName: jobName,
                    runId: id
                )
            )

            return output.jobRun
        } catch {
            return nil
        }
    }
```

- Per i dettagli sull'API, consulta la GetJobRunguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

## Utilizzo **`GetJobRuns`** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare `GetJobRuns`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Informazioni di base

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// Get information about all AWS Glue runs of a specific job.
    /// </summary>
    /// <param name="jobName">The name of the job.</param>
    /// <returns>A list of JobRun objects.</returns>
    public async Task<List<JobRun>> GetJobRunsAsync(string jobName)
    {
        var jobRuns = new List<JobRun>();

        var request = new GetJobRunsRequest
        {
            JobName = jobName,
        };

        // No need to loop to get all the log groups--the SDK does it for us
behind the scenes
        var paginatorForJobRuns =
            _amazonGlue.Paginators.GetJobRuns(request);
```

```
        await foreach (var response in paginatorForJobRuns.Responses)
        {
            response.JobRuns.ForEach(jobRun =>
            {
                jobRuns.Add(jobRun);
            });
        }

        return jobRuns;
    }
```

- Per i dettagli sull'API, GetJobRunsconsulta AWS SDK per .NET API Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
       Aws::Client::ClientConfiguration clientConfig;
       // Optional: Set to the AWS Region in which the bucket was created
(overrides config file).
       // clientConfig.region = "us-east-1";

   Aws::Glue::GlueClient client(clientConfig);

       Aws::Glue::Model::GetJobRunsRequest getJobRunsRequest;
       getJobRunsRequest.SetJobName(jobName);

       Aws::String nextToken; // Used for pagination.
       std::vector<Aws::Glue::Model::JobRun> allJobRuns;
       do {
           if (!nextToken.empty()) {
               getJobRunsRequest.SetNextToken(nextToken);
           }
```

```
            Aws::Glue::Model::GetJobRunsOutcome jobRunsOutcome =
client.GetJobRuns(
                    getJobRunsRequest);

            if (jobRunsOutcome.IsSuccess()) {
                const std::vector<Aws::Glue::Model::JobRun> &jobRuns =
jobRunsOutcome.GetResult().GetJobRuns();
                allJobRuns.insert(allJobRuns.end(), jobRuns.begin(),
jobRuns.end());

                nextToken = jobRunsOutcome.GetResult().GetNextToken();
            }
            else {
                std::cerr << "Error getting job runs. "
                          << jobRunsOutcome.GetError().GetMessage()
                          << std::endl;
                break;
            }
        } while (!nextToken.empty());
```

- Per i dettagli sull'API, GetJobRunsconsulta AWS SDK per C++ API Reference.

CLI

AWS CLI

Come ottenere informazioni su tutte le esecuzioni di processo per un determinato processo

L'esempio `get-job-runs` seguente recupera informazioni sulle esecuzioni di processo per un determinato processo.

```
aws glue get-job-runs \
    --job-name "my-testing-job"
```

Output:

```
{
    "JobRuns": [
        {
            "Id":
 "jr_012e176506505074d94d761755e5c62538ee1aad6f17d39f527e9140cf0c9a5e",
```

```
            "Attempt": 0,
            "JobName": "my-testing-job",
            "StartedOn": 1602873931.255,
            "LastModifiedOn": 1602874075.985,
            "CompletedOn": 1602874075.985,
            "JobRunState": "SUCCEEDED",
            "Arguments": {
                "--enable-continuous-cloudwatch-log": "true",
                "--enable-metrics": "",
                "--enable-spark-ui": "true",
                "--job-bookmark-option": "job-bookmark-enable",
                "--spark-event-logs-path": "s3://aws-glue-assets-111122223333-us-
east-1/sparkHistoryLogs/"
            },
            "PredecessorRuns": [],
            "AllocatedCapacity": 10,
            "ExecutionTime": 117,
            "Timeout": 2880,
            "MaxCapacity": 10.0,
            "WorkerType": "G.1X",
            "NumberOfWorkers": 10,
            "LogGroupName": "/aws-glue/jobs",
            "GlueVersion": "2.0"
        },
        {
            "Id":
 "jr_03cc19ddab11c4e244d3f735567de74ff93b0b3ef468a713ffe73e53d1aec08f_attempt_2",
            "Attempt": 2,
            "PreviousRunId":
 "jr_03cc19ddab11c4e244d3f735567de74ff93b0b3ef468a713ffe73e53d1aec08f_attempt_1",
            "JobName": "my-testing-job",
            "StartedOn": 1602811168.496,
            "LastModifiedOn": 1602811282.39,
            "CompletedOn": 1602811282.39,
            "JobRunState": "FAILED",
            "ErrorMessage": "An error occurred while calling
 o122.pyWriteDynamicFrame.
                Access Denied (Service: Amazon S3; Status Code: 403; Error Code:
 AccessDenied;
                Request ID: 021AAB703DB20A2D;
                S3 Extended Request ID: teZk24Y09TkXzBvMPG502L5VJBhe9DJuWA9/
TXtuGOqfByajkfL/Tlqt5JBGdEGpigAqzdMDM/U=)",
            "PredecessorRuns": [],
            "AllocatedCapacity": 10,
```

```
            "ExecutionTime": 110,
            "Timeout": 2880,
            "MaxCapacity": 10.0,
            "WorkerType": "G.1X",
            "NumberOfWorkers": 10,
            "LogGroupName": "/aws-glue/jobs",
            "GlueVersion": "2.0"
        },
        {
            "Id":
 "jr_03cc19ddab11c4e244d3f735567de74ff93b0b3ef468a713ffe73e53d1aec08f_attempt_1",
            "Attempt": 1,
            "PreviousRunId":
 "jr_03cc19ddab11c4e244d3f735567de74ff93b0b3ef468a713ffe73e53d1aec08f",
            "JobName": "my-testing-job",
            "StartedOn": 1602811020.518,
            "LastModifiedOn": 1602811138.364,
            "CompletedOn": 1602811138.364,
            "JobRunState": "FAILED",
            "ErrorMessage": "An error occurred while calling
 o122.pyWriteDynamicFrame.
                 Access Denied (Service: Amazon S3; Status Code: 403; Error Code:
 AccessDenied;
                 Request ID: 2671D37856AE7ABB;
                 S3 Extended Request ID: RLJCJw20brV
+PpC6GpORahyF2fp9flB5SSb2bTGPnUSPVizLXRl1PN3QZldb+v1o9qRVktNYbW8=)",
            "PredecessorRuns": [],
            "AllocatedCapacity": 10,
            "ExecutionTime": 113,
            "Timeout": 2880,
            "MaxCapacity": 10.0,
            "WorkerType": "G.1X",
            "NumberOfWorkers": 10,
            "LogGroupName": "/aws-glue/jobs",
            "GlueVersion": "2.0"
        }
    ]
}
```


Per ulteriori informazioni, consulta Esecuzioni di processi nella Guida per gli sviluppatori di AWS Glue.

- Per i dettagli sull'API, consulta GetJobRuns AWS CLICommand Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
   /**
    * Retrieves the job runs for a given Glue job and prints the status of the
job runs.
    *
    * @param glueClient the Glue client used to make API calls
    * @param jobName    the name of the Glue job to retrieve the job runs for
    */
   public static void getJobRuns(GlueClient glueClient, String jobName) {
       try {
           GetJobRunsRequest runsRequest = GetJobRunsRequest.builder()
               .jobName(jobName)
               .maxResults(20)
               .build();

           boolean jobDone = false;
           while (!jobDone) {
               GetJobRunsResponse response = glueClient.getJobRuns(runsRequest);
               List<JobRun> jobRuns = response.jobRuns();
               for (JobRun jobRun : jobRuns) {
                   String jobState = jobRun.jobRunState().name();
                   if (jobState.compareTo("SUCCEEDED") == 0) {
                       System.out.println(jobName + " has succeeded");
                       jobDone = true;

                   } else if (jobState.compareTo("STOPPED") == 0) {
                       System.out.println("Job run has stopped");
                       jobDone = true;

                   } else if (jobState.compareTo("FAILED") == 0) {
                       System.out.println("Job run has failed");
                       jobDone = true;
```

```
                    } else if (jobState.compareTo("TIMEOUT") == 0) {
                        System.out.println("Job run has timed out");
                        jobDone = true;

                    } else {
                        System.out.println("*** Job run state is " +
jobRun.jobRunState().name());
                        System.out.println("Job run Id is " + jobRun.id());
                        System.out.println("The Glue version is " +
jobRun.glueVersion());
                    }
                    TimeUnit.SECONDS.sleep(5);
                }
            }

        } catch (GlueException e) {
            throw e;
        } catch (InterruptedException e) {
            throw new RuntimeException(e);
        }
    }
```

- Per i dettagli sull'API, GetJobRunsconsulta AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
const getJobRuns = (jobName) => {
  const client = new GlueClient({});
  const command = new GetJobRunsCommand({
    JobName: jobName,
  });

  return client.send(command);
```

```
};
```

- Per i dettagli sull'API, GetJobRunsconsulta AWS SDK per JavaScript API Reference.

PHP

SDK per PHP

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
        $jobName = 'test-job-' . $uniqid;

        $jobRuns = $glueService->getJobRuns($jobName);

    public function getJobRuns($jobName, $maxResults = 0, $nextToken = ''):
Result
    {
        $arguments = ['JobName' => $jobName];
        if ($maxResults) {
            $arguments['MaxResults'] = $maxResults;
        }
        if ($nextToken) {
            $arguments['NextToken'] = $nextToken;
        }
        return $this->glueClient->getJobRuns($arguments);
    }
```

- Per i dettagli sull'API, GetJobRunsconsulta AWS SDK per PHP API Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
class GlueWrapper:
    """Encapsulates AWS Glue actions."""

    def __init__(self, glue_client):
        """
        :param glue_client: A Boto3 Glue client.
        """
        self.glue_client = glue_client


    def get_job_runs(self, job_name):
        """
        Gets information about runs that have been performed for a specific job
        definition.

        :param job_name: The name of the job definition to look up.
        :return: The list of job runs.
        """
        try:
            response = self.glue_client.get_job_runs(JobName=job_name)
        except ClientError as err:
            logger.error(
                "Couldn't get job runs for %s. Here's why: %s: %s",
                job_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return response["JobRuns"]
```

- Per i dettagli sull'API, consulta GetJobRuns AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
# The `GlueWrapper` class serves as a wrapper around the AWS Glue API, providing
 a simplified interface for common operations.
# It encapsulates the functionality of the AWS SDK for Glue and provides methods
 for interacting with Glue crawlers, databases, tables, jobs, and S3 resources.
# The class initializes with a Glue client and a logger, allowing it to make API
 calls and log any errors or informational messages.
class GlueWrapper
  def initialize(glue_client, logger)
    @glue_client = glue_client
    @logger = logger
  end

  # Retrieves a list of job runs for the specified job.
  #
  # @param job_name [String] The name of the job to retrieve job runs for.
  # @return [Array<Aws::Glue::Types::JobRun>]
  def get_job_runs(job_name)
    response = @glue_client.get_job_runs(job_name: job_name)
    response.job_runs
  rescue Aws::Glue::Errors::GlueException => e
    @logger.error("Glue could not get job runs: \n#{e.message}")
  end
```

- Per i dettagli sull'API, GetJobRunsconsulta AWS SDK per Ruby API Reference.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
TRY.
    " iv_job_name = 'my-etl-job'
    oo_result = lo_glu->getjobruns( iv_jobname = iv_job_name ).
    DATA(lt_job_runs) = oo_result->get_jobruns( ).
    MESSAGE 'Job runs retrieved successfully.' TYPE 'I'.
  CATCH /aws1/cx_gluentitynotfoundex.
    MESSAGE 'Job does not exist.' TYPE 'E'.
  CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).
    DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).
    MESSAGE lv_invalid_error TYPE 'E'.
  CATCH /aws1/cx_gluinternalserviceex INTO DATA(lo_internal_ex).
    DATA(lv_internal_error) = lo_internal_ex->if_message~get_longtext( ).
    MESSAGE lv_internal_error TYPE 'E'.
  CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
    DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
    MESSAGE lv_timeout_error TYPE 'E'.
ENDTRY.
```

- Per i dettagli sulle API, GetJobRunsconsulta AWS SDK for SAP ABAP API reference.

Swift

SDK per Swift

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import AWSClientRuntime
import AWSGlue

    /// Return a list of the job runs for the specified job.
    ///
    /// - Parameters:
    ///   - glueClient: The AWS Glue client to use.
    ///   - jobName: The name of the job for which to return its job runs.
    ///   - maxResults: The maximum number of job runs to return (default:
    ///     1000).
    ///
    /// - Returns: An array of `GlueClientTypes.JobRun` objects describing
    ///   each job run.
    func getJobRuns(glueClient: GlueClient, name jobName: String, maxResults:
 Int? = nil) async -> [GlueClientTypes.JobRun] {
        do {
            let output = try await glueClient.getJobRuns(
                input: GetJobRunsInput(
                    jobName: jobName,
                    maxResults: maxResults
                )
            )

            guard let jobRuns = output.jobRuns else {
                print("*** No job runs found.")
                return []
            }

            return jobRuns
        } catch is EntityNotFoundException {
            print("*** The specified job name, \(jobName), doesn't exist.")
            return []
        } catch {
            print("*** Unexpected error getting job runs:")
            dump(error)
            return []
        }
    }
```

- Per i dettagli sull'API, consulta la GetJobRunsguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

## Utilizzo **`GetTables`** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare `GetTables`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Informazioni di base

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
/// <summary>
/// Get a list of tables for an AWS Glue database.
/// </summary>
/// <param name="dbName">The name of the database.</param>
/// <returns>A list of Table objects.</returns>
public async Task<List<Table>> GetTablesAsync(string dbName)
{
    var request = new GetTablesRequest { DatabaseName = dbName };
    var tables = new List<Table>();

    // Get a paginator for listing the tables.
    var tablePaginator = _amazonGlue.Paginators.GetTables(request);

    await foreach (var response in tablePaginator.Responses)
    {
        tables.AddRange(response.TableList);
    }
```

```
        return tables;
    }
```

- Per i dettagli sull'API, GetTablesconsulta AWS SDK per .NET API Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
       Aws::Client::ClientConfiguration clientConfig;
       // Optional: Set to the AWS Region in which the bucket was created
(overrides config file).
       // clientConfig.region = "us-east-1";

   Aws::Glue::GlueClient client(clientConfig);

       Aws::Glue::Model::GetTablesRequest request;
       request.SetDatabaseName(CRAWLER_DATABASE_NAME);
       std::vector<Aws::Glue::Model::Table> all_tables;
       Aws::String nextToken; // Used for pagination.
       do {
           Aws::Glue::Model::GetTablesOutcome outcome =
client.GetTables(request);

           if (outcome.IsSuccess()) {
               const std::vector<Aws::Glue::Model::Table> &tables =
outcome.GetResult().GetTableList();
               all_tables.insert(all_tables.end(), tables.begin(),
tables.end());
               nextToken = outcome.GetResult().GetNextToken();
           }
           else {
               std::cerr << "Error getting the tables. "
                         << outcome.GetError().GetMessage()
```

```
                        << std::endl;
                deleteAssets(CRAWLER_NAME, CRAWLER_DATABASE_NAME, "", bucketName,
                             clientConfig);
                return false;
            }
        } while (!nextToken.empty());

        std::cout << "The database contains " << all_tables.size()
                  << (all_tables.size() == 1 ?
                      " table." : "tables.") << std::endl;
        std::cout << "Here is a list of the tables in the database.";
        for (size_t index = 0; index < all_tables.size(); ++index) {
            std::cout << "    " << index + 1 << ":  " <<
all_tables[index].GetName()
                      << std::endl;
        }

        if (!all_tables.empty()) {
            int tableIndex = askQuestionForIntRange(
                    "Enter an index to display the database detail ",
                    1, static_cast<int>(all_tables.size()));
            std::cout << all_tables[tableIndex -
1].Jsonize().View().WriteReadable()
                      << std::endl;

            tableName = all_tables[tableIndex - 1].GetName();
        }
```

- Per i dettagli sull'API, GetTablesconsulta AWS SDK per C++ API Reference.

CLI

AWS CLI

Come elencare le definizioni di alcune o tutte le tabelle del database specificato

L'esempio `get-tables` seguente restituisce le informazioni relative alle tabelle del database specificato.

```
aws glue get-tables --database-name 'tempdb'
```

Output:

```
{
    "TableList": [
        {
            "Name": "my-s3-sink",
            "DatabaseName": "tempdb",
            "CreateTime": 1602730539.0,
            "UpdateTime": 1602730539.0,
            "Retention": 0,
            "StorageDescriptor": {
                "Columns": [
                    {
                        "Name": "sensorid",
                        "Type": "int"
                    },
                    {
                        "Name": "currenttemperature",
                        "Type": "int"
                    },
                    {
                        "Name": "status",
                        "Type": "string"
                    }
                ],
                "Location": "s3://janetst-bucket-01/test-s3-output/",
                "Compressed": false,
                "NumberOfBuckets": 0,
                "SerdeInfo": {
                    "SerializationLibrary": "org.openx.data.jsonserde.JsonSerDe"
                },
                "SortColumns": [],
                "StoredAsSubDirectories": false
            },
            "Parameters": {
                "classification": "json"
            },
            "CreatedBy": "arn:aws:iam::007436865787:user/JRSTERN",
            "IsRegisteredWithLakeFormation": false,
            "CatalogId": "007436865787"
        },
        {
            "Name": "s3-source",
            "DatabaseName": "tempdb",
```

```
            "CreateTime": 1602730658.0,
            "UpdateTime": 1602730658.0,
            "Retention": 0,
            "StorageDescriptor": {
                "Columns": [
                    {
                        "Name": "sensorid",
                        "Type": "int"
                    },
                    {
                        "Name": "currenttemperature",
                        "Type": "int"
                    },
                    {
                        "Name": "status",
                        "Type": "string"
                    }
                ],
                "Location": "s3://janetst-bucket-01/",
                "Compressed": false,
                "NumberOfBuckets": 0,
                "SortColumns": [],
                "StoredAsSubDirectories": false
            },
            "Parameters": {
                "classification": "json"
            },
            "CreatedBy": "arn:aws:iam::007436865787:user/JRSTERN",
            "IsRegisteredWithLakeFormation": false,
            "CatalogId": "007436865787"
        },
        {
            "Name": "test-kinesis-input",
            "DatabaseName": "tempdb",
            "CreateTime": 1601507001.0,
            "UpdateTime": 1601507001.0,
            "Retention": 0,
            "StorageDescriptor": {
                "Columns": [
                    {
                        "Name": "sensorid",
                        "Type": "int"
                    },
                    {
```

```
                        "Name": "currenttemperature",
                        "Type": "int"
                    },
                    {
                        "Name": "status",
                        "Type": "string"
                    }
                ],
                "Location": "my-testing-stream",
                "Compressed": false,
                "NumberOfBuckets": 0,
                "SerdeInfo": {
                    "SerializationLibrary": "org.openx.data.jsonserde.JsonSerDe"
                },
                "SortColumns": [],
                "Parameters": {
                    "kinesisUrl": "https://kinesis.us-east-1.amazonaws.com",
                    "streamName": "my-testing-stream",
                    "typeOfData": "kinesis"
                },
                "StoredAsSubDirectories": false
            },
            "Parameters": {
                "classification": "json"
            },
            "CreatedBy": "arn:aws:iam::007436865787:user/JRSTERN",
            "IsRegisteredWithLakeFormation": false,
            "CatalogId": "007436865787"
        }
    ]
}
```

Per ulteriori informazioni, consulta Definizione delle tabelle nel AWS Glue Data Catalog nella AWS Glue Developer Guide.

- Per i dettagli sulle API, consulta GetTables AWS CLICommand Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
   /**
    * Retrieves the names of the tables in the specified Glue database.
    *
    * @param glueClient the Glue client to use for the operation
    * @param dbName     the name of the Glue database to retrieve the table
names from
    * @return the name of the first table retrieved, or an empty string if no
tables were found
    */
   public static String getGlueTables(GlueClient glueClient, String dbName) {
       String myTableName = "";
       try {
           GetTablesRequest tableRequest = GetTablesRequest.builder()
               .databaseName(dbName)
               .build();

           GetTablesResponse response = glueClient.getTables(tableRequest);
           List<Table> tables = response.tableList();
           if (tables.isEmpty()) {
               System.out.println("No tables were returned");
           } else {
               for (Table table : tables) {
                   myTableName = table.name();
                   System.out.println("Table name is: " + myTableName);
               }
           }

       } catch (GlueException e) {
           throw e;
       }
       return myTableName;
```

```
    }
```

- Per i dettagli sull'API, GetTablesconsulta AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
const getTables = (databaseName) => {
  const client = new GlueClient({});

  const command = new GetTablesCommand({
    DatabaseName: databaseName,
  });

  return client.send(command);
};
```

- Per i dettagli sull'API, GetTablesconsulta AWS SDK per JavaScript API Reference.

PHP

SDK per PHP

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
        $databaseName = "doc-example-database-$uniqid";
```

```
        $tables = $glueService->getTables($databaseName);

    public function getTables($databaseName): Result
    {
        return $this->glueClient->getTables([
            'DatabaseName' => $databaseName,
        ]);
    }
```

- Per i dettagli sull'API, GetTablesconsulta AWS SDK per PHP API Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
class GlueWrapper:
    """Encapsulates AWS Glue actions."""

    def __init__(self, glue_client):
        """
        :param glue_client: A Boto3 Glue client.
        """
        self.glue_client = glue_client


    def get_tables(self, db_name):
        """
        Gets a list of tables in a Data Catalog database.

        :param db_name: The name of the database to query.
        :return: The list of tables in the database.
        """
        try:
            response = self.glue_client.get_tables(DatabaseName=db_name)
```

```
        except ClientError as err:
            logger.error(
                "Couldn't get tables %s. Here's why: %s: %s",
                db_name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return response["TableList"]
```

- Per i dettagli sull'API, consulta GetTables AWSSDK for Python (Boto3) API Reference.

## Ruby

### SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
# The `GlueWrapper` class serves as a wrapper around the AWS Glue API, providing
 a simplified interface for common operations.
# It encapsulates the functionality of the AWS SDK for Glue and provides methods
 for interacting with Glue crawlers, databases, tables, jobs, and S3 resources.
# The class initializes with a Glue client and a logger, allowing it to make API
 calls and log any errors or informational messages.
class GlueWrapper
  def initialize(glue_client, logger)
    @glue_client = glue_client
    @logger = logger
  end

  # Retrieves a list of tables in the specified database.
  #
  # @param db_name [String] The name of the database to retrieve tables from.
  # @return [Array<Aws::Glue::Types::Table>]
```

```
def get_tables(db_name)
  response = @glue_client.get_tables(database_name: db_name)
  response.table_list
rescue Aws::Glue::Errors::GlueException => e
  @logger.error("Glue could not get tables #{db_name}: \n#{e.message}")
  raise
end
```

- Per i dettagli sull'API, GetTablesconsulta AWS SDK per Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
        let tables = glue
            .get_tables()
            .database_name(self.database())
            .send()
            .await
            .map_err(GlueMvpError::from_glue_sdk)?;

        let tables = tables.table_list();
```

- Per i dettagli sulle API, consulta il riferimento GetTablesall'API AWS SDK for Rust.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
TRY.
    " iv_database_name = 'my-database'
    oo_result = lo_glu->gettables( iv_databasename = iv_database_name ).
    DATA(lt_tables) = oo_result->get_tablelist( ).
    MESSAGE 'Tables retrieved successfully.' TYPE 'I'.
  CATCH /aws1/cx_gluentitynotfoundex.
    MESSAGE 'Database does not exist.' TYPE 'E'.
  CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).
    DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).
    MESSAGE lv_invalid_error TYPE 'E'.
  CATCH /aws1/cx_gluinternalserviceex INTO DATA(lo_internal_ex).
    DATA(lv_internal_error) = lo_internal_ex->if_message~get_longtext( ).
    MESSAGE lv_internal_error TYPE 'E'.
  CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
    DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
    MESSAGE lv_timeout_error TYPE 'E'.
ENDTRY.
```

- Per i dettagli sulle API, GetTablesconsulta AWS SDK for SAP ABAP API reference.

Swift

SDK per Swift

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import AWSClientRuntime
import AWSGlue

    /// Returns a list of the tables in the specified database.
    ///
    /// - Parameters:
    ///   - glueClient: The AWS Glue client to use.
    ///   - databaseName: The name of the database whose tables are to be
    ///     returned.
    ///
    /// - Returns: An array of `GlueClientTypes.Table` objects, each
    ///   describing one table in the named database. An empty array indicates
    ///   that there are either no tables in the database, or an error
    ///   occurred before any tables could be found.
    func getTablesInDatabase(glueClient: GlueClient, databaseName: String) async
 -> [GlueClientTypes.Table] {
        var tables: [GlueClientTypes.Table] = []
        var nextToken: String?

        repeat {
            do {
                let output = try await glueClient.getTables(
                    input: GetTablesInput(
                        databaseName: databaseName,
                        nextToken: nextToken
                    )
                )

                guard let tableList = output.tableList else {
                    return tables
                }

                tables = tables + tableList
                nextToken = output.nextToken
            } catch {
                return tables
            }
        } while nextToken != nil

        return tables
    }
```

- Per i dettagli sull'API, consulta la GetTablesguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

## Utilizzo **`ListJobs`** con un AWS SDK

Gli esempi di codice seguenti mostrano come utilizzare `ListJobs`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Informazioni di base

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /// <summary>
    /// List AWS Glue jobs using a paginator.
    /// </summary>
    /// <returns>A list of AWS Glue job names.</returns>
    public async Task<List<string>> ListJobsAsync()
    {
        var jobNames = new List<string>();

        var listJobsPaginator = _amazonGlue.Paginators.ListJobs(new
ListJobsRequest { MaxResults = 10 });
        await foreach (var response in listJobsPaginator.Responses)
        {
            jobNames.AddRange(response.JobNames);
        }
```

```
        return jobNames;
    }
```

- Per i dettagli sull'API, ListJobsconsulta AWS SDK per .NET API Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
       Aws::Client::ClientConfiguration clientConfig;
       // Optional: Set to the AWS Region in which the bucket was created
(overrides config file).
       // clientConfig.region = "us-east-1";

   Aws::Glue::GlueClient client(clientConfig);

       Aws::Glue::Model::ListJobsRequest listJobsRequest;

       Aws::String nextToken;
       std::vector<Aws::String> allJobNames;

       do {
           if (!nextToken.empty()) {
               listJobsRequest.SetNextToken(nextToken);
           }
           Aws::Glue::Model::ListJobsOutcome listRunsOutcome = client.ListJobs(
                   listJobsRequest);

           if (listRunsOutcome.IsSuccess()) {
               const std::vector<Aws::String> &jobNames =
listRunsOutcome.GetResult().GetJobNames();
               allJobNames.insert(allJobNames.end(), jobNames.begin(),
jobNames.end());
               nextToken = listRunsOutcome.GetResult().GetNextToken();
```

```
            }
            else {
                std::cerr << "Error listing jobs. "
                          << listRunsOutcome.GetError().GetMessage()
                          << std::endl;
            }
        } while (!nextToken.empty());
```

- Per i dettagli sull'API, ListJobsconsulta AWS SDK per C++ API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
const listJobs = () => {
  const client = new GlueClient({});

  const command = new ListJobsCommand({});

  return client.send(command);
};
```

- Per i dettagli sull'API, ListJobsconsulta AWS SDK per JavaScript API Reference.

PHP

SDK per PHP

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
        $jobs = $glueService->listJobs();
        echo "Current jobs:\n";
        foreach ($jobs['JobNames'] as $jobsName) {
            echo "{$jobsName}\n";
        }

    public function listJobs($maxResults = null, $nextToken = null, $tags = []):
Result
    {
        $arguments = [];
        if ($maxResults) {
            $arguments['MaxResults'] = $maxResults;
        }
        if ($nextToken) {
            $arguments['NextToken'] = $nextToken;
        }
        if (!empty($tags)) {
            $arguments['Tags'] = $tags;
        }
        return $this->glueClient->listJobs($arguments);
    }
```

- Per i dettagli sull'API, ListJobsconsulta AWS SDK per PHP API Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
class GlueWrapper:
    """Encapsulates AWS Glue actions."""

    def __init__(self, glue_client):
        """
        :param glue_client: A Boto3 Glue client.
        """
        self.glue_client = glue_client


    def list_jobs(self):
        """
        Lists the names of job definitions in your account.

        :return: The list of job definition names.
        """
        try:
            response = self.glue_client.list_jobs()
        except ClientError as err:
            logger.error(
                "Couldn't list jobs. Here's why: %s: %s",
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return response["JobNames"]
```

- Per i dettagli sull'API, consulta ListJobs AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
# The `GlueWrapper` class serves as a wrapper around the AWS Glue API, providing
 a simplified interface for common operations.
# It encapsulates the functionality of the AWS SDK for Glue and provides methods
 for interacting with Glue crawlers, databases, tables, jobs, and S3 resources.
# The class initializes with a Glue client and a logger, allowing it to make API
 calls and log any errors or informational messages.
class GlueWrapper
  def initialize(glue_client, logger)
    @glue_client = glue_client
    @logger = logger
  end

  # Retrieves a list of jobs in AWS Glue.
  #
  # @return [Aws::Glue::Types::ListJobsResponse]
  def list_jobs
    @glue_client.list_jobs
  rescue Aws::Glue::Errors::GlueException => e
    @logger.error("Glue could not list jobs: \n#{e.message}")
    raise
  end
```

- Per i dettagli sull'API, ListJobsconsulta AWS SDK per Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    let mut list_jobs = glue.list_jobs().into_paginator().send();
    while let Some(list_jobs_output) = list_jobs.next().await {
        match list_jobs_output {
            Ok(list_jobs) => {
                let names = list_jobs.job_names();
                info!(?names, "Found these jobs")
            }
            Err(err) => return Err(GlueMvpError::from_glue_sdk(err)),
        }
    }
```

- Per i dettagli sulle API, consulta il riferimento ListJobsall'API AWS SDK for Rust.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
  TRY.
      oo_result = lo_glu->listjobs( ).
      DATA(lt_job_names) = oo_result->get_jobnames( ).
      MESSAGE 'Job list retrieved successfully.' TYPE 'I'.
    CATCH /aws1/cx_gluentitynotfoundex.
      MESSAGE 'No jobs found.' TYPE 'I'.
```

```
      CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).
        DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).
        MESSAGE lv_invalid_error TYPE 'E'.
      CATCH /aws1/cx_gluinternalserviceex INTO DATA(lo_internal_ex).
        DATA(lv_internal_error) = lo_internal_ex->if_message~get_longtext( ).
        MESSAGE lv_internal_error TYPE 'E'.
      CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
        DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
        MESSAGE lv_timeout_error TYPE 'E'.
    ENDTRY.
```

- Per i dettagli sulle API, ListJobsconsulta AWS SDK for SAP ABAP API reference.

## Swift

### SDK per Swift

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import AWSClientRuntime
import AWSGlue

    /// Return a list of the AWS Glue jobs listed on the user's account.
    ///
    /// - Parameters:
    ///   - glueClient: The AWS Glue client to use.
    ///   - maxJobs: The maximum number of jobs to return (default: 100).
    ///
    /// - Returns: An array of strings listing the names of all available AWS
    ///   Glue jobs.
    func listJobs(glueClient: GlueClient, maxJobs: Int = 100) async -> [String] {
        var jobList: [String] = []
        var nextToken: String?

        repeat {
            do {
                let output = try await glueClient.listJobs(
```

```
                input: ListJobsInput(
                    maxResults: maxJobs,
                    nextToken: nextToken
                )
            )

            guard let jobs = output.jobNames else {
                return jobList
            }

            jobList = jobList + jobs
            nextToken = output.nextToken
        } catch {
            return jobList
        }
    } while (nextToken != nil)

    return jobList
}
```

- Per i dettagli sull'API, consulta la ListJobsguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

## Utilizzo **`StartCrawler`** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare `StartCrawler`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Informazioni di base

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
/// <summary>
/// Start an AWS Glue crawler.
/// </summary>
/// <param name="crawlerName">The name of the crawler.</param>
/// <returns>A Boolean value indicating the success of the action.</returns>
public async Task<bool> StartCrawlerAsync(string crawlerName)
{
    var crawlerRequest = new StartCrawlerRequest
    {
        Name = crawlerName,
    };

    var response = await _amazonGlue.StartCrawlerAsync(crawlerRequest);

    return response.HttpStatusCode == System.Net.HttpStatusCode.OK;
}
```

- Per i dettagli sull'API, StartCrawlerconsulta AWS SDK per .NET API Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
       Aws::Client::ClientConfiguration clientConfig;
       // Optional: Set to the AWS Region in which the bucket was created
(overrides config file).
       // clientConfig.region = "us-east-1";

   Aws::Glue::GlueClient client(clientConfig);

       Aws::Glue::Model::StartCrawlerRequest request;
       request.SetName(CRAWLER_NAME);

       Aws::Glue::Model::StartCrawlerOutcome outcome =
client.StartCrawler(request);


       if (outcome.IsSuccess() || (Aws::Glue::GlueErrors::CRAWLER_RUNNING ==
                                   outcome.GetError().GetErrorType())) {
           if (!outcome.IsSuccess()) {
               std::cout << "Crawler was already started." << std::endl;
           }
           else {
               std::cout << "Successfully started crawler." << std::endl;
           }

           std::cout << "This may take a while to run." << std::endl;

           Aws::Glue::Model::CrawlerState crawlerState =
Aws::Glue::Model::CrawlerState::NOT_SET;
           int iterations = 0;
           while (Aws::Glue::Model::CrawlerState::READY != crawlerState) {
               std::this_thread::sleep_for(std::chrono::seconds(1));
               ++iterations;
               if ((iterations % 10) == 0) { // Log status every 10 seconds.
                   std::cout << "Crawler status " <<

Aws::Glue::Model::CrawlerStateMapper::GetNameForCrawlerState(
                                       crawlerState)
                             << ". After " << iterations
                             << " seconds elapsed."
                             << std::endl;
               }
               Aws::Glue::Model::GetCrawlerRequest getCrawlerRequest;
               getCrawlerRequest.SetName(CRAWLER_NAME);
```

```
                Aws::Glue::Model::GetCrawlerOutcome getCrawlerOutcome =
client.GetCrawler(
                        getCrawlerRequest);

                if (getCrawlerOutcome.IsSuccess()) {
                    crawlerState =
getCrawlerOutcome.GetResult().GetCrawler().GetState();
                }
                else {
                    std::cerr << "Error getting crawler.  "
                              << getCrawlerOutcome.GetError().GetMessage() <<
std::endl;
                    break;
                }
            }

            if (Aws::Glue::Model::CrawlerState::READY == crawlerState) {
                std::cout << "Crawler finished running after " << iterations
                          << " seconds."
                          << std::endl;
            }
        }
        else {
            std::cerr << "Error starting a crawler.  "
                      << outcome.GetError().GetMessage()
                      << std::endl;

            deleteAssets(CRAWLER_NAME, CRAWLER_DATABASE_NAME, "", bucketName,
                         clientConfig);
            return false;
        }
```

- Per i dettagli sull'API, StartCrawlerconsulta AWS SDK per C++ API Reference.

## CLI

### AWS CLI

#### Come avviare un crawler

L'esempio `start-crawler` seguente avvia un crawler.

```
aws glue start-crawler --name my-crawler
```

Output:

```
None
```

Per ulteriori informazioni, consulta Definizione di crawler nella Guida per gli sviluppatori di AWS Glue.

- Per i dettagli sull'API, consulta StartCrawler AWS CLICommand Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /**
     * Starts a specific AWS Glue crawler.
     *
     * @param glueClient  the AWS Glue client to use for the crawler operation
     * @param crawlerName the name of the crawler to start
     * @throws GlueException if there is an error starting the crawler
     */
    public static void startSpecificCrawler(GlueClient glueClient, String
crawlerName) {
        try {
            StartCrawlerRequest crawlerRequest = StartCrawlerRequest.builder()
                .name(crawlerName)
                .build();

            glueClient.startCrawler(crawlerRequest);
            System.out.println(crawlerName + " was successfully started!");

        } catch (GlueException e) {
            throw e;
        }
```

```
    }
```

- Per i dettagli sull'API, StartCrawlerconsulta AWS SDK for Java 2.x API Reference.

JavaScript

SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
const startCrawler = (name) => {
  const client = new GlueClient({});

  const command = new StartCrawlerCommand({
    Name: name,
  });

  return client.send(command);
};
```

- Per i dettagli sull'API, StartCrawlerconsulta AWS SDK per JavaScript API Reference.

Kotlin

SDK per Kotlin

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
suspend fun startSpecificCrawler(crawlerName: String?) {
```

```
    val request =
        StartCrawlerRequest {
            name = crawlerName
        }

    GlueClient.fromEnvironment { region = "us-west-2" }.use { glueClient ->
        glueClient.startCrawler(request)
        println("$crawlerName was successfully started.")
    }
}
```

- Per i dettagli sull'API, StartCrawlerconsulta AWS SDK for Kotlin API reference.

PHP

SDK per PHP

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
        $crawlerName = "example-crawler-test-" . $uniqid;

        $databaseName = "doc-example-database-$uniqid";

        $glueService->startCrawler($crawlerName);

    public function startCrawler($crawlerName): Result
    {
        return $this->glueClient->startCrawler([
            'Name' => $crawlerName,
        ]);
    }
```

- Per i dettagli sull'API, StartCrawlerconsulta AWS SDK per PHP API Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
class GlueWrapper:
    """Encapsulates AWS Glue actions."""

    def __init__(self, glue_client):
        """
        :param glue_client: A Boto3 Glue client.
        """
        self.glue_client = glue_client


    def start_crawler(self, name):
        """
        Starts a crawler. The crawler crawls its configured target and creates
        metadata that describes the data it finds in the target data source.

        :param name: The name of the crawler to start.
        """
        try:
            self.glue_client.start_crawler(Name=name)
        except ClientError as err:
            logger.error(
                "Couldn't start crawler %s. Here's why: %s: %s",
                name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
```

- Per i dettagli sull'API, consulta StartCrawler AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
# The `GlueWrapper` class serves as a wrapper around the AWS Glue API, providing
 a simplified interface for common operations.
# It encapsulates the functionality of the AWS SDK for Glue and provides methods
 for interacting with Glue crawlers, databases, tables, jobs, and S3 resources.
# The class initializes with a Glue client and a logger, allowing it to make API
 calls and log any errors or informational messages.
class GlueWrapper
  def initialize(glue_client, logger)
    @glue_client = glue_client
    @logger = logger
  end

  # Starts a crawler with the specified name.
  #
  # @param name [String] The name of the crawler to start.
  # @return [void]
  def start_crawler(name)
    @glue_client.start_crawler(name: name)
  rescue Aws::Glue::Errors::ServiceError => e
    @logger.error("Glue could not start crawler #{name}: \n#{e.message}")
    raise
  end
```

- Per i dettagli sull'API, StartCrawlerconsulta AWS SDK per Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
        let start_crawler =
glue.start_crawler().name(self.crawler()).send().await;

        match start_crawler {
            Ok(_) => Ok(()),
            Err(err) => {
                let glue_err: aws_sdk_glue::Error = err.into();
                match glue_err {
                    aws_sdk_glue::Error::CrawlerRunningException(_) => Ok(()),
                    _ => Err(GlueMvpError::GlueSdk(glue_err)),
                }
            }
        }?;
```

- Per i dettagli sulle API, consulta il riferimento StartCrawlerall'API AWS SDK for Rust.

SAP ABAP

SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    TRY.
        " iv_crawler_name = 'my-crawler'
        lo_glu->startcrawler( iv_name = iv_crawler_name ).
```

```
        MESSAGE 'Crawler started successfully.' TYPE 'I'.
      CATCH /aws1/cx_glucrawlerrunningex.
        MESSAGE 'Crawler is already running.' TYPE 'I'.
      CATCH /aws1/cx_gluentitynotfoundex.
        MESSAGE 'Crawler does not exist.' TYPE 'E'.
      CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
        DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
        MESSAGE lv_timeout_error TYPE 'E'.
    ENDTRY.
```

- Per i dettagli sulle API, StartCrawlerconsulta AWS SDK for SAP ABAP API reference.

## Swift

### SDK per Swift

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import AWSClientRuntime
import AWSGlue

    /// Start running an AWS Glue crawler.
    ///
    /// - Parameters:
    ///   - glueClient: The AWS Glue client to use when starting the crawler.
    ///   - name: The name of the crawler to start running.
    ///
    /// - Returns: `true` if the crawler is started successfully, otherwise
 `false`.
    func startCrawler(glueClient: GlueClient, name: String) async -> Bool {
        do {
            _ = try await glueClient.startCrawler(
                input: StartCrawlerInput(name: name)
            )
        } catch {
            print("*** An unexpected error occurred starting the crawler.")
            return false
```

```
        }

        return true
    }
```

- Per i dettagli sull'API, consulta la StartCrawlerguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

## Utilizzo **`StartJobRun`** con un AWS SDK o una CLI

Gli esempi di codice seguenti mostrano come utilizzare `StartJobRun`.

Gli esempi di operazioni sono estratti di codice da programmi più grandi e devono essere eseguiti nel contesto. È possibile visualizzare questa operazione nel contesto nel seguente esempio di codice:

- Informazioni di base

.NET

SDK per .NET

> **Note**
>
> C'è altro su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
/// <summary>
/// Start an AWS Glue job run.
/// </summary>
/// <param name="jobName">The name of the job.</param>
/// <returns>A string representing the job run Id.</returns>
public async Task<string> StartJobRunAsync(
    string jobName,
    string inputDatabase,
    string inputTable,
```

```
        string bucketName)
    {
        var request = new StartJobRunRequest
        {
            JobName = jobName,
            Arguments = new Dictionary<string, string>
            {
                {"--input_database", inputDatabase},
                {"--input_table", inputTable},
                {"--output_bucket_url", $"s3://{bucketName}/"}
            }
        };

        var response = await _amazonGlue.StartJobRunAsync(request);
        return response.JobRunId;
    }
```

- Per i dettagli sull'API, StartJobRunconsulta AWS SDK per .NET API Reference.

C++

SDK per C++

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
        Aws::Client::ClientConfiguration clientConfig;
        // Optional: Set to the AWS Region in which the bucket was created
(overrides config file).
        // clientConfig.region = "us-east-1";

    Aws::Glue::GlueClient client(clientConfig);

        Aws::Glue::Model::StartJobRunRequest request;
        request.SetJobName(JOB_NAME);

        Aws::Map<Aws::String, Aws::String> arguments;
```

```
        arguments["--input_database"] = CRAWLER_DATABASE_NAME;
        arguments["--input_table"] = tableName;
        arguments["--output_bucket_url"] = Aws::String("s3://") + bucketName +
"/";
        request.SetArguments(arguments);

        Aws::Glue::Model::StartJobRunOutcome outcome =
client.StartJobRun(request);

        if (outcome.IsSuccess()) {
            std::cout << "Successfully started the job." << std::endl;

            Aws::String jobRunId = outcome.GetResult().GetJobRunId();

            int iterator = 0;
            bool done = false;
            while (!done) {
                ++iterator;
                std::this_thread::sleep_for(std::chrono::seconds(1));
                Aws::Glue::Model::GetJobRunRequest jobRunRequest;
                jobRunRequest.SetJobName(JOB_NAME);
                jobRunRequest.SetRunId(jobRunId);

                Aws::Glue::Model::GetJobRunOutcome jobRunOutcome =
client.GetJobRun(
                        jobRunRequest);

                if (jobRunOutcome.IsSuccess()) {
                    const Aws::Glue::Model::JobRun &jobRun =
jobRunOutcome.GetResult().GetJobRun();
                    Aws::Glue::Model::JobRunState jobRunState =
jobRun.GetJobRunState();

                    if ((jobRunState == Aws::Glue::Model::JobRunState::STOPPED)
||
                        (jobRunState == Aws::Glue::Model::JobRunState::FAILED) ||
                        (jobRunState == Aws::Glue::Model::JobRunState::TIMEOUT))
{
                        std::cerr << "Error running job. "
                                  << jobRun.GetErrorMessage()
                                  << std::endl;
                        deleteAssets(CRAWLER_NAME, CRAWLER_DATABASE_NAME,
JOB_NAME,
                                     bucketName,
```

```
                                    clientConfig);
                        return false;
                    }
                    else if (jobRunState ==
                             Aws::Glue::Model::JobRunState::SUCCEEDED) {
                        std::cout << "Job run succeeded after  " << iterator <<
                                  " seconds elapsed." << std::endl;
                        done = true;
                    }
                    else if ((iterator % 10) == 0) { // Log status every 10
seconds.
                        std::cout << "Job run status " <<

Aws::Glue::Model::JobRunStateMapper::GetNameForJobRunState(
                                        jobRunState) <<
                                  ". " << iterator <<
                                  " seconds elapsed." << std::endl;
                    }
                }
                else {
                    std::cerr << "Error retrieving job run state. "
                              << jobRunOutcome.GetError().GetMessage()
                              << std::endl;
                    deleteAssets(CRAWLER_NAME, CRAWLER_DATABASE_NAME, JOB_NAME,
                                 bucketName, clientConfig);
                    return false;
                }
            }
        }
        else {
            std::cerr << "Error starting a job. " <<
outcome.GetError().GetMessage()
                      << std::endl;
            deleteAssets(CRAWLER_NAME, CRAWLER_DATABASE_NAME, JOB_NAME,
bucketName,
                         clientConfig);
            return false;
        }
```

- Per i dettagli sull'API, StartJobRunconsulta AWS SDK per C++ API Reference.

CLI

AWS CLI

Come avviare l'esecuzione di un processo

L'esempio `start-job-run` seguente avvia un processo.

```
aws glue start-job-run \
    --job-name my-job
```

Output:

```
{
    "JobRunId":
 "jr_22208b1f44eb5376a60569d4b21dd20fcb8621e1a366b4e7b2494af764b82ded"
}
```

Per ulteriori informazioni, consulta Creazione di processi nella Guida per gli sviluppatori di AWS Glue.

- Per i dettagli sull'API, consulta StartJobRun AWS CLICommand Reference.

Java

SDK per Java 2.x

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    /**
     * Starts a job run in AWS Glue.
     *
     * @param glueClient    the AWS Glue client to use for the job run
     * @param jobName       the name of the Glue job to run
     * @param inputDatabase the name of the input database
     * @param inputTable    the name of the input table
```

```
     * @param outBucket     the URL of the output S3 bucket
     * @throws GlueException if there is an error starting the job run
     */
    public static void startJob(GlueClient glueClient, String jobName, String
inputDatabase, String inputTable,
                                String outBucket) {
        try {
            Map<String, String> myMap = new HashMap<>();
            myMap.put("--input_database", inputDatabase);
            myMap.put("--input_table", inputTable);
            myMap.put("--output_bucket_url", outBucket);

            StartJobRunRequest runRequest = StartJobRunRequest.builder()
                .workerType(WorkerType.G_1_X)
                .numberOfWorkers(10)
                .arguments(myMap)
                .jobName(jobName)
                .build();

            StartJobRunResponse response = glueClient.startJobRun(runRequest);
            System.out.println("The request Id of the job is " +
response.responseMetadata().requestId());

        } catch (GlueException e) {
            throw e;
        }
    }
```

- Per i dettagli sull'API, StartJobRunconsulta AWS SDK for Java 2.x API Reference.

## JavaScript

### SDK per JavaScript (v3)

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
const startJobRun = (jobName, dbName, tableName, bucketName) => {
```

```
  const client = new GlueClient({});

  const command = new StartJobRunCommand({
    JobName: jobName,
    Arguments: {
      "--input_database": dbName,
      "--input_table": tableName,
      "--output_bucket_url": `s3://${bucketName}/`,
    },
  });

  return client.send(command);
};
```

- Per i dettagli sull'API, StartJobRunconsulta AWS SDK per JavaScript API Reference.

PHP

SDK per PHP

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
        $jobName = 'test-job-' . $uniqid;

        $databaseName = "doc-example-database-$uniqid";

        $tables = $glueService->getTables($databaseName);

        $outputBucketUrl = "s3://$bucketName";
        $runId = $glueService->startJobRun($jobName, $databaseName, $tables,
$outputBucketUrl)['JobRunId'];

    public function startJobRun($jobName, $databaseName, $tables,
$outputBucketUrl): Result
    {
        return $this->glueClient->startJobRun([
            'JobName' => $jobName,
```

```
            'Arguments' => [
                'input_database' => $databaseName,
                'input_table' => $tables['TableList'][0]['Name'],
                'output_bucket_url' => $outputBucketUrl,
                '--input_database' => $databaseName,
                '--input_table' => $tables['TableList'][0]['Name'],
                '--output_bucket_url' => $outputBucketUrl,
            ],
        ]);
    }
```

- Per i dettagli sull'API, StartJobRunconsulta AWS SDK per PHP API Reference.

Python

SDK per Python (Boto3)

> **Note**
>
> C'è altro su GitHub. Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
class GlueWrapper:
    """Encapsulates AWS Glue actions."""

    def __init__(self, glue_client):
        """
        :param glue_client: A Boto3 Glue client.
        """
        self.glue_client = glue_client


    def start_job_run(self, name, input_database, input_table,
 output_bucket_name):
        """
        Starts a job run. A job run extracts data from the source, transforms it,
        and loads it to the output bucket.

        :param name: The name of the job definition.
```

```
        :param input_database: The name of the metadata database that contains
tables
                              that describe the source data. This is typically
created
                              by a crawler.
        :param input_table: The name of the table in the metadata database that
                            describes the source data.
        :param output_bucket_name: The S3 bucket where the output is written.
        :return: The ID of the job run.
        """
        try:
            # The custom Arguments that are passed to this function are used by
the
            # Python ETL script to determine the location of input and output
data.
            response = self.glue_client.start_job_run(
                JobName=name,
                Arguments={
                    "--input_database": input_database,
                    "--input_table": input_table,
                    "--output_bucket_url": f"s3://{output_bucket_name}/",
                },
            )
        except ClientError as err:
            logger.error(
                "Couldn't start job run %s. Here's why: %s: %s",
                name,
                err.response["Error"]["Code"],
                err.response["Error"]["Message"],
            )
            raise
        else:
            return response["JobRunId"]
```

- Per i dettagli sull'API, consulta StartJobRun AWSSDK for Python (Boto3) API Reference.

Ruby

SDK per Ruby

> **Note**
>
> C'è di più su. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
# The `GlueWrapper` class serves as a wrapper around the AWS Glue API, providing
 a simplified interface for common operations.
# It encapsulates the functionality of the AWS SDK for Glue and provides methods
 for interacting with Glue crawlers, databases, tables, jobs, and S3 resources.
# The class initializes with a Glue client and a logger, allowing it to make API
 calls and log any errors or informational messages.
class GlueWrapper
  def initialize(glue_client, logger)
    @glue_client = glue_client
    @logger = logger
  end

  # Starts a job run for the specified job.
  #
  # @param name [String] The name of the job to start the run for.
  # @param input_database [String] The name of the input database for the job.
  # @param input_table [String] The name of the input table for the job.
  # @param output_bucket_name [String] The name of the output S3 bucket for the
 job.
  # @return [String] The ID of the started job run.
  def start_job_run(name, input_database, input_table, output_bucket_name)
    response = @glue_client.start_job_run(
      job_name: name,
      arguments: {
        '--input_database': input_database,
        '--input_table': input_table,
        '--output_bucket_url': "s3://#{output_bucket_name}/"
      }
    )
    response.job_run_id
  rescue Aws::Glue::Errors::GlueException => e
```

```
    @logger.error("Glue could not start job run #{name}: \n#{e.message}")
    raise
  end
```

- Per i dettagli sull'API, StartJobRunconsulta AWS SDK per Ruby API Reference.

Rust

SDK per Rust

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
        let job_run_output = glue
            .start_job_run()
            .job_name(self.job())
            .arguments("--input_database", self.database())
            .arguments(
                "--input_table",
                self.tables
                    .first()
                    .ok_or_else(|| GlueMvpError::Unknown("Missing crawler
table".into()))?
                    .name(),
            )
            .arguments("--output_bucket_url", self.bucket())
            .send()
            .await
            .map_err(GlueMvpError::from_glue_sdk)?;

        let job = job_run_output
            .job_run_id()
            .ok_or_else(|| GlueMvpError::Unknown("Missing run id from just
started job".into()))?
            .to_string();
```

- Per i dettagli sulle API, consulta il riferimento StartJobRunall'API AWS SDK for Rust.

## SAP ABAP

### SDK per SAP ABAP

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
    TRY.
        " iv_job_name = 'my-etl-job'
        " iv_input_database = 'my-database'
        " iv_input_table = 'my-table'
        " iv_output_bucket_url = 's3://example-output-bucket/'

        DATA lt_arguments TYPE /aws1/cl_glugenericmap_w=>tt_genericmap.
        lt_arguments = VALUE #(
          ( VALUE /aws1/cl_glugenericmap_w=>ts_genericmap_maprow(
            key = '--input_database'
            value = NEW /aws1/cl_glugenericmap_w( iv_value =
iv_input_database ) ) )
          ( VALUE /aws1/cl_glugenericmap_w=>ts_genericmap_maprow(
            key = '--input_table'
            value = NEW /aws1/cl_glugenericmap_w( iv_value = iv_input_table ) ) )
          ( VALUE /aws1/cl_glugenericmap_w=>ts_genericmap_maprow(
            key = '--output_bucket_url'
            value = NEW /aws1/cl_glugenericmap_w( iv_value =
iv_output_bucket_url ) ) ) ).

        DATA(oo_result) = lo_glu->startjobrun(
          iv_jobname = iv_job_name
          it_arguments = lt_arguments ).
        ov_job_run_id = oo_result->get_jobrunid( ).
        MESSAGE 'Job run started successfully.' TYPE 'I'.
      CATCH /aws1/cx_gluconcurrentrunsex00.
        MESSAGE 'Maximum concurrent runs exceeded.' TYPE 'E'.
      CATCH /aws1/cx_gluentitynotfoundex.
        MESSAGE 'Job does not exist.' TYPE 'E'.
      CATCH /aws1/cx_gluinvalidinputex INTO DATA(lo_invalid_ex).
```

```
        DATA(lv_invalid_error) = lo_invalid_ex->if_message~get_longtext( ).
        MESSAGE lv_invalid_error TYPE 'E'.
      CATCH /aws1/cx_gluinternalserviceex INTO DATA(lo_internal_ex).
        DATA(lv_internal_error) = lo_internal_ex->if_message~get_longtext( ).
        MESSAGE lv_internal_error TYPE 'E'.
      CATCH /aws1/cx_gluoperationtimeoutex INTO DATA(lo_timeout_ex).
        DATA(lv_timeout_error) = lo_timeout_ex->if_message~get_longtext( ).
        MESSAGE lv_timeout_error TYPE 'E'.
      CATCH /aws1/cx_gluresrcnumlmtexcdex INTO DATA(lo_limit_ex).
        DATA(lv_limit_error) = lo_limit_ex->if_message~get_longtext( ).
        MESSAGE lv_limit_error TYPE 'E'.
    ENDTRY.
```

- Per i dettagli sulle API, StartJobRunconsulta AWS SDK for SAP ABAP API reference.

## Swift

### SDK per Swift

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri di più sulla configurazione e l'esecuzione nel Repository di esempi di codice AWS.

```
import AWSClientRuntime
import AWSGlue

    /// Start an AWS Glue job run.
    ///
    /// - Parameters:
    ///   - glueClient: The AWS Glue client to use.
    ///   - jobName: The name of the job to run.
    ///   - databaseName: The name of the AWS Glue database to run the job
 against.
    ///   - tableName: The name of the table in the database to run the job
 against.
    ///   - outputURL: The AWS S3 URI of the bucket location into which to
    ///     write the resulting output.
    ///
```

```
    /// - Returns: `true` if the job run is started successfully, otherwise
`false`.
    func startJobRun(glueClient: GlueClient, name jobName: String, databaseName:
String,
                     tableName: String, outputURL: String) async -> String? {
        do {
            let output = try await glueClient.startJobRun(
                input: StartJobRunInput(
                    arguments: [
                        "--input_database": databaseName,
                        "--input_table": tableName,
                        "--output_bucket_url": outputURL
                    ],
                    jobName: jobName,
                    numberOfWorkers: 10,
                    workerType: .g1x
                )
            )

            guard let id = output.jobRunId else {
                return nil
            }

            return id
        } catch {
            return nil
        }
    }
```

- Per i dettagli sull'API, consulta la StartJobRunguida di riferimento all'API AWS SDK for Swift.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. Utilizzo di questo servizio con un AWS SDK Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

## Scenari per l' AWS Glue utilizzo degli AWS SDK

I seguenti esempi di codice mostrano come implementare scenari comuni AWS Glue con gli AWS SDK. Questi scenari mostrano come eseguire attività specifiche richiamando più funzioni all'interno

AWS Glue o combinandole con altre. Servizi AWS Ogni scenario include un collegamento al codice sorgente completo, dove è possibile trovare le istruzioni su come configurare ed eseguire il codice.

Gli scenari sono relativi a un livello intermedio di esperienza per aiutarti a comprendere le azioni di servizio nel contesto.

Esempi

- Guida introduttiva al Glue Data Catalog

## Guida introduttiva al Glue Data Catalog

L'esempio di codice seguente mostra come:

- Creare un database
- Creare una tabella
- Eseguire la pulizia delle risorse

Bash

AWS CLI con lo script Bash

> **Note**
>
> C'è altro da fare. GitHub Trova l'esempio completo e scopri come eseguire la configurazione e l'esecuzione nel repository dei tutorial sugli esempi di codice per gli sviluppatori.

```
#!/bin/bash

# AWS Glue Data Catalog Tutorial Script
# This script demonstrates how to create and manage AWS Glue Data Catalog
 resources using the AWS CLI
# Cost improvements: Reduced API calls, optimized queries, eliminated redundant
 operations
# Reliability improvements: Enhanced error handling, input validation, resource
 tracking

set -euo pipefail
```

```
# Setup logging
LOG_FILE="glue-tutorial-$(date +%Y%m%d-%H%M%S).log"
exec > >(tee -a "$LOG_FILE") 2>&1

echo "Starting AWS Glue Data Catalog tutorial script at $(date)"
echo "All operations will be logged to $LOG_FILE"

# Generate a unique identifier for resource names
UNIQUE_ID=$(openssl rand -hex 4)
DB_NAME="tutorial-db-${UNIQUE_ID}"
TABLE_NAME="flights-data-${UNIQUE_ID}"
TABLE_INPUT_FILE="table-input-${UNIQUE_ID}.json"

# Track created resources
declare -a CREATED_RESOURCES=()

# Set default region if not provided
AWS_REGION="${AWS_REGION:-us-east-1}"

# Flag to track if database was successfully created
DATABASE_CREATED=false

# Trap to ensure cleanup on exit
trap cleanup_resources EXIT

# Function to check command status
check_status() {
    if [ $? -ne 0 ]; then
        echo "ERROR: $1 failed." >&2
        exit 1
    fi
}

# Function to cleanup resources
cleanup_resources() {
    local exit_code=$?
    echo "Attempting to clean up resources..."

    # Delete resources in reverse order
    for ((i=${#CREATED_RESOURCES[@]}-1; i>=0; i--)); do
        resource=${CREATED_RESOURCES[$i]}
        resource_type=$(echo "$resource" | cut -d':' -f1)
        resource_name=$(echo "$resource" | cut -d':' -f2)
```

```
        echo "Deleting $resource_type: $resource_name"

        case $resource_type in
            "table")
                if [ "$DATABASE_CREATED" = true ]; then
                    aws glue delete-table \
                        --database-name "$DB_NAME" \
                        --name "$resource_name" \
                        --region "$AWS_REGION" \
                        2>/dev/null || echo "Warning: Failed to delete table
 $resource_name"
                fi
                ;;
            "database")
                aws glue delete-database \
                    --name "$resource_name" \
                    --region "$AWS_REGION" \
                    2>/dev/null || echo "Warning: Failed to delete database
 $resource_name"
                ;;
            *)
                echo "Unknown resource type: $resource_type" >&2
                ;;
        esac
    done

    # Clean up temporary files securely
    if [ -f "$TABLE_INPUT_FILE" ]; then
        if command -v shred &> /dev/null; then
            shred -vfz -n 3 "$TABLE_INPUT_FILE" 2>/dev/null || rm -f
 "$TABLE_INPUT_FILE"
        else
            rm -f "$TABLE_INPUT_FILE"
        fi
    fi

    echo "Cleanup completed."
    exit $exit_code
}

# Function to validate prerequisites
validate_prerequisites() {
    # Validate AWS CLI is available
```

```
    if ! command -v aws &> /dev/null; then
        echo "ERROR: AWS CLI is not installed or not in PATH" >&2
        exit 1
    fi

    # Validate AWS CLI version
    local AWS_CLI_VERSION
    AWS_CLI_VERSION=$(aws --version 2>&1 | cut -d' ' -f1 | cut -d'/' -f2 | cut -
d'.' -f1)
    if [ "$AWS_CLI_VERSION" -lt 1 ]; then
        echo "ERROR: AWS CLI is required" >&2
        exit 1
    fi

    # Validate jq is available for JSON validation
    if ! command -v jq &> /dev/null; then
        echo "ERROR: jq is not installed or not in PATH" >&2
        exit 1
    fi

    # Validate AWS credentials and get account identity in single call (cost
 optimization)
    local CALLER_IDENTITY
    CALLER_IDENTITY=$(aws sts get-caller-identity --region "$AWS_REGION" --query
 'Account' --output text 2>/dev/null) || {
        echo "ERROR: Failed to get AWS caller identity. Check credentials and
 permissions." >&2
        exit 1
    }

    if [ -z "$CALLER_IDENTITY" ] || [ "$CALLER_IDENTITY" == "None" ]; then
        echo "ERROR: Unable to determine AWS account identity" >&2
        exit 1
    fi
    echo "Using AWS Account: $CALLER_IDENTITY"
    echo "Using Region: $AWS_REGION"
}

# Function to create database with verification
create_database() {
    echo "Step 1: Creating a database named $DB_NAME"

    if ! aws glue create-database \
```

```
        --database-input "Name=$DB_NAME,Description=Database for AWS Glue
 tutorial" \
        --region "$AWS_REGION" \
        --output json > /dev/null 2>&1; then
        echo "ERROR: Failed to create database $DB_NAME" >&2
        exit 1
    fi

    ACCOUNT_ID=$(aws sts get-caller-identity --query 'Account' --output text)
    aws glue tag-resource \
        --resource-arn "arn:aws:glue:${AWS_REGION}:${ACCOUNT_ID}:database/
${DB_NAME}" \
        --tags-to-add Key=project,Value=doc-smith Key=tutorial,Value=glue-gs \
        --region "$AWS_REGION" \
        2>/dev/null || true

    DATABASE_CREATED=true
    CREATED_RESOURCES+=("database:$DB_NAME")
    echo "Database $DB_NAME created successfully."
}

# Function to prepare table input JSON
prepare_table_input() {
    # Create a temporary JSON file for table input with restricted permissions
    if ! touch "$TABLE_INPUT_FILE" 2>/dev/null; then
        echo "ERROR: Failed to create temporary file $TABLE_INPUT_FILE" >&2
        exit 1
    fi

    if ! chmod 600 "$TABLE_INPUT_FILE" 2>/dev/null; then
        echo "ERROR: Failed to set permissions on $TABLE_INPUT_FILE" >&2
        rm -f "$TABLE_INPUT_FILE"
        exit 1
    fi

    cat > "$TABLE_INPUT_FILE" << 'EOF'
{
  "Name": "TABLE_NAME_PLACEHOLDER",
  "StorageDescriptor": {
    "Columns": [
      {
        "Name": "year",
        "Type": "bigint"
      },
```

```
      {
        "Name": "quarter",
        "Type": "bigint"
      }
    ],
    "Location": "s3://crawler-public-us-west-2/flight/2016/csv",
    "InputFormat": "org.apache.hadoop.mapred.TextInputFormat",
    "OutputFormat": "org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat",
    "Compressed": false,
    "NumberOfBuckets": -1,
    "SerdeInfo": {
      "SerializationLibrary":
 "org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe",
      "Parameters": {
        "field.delim": ",",
        "serialization.format": ","
      }
    }
  },
  "PartitionKeys": [
    {
      "Name": "mon",
      "Type": "string"
    }
  ],
  "TableType": "EXTERNAL_TABLE",
  "Parameters": {
    "EXTERNAL": "TRUE",
    "classification": "csv",
    "columnsOrdered": "true",
    "compressionType": "none",
    "delimiter": ",",
    "skip.header.line.count": "1",
    "typeOfData": "file"
  }
}
EOF

    # Replace placeholder with actual table name
    if ! sed -i "s/TABLE_NAME_PLACEHOLDER/$TABLE_NAME/g" "$TABLE_INPUT_FILE" 2>/
dev/null; then
        echo "ERROR: Failed to substitute table name in JSON file" >&2
        rm -f "$TABLE_INPUT_FILE"
        exit 1
```

```
    fi

    # Validate JSON syntax before using it
    if ! jq empty "$TABLE_INPUT_FILE" 2>/dev/null; then
        echo "ERROR: Invalid JSON in table input file" >&2
        rm -f "$TABLE_INPUT_FILE"
        exit 1
    fi
}

# Function to create table
create_table() {
    echo "Step 2: Creating a table named $TABLE_NAME in database $DB_NAME"

    prepare_table_input

    local TABLE_ARN
    if ! aws glue create-table \
        --database-name "$DB_NAME" \
        --table-input "file://${TABLE_INPUT_FILE}" \
        --region "$AWS_REGION" \
        --output json > /dev/null 2>&1; then
        echo "ERROR: Failed to create table $TABLE_NAME" >&2
        rm -f "$TABLE_INPUT_FILE"
        exit 1
    fi

    aws glue tag-resource \
        --resource-arn "arn:aws:glue:${AWS_REGION}:${ACCOUNT_ID}:table/
${DB_NAME}/${TABLE_NAME}" \
        --tags-to-add Key=project,Value=doc-smith Key=tutorial,Value=glue-gs \
        --region "$AWS_REGION" \
        2>/dev/null || true

    CREATED_RESOURCES+=("table:$TABLE_NAME")
    echo "Table $TABLE_NAME created successfully."
}

# Function to get and display table details
display_table_details() {
    echo "Step 3: Getting details of table $TABLE_NAME"

    if ! aws glue get-table \
        --database-name "$DB_NAME" \
```

```
        --name "$TABLE_NAME" \
        --region "$AWS_REGION" \
        --output json; then
        echo "ERROR: Failed to retrieve table details" >&2
        exit 1
    fi
}

# Function to display summary
display_summary() {
    echo ""
    echo "==========================================="
    echo "RESOURCES CREATED"
    echo "==========================================="
    echo "Database: $DB_NAME"
    echo "Table: $TABLE_NAME"
    echo "==========================================="
}

# Main execution flow
validate_prerequisites
create_database
create_table
display_table_details
display_summary

echo ""
echo "==========================================="
echo "CLEANUP CONFIRMATION"
echo "==========================================="
echo "Starting cleanup process..."

echo "Script completed at $(date)"
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella documentazione di riferimento dei comandi della AWS CLI .
  - CreateDatabase
  - CreateTable
  - DeleteDatabase
  - DeleteTable
  - GetDatabase

- GetTable

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consultaUtilizzo di questo servizio con un AWS SDK. Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

# Sicurezza in AWS Glue

La sicurezza del cloud AWS è la massima priorità. In qualità di AWS cliente, puoi beneficiare di un data center e di un'architettura di rete progettati per soddisfare i requisiti delle organizzazioni più sensibili alla sicurezza.

La sicurezza è una responsabilità condivisa tra AWS te e te. Il modello di responsabilità condivisa descrive questo come sicurezza del cloud e sicurezza nel cloud:

- Sicurezza del cloud: AWS è responsabile della protezione dell'infrastruttura che gestisce AWS i servizi nel AWS cloud. AWS ti fornisce anche servizi che puoi utilizzare in modo sicuro. I revisori di terze parti testano e verificano regolarmente l'efficacia della sicurezza come parte dei programmi di conformitàAWS. Per ulteriori informazioni sui programmi di conformità applicabili AWS Glue, consulta AWS Services in Scope by Compliance Program.
- Sicurezza nel cloud: la tua responsabilità è determinata dal AWS servizio che utilizzi. L'utente è anche responsabile di altri fattori, tra cui la riservatezza dei dati, i requisiti della propria azienda e le leggi e normative vigenti.

Questa documentazione serve a facilitare la comprensione dell'applicazione del modello di responsabilità condivisa quando si utilizza l'AWS Glue. I seguenti argomenti illustrano come configurare l'AWS Glue per soddisfare gli obiettivi di sicurezza e conformità. Imparerai anche a utilizzare altri AWS servizi che ti aiutano a monitorare e proteggere AWS Glue le tue risorse.

Argomenti


- Protezione dei dati in AWS Glue
- Gestione delle identità e degli accessi per AWS Glue
- AWS Lake Formation modelli di controllo degli accessi
- Utilizzo di Amazon S3 Access Grants con AWS Glue
- Propagazione affidabile delle identità con ETL AWS Glue
- Registrazione e monitoraggio AWS Glue
- Convalida della conformità per AWS Glue
- Resilienza in AWS Glue
- Sicurezza dell'infrastruttura nell' AWS Glue

# Protezione dei dati in AWS Glue

AWS Glue offre diverse caratteristiche che permettono di proteggere i dati.

Argomenti


- Crittografia dei dati a riposo
- Crittografia dei dati in transito
- Conformità a FIPS
- Gestione delle chiavi
- AWS Gluedipendenza da altri servizi AWS
- Endpoint di sviluppo


## Crittografia dei dati a riposo

AWS Glue supporta la crittografia dei dati a riposo per Creazione di processi ETL visivi eUtilizzo di endpoint per lo sviluppo di script. È possibile configurare i processi di estrazione, trasformazione e caricamento (ETL) e gli endpoint di sviluppo per usare AWS Key Management Service (AWS KMS) durante la scrittura dei dati inattivi criptati. Puoi anche crittografare i metadati archiviati nelle chiavi di AWS Glue Data Catalog utilizzo con cui gestisci. AWS KMSInoltre, è possibile utilizzare AWS KMS le chiavi per crittografare i segnalibri dei lavori e i log generati dai crawler e dai job ETL.

Oltre ai dati scritti su Amazon Simple Storage Service (Amazon S3) e Amazon Logs, puoi crittografare gli oggetti di metadati AWS Glue Data Catalog presenti nel tuo computer oltre ai dati scritti in Amazon Simple Storage Service (Amazon S3) e CloudWatch Amazon Logs per job, crawler ed endpoint di sviluppo. Quando si creano processi, crawler ed endpoint di sviluppo in AWS Glue, è possibile fornire impostazioni di crittografia collegando una configurazione di protezione. Le configurazioni di sicurezza contengono chiavi di crittografia lato server gestite da Amazon S3 (SSE-S3) o chiavi master del cliente () archiviate in (SSE-KMS). CMKs AWS KMS È possibile creare configurazioni di sicurezza utilizzando la console AWS Glue.

Puoi attivare la crittografia dell'intero catalogo dati nel tuo account. Puoi CMKs farlo specificando stored in. AWS KMS

> **Important**
>
> AWS Glue supporta solo chiavi gestite del cliente simmetriche. Per ulteriori informazioni, consulta Customer Managed Keys (CMKs) nella Guida per gli AWS Key Management Service sviluppatori.

Con la crittografia attivata, quando aggiungi oggetti del catalogo dati, esegui crawler o processi o avvii endpoint di sviluppo, vengono usate chiavi SSE-S3 o SSE-KMS per scrivere i dati a riposo. Puoi anche configurare AWS Glue per accedere ai datastore Java Database Connectivity (JDBC) solo attraverso un protocollo Transport Layer Security (TLS) attendibile.

In AWS Glue è possibile controllare le impostazioni di crittografia nelle posizioni seguenti:

- Le impostazioni del catalogo dati.
- Configurazioni della sicurezza create.
- Impostazione di crittografia lato server (SSE-S3 o SSE-KMS) passata come parametro al processo ETL (estrazione, trasformazione e caricamento) di AWS Glue.

Per ulteriori informazioni su come configurare le crittografia, consulta Configurazione della crittografia in AWS Glue.

Argomenti


- Crittografia del catalogo dati
- Crittografia delle password di connessione
- Crittografia dei dati scritti da AWS Glue


## Crittografia del catalogo dati

AWS Glue Data Catalog la crittografia offre una maggiore sicurezza per i dati sensibili. AWS Glue si integra con AWS Key Management Service (AWS KMS) per crittografare i metadati archiviati nel Data Catalog. È possibile abilitare o disabilitare le impostazioni di crittografia per le risorse nel Data Catalog utilizzando la AWS Glue console o il. AWS CLI

Quando abiliti la crittografia per il tuo Catalogo dati, tutti i nuovi oggetti che crei verranno crittografati. Quando disabiliti la crittografia, i nuovi oggetti che crei non verranno crittografati, ma gli oggetti crittografati esistenti rimarranno crittografati.

È possibile crittografare l'intero Data Catalog utilizzando chiavi di crittografia AWS gestite o chiavi di crittografia gestite dal cliente. Per ulteriori informazioni sui tipi e sugli stati delle chiavi, consulta AWS Key Management Service i concetti nella Guida per gli AWS Key Management Service sviluppatori.

> **Note**
>
> Quando si utilizza il Catalogo dati crittografato con un crawler, è necessario mantenere le impostazioni di crittografia. La rimozione delle impostazioni di crittografia dopo che un crawler ha elaborato un catalogo crittografato genera errori. Se devi rimuovere le impostazioni di crittografia, crea un nuovo crawler anziché modificare quello esistente.

AWS chiavi gestite

AWS le chiavi gestite sono chiavi KMS presenti nel tuo account che vengono create, gestite e utilizzate per tuo conto da un AWS servizio integrato con. AWS KMS Puoi visualizzare le chiavi AWS gestite nel tuo account, visualizzare le relative politiche chiave e verificarne l'utilizzo nei AWS CloudTrail log. Tuttavia, non puoi gestire queste chiavi o modificarne le autorizzazioni.

Encryption at rest si integra automaticamente con AWS KMS la gestione delle chiavi AWS gestite AWS Glue utilizzate per crittografare i metadati. Se non esiste una chiave AWS gestita quando abiliti la crittografia dei metadati, crea AWS KMS automaticamente una nuova chiave per te.

Per ulteriori informazioni, consulta le AWS chiavi gestite da .

Chiavi gestite dal cliente

Le chiavi gestite dal cliente sono chiavi KMS Account AWS che crei, possiedi e gestisci. Hai il pieno controllo su queste chiavi KMS. Puoi:

- Stabilire e mantenere le policy delle chiavi, le policy IAM e le concessioni
- Abilitarle e disabilitarle
- Ruotare il loro materiale crittografico
- Aggiunta di tag
- Creazione di alias che si riferiscono ad essi
- Pianificazione dell'eliminazione delle chiavi

Per ulteriori informazioni relative alla gestione delle autorizzazioni di una chiave gestita dal cliente, consulta Customer managed keys.

> **Important**
>
> AWS Glue supporta solo chiavi simmetriche gestite dal cliente. L'elenco di chiavi KMS mostra solo le chiavi simmetriche. Tuttavia, selezionando Scegli un ARN chiave KMS, la console consente di inserire un ARN per qualsiasi tipo di chiave. Assicurati di inserire solo chiavi ARNs simmetriche.
> Per creare una chiave simmetrica gestita del cliente, segui la procedura per la creazione di chiavi simmetriche gestite dal cliente nella Guida per gli sviluppatori di AWS Key Management Service .

Quando abiliti la crittografia del Catalogo dati inattivo, i seguenti tipi di risorse vengono crittografati utilizzando chiavi KMS:

- Database
- Tabelle
- Partizioni
- Versioni di tabella
- Statistiche delle colonne
- Funzioni definite dall'utente
- Viste di Catalogo dati

AWS Glue contesto di crittografia

Un contesto di crittografia è un set opzionale di coppie chiave-valore che contiene ulteriori informazioni contestuali sui dati. AWS KMS utilizza il contesto di crittografia come dati autenticati aggiuntivi per supportare crittografia autenticata. Quando includi un contesto di crittografia in una richiesta di crittografia dei dati, AWS KMS associa il contesto di crittografia ai dati crittografati. Per decrittografare i dati, includi lo stesso contesto di crittografia nella richiesta. AWS Glue utilizza lo stesso contesto di crittografia in tutte le operazioni AWS KMS crittografiche, in cui la chiave è `glue_catalog_id` e il valore è il. `catalogId`

```
"encryptionContext": {
```

```
    "glue_catalog_id": "111122223333"
}
```

Quando si utilizza una chiave AWS gestita o una chiave simmetrica gestita dal cliente per crittografare il catalogo dati, è possibile utilizzare il contesto di crittografia anche nei record e nei registri di controllo per identificare come viene utilizzata la chiave. Il contesto di crittografia viene visualizzato anche nei log generati da or logs. AWS CloudTrail Amazon CloudWatch

Abilitazione della crittografia

È possibile abilitare la crittografia per AWS Glue Data Catalog gli oggetti nelle impostazioni del Data Catalog nella AWS Glue console o utilizzando il. AWS CLI

Console

Per abilitare la crittografia usando la console

1. Accedi a Console di gestione AWS e apri la AWS Glue console all'indirizzo https://console.aws.amazon.com/glue/.
2. Nel pannello di navigazione, seleziona Catalogo dati.
3. Nella pagina delle impostazioni del Data Catalog, seleziona la casella di controllo Crittografia dei metadati e scegli una AWS KMS chiave.

   Quando abiliti la crittografia, se non specifichi una chiave gestita dal cliente, le impostazioni di crittografia utilizzano una chiave KMS AWS gestita.
4. (Facoltativo) Quando utilizzi una chiave gestita dal cliente per crittografare il tuo Catalogo dati, il Catalogo dati offre la possibilità di registrare un ruolo IAM per crittografare e decrittografare le risorse. Devi concedere al tuo ruolo IAM le autorizzazioni che AWS Glue puoi assumere per tuo conto. Ciò include AWS KMS le autorizzazioni per crittografare e decrittografare i dati.

   Quando crei una nuova risorsa nel Data Catalog, AWS Glue assume il ruolo IAM fornito per crittografare i dati. Allo stesso modo, quando un consumatore accede alla risorsa, AWS Glue assume il ruolo IAM di decrittografare i dati. Se registri un ruolo IAM con le autorizzazioni richieste, il principale chiamante non necessita più delle autorizzazioni per accedere alla chiave e decrittografare i dati.

> **Important**
>
> Puoi delegare le operazioni KMS a un ruolo IAM solo quando utilizzi una chiave gestita dal cliente per crittografare le risorse del Catalogo dati. Al momento, la funzionalità di delega dei ruoli KMS non supporta l'utilizzo di AWS chiavi gestite per la crittografia delle risorse del Catalogo dati.

> **Warning**
>
> Quando abiliti un ruolo IAM per delegare le operazioni KMS, non puoi più accedere alle risorse del Data Catalog che in precedenza erano crittografate con una chiave gestita. AWS

a. Per abilitare un ruolo IAM che AWS Glue può pretendere di crittografare e decrittografare i dati per tuo conto, seleziona l'opzione Delega le operazioni KMS a un ruolo IAM.

b. Successivamente, Scegli un ruolo IAM.

   Per creare un ruolo IAM, consulta l'argomento relativo alla creazione di ruoli IAM per AWS Glue.

   Il ruolo IAM che AWS Glue presuppone l'accesso al Data Catalog deve disporre delle autorizzazioni per crittografare e decrittografare i metadati nel Data Catalog. Puoi creare un ruolo IAM e collegare le seguenti policy in linea:

   - Aggiungi la seguente policy per includere le AWS KMS autorizzazioni per crittografare e decrittografare il Data Catalog.

     JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt",
        "kms:Encrypt",
```

```
        "kms:GenerateDataKey"
      ],
      "Resource": "arn:aws:kms:us-east-1:111122223333:key/<key-id>"
    }
  ]
}
```

- Quindi, aggiungi la seguente policy di fiducia al ruolo affinché il AWS Glue servizio assuma il ruolo IAM.

  JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "glue.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

- Quindi, aggiungi l'autorizzazione `iam:PassRole` al ruolo IAM.

  JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": [
        "arn:aws:iam::111122223333:role/<encryption-role-name>"
      ]
    }
  ]
```

```
}
```

Quando abiliti la crittografia, se non hai specificato un ruolo IAM AWS Glue da assumere, il principale che accede al Data Catalog deve disporre delle autorizzazioni per eseguire le seguenti operazioni API:

- `kms:Decrypt`
- `kms:Encrypt`
- `kms:GenerateDataKey`

AWS CLI

Per abilitare la crittografia utilizzando l'SDK o AWS CLI

- Usa l'operazione API `PutDataCatalogEncryptionSettings`. Se non viene specificata alcuna chiave, AWS Glue utilizza la chiave di crittografia AWS gestita per l'account cliente per crittografare il Data Catalog.

```
aws glue put-data-catalog-encryption-settings \
  --data-catalog-encryption-settings '{
      "EncryptionAtRest": {
      "CatalogEncryptionMode": "SSE-KMS-WITH-SERVICE-ROLE",
      "SseAwsKmsKeyId": "arn:aws:kms:<region>:<account-id>:key/<key-id>",
      "CatalogEncryptionServiceRole":"arn:aws:iam::<account-
id>:role/<encryption-role-name>"
    }

  }'
```

Quando abiliti la crittografia, tutti i nuovi oggetti che crei tra gli oggetti del Catalogo dati verranno crittografati. Se cancelli questa impostazione, gli oggetti creati nel Catalogo dati non vengono più crittografati. Puoi continuare ad accedere agli oggetti crittografati esistenti nel Catalogo dati con le autorizzazioni KMS richieste.

> **Important**
>
> La AWS KMS chiave deve rimanere disponibile nell'archivio delle AWS KMS chiavi per tutti gli oggetti crittografati con essa nel Data Catalog. Se rimuovi la chiave, non sarà più possibile decrittografare gli oggetti. In alcuni scenari ciò potrebbe essere necessario per impedire l'accesso ai metadati del catalogo dati.

### Monitoraggio delle chiavi KMS per AWS Glue

Quando utilizzi le chiavi KMS con le risorse del tuo Data Catalog, puoi utilizzare AWS CloudTrail or Amazon CloudWatch Logs per tenere traccia delle richieste inviate AWS Glue a. AWS KMS AWS CloudTrail monitora e registra le operazioni KMS relative alle AWS Glue chiamate di accesso ai dati crittografati dalle tue chiavi KMS.

Gli esempi seguenti sono AWS CloudTrail gli eventi relativi alle operazioni and. `Decrypt` `GenerateDataKey`

Decrypt


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAXPHTESTANDEXAMPLE:Sampleuser01",
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/Sampleuser01",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAXPHTESTANDEXAMPLE",
                "arn": "arn:aws:iam::111122223333:role/Admin",
                "accountId": "111122223333",
                "userName": "Admin"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2024-01-10T14:33:56Z",
                "mfaAuthenticated": "false"
```

```
            }
        },
        "invokedBy": "glue.amazonaws.com"
    },
    "eventTime": "2024-01-10T15:18:11Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "eu-west-2",
    "sourceIPAddress": "glue.amazonaws.com",
    "userAgent": "glue.amazonaws.com",
    "requestParameters": {
        "encryptionContext": {
            "glue_catalog_id": "111122223333"
        },
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT"
    },
    "responseElements": null,
    "requestID": "43b019aa-34b8-4798-9b98-ee968b2d63df",
    "eventID": "d7614763-d3fe-4f84-a1e1-3ca4d2a5bbd5",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:<region>:111122223333:key/<key-id>"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management",
    "sessionCredentialFromConsole": "true"
}
```

## GenerateDataKey

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId":
 "AROAXPHTESTANDEXAMPLE:V_00_GLUE_KMS_GENERATE_DATA_KEY_111122223333",
```

```
        "arn": "arn:aws:sts::111122223333:assumed-role/Admin/
V_00_GLUE_KMS_GENERATE_DATA_KEY_111122223333",
        "accountId": "111122223333",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAXPHTESTANDEXAMPLE",
                "arn": "arn:aws:iam::111122223333:role/Admin",
                "accountId": "AKIAIOSFODNN7EXAMPLE",
                "userName": "Admin"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2024-01-05T21:15:47Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "glue.amazonaws.com"
    },
    "eventTime": "2024-01-05T21:15:47Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKey",
    "awsRegion": "eu-west-2",
    "sourceIPAddress": "glue.amazonaws.com",
    "userAgent": "glue.amazonaws.com",
    "requestParameters": {
        "keyId": "arn:aws:kms:eu-west-2:AKIAIOSFODNN7EXAMPLE:key/
AKIAIOSFODNN7EXAMPLE",
        "encryptionContext": {
            "glue_catalog_id": "111122223333"
        },
        "keySpec": "AES_256"
    },
    "responseElements": null,
    "requestID": "64d1783a-4b62-44ba-b0ab-388b50188070",
    "eventID": "1c73689b-2ef2-443b-aed7-8c126585ca5e",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:eu-west-2:111122223333:key/AKIAIOSFODNN7EXAMPLE"
        }
```

```
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

## Crittografia delle password di connessione

È possibile recuperare le password di connessione AWS Glue Data Catalog utilizzando le operazioni `GetConnection` e `GetConnections` API. Queste password sono memorizzate nella connessione del catalogo dati e vengono utilizzate quando AWS Glue si connette a un datastore JDBC (Java Database Connectivity). Quando la connessione è stata creata o aggiornata, un'opzione nelle impostazioni del Data Catalog ha determinato se la password è stata crittografata e, in caso affermativo, quale AWS Key Management Service (AWS KMS) chiave è stata specificata.

Sulla console AWS Glue, puoi attivare questa opzione nella pagina Data catalog settings (Impostazioni catalogo dati).

Crittografare le password di connessione

1. Accedi a Console di gestione AWS e apri la AWS Glue console all'indirizzo https://console.aws.amazon.com/glue/.
2. Scegliere Settings (Impostazioni) nel riquadro di navigazione.
3. Nella pagina Data catalog settings (Impostazione catalogo dati), seleziona la casella di controllo Encrypt connection passwords (Crittografa password di connessione) e scegli una chiave AWS KMS .

> **Important**
>
> AWS Gluesupporta solo le chiavi master simmetriche del cliente ()CMKs. L'elenco di chiavi AWS KMS  mostra solo le chiavi simmetriche. Tuttavia, se si seleziona Scegli un

ARN AWS KMS per una chiave, la console consente di inserire un ARN per qualsiasi tipo di chiave. Assicurati di inserire solo chiavi ARNs simmetriche.

Per ulteriori informazioni, consulta Impostazioni del Catalogo dati.

## Crittografia dei dati scritti da AWS Glue

Una configurazione della sicurezza è un set di proprietà di sicurezza che AWS Glue può usare. Puoi usare una configurazione della sicurezza per crittografare i dati inattivi. Gli scenari seguenti mostrano alcuni modi in cui è possibile usare una configurazione della sicurezza.

- Collega una configurazione di sicurezza a un AWS Glue crawler per scrivere Amazon CloudWatch Logs crittografati. Per ulteriori informazioni sull'aggiunta di configurazioni di sicurezza ai crawler, consulta the section called "Configurazione delle impostazioni di sicurezza".
- Collega una configurazione di sicurezza a un processo di estrazione, trasformazione e caricamento (ETL) per scrivere obiettivi Amazon Simple Storage Service (Amazon S3) crittografati e log crittografati. CloudWatch
- Collegare una configurazione della sicurezza a un processo ETL per scrivere i segnalibri dei processi come dati Amazon S3 crittografati.
- Collegare una configurazione della sicurezza a un endpoint di sviluppo per scrivere destinazioni Amazon S3 crittografate.

**Important**

Attualmente, una configurazione della sicurezza sostituisce qualsiasi impostazione di crittografia lato server (SSE-S3) passata come parametro di un processo ETL. Pertanto, se a un processo sono associati sia una configurazione della sicurezza che un parametro SSE-S3, il parametro SSE-S3 viene ignorato.

Per ulteriori informazioni sulle configurazioni della sicurezza, consulta Gestire le configurazioni di sicurezza nella console AWS Glue.

Argomenti

- Impostazione di AWS Glue per l'uso di configurazioni della sicurezza

- Creazione di un percorso AWS KMS per i job e i crawler VPC
- Gestire le configurazioni di sicurezza nella console AWS Glue

Impostazione di AWS Glue per l'uso di configurazioni della sicurezza

Segui queste fasi per impostare l'ambiente AWS Glue per l'uso di configurazioni della sicurezza.

1. Crea o aggiorna le tue chiavi AWS Key Management Service (AWS KMS) per concedere le AWS KMS autorizzazioni ai ruoli IAM che vengono passate ai AWS Glue crawler e ai job per crittografare i log. CloudWatch Per ulteriori informazioni, consulta Encrypt Log Data in CloudWatch Logs Using AWS KMS nella Amazon CloudWatch Logs User Guide.

   Nell'esempio seguente *"role1""role2"*, e *"role3"* sono i ruoli IAM che vengono passati ai crawler e ai job.

   ```
   {
          "Effect": "Allow",
          "Principal": { "Service": "logs.region.amazonaws.com",
          "AWS": [
                   "role1",
                   "role2",
                   "role3"
                ] },
                      "Action": [
                             "kms:Encrypt*",
                             "kms:Decrypt*",
                             "kms:ReEncrypt*",
                             "kms:GenerateDataKey*",
                             "kms:Describe*"
                      ],
                      "Resource": "*"
   }
   ```

   L'`Service`istruzione, mostrata come`"Service": "logs.region.amazonaws.com"`, è obbligatoria se si utilizza la chiave per CloudWatch crittografare i log.

2. Assicuratevi che la AWS KMS chiave sia presente `ENABLED` prima di essere utilizzata.

> **Note**
>
> Se utilizzi Iceberg come framework di data lake, le tabelle Iceberg dispongono di meccanismi propri per abilitare la crittografia lato server. È necessario abilitare queste configurazioni in aggiunta alle configurazioni AWS Glue di sicurezza. Per abilitare la crittografia lato server sulle tabelle Iceberg, consulta le indicazioni contenute nella documentazione di Iceberg.

Creazione di un percorso AWS KMS per i job e i crawler VPC

Puoi connetterti direttamente a AWS KMS attraverso un endpoint privato nel cloud privato virtuale (VPC, Virtual Private Cloud) invece che tramite Internet. Quando utilizzi un endpoint VPC, la comunicazione tra il tuo VPC e il tuo VPC AWS KMS viene condotta interamente all'interno della rete. AWS

Puoi creare un endpoint AWS KMS VPC all'interno di un VPC. Senza questa fase, i processi o i crawler potrebbero avere esito negativo, con un errore `kms timeout` nei processi o un'eccezione `internal service exception` nei crawler. Per istruzioni dettagliate, consulta Connessione a AWS KMS un endpoint VPC nella Guida per gli AWS Key Management Service sviluppatori.

Mentre segui le istruzioni, nella console VPC esegui queste operazioni:

- Selezionare Abilita nome DNS privato.
- Scegli il gruppo di sicurezza (con regola autoreferenziale) da usare per il processo o il crawler che accede a JDBC (Java Database Connectivity). Per ulteriori informazioni sulle connessioni AWS Glue, consulta Connessione ai dati.

Quando aggiungi una configurazione di sicurezza a un crawler o a un job che accede agli archivi dati JDBC, AWS Glue devi disporre di un percorso verso l'endpoint. AWS KMS È possibile fornire il percorso con un gateway NAT (Network Address Translation) o con un endpoint AWS KMS VPC. Per creare un gateway NAT, consulta Gateway NAT nella Guida per l'utente di Amazon VPC.

Gestire le configurazioni di sicurezza nella console AWS Glue

> **Warning**
>
> AWS Glue le configurazioni di sicurezza non sono attualmente supportate nei lavori Ray.

Una configurazione di sicurezza in AWS Glue contiene le proprietà necessarie per la scrittura di dati crittografati. Puoi creare configurazioni di sicurezza nella console AWS Glue per specificare le proprietà di crittografia usate da crawler, processi ed endpoint di sviluppo.

Per visualizzare un elenco delle configurazioni di sicurezza che hai creato, apri la console AWS Glue all'indirizzo https://console.aws.amazon.com/glue/ e scegli Security configurations (Configurazioni di sicurezza) nel riquadro di navigazione.

L'elenco Security configurations (Configurazioni di sicurezza) visualizza le proprietà seguenti su ogni configurazione:

Name

Nome univoco che hai assegnato quando hai creato la configurazione. Il nome può contenere un massimo di 255 caratteri, tra cui lettere (A-Z), numeri (0-9), trattini (-) o caratteri di sottolineatura (_).

Attivazione della crittografia in Amazon S3

Se attivata, la modalità di crittografia Amazon Simple Storage Service (Amazon S3), ad esempio `SSE-KMS` o `SSE-S3`, è abilitata per l'archiviazione di metadati nel catalogo dati.

Abilita la crittografia dei CloudWatch log di Amazon

Se attivata, la modalità di crittografia di Amazon S3, ad esempio, `SSE-KMS` è abilitata durante la scrittura di log su Amazon. CloudWatch

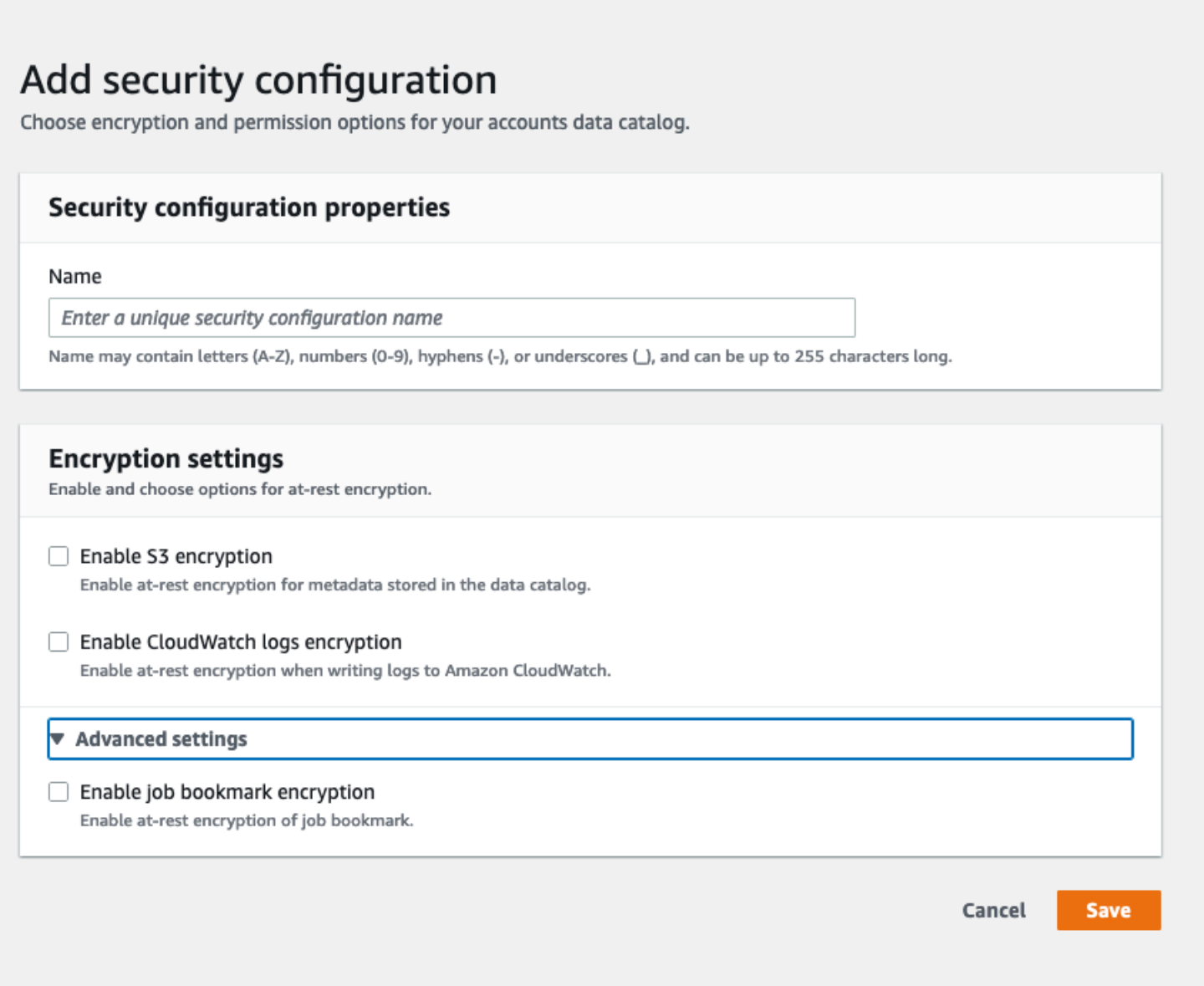
Impostazioni avanzate: abilitazione della crittografia dei segnalibri del processo

Se attivata, la modalità di crittografia di Amazon S3, ad esempio `CSE-KMS`, è abilitata quando i processi vengono aggiunti ai segnalibri.

Puoi aggiungere o eliminare configurazioni nella sezione Security configurations (Configurazioni di sicurezza) nella console. Per visualizzare altri dettagli relativi a una configurazione, scegli il nome della configurazione nell'elenco. I dettagli includono le informazioni che hai definito quando hai creato la configurazione.

Aggiunta di una configurazione di sicurezza

Per aggiungere una configurazione di sicurezza usando la console AWS Glue, nella pagina Security configurations (Configurazioni di sicurezza) scegli Add security configuration (Aggiungi configurazione di sicurezza.




Proprietà di configurazione di sicurezza

Inserisci un nome univoco per la configurazione di sicurezza. Il nome può contenere un massimo di 255 caratteri, tra cui lettere (A-Z), numeri (0-9), trattini (-) o caratteri di sottolineatura (_).

Impostazioni di crittografia

Puoi abilitare la crittografia a riposo per i metadati archiviati nel Data Catalog in Amazon S3 e i log in Amazon. CloudWatch Per configurare la crittografia di dati e metadati con le chiavi AWS Key Management Service (AWS KMS) sulla AWS Glue console, aggiungi una policy all'utente della console. Questa policy deve specificare le risorse consentite come Amazon Resource Names (ARNs) chiave utilizzate per crittografare gli archivi di dati Amazon S3, come nell'esempio seguente.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "kms:GenerateDataKey",
      "kms:Decrypt",
      "kms:Encrypt"
    ],
    "Resource": "arn:aws:kms:us-east-1:111122223333:key/key-id"
  }
}
```

> **Important**
>
> Quando una configurazione di sicurezza è collegata a un crawler o a un job, il ruolo IAM passato deve disporre di autorizzazioni. AWS KMS Per ulteriori informazioni, consulta Crittografia dei dati scritti da AWS Glue.

Quando definisci una configurazione, puoi specificare i valori per le proprietà seguenti:

Abilitazione della crittografia in S3

Quando scrivi dati Amazon S3, utilizzi la crittografia lato server con chiavi gestite di Amazon S3 (SSE-S3) o la crittografia lato server con chiavi gestite (SSE-KMS). AWS KMS Questo campo è facoltativo. Per consentire l'accesso ad Amazon S3, scegli una AWS KMS chiave o scegli Inserisci un codice ARN e fornisci l'ARN per la chiave. Immetti l'ARN usando questo formato: `arn:aws:kms:region:account-id:key/key-id`. Puoi specificare l'ARN anche sotto forma di alias di chiavi, ad esempio `arn:aws:kms:region:account-id:alias/alias-name`.

Se abiliti l'interfaccia utente Spark per il processo, il file di log dell'interfaccia utente di Spark caricato su Amazon S3 verrà applicato con la stessa crittografia.

> **Important**
>
> AWS Gluesupporta solo chiavi master simmetriche per i clienti (). CMKs L'elenco di chiavi AWS KMS  mostra solo le chiavi simmetriche. Tuttavia, se si seleziona Scegli un ARN AWS KMS per una chiave, la console consente di inserire un ARN per qualsiasi tipo di chiave. Assicurati di inserire solo chiavi ARNs simmetriche.

Abilita la crittografia CloudWatch dei log

La crittografia lato server (SSE-KMS) viene utilizzata per crittografare i log. CloudWatch Questo campo è facoltativo. Per attivarlo, scegli una AWS KMS chiave o scegli Inserisci una chiave ARN e fornisci l'ARN per la chiave. Immetti l'ARN usando questo formato: `arn:aws:kms:`*`region`*`:`*`account-id`*`:key/`*`key-id`*. Puoi specificare l'ARN anche sotto forma di alias di chiavi, ad esempio `arn:aws:kms:`*`region`*`:`*`account-id`*`:alias/`*`alias-name`*.

Impostazioni avanzate: crittografia dei segnalibri del processo

Crittografia lato client (CSE-KMS) usata per crittografare segnalibri dei processi. Questo campo è facoltativo. I dati dei segnalibri vengono crittografati prima di essere inviati ad Amazon S3 per lo storage. Per attivarlo, scegli una AWS KMS chiave o scegli Inserisci una chiave ARN e fornisci l'ARN per la chiave. Immetti l'ARN usando questo formato: `arn:aws:kms:`*`region`*`:`*`account-id`*`:key/`*`key-id`*. Puoi specificare l'ARN anche sotto forma di alias di chiavi, ad esempio `arn:aws:kms:`*`region`*`:`*`account-id`*`:alias/`*`alias-name`*.

Per ulteriori informazioni, consulta i seguenti argomenti nella Guida per l'utente di Amazon Simple Storage Service:

- Per informazioni su `SSE-S3`, consulta Protezione dei dati mediante la crittografia lato server con chiavi crittografia gestite da Amazon S3 (SSE-S3).
- Per informazioni su`SSE-KMS`, consulta Protezione dei dati utilizzando la crittografia lato server con. AWS KMS keys
- Per ulteriori informazioni su `CSE-KMS`, consulta la pagina Using a KMS key stored in AWS KMS.

## Crittografia dei dati in transito

AWS fornisce la crittografia Transport Layer Security (TLS) per i dati in movimento. È possibile configurare le impostazioni di crittografia per crawler, processi ETL ed endpoint di sviluppo utilizzando le configurazioni di sicurezza in AWS Glue. È possibile attivare la AWS Glue Data Catalog crittografia tramite le impostazioni del Data Catalog.

A partire dal 4 settembre 2018, AWS KMS (porta la tua chiave e la crittografia lato server) per AWS Glue ETL è supportata AWS Glue Data Catalog .

## Conformità a FIPS

Se hai bisogno di moduli crittografici convalidati FIPS 140-2 per l'accesso AWS tramite un'interfaccia a riga di comando o un'API, usa un endpoint FIPS. Per ulteriori informazioni sugli endpoint FIPS disponibili, consulta il Federal Information Processing Standard (FIPS) 140-2.

## Gestione delle chiavi

Puoi utilizzare AWS Identity and Access Management (IAM) con AWS Glue per definire utenti, AWS risorse, gruppi, ruoli e politiche granulari relative all'accesso, al rifiuto e altro ancora.

È possibile definire l'accesso ai metadati utilizzando policy basate sulle risorse e basate sull'identità, a seconda delle esigenze dell'organizzazione. Le policy basate sulle risorse elencano le entità alle quali viene concesso o negato l'accesso alle risorse, permettendo di configurare policy come l'accesso multi-account. Le policy basate sull'identità sono specificamente collegate a utenti, gruppi e ruoli all'interno di IAM.

step-by-stepAd esempio, consulta Limita l'accesso ai tuoi AWS Glue Data Catalog con autorizzazioni IAM a livello di risorsa e politiche basate sulle risorse sul Big Data Blog. AWS

La parte dedicata all'accesso granulare della policy è definita all'interno della clausola `Resource`. Questa parte definisce sia l' AWS Glue Data Catalog oggetto su cui può essere eseguita l'azione, sia gli oggetti risultanti che vengono restituiti da tale operazione.

Un endpoint di sviluppo è un ambiente che puoi usare per sviluppare e testare gli script AWS Glue. Puoi aggiungere, eliminare o ruotare la chiave SSH di un endpoint di sviluppo.

A partire dal 4 settembre 2018, AWS KMS (porta la tua chiave e la crittografia lato server) per AWS Glue ETL and the AWS Glue Data Catalog è supportata.

# AWS Gluedipendenza da altri servizi AWS

Affinché un utente possa utilizzare la AWS Glue console, deve disporre di un set minimo di autorizzazioni che gli consentano di utilizzare le AWS Glue risorse del proprio AWS account. Oltre a queste autorizzazioni AWS Glue, la console richiede le autorizzazioni dei servizi seguenti:

- Autorizzazioni Amazon CloudWatch Logs per visualizzare i log.
- AWS Identity and Access Management (IAM) autorizzazioni per elencare e trasferire ruoli.
- CloudFormation autorizzazioni per lavorare con gli stack.
- Autorizzazioni Amazon Elastic Compute Cloud (Amazon EC2) per elencare cloud privati virtuali (VPCs), sottoreti, gruppi di sicurezza, istanze e altri oggetti (per configurare EC2 elementi Amazon, ad esempio VPCs durante l'esecuzione di job, crawler e la creazione di endpoint di sviluppo).
- Autorizzazioni Amazon Simple Storage Service (Amazon S3) per elencare bucket e oggetti e per recuperare e salvare script.
- Autorizzazioni Amazon Redshift necessarie per l'utilizzo dei cluster.
- Autorizzazioni Amazon Relational Database Service (Amazon RDS) per elencare le istanze.

## Endpoint di sviluppo

Un endpoint di sviluppo è un ambiente che puoi usare per sviluppare e testare gli script AWS Glue. È possibile utilizzare AWS Glue per creare, modificare ed eliminare gli endpoint di sviluppo. È possibile elencare tutti gli endpoint di sviluppo che vengono creati. Puoi aggiungere, eliminare o ruotare la chiave SSH di un endpoint di sviluppo. Puoi anche creare notebook che utilizzano l'endpoint di sviluppo.

Puoi fornire i valori di configurazione per effettuare il provisioning degli ambienti di sviluppo. Questi valori indicano a AWS Glue come configurare la rete in modo da poter accedere all'endpoint di sviluppo in modo sicuro e da poter accedere ai datastore. Quindi, è possibile creare un notebook che si colleghi all'endpoint di sviluppo. È possibile utilizzare il notebook per creare e testare lo script ETL.

Utilizza un ruolo AWS Identity and Access Management (IAM) con autorizzazioni simili al ruolo IAM che usi per eseguire lavori AWS Glue ETL. Utilizzare un cloud privato virtuale (VPC), una sottorete e un gruppo di sicurezza per creare un endpoint di sviluppo che sia in grado di connettersi alle risorse dati in modo sicuro. È possibile generare una coppia di chiavi SSH per connettersi all'ambiente di sviluppo tramite SSH.

È possibile creare endpoint di sviluppo per i dati Amazon S3 e all'interno di un VPC che è possibile utilizzare per accedere ai set di dati utilizzando JDBC.

È possibile installare un notebook Jupyter sul computer locale e utilizzarlo per eseguire il debug e testare gli script ETL in un endpoint di sviluppo. In alternativa, puoi usare un notebook Sagemaker per creare script ETL. JupyterLab AWS Vedi Usare un SageMaker notebook con il tuo endpoint di sviluppo.

AWS Gluecontrassegna EC2 le istanze Amazon con un nome preceduto da. `aws-glue-dev-endpoint`

È possibile configurare un server notebook su un endpoint di sviluppo per l'esecuzione PySpark con estensioni. AWS Glue

# Gestione delle identità e degli accessi per AWS Glue

AWS Identity and Access Management (IAM) è uno strumento Servizio AWS che aiuta un amministratore a controllare in modo sicuro l'accesso alle AWS risorse. Gli amministratori IAM controllano chi può essere autenticato (effettuato l'accesso) e autorizzato (dispone delle autorizzazioni) a utilizzare le risorse AWS Glue. IAM è uno Servizio AWS strumento che puoi utilizzare senza costi aggiuntivi.

> **Note**
>
> Puoi concedere l'accesso ai tuoi dati nel AWS Glue Data Catalog utilizzando AWS Glue metodi o AWS Lake Formation concessioni. Puoi utilizzare le policy AWS Identity and Access Management (IAM) per impostare un controllo granulare degli accessi con metodi. AWS Glue Lake Formation utilizza un modello di autorizzazioni `GRANT/REVOKE` più semplice, che è simile ai comandi `GRANT/REVOKE` in un sistema di database relazionale.
> Questa sezione include informazioni su come utilizzare i metodi AWS Glue. Per ulteriori informazioni sull'utilizzo delle autorizzazioni Lake Formation, consulta Concedere autorizzazioni Lake Formation nella Guida per gli sviluppatori di AWS Lake Formation .

Argomenti

- Destinatari
- Autenticazione con identità

- Gestione dell'accesso tramite policy
- Come funziona AWS Glue con IAM
- Configurazione delle autorizzazioni IAM per AWS Glue
- AWS Esempi di politiche di controllo degli accessi di Glue
- Concessione di politiche AWS gestite per AWS Glue
- Concessione di policy con ambito dinamico per l'esecuzione dei processi
- Specificare la risorsa AWS Glue ARNs
- Come concedere l'accesso multi-account
- Risoluzione dei problemi relativi all'identità e all'accesso a AWS Glue

## Destinatari

Il modo in cui utilizzi AWS Identity and Access Management (IAM) varia in base al tuo ruolo:

- Utente del servizio: richiedi le autorizzazioni all'amministratore se non riesci ad accedere alle funzionalità (consulta Risoluzione dei problemi relativi all'identità e all'accesso a AWS Glue)
- Amministratore del servizio: determina l'accesso degli utenti e invia le richieste di autorizzazione (consulta Come funziona AWS Glue con IAM)
- Amministratore IAM: scrivi policy per gestire l'accesso (consulta Esempi di policy basate sull'identità per Glue AWS)

## Autenticazione con identità

L'autenticazione è il modo in cui accedi AWS utilizzando le tue credenziali di identità. Devi autenticarti come utente IAM o assumendo un ruolo IAM. Utente root dell'account AWS

Puoi accedere come identità federata utilizzando credenziali provenienti da una fonte di identità come AWS IAM Identity Center (IAM Identity Center), autenticazione Single Sign-On o credenziali. Google/ Facebook Per ulteriori informazioni sull'accesso, consulta Come accedere all' Account AWS nella Guida per l'utente di Accedi ad AWS .

Per l'accesso programmatico, AWS fornisce un SDK e una CLI per firmare crittograficamente le richieste. Per ulteriori informazioni, consulta AWS Signature Version 4 per le richieste API nella Guida per l'utente di IAM.

## Account AWS utente root

Quando si crea un Account AWS, si inizia con un'identità di accesso denominata utente Account AWS root che ha accesso completo a tutte Servizi AWS le risorse. Consigliamo vivamente di non utilizzare l'utente root per le attività quotidiane. Per le attività che richiedono le credenziali dell'utente root, consulta Attività che richiedono le credenziali dell'utente root nella Guida per l'utente IAM.

## Identità federata

Come procedura ottimale, richiedi agli utenti umani di utilizzare la federazione con un provider di identità per accedere Servizi AWS utilizzando credenziali temporanee.

Un'identità federata è un utente della directory aziendale, del provider di identità Web o Directory Service che accede Servizi AWS utilizzando le credenziali di una fonte di identità. Le identità federate assumono ruoli che forniscono credenziali temporanee.

Per la gestione centralizzata degli accessi, si consiglia di utilizzare AWS IAM Identity Center. Per ulteriori informazioni, consulta Che cos'è il Centro identità IAM? nella Guida per l'utente di AWS IAM Identity Center .

## Utenti e gruppi IAM

Un utente IAM è una identità che dispone di autorizzazioni specifiche per una singola persona o applicazione. Ti consigliamo di utilizzare credenziali temporanee invece di utenti IAM con credenziali a lungo termine. Per ulteriori informazioni, consulta Richiedere agli utenti umani di utilizzare la federazione con un provider di identità per accedere AWS utilizzando credenziali temporanee nella Guida per l'utente IAM.

Un gruppo IAM specifica una raccolta di utenti IAM e semplifica la gestione delle autorizzazioni per gestire gruppi di utenti di grandi dimensioni. Per ulteriori informazioni, consulta Casi d'uso per utenti IAM nella Guida per l'utente di IAM.

## Ruoli IAM

Un ruolo IAM è un'identità con autorizzazioni specifiche che fornisce credenziali temporanee. Puoi assumere un ruolo passando da un ruolo utente a un ruolo IAM (console) o chiamando un'operazione AWS CLI o AWS API. Per ulteriori informazioni, consulta Metodi per assumere un ruolo nella Guida per l'utente di IAM.

I ruoli IAM sono utili per l'accesso degli utenti federati, le autorizzazioni utente IAM temporanee, l'accesso multi-account, l'accesso multi-servizio e le applicazioni in esecuzione su Amazon EC2. Per

maggiori informazioni, consultare Accesso a risorse multi-account in IAM nella Guida per l'utente IAM.

## Gestione dell'accesso tramite policy

Puoi controllare l'accesso AWS creando policy e associandole a AWS identità o risorse. Una policy definisce le autorizzazioni quando è associata a un'identità o a una risorsa. AWS valuta queste politiche quando un preside effettua una richiesta. La maggior parte delle politiche viene archiviata AWS come documenti JSON. Per maggiori informazioni sui documenti delle policy JSON, consulta Panoramica delle policy JSON nella Guida per l'utente IAM.

Utilizzando le policy, gli amministratori specificano chi ha accesso a cosa definendo quale principale può eseguire azioni su quali risorse e in quali condizioni.

Per impostazione predefinita, utenti e ruoli non dispongono di autorizzazioni. Un amministratore IAM crea le policy IAM e le aggiunge ai ruoli, che gli utenti possono quindi assumere. Le policy IAM definiscono le autorizzazioni indipendentemente dal metodo utilizzato per eseguirle.

### Policy basate sull'identità

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile collegare a un'identità (utente, gruppo o ruolo). Tali policy controllano le operazioni autorizzate per l'identità, nonché le risorse e le condizioni in cui possono essere eseguite. Per informazioni su come creare una policy basata su identità, consultare Definizione di autorizzazioni personalizzate IAM con policy gestite dal cliente nella Guida per l'utente IAM.

Le policy basate su identità possono essere policy in linea (con embedding direttamente in una singola identità) o policy gestite (policy autonome collegate a più identità). Per informazioni su come scegliere tra una policy gestita o una policy inline, consulta Scegliere tra policy gestite e policy in linea nella Guida per l'utente di IAM.

### Policy basate sulle risorse

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Gli esempi includono le policy di trust dei ruoli IAM e le policy dei bucket di Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. In una policy basata sulle risorse è obbligatorio specificare un'entità principale.

Le policy basate sulle risorse sono policy inline che si trovano in tale servizio. Non è possibile utilizzare le policy AWS gestite di IAM in una policy basata sulle risorse.

### Altri tipi di policy

AWS supporta tipi di policy aggiuntivi che possono impostare le autorizzazioni massime concesse dai tipi di policy più comuni:

- Limiti delle autorizzazioni: imposta il numero massimo di autorizzazioni che una policy basata su identità ha la possibilità di concedere a un'entità IAM. Per ulteriori informazioni, consulta Limiti delle autorizzazioni per le entità IAM nella Guida per l'utente di IAM.
- Politiche di controllo del servizio (SCPs): specificano le autorizzazioni massime per un'organizzazione o un'unità organizzativa in. AWS Organizations Per ulteriori informazioni, consultare Policy di controllo dei servizi nella Guida per l'utente di AWS Organizations .
- Politiche di controllo delle risorse (RCPs): imposta le autorizzazioni massime disponibili per le risorse nei tuoi account. Per ulteriori informazioni, consulta Politiche di controllo delle risorse (RCPs) nella Guida per l'AWS Organizations utente.
- Policy di sessione: policy avanzate passate come parametro quando si crea una sessione temporanea per un ruolo o un utente federato. Per maggiori informazioni, consultare Policy di sessione nella Guida per l'utente IAM.

### Più tipi di policy

Quando a una richiesta si applicano più tipi di policy, le autorizzazioni risultanti sono più complicate da comprendere. Per scoprire come si AWS determina se consentire o meno una richiesta quando sono coinvolti più tipi di policy, consulta Logica di valutazione delle policy nella IAM User Guide.

## Come funziona AWS Glue con IAM

Prima di utilizzare IAM per gestire l'accesso a AWS Glue, scopri quali funzionalità IAM sono disponibili per l'uso con AWS Glue.

Funzionalità IAM che puoi usare con AWS Glue

| Funzionalità IAM | AWS Supporto Glue |
| --- | --- |
| Policy basate sull'identità | Sì |

| Funzionalità IAM | AWS Supporto Glue |
| --- | --- |
| Policy basate su risorse | Parziale |
| Operazioni di policy | Sì |
| Risorse relative alle policy | Sì |
| Chiavi di condizione della policy (specifica del servizio) | Sì |
| ACLs | No |
| ABAC (tag nelle policy) | Parziale |
| Credenziali temporanee | Sì |
| Autorizzazioni del principale | No |
| Ruoli di servizio | Sì |
| Ruoli collegati al servizio | No |

Per avere una visione di alto livello di come AWS Glue e gli altri AWS servizi funzionano con la maggior parte delle funzionalità IAM, consulta AWS i servizi che funzionano con IAM nella IAM User Guide.

## Politiche basate sull'identità per Glue AWS

Supporta le policy basate sull'identità: sì

Le policy basate sull'identità sono documenti di policy di autorizzazione JSON che è possibile allegare a un'identità (utente, gruppo di utenti o ruolo IAM). Tali policy definiscono le operazioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una policy basata su identità, consulta Definizione di autorizzazioni personalizzate IAM con policy gestite dal cliente nella Guida per l'utente di IAM.

Con le policy basate sull'identità di IAM, è possibile specificare quali operazioni e risorse sono consentite o respinte, nonché le condizioni in base alle quali le operazioni sono consentite o respinte.

Per informazioni su tutti gli elementi utilizzabili in una policy JSON, consulta Guida di riferimento agli elementi delle policy JSON IAM nella Guida per l'utente IAM.

AWS Glue supporta le policy basate su identità (policy IAM) per tutte le operazioni AWS Glue. Collegando una policy, puoi concedere le autorizzazioni per creare, accedere o modificare una risorsa AWS Glue, ad esempio una tabella nel AWS Glue Data Catalog.

Esempi di policy basate sull'identità per Glue AWS

Per visualizzare esempi di politiche basate sull'identità di AWS Glue, vedere. Esempi di policy basate sull'identità per Glue AWS

## Politiche basate sulle risorse all'interno di Glue AWS

Supporta le policy basate su risorse: parziale

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Esempi di policy basate sulle risorse sono le policy di attendibilità dei ruoli IAM e le policy di bucket Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. Quando è collegata a una risorsa, una policy definisce le operazioni che un principale può eseguire su tale risorsa e a quali condizioni. In una policy basata sulle risorse è obbligatorio specificare un'entità principale. I principali possono includere account, utenti, ruoli, utenti federati o. Servizi AWS

Per consentire l'accesso multi-account, è possibile specificare un intero account o entità IAM in un altro account come entità principale in una policy basata sulle risorse. Per ulteriori informazioni, consulta Accesso a risorse multi-account in IAM nella Guida per l'utente IAM.

> **Note**
>
> Una policy delle risorse AWS Glue può essere utilizzata solo per gestire le autorizzazioni per le risorse di catalogo dati. Non è possibile collegarla ad altre risorse AWS Glue, ad esempio processi, trigger, endpoint di sviluppo, crawler o classificatori.
> È ammessa solo una policy sulle risorse per catalogo e la sua dimensione è limitata a 10 KB.

In AWS Glue, una politica delle risorse è allegata a un catalogo, che è un contenitore virtuale per tutti i tipi di risorse del Data Catalog menzionati in precedenza. Ogni AWS account possiede un singolo

catalogo in una AWS regione il cui ID di catalogo è uguale all'ID dell' AWS account. Non è possibile eliminare o modificare un catalogo.

Una policy della risorsa viene valutata per tutte le chiamate al catalogo effettuate dall'API. In questo caso, il principale del chiamante è incluso nel blocco `"Principal"` del documento delle policy.

Per visualizzare esempi di politiche basate sulle risorse di AWS Glue, vedere. Esempi di policy basate sulle risorse per Glue AWS

## Azioni politiche per AWS Glue

Supporta le operazioni di policy: si

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale entità principale può eseguire operazioni su quali risorse e in quali condizioni.

L'elemento `Action` di una policy JSON descrive le operazioni che è possibile utilizzare per consentire o negare l'accesso in una policy. Includere le operazioni in una policy per concedere le autorizzazioni a eseguire l'operazione associata.

Per visualizzare un elenco delle azioni AWS Glue, vedere Actions defined by AWS Glue nel Service Authorization Reference.

Le azioni politiche in AWS Glue utilizzano il seguente prefisso prima dell'azione:

```
glue
```

Per specificare più operazioni in una sola istruzione, occorre separarle con la virgola.

```
"Action": [
      "glue:action1",
      "glue:action2"
         ]
```

È possibile specificare più azioni tramite caratteri jolly (*). Ad esempio, per specificare tutte le azioni che iniziano con la parola `Get`, includi la seguente azione:

```
"Action": "glue:Get*"
```

Per visualizzare le policy di esempio, consulta AWS Esempi di politiche di controllo degli accessi di Glue.

## Risorse politiche per AWS Glue

Supporta le risorse relative alle policy: sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale entità principale può eseguire operazioni su quali risorse e in quali condizioni.

L'elemento JSON `Resource` della policy specifica l'oggetto o gli oggetti ai quali si applica l'operazione. Come best practice, specifica una risorsa utilizzando il suo nome della risorsa Amazon (ARN). Per le azioni che non supportano le autorizzazioni a livello di risorsa, si utilizza un carattere jolly (*) per indicare che l'istruzione si applica a tutte le risorse.

```
"Resource": "*"
```

Per ulteriori informazioni su come controllare l'accesso alle risorse AWS Glue utilizzando ARNs, vedereSpecificare la risorsa AWS Glue ARNs.

Per visualizzare un elenco dei tipi di risorse AWS Glue e relativi ARNs, vedere Resources defined by AWS Glue nel Service Authorization Reference. Per sapere quali azioni puoi usare per specificare l'ARN di ogni risorsa, vedi Azioni definite da AWS Glue.

## Chiavi relative alle condizioni della policy per AWS Glue

Supporta le chiavi di condizione delle policy specifiche del servizio: sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale entità principale può eseguire operazioni su quali risorse e in quali condizioni.

L'elemento `Condition` specifica quando le istruzioni vengono eseguite in base a criteri definiti. È possibile compilare espressioni condizionali che utilizzano operatori di condizione, ad esempio uguale a o minore di, per soddisfare la condizione nella policy con i valori nella richiesta. Per visualizzare tutte le chiavi di condizione AWS globali, consulta le chiavi di contesto delle condizioni AWS globali nella Guida per l'utente IAM.

Per visualizzare un elenco delle chiavi di condizione di AWS Glue, vedere Condition keys for AWS Glue nel Service Authorization Reference. Per sapere con quali azioni e risorse puoi usare una chiave di condizione, vedi Actions defined by AWS Glue.

Per visualizzare le policy di esempio, consulta Controllo delle impostazioni utilizzando le chiavi di condizione o di contesto.

## ACLs in AWS Glue

Supporti ACLs: No

Le liste di controllo degli accessi (ACLs) controllano quali principali (membri dell'account, utenti o ruoli) dispongono delle autorizzazioni per accedere a una risorsa. ACLs sono simili alle politiche basate sulle risorse, sebbene non utilizzino il formato del documento di policy JSON.

## ABAC con Glue AWS

Supporta ABAC (tag nelle policy): parzialmente

Il controllo degli accessi basato su attributi (ABAC) è una strategia di autorizzazione che definisce le autorizzazioni in base agli attributi, chiamati tag. Puoi allegare tag a entità e AWS risorse IAM, quindi progettare politiche ABAC per consentire operazioni quando il tag del principale corrisponde al tag sulla risorsa.

Per controllare l'accesso basato su tag, fornire informazioni sui tag nell'elemento condizione di una policy utilizzando le chiavi di condizione `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` o `aws:TagKeys`.

Se un servizio supporta tutte e tre le chiavi di condizione per ogni tipo di risorsa, il valore per il servizio è Sì. Se un servizio supporta tutte e tre le chiavi di condizione solo per alcuni tipi di risorsa, allora il valore sarà Parziale.

Per maggiori informazioni su ABAC, consulta Definizione delle autorizzazioni con autorizzazione ABAC nella Guida per l'utente di IAM. Per visualizzare un tutorial con i passaggi per l'impostazione di ABAC, consulta Utilizzo del controllo degli accessi basato su attributi (ABAC) nella Guida per l'utente di IAM.

> **Important**
>
> Le chiavi di contesto della condizione si applicano solo alle operazioni API AWS Glue su crawler, processi, trigger ed endpoint di sviluppo. Per ulteriori informazioni sulle operazioni API interessate, consulta Condition keys for AWS Glue.
> Le operazioni dell'API AWS Glue catalogo dati attualmente non supportano le chiavi di contesto delle condizioni globali `aws:referer` e `aws:UserAgent`.

Per visualizzare una policy basata sulle identità di esempio per limitare l'accesso a una risorsa basata su tag su tale risorsa, consulta Autorizzazione dell'accesso utilizzando tag.

## Utilizzo di credenziali temporanee con AWS Glue

Supporta le credenziali temporanee: sì

Le credenziali temporanee forniscono un accesso a breve termine alle AWS risorse e vengono create automaticamente quando si utilizza la federazione o si cambia ruolo. AWS consiglia di generare dinamicamente credenziali temporanee anziché utilizzare chiavi di accesso a lungo termine. Per ulteriori informazioni, consulta Credenziali di sicurezza temporanee in IAM e Servizi AWS compatibili con IAM nella Guida per l'utente IAM.

## Autorizzazioni principali multiservizio per Glue AWS

Supporta l'inoltro delle sessioni di accesso (FAS): no

Le sessioni di accesso inoltrato (FAS) utilizzano le autorizzazioni del principale che chiama an Servizio AWS, combinate con la richiesta di effettuare richieste Servizio AWS ai servizi downstream. Per i dettagli delle policy relative alle richieste FAS, consulta Forward access sessions.

## Ruoli di servizio per AWS Glue

Supporta i ruoli di servizio: sì

Un ruolo di servizio è un ruolo IAM che un servizio assume per eseguire operazioni per tuo conto. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per ulteriori informazioni, consulta Create a role to delegate permissions to an Servizio AWS nella Guida per l'utente IAM.

> **Warning**
>
> La modifica delle autorizzazioni per un ruolo di servizio potrebbe interrompere AWS la funzionalità Glue. Modifica i ruoli di servizio solo quando AWS Glue fornisce indicazioni in tal senso.

Per istruzioni dettagliate sulla creazione di un ruolo di servizio per AWS Glue, vedere Fase 1: creare una policy IAM per il servizio AWS Glue eFase 2: creare un ruolo IAM per AWS Glue.

### Ruoli collegati ai servizi per Glue AWS

Supporta i ruoli collegati ai servizi: no

Un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un. Servizio AWS Il servizio può assumere il ruolo per eseguire un'operazione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati al servizio, ma non modificarle.

Per ulteriori informazioni su come creare e gestire i ruoli collegati ai servizi, consulta Servizi AWS supportati da IAM. Trova un servizio nella tabella che include un `Yes` nella colonna Service-linked role (Ruolo collegato ai servizi). Scegli il collegamento Sì per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.

## Configurazione delle autorizzazioni IAM per AWS Glue

Utilizzi AWS Identity and Access Management (IAM) per definire le politiche e i ruoli da utilizzare per accedere alle risorse. AWS Glue I passaggi seguenti illustrano varie opzioni per la configurazione delle autorizzazioni per AWS Glue. A seconda delle necessità aziendali, potrebbe essere necessario aggiungere o ridurre l'accesso alle risorse.

> **Note**
>
> Per iniziare AWS Glue invece con le autorizzazioni IAM di base, consultaConfigurazione delle autorizzazioni IAM per AWS Glue.

1. Crea una policy IAM per il servizio AWS Glue: crea una policy del servizio che permette l'accesso alle risorse AWS Glue.
2. Crea un ruolo IAM per AWS Glue: crea un ruolo IAM e collega la policy di servizio AWS Glue e una policy per le risorse Amazon Simple Storage Service (Amazon S3) utilizzate da AWS Glue.
3. Collega una policy agli utenti o ai gruppi che accedono a AWS Glue: collega le policy a qualsiasi utente o gruppo che accede alla console AWS Glue.
4. Crea una policy IAM per i notebook: crea una policy di server notebook da usare per la creazione di server notebook negli endpoint di sviluppo.
5. Crea un ruolo IAM per i notebook: crea un ruolo IAM e collega la policy del server notebook.
6. Crea una policy IAM per i notebook Amazon SageMaker AI: crea una policy IAM da utilizzare durante la creazione di notebook Amazon SageMaker AI sugli endpoint di sviluppo.

7. Crea un ruolo IAM per i notebook Amazon SageMaker AI: crea un ruolo IAM e allega la policy per concedere le autorizzazioni durante la creazione di notebook Amazon SageMaker AI sugli endpoint di sviluppo.

## Fase 1: creare una policy IAM per il servizio AWS Glue

Per qualsiasi operazione che accede ai dati su un'altra AWS risorsa, come l'accesso ai tuoi oggetti in Amazon S3AWS Glue, è necessaria l'autorizzazione per accedere alla risorsa per tuo conto. Fornisci tali autorizzazioni utilizzando AWS Identity and Access Management (IAM).

> **Note**
>
> Puoi saltare questo passaggio se utilizzi la politica AWS gestita. **`AWSGlueServiceRole`**

In questa fase, crei una policy simile a `AWSGlueServiceRole`. Puoi trovare la versione più recente di `AWSGlueServiceRole` nella console IAM.

Per creare una policy IAM per AWS Glue

Questa policy concede l'autorizzazione per alcune operazioni Amazon S3 per gestire le risorse nell'account richieste da AWS Glue quando assume il ruolo usando la policy. Alcune delle risorse specificate in questa politica si riferiscono a nomi predefiniti utilizzati dai bucket Amazon S3, dagli AWS Glue script ETL di Amazon S3, dai log e dalle risorse Amazon EC2. CloudWatch Per semplicità, AWS Glue scrive alcuni oggetti Amazon S3 nei bucket nell'account con il prefisso `aws-glue-*` per impostazione predefinita.

1. Accedi e apri la console IAM all'indirizzo. Console di gestione AWS https://console.aws.amazon.com/iam/
2. Nel riquadro di navigazione sinistro, scegli Policy.
3. Scegli Crea policy.
4. Nella schermata Create Policy (Crea policy), passa alla scheda per modificare JSON. Crea un documento di policy con le seguenti istruzioni JSON, quindi scegli Review policy (Verifica policy).

> **Note**
>
> Aggiungi le autorizzazioni necessarie per le risorse Amazon S3. Potresti voler esplorare la sezione delle risorse della policy di accesso solo con le risorse necessarie.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:*",
        "s3:GetBucketLocation",
        "s3:ListBucket",
        "s3:ListAllMyBuckets",
        "s3:GetBucketAcl",
        "ec2:DescribeVpcEndpoints",
        "ec2:DescribeRouteTables",
        "ec2:CreateNetworkInterface",
        "ec2:DeleteNetworkInterface",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeSubnets",
        "ec2:DescribeVpcAttribute",
        "iam:ListRolePolicies",
        "iam:GetRole",
        "iam:GetRolePolicy",
        "cloudwatch:PutMetricData"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:CreateBucket",
        "s3:PutBucketPublicAccessBlock"
```

```
      ],
      "Resource": [
        "arn:aws:s3:::aws-glue-*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:PutObject",
        "s3:DeleteObject"
      ],
      "Resource": [
        "arn:aws:s3:::aws-glue-*/*",
        "arn:aws:s3:::*/*aws-glue-*/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::crawler-public*",
        "arn:aws:s3:::aws-glue-*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:AssociateKmsKey"
      ],
      "Resource": [
        "arn:aws:logs:*:*:log-group:/aws-glue/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateTags",
        "ec2:DeleteTags"
```

```
      ],
      "Condition": {
        "ForAllValues:StringEquals": {
          "aws:TagKeys": [
            "aws-glue-service-resource"
          ]
        }
      },
      "Resource": [
        "arn:aws:ec2:*:*:network-interface/*",
        "arn:aws:ec2:*:*:security-group/*",
        "arn:aws:ec2:*:*:instance/*"
      ]
    }
  ]
}
```

La tabella seguente descrive le autorizzazioni concesse dalla policy.

| Azione | Risorsa | Descrizione |
| --- | --- | --- |
| `"glue:*"` | `"*"` | Concede l'autorizzazione per eseguire tutte le operazioni API AWS Glue. |
| `"s3:GetBucketLocation", "s3:ListBucket", "s3:ListAllMyBuckets", "s3:GetBucketAcl",` | `"*"` | Permette di elencare i bucket Amazon S3 da crawler, processi, endpoint di sviluppo e server notebook. |

| Azione | Risorsa | Descrizione |
| --- | --- | --- |
| `"ec2:DescribeVpcEndpoints", "ec2:DescribeRouteTables", "ec2:CreateNetworkInterface", "ec2:DeleteNetworkInterface", "ec2:DescribeNetworkInterfaces", "ec2:DescribeSecurityGroups", "ec2:DescribeSubnets", "ec2:DescribeVpcAttribute",` | `"*"` | Consente la configurazione di elementi di rete Amazon EC2, come cloud privati virtuali (VPCs) durante l'esecuzione di job, crawler ed endpoint di sviluppo. |
| `"iam:ListRolePolicies", "iam:GetRole", "iam:GetRolePolicy"` | `"*"` | Permette di elencare i ruoli IAM da crawler, processi, endpoint di sviluppo e server notebook. |
| `"cloudwatch:PutMetricData"` | `"*"` | Consente di scrivere metriche per CloudWatch i lavori. |
| `"s3:CreateBucket", "s3:PutBucketPublicAccessBlock"` | `"arn:aws:s3:::aws-glue-*"` | Consente la creazione di bucket Amazon S3 nel tuo account da processi e server notebook.<br><br>Convenzione per la denominazione: utilizza cartelle Amazon S3 denominate aws-glue-.<br><br>Consente ad AWS Glue di creare i bucket che bloccano l'accesso pubblico. |

| Azione | Risorsa | Descrizione |
| --- | --- | --- |
| `"s3:GetObject", "s3:PutObject", "s3:DeleteObject"` | `"arn:aws:s3:::aws-glue-*/*", "arn:aws:s3:::*/*aws-glue-*/*"` | Permette di ottenere, inserire ed eliminare oggetti Amazon S3 nell'account quando vengono archiviat i oggetti come script ETL e posizioni dei server notebook.<br><br>Convenzione per la denominazione: concede l'autorizzazione alle cartelle o ai bucket Amazon S3 i cui nomi hanno il prefisso aws-glue-. |
| `"s3:GetObject"` | `"arn:aws:s3:::crawler-public*", "arn:aws:s3:::aws-glue-*"` | Permette di ottenere gli oggetti Amazon S3 usati da esempi e tutorial da crawler e processi.<br><br>Convenzione per la denominazione: i nomi di bucket Amazon S3 iniziano con crawler-public e aws-glue-. |
| `"logs:CreateLogGroup", "logs:CreateLogStream", "logs:PutLogEvents"` | `"arn:aws:logs:*:*:log-group:/aws-glue/*"` | Consente di scrivere log su Logs. CloudWatch<br><br>Convenzione di denominazione: AWS Glue scrive i log in gruppi di log i cui nomi iniziano con aws-glue. |

| Azione | Risorsa | Descrizione |
| --- | --- | --- |
| `"ec2:CreateTags", "ec2:DeleteTags"` | `"arn:aws:ec2:*:*:network-interface/*", "arn:aws:ec2:*:*:security-group/*", "arn:aws:ec2:*:*:instance/*"` | Consente il tagging delle risorse Amazon EC2 create per gli endpoint di sviluppo.<br><br>Convenzione di denominazione: AWS Glue etichetta con le interfacce di rete, i gruppi di sicurezza e le istanze di Amazon EC2. aws-glue-service-resource |

5. Nella schermata Verifica policy, inserisci il Nome policy, ad esempio GlueServiceRolePolicy. Digita una descrizione facoltativa e, al termine, seleziona Create policy (Crea policy).

## Fase 2: creare un ruolo IAM per AWS Glue

A questo punto devi concedere le autorizzazioni del ruolo IAM che AWS Glue può assumere quando chiama altri servizi per tuo conto. Ciò include l'accesso ad Amazon S3 per le origini, le destinazioni, gli script e le directory temporanee che usi con AWS Glue. Le autorizzazioni sono necessarie per crawler, processi ed endpoint di sviluppo.

Fornisci tali autorizzazioni utilizzando AWS Identity and Access Management (IAM). Aggiungi una policy al ruolo IAM passato a AWS Glue.

Per creare un ruolo IAM all'interno del job editor

1. Quando crei un lavoro nella AWS Glue console, individua la sezione relativa al ruolo.
2. Scegli Create new role (Crea nuovo ruolo).
3. Si apre un modulo in linea per la creazione di ruoli, che consente di:
   - Specificate il nome del ruolo, `AWSGlueServiceRoleDefault` ad esempio.
   - La politica gestita `AWSGlueServiceRole` viene selezionata automaticamente.
   - Rivedi la politica di fiducia per assumere il ruolo.
   - Aggiungi tag opzionali per i metadati.
4. Scegli Crea ruolo.

5. Il ruolo appena creato viene selezionato automaticamente per il tuo lavoro.

In alternativa, puoi utilizzare la console IAM per creare il ruolo:

Per creare un ruolo IAM per AWS Glue l'utilizzo della console IAM

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione a sinistra seleziona Ruoli.
3. Scegli Crea ruolo.
4. Per il tipo di entità attendibile, scegli il servizio AWS . Quindi, per l'assistenza o il caso d'uso, trova e seleziona AWS Glue. Scegli Next (Successivo).
5. Nella pagina Aggiungi autorizzazioni, scegli le politiche che contengono le autorizzazioni richieste; ad esempio, la politica gestita per le autorizzazioni generali AWS Glue e la politica AWS gestita AmazonS3 **`AWSGlueServiceRole`** per l' AWS accesso FullAccess alle risorse Amazon S3. Quindi scegli Successivo.

> **Note**
>
> Verifica che una delle policy in questo ruolo conceda le autorizzazioni per le origini e le destinazioni Amazon S3. Puoi fornire una policy personalizzata per l'accesso a risorse Amazon S3 specifiche. Le origini dati richiedono le autorizzazioni `s3:ListBucket` e `s3:GetObject`. Le destinazioni dati richiedono le autorizzazioni `s3:ListBucket`, `s3:PutObject` e `s3:DeleteObject`. Per ulteriori informazioni sulla creazione di una policy Amazon S3 per le risorse, vedi Specificare le risorse in una policy. Per un esempio di policy Amazon S3, consulta la pagina relativa alla scrittura di policy IAM per concedere l'accesso a un bucket Amazon S3.
> Se prevedi di accedere a origini e destinazioni Amazon S3 crittografate con SSE-KMS, collega una policy che permetta a crawler, processi ed endpoint di sviluppo AWS Glue di decrittografare i dati. Per ulteriori informazioni, consulta Protezione dei dati utilizzando la crittografia lato server con chiavi gestite (SSE-KMS). AWS KMS
> Di seguito è riportato un esempio di :
>
> JSON
>
> ```
> {
> ```

```
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:*:111122223333:key/key-id"
      ]
    }
  ]
}
```

6. Assegna un nome al tuo ruolo e aggiungi una descrizione (facoltativa), quindi rivedi la policy di attendibilità e le autorizzazioni. Per Role name (Nome ruolo), digita un nome per il ruolo, ad esempio `AWSGlueServiceRoleDefault`. Crea il ruolo usando come prefisso del nome la stringa `AWSGlueServiceRole` per permettere il passaggio del ruolo dagli utenti della console al servizio. Le policy fornite da AWS Glue prevedono ruoli di servizio IAM che iniziano con `AWSGlueServiceRole`. In caso contrario, è necessario aggiungere una policy per concedere agli utenti l'autorizzazione `iam:PassRole` per i ruoli IAM in modo da soddisfare la convenzione per la denominazione. Selezionare Crea ruolo.

> **Note**
>
> Quando crei un notebook con un ruolo, tale ruolo viene passato alle sessioni interattive in modo che lo stesso ruolo possa essere utilizzato in entrambe le posizioni. Come tale, il permesso `iam:PassRole` deve essere parte della policy del ruolo.
> Crea una nuova policy per il tuo ruolo utilizzando l'esempio seguente. Sostituisci il numero di account con il tuo e il nome del ruolo.
>
> JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
```

```
      "Resource": "arn:aws:iam::090000000210:role/<role_name>"
    }
  ]
}
```

7. Aggiungi tag al tuo ruolo (facoltativo). I tag sono coppie chiave-valore che è possibile aggiungere alle AWS risorse per identificare, organizzare o cercare risorse. Quindi seleziona Create role (Crea ruolo).

## Fase 3: Collegamento di una policy agli utenti o ai gruppi che accedono a AWS Glue

L'amministratore deve assegnare le autorizzazioni a qualsiasi utente, gruppo o ruolo utilizzando la AWS Glue console o AWS Command Line Interface ()AWS CLI. Puoi fornire queste autorizzazioni usando AWS Identity and Access Management (IAM), tramite le policy. Questa fase descrive l'assegnazione di autorizzazioni a utenti o gruppi.

Una volta completata questa fase, all'utente o al gruppo sono collegate le policy seguenti:

- La politica AWS gestita `AWSGlueConsoleFullAccess` o la politica personalizzata GlueConsoleAccessPolicy
- **`AWSGlueConsoleSageMakerNotebookFullAccess`**
- **`CloudWatchLogsReadOnlyAccess`**
- **`AWSCloudFormationReadOnlyAccess`**
- **`AmazonAthenaFullAccess`**

Per collegare una policy inline e incorporarla in un utente o in un gruppo

È possibile allegare una policy AWS gestita o una policy in linea a un utente o gruppo per accedere alla AWS Glue console. Alcune delle risorse specificate in questa politica si riferiscono a nomi predefiniti utilizzati dai AWS Glue bucket Amazon S3, dagli script ETL di Amazon S3 CloudWatch , dai log e dalle risorse Amazon EC2. CloudFormation Per semplicità, AWS Glue scrive alcuni oggetti Amazon S3 nei bucket nell'account con il prefisso `aws-glue-*` per impostazione predefinita.

> **Note**
>
> Puoi saltare questo passaggio se utilizzi la policy gestita. AWS **`AWSGlueConsoleFullAccess`**

> **Important**
>
> AWS Glue richiede l'autorizzazione per assumere un ruolo usato per eseguire il lavoro per tuo conto. A tale scopo, aggiungi le autorizzazioni **`iam:PassRole`** agli utenti o ai gruppi AWS Glue. Questa policy concede l'autorizzazione ai ruoli che iniziano con `AWSGlueServiceRole` per i ruoli di servizio AWS Glue e `AWSGlueServiceNotebookRole` per i ruoli necessari al momento della creazione di un server notebook. Puoi anche creare una policy per le autorizzazioni `iam:PassRole` che segue la convenzione per la denominazione.
> In base alle best practice di sicurezza, si consiglia di limitare l'accesso rafforzando le policy per limitare ulteriormente l'accesso ai bucket Amazon CloudWatch e ai gruppi di log di Amazon S3. Per un esempio di policy Amazon S3, consulta la pagina relativa alla scrittura di policy IAM per concedere l'accesso a un bucket Amazon S3.

In questa fase, crei una policy simile a `AWSGlueConsoleFullAccess`. Puoi trovare la versione più recente di `AWSGlueConsoleFullAccess` nella console IAM.

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo. https://console.aws.amazon.com/iam/
2. Nel pannello di navigazione, scegli Utenti o Gruppi di utenti.
3. Nell'elenco, scegli il nome dell'utente o del gruppo in cui integrare una policy.
4. Scegliere la scheda Permissions (Autorizzazioni) e, se necessario, espandere la sezione Permissions policies (Policy autorizzazioni).
5. Scegli il collegamento Add Inline policy (Aggiungi policy inline).
6. Nella schermata Create Policy (Crea policy), passa alla scheda per modificare JSON. Crea un documento di policy con le seguenti istruzioni JSON, quindi scegli Review policy (Verifica policy).

   JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:*",
```

```
        "redshift:DescribeClusters",
        "redshift:DescribeClusterSubnetGroups",
        "iam:ListRoles",
        "iam:ListUsers",
        "iam:ListGroups",
        "iam:ListRolePolicies",
        "iam:GetRole",
        "iam:GetRolePolicy",
        "iam:ListAttachedRolePolicies",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeSubnets",
        "ec2:DescribeVpcs",
        "ec2:DescribeVpcEndpoints",
        "ec2:DescribeRouteTables",
        "ec2:DescribeVpcAttribute",
        "ec2:DescribeKeyPairs",
        "ec2:DescribeInstances",
        "rds:DescribeDBInstances",
        "rds:DescribeDBClusters",
        "rds:DescribeDBSubnetGroups",
        "s3:ListAllMyBuckets",
        "s3:ListBucket",
        "s3:GetBucketAcl",
        "s3:GetBucketLocation",
        "cloudformation:DescribeStacks",
        "cloudformation:GetTemplateSummary",
        "dynamodb:ListTables",
        "kms:ListAliases",
        "kms:DescribeKey",
        "cloudwatch:GetMetricData",
        "cloudwatch:ListDashboards"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::*/*aws-glue-*/*",
```

```
        "arn:aws:s3:::aws-glue-*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "tag:GetResources"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:CreateBucket",
        "s3:PutBucketPublicAccessBlock"
      ],
      "Resource": [
        "arn:aws:s3:::aws-glue-*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "logs:GetLogEvents"
      ],
      "Resource": [
        "arn:aws:logs:*:*:/aws-glue/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "cloudformation:CreateStack",
        "cloudformation:DeleteStack"
      ],
      "Resource": "arn:aws:cloudformation:*:*:stack/aws-glue*/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:RunInstances"
      ],
```

```
      "Resource": [
        "arn:aws:ec2:*:*:instance/*",
        "arn:aws:ec2:*:*:key-pair/*",
        "arn:aws:ec2:*:*:image/*",
        "arn:aws:ec2:*:*:security-group/*",
        "arn:aws:ec2:*:*:network-interface/*",
        "arn:aws:ec2:*:*:subnet/*",
        "arn:aws:ec2:*:*:volume/*"
      ]
    },
    {
      "Action": [
        "iam:PassRole"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:iam::*:role/AWSGlueServiceRole*",
      "Condition": {
        "StringLike": {
          "iam:PassedToService": [
            "glue.amazonaws.com"
          ]
        }
      }
    },
    {
      "Action": [
        "iam:PassRole"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:iam::*:role/AWSGlueServiceNotebookRole*",
      "Condition": {
        "StringLike": {
          "iam:PassedToService": [
            "ec2.amazonaws.com"
          ]
        }
      }
    },
    {
      "Action": [
        "iam:PassRole"
      ],
      "Effect": "Allow",
      "Resource": [
```

```
        "arn:aws:iam::*:role/service-role/AWSGlueServiceRole*"
      ],
      "Condition": {
        "StringLike": {
          "iam:PassedToService": [
            "glue.amazonaws.com"
          ]
        }
      }
    }
  ]
}
```

La tabella seguente descrive le autorizzazioni concesse dalla policy.

| Azione | Risorsa | Descrizione |
| --- | --- | --- |
| `"glue:*"` | `"*"` | Concede l'autorizzazione per eseguire tutte le operazioni API AWS Glue.<br><br>Se in precedenza la policy è stata creata senza l'operazione `"glue:*"`, è necessario aggiungere le seguenti autorizzazioni individuali alla policy:<br><br>• «collaListCrawlers»:<br>• «collaBatchGetCrawlers»:<br>• «collaListTriggers»:<br>• «collaBatchGetTriggers»:<br>• «collaListDevEndpoints»:<br>• «collaBatchGetDevEndpoints»:<br>• «collaListJobs»:<br>• «collaBatchGetJobs»: |
| `"redshift:DescribeClusters", "redshift:DescribeClusterSubnetGroups"` | `"*"` | Permette la creazione di connessioni ad Amazon Redshift. |

| Azione | Risorsa | Descrizione |
| --- | --- | --- |
| `"iam:ListRoles", "iam:ListRolePolicies", "iam:GetRole", "iam:GetRolePolicy", "iam:ListAttachedRolePolicies"` | `"*"` | Consente l'elenco dei ruoli IAM quando si utilizzano crawler, processi, endpoint di sviluppo e server notebook. |
| `"ec2:DescribeSecurityGroups", "ec2:DescribeSubnets", "ec2:DescribeVpcs", "ec2:DescribeVpcEndpoints", "ec2:DescribeRouteTables", "ec2:DescribeVpcAttribute", "ec2:DescribeKeyPairs", "ec2:DescribeInstances"` | `"*"` | Consente la configurazione degli elementi della rete Amazon EC2, ad esempio durante l'esecuzione di job VPCs, crawler ed endpoint di sviluppo. |
| `"rds:DescribeDBInstances"` | `"*"` | Consente la creazione di connessioni ad Amazon RDS. |
| `"s3:ListAllMyBuckets", "s3:ListBucket", "s3:GetBucketAcl", "s3:GetBucketLocation"` | `"*"` | Permette di elencare i bucket Amazon S3 quando vengono usati crawler, processi, endpoint di sviluppo e server notebook. |
| `"dynamodb:ListTables"` | `"*"` | Permette di elencare tabelle DynamoDB. |
| `"kms:ListAliases", "kms:DescribeKey"` | `"*"` | Permette di usare le chiavi KMS. |

| Azione | Risorsa | Descrizione |
| --- | --- | --- |
| `"cloudwatch:GetMetricData", "cloudwatch:ListDashboards"` | `"*"` | Consente di lavorare con le metriche. CloudWatch |
| `"s3:GetObject", "s3:PutObject"` | `"arn:aws:s3::: aws-glue-*/*", "arn:aws:s3::: */*aws-glue-*/*", "arn:aws:s3::: aws-glue-*"` | Permette di ottenere e inserire oggetti Amazon S3 nell'account quando vengono archiviati oggetti come script ETL e posizioni dei server notebook.<br><br>Convenzione per la denominazione: concede l'autorizzazione alle cartelle o ai bucket Amazon S3 i cui nomi hanno il prefisso aws-glue-. |
| `"tag:GetResources"` | `"*"` | Consente il recupero dei tag. AWS |

| Azione | Risorsa | Descrizione |
| --- | --- | --- |
| `"s3:CreateBucket", "s3:PutBucketPublicAccessBlock"` | `"arn:aws:s3::: aws-glue-*"` | Permette di creare un bucket Amazon S3 nell'account quando vengono archiviati oggetti come script ETL e posizioni dei server notebook.<br><br>Convenzione per la denominazione: concede l'autorizzazione alle cartelle o ai bucket Amazon S3 i cui nomi hanno il prefisso aws-glue-.<br><br>Consente ad AWS Glue di creare i bucket che bloccano l'accesso pubblico. |
| `"logs:GetLogEvents"` | `"arn:aws:logs:*:*: /aws-glue/*"` | Consente il recupero dei log. CloudWatch<br><br>Convenzione di denominazione: AWS Glue scrive i log in gruppi di log i cui nomi iniziano con aws-glue-. |

| Azione | Risorsa | Descrizione |
| --- | --- | --- |
| `"cloudformation:CreateStack", "cloudformation:DeleteStack"` | `"arn:aws:cloudformation:*:*:stack/aws-glue*/*"` | Consente la gestione degli CloudFormation stack quando si lavora con server notebook.<br><br>Convenzione di denominazione: AWS Glue crea stack i cui nomi iniziano con aws-glue. |
| `"ec2:RunInstances"` | `"arn:aws:ec2:*:*:instance/*", "arn:aws:ec2:*:*:key-pair/*", "arn:aws:ec2:*:*:image/*", "arn:aws:ec2:*:*:security-group/*", "arn:aws:ec2:*:*:network-interface/*", "arn:aws:ec2:*:*:subnet/*", "arn:aws:ec2:*:*:volume/*"` | Consente l'esecuzione di endpoint di sviluppo e server notebook. |
| `"iam:PassRole"` | `"arn:aws:iam::*:role/ AWSGlueServiceRole*"` | Permette a AWS Glue di usare l'autorizzazione `PassRole` per i ruoli che iniziano con `AWSGlueServiceRole` . |

| Azione | Risorsa | Descrizione |
| --- | --- | --- |
| `"iam:PassRole"` | `"arn:aws:iam::*:role/ AWSGlueServiceNotebookRole*"` | Permette ad Amazon EC2 di usare l'autorizzazione `PassRole` per i ruoli che iniziano con `AWSGlueServiceNotebookRole` . |
| `"iam:PassRole"` | `"arn:aws:iam::*:role/service-role/AWSGlueServiceRole*"` | Permette a AWS Glue di usare l'autorizzazione `PassRole` per i ruoli che iniziano con `service-role/AWSGlueServiceRole` . |

7. Nella schermata Revisione della policy, inserisci un nome per la policy, ad esempio GlueConsoleAccessPolicy. Al termine, scegliere Create policy (Crea policy). Assicurati che non siano presenti errori in una casella rossa nella parte superiore dello schermo. Correggi gli eventuali errori segnalati.

> **Note**
>
> Se Use autoformatting (Usa formattazione automatica) è selezionato, la policy viene riformattata ogni volta che pari una policy oppure scegli Validate Policy (Convalida policy).

Per allegare la politica AWSGlue ConsoleFullAccess gestita

Puoi collegare la policy `AWSGlueConsoleFullAccess` per fornire le autorizzazioni necessarie all'utente della console AWS Glue.

> **Note**
>
> Puoi ignorare questa fase se hai creato una policy personalizzata per l'accesso alla console AWS Glue.

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, scegli Policy.
3. Nell'elenco delle politiche, seleziona la casella di controllo accanto a AWSGlueConsoleFullAccess. Puoi utilizzare il menu Filtro e la casella di ricerca per filtrare l'elenco di policy.
4. Scegli Operazioni di policy, quindi Collega.
5. Sceglie l'utente a cui collegare la policy. Puoi usare il menu Filtro e la casella di ricerca per filtrare l'elenco delle entità principali. Dopo aver scelto l'utente a cui collegare la policy, scegli Attach policy (Collega policy).

Per collegare la policy gestita **`AWSGlueConsoleSageMakerNotebookFullAccess`**.

Puoi allegare la `AWSGlueConsoleSageMakerNotebookFullAccess` policy a un utente per gestire i notebook SageMaker AI creati sulla console. AWS Glue Oltre alle altre autorizzazioni richieste per la AWS Glue console, questa politica consente l'accesso alle risorse necessarie per gestire i notebook AI. SageMaker

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo. https://console.aws.amazon.com/iam/
2. Nel riquadro di navigazione, scegli Policy.
3. Nell'elenco delle politiche, seleziona la casella di controllo accanto a AWSGlueConsoleSageMakerNotebookFullAccess. Puoi utilizzare il menu Filtro e la casella di ricerca per filtrare l'elenco di policy.
4. Scegli Operazioni di policy, quindi Collega.
5. Sceglie l'utente a cui collegare la policy. Puoi usare il menu Filtro e la casella di ricerca per filtrare l'elenco delle entità principali. Dopo aver scelto l'utente a cui collegare la policy, scegli Attach policy (Collega policy).

Per allegare la politica CloudWatchLogsReadOnlyAccess gestita

È possibile allegare la CloudWatchLogsReadOnlyAccesspolicy a un utente per visualizzare i log creati da AWS Glue nella console CloudWatch Logs.

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo. https://console.aws.amazon.com/iam/
2. Nel riquadro di navigazione, scegli Policy.
3. Nell'elenco delle politiche, seleziona la casella di controllo accanto a CloudWatchLogsReadOnlyAccess. Puoi utilizzare il menu Filtro e la casella di ricerca per filtrare l'elenco di policy.
4. Scegli Operazioni di policy, quindi Collega.
5. Sceglie l'utente a cui collegare la policy. Puoi usare il menu Filtro e la casella di ricerca per filtrare l'elenco delle entità principali. Dopo aver scelto l'utente a cui collegare la policy, scegli Attach policy (Collega policy).

Per allegare la politica AWSCloud FormationReadOnlyAccess gestita

È possibile allegare la AWSCloudFormationReadOnlyAccesspolicy a un utente per visualizzare gli CloudFormation stack utilizzati AWS Glue sulla CloudFormation console.

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, scegli Policy.
3. Nell'elenco delle politiche, seleziona la casella di controllo accanto a AWSCloudFormationReadOnlyAccess. Puoi utilizzare il menu Filtro e la casella di ricerca per filtrare l'elenco di policy.
4. Scegli Operazioni di policy, quindi Collega.
5. Sceglie l'utente a cui collegare la policy. Puoi usare il menu Filtro e la casella di ricerca per filtrare l'elenco delle entità principali. Dopo aver scelto l'utente a cui collegare la policy, scegli Attach policy (Collega policy).

Per allegare la politica AmazonAthenaFullAccess gestita

Puoi allegare la AmazonAthenaFullAccesspolicy a un utente per visualizzare i dati di Amazon S3 nella console Athena.

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo. https://console.aws.amazon.com/iam/
2. Nel riquadro di navigazione, scegli Policy.
3. Nell'elenco delle politiche, seleziona la casella di controllo accanto a AmazonAthenaFullAccess. Puoi utilizzare il menu Filtro e la casella di ricerca per filtrare l'elenco di policy.
4. Scegli Operazioni di policy, quindi Collega.
5. Sceglie l'utente a cui collegare la policy. Puoi usare il menu Filtro e la casella di ricerca per filtrare l'elenco delle entità principali. Dopo aver scelto l'utente a cui collegare la policy, scegli Attach policy (Collega policy).

## Fase 4: creare una policy IAM per i server notebook

Se prevedi di utilizzare i notebook con gli endpoint di sviluppo, devi specificare le autorizzazioni quando crei il server notebook. È possibile fornire queste autorizzazioni usando AWS Identity and Access Management (IAM).

Questa policy concede l'autorizzazione per alcune operazioni Amazon S3 per gestire le risorse nell'account richieste da AWS Glue quando assume il ruolo usando la policy. Alcune delle risorse specificate nella policy fanno riferimento a nomi predefiniti usati da AWS Glue per i bucket Amazon S3, gli script ETL Amazon S3 e le risorse Amazon EC2. Per semplicità, AWS Glue scrive alcuni oggetti Amazon S3 nei bucket nell'account con il prefisso `aws-glue-*` per impostazione predefinita.

> **Note**
>
> È possibile saltare questo passaggio se si utilizza la policy AWS **`AWSGlueServiceNotebookRole`**gestita.

In questa fase, crei una policy simile a `AWSGlueServiceNotebookRole`. Puoi trovare la versione più recente di `AWSGlueServiceNotebookRole` nella console IAM.

Per creare una policy IAM per i notebook

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione sinistro, scegli Policy.

3. Scegli Crea policy.
4. Nella schermata Create Policy (Crea policy), passa alla scheda per modificare JSON. Crea un documento di policy con le seguenti istruzioni JSON, quindi scegli Review policy (Verifica policy).

   JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:CreateDatabase",
        "glue:CreatePartition",
        "glue:CreateTable",
        "glue:DeleteDatabase",
        "glue:DeletePartition",
        "glue:DeleteTable",
        "glue:GetDatabase",
        "glue:GetDatabases",
        "glue:GetPartition",
        "glue:GetPartitions",
        "glue:GetTable",
        "glue:GetTableVersions",
        "glue:GetTables",
        "glue:UpdateDatabase",
        "glue:UpdatePartition",
        "glue:UpdateTable",
        "glue:GetJobBookmark",
        "glue:ResetJobBookmark",
        "glue:CreateConnection",
        "glue:CreateJob",
        "glue:DeleteConnection",
        "glue:DeleteJob",
        "glue:GetConnection",
        "glue:GetConnections",
        "glue:GetDevEndpoint",
        "glue:GetDevEndpoints",
        "glue:GetJob",
        "glue:GetJobs",
        "glue:UpdateJob",
        "glue:BatchDeleteConnection",
```

```
        "glue:UpdateConnection",
        "glue:GetUserDefinedFunction",
        "glue:UpdateUserDefinedFunction",
        "glue:GetUserDefinedFunctions",
        "glue:DeleteUserDefinedFunction",
        "glue:CreateUserDefinedFunction",
        "glue:BatchGetPartition",
        "glue:BatchDeletePartition",
        "glue:BatchCreatePartition",
        "glue:BatchDeleteTable",
        "glue:UpdateDevEndpoint",
        "s3:GetBucketLocation",
        "s3:ListBucket",
        "s3:ListAllMyBuckets",
        "s3:GetBucketAcl"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::crawler-public*",
        "arn:aws:s3:::aws-glue*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:DeleteObject"
      ],
      "Resource": [
        "arn:aws:s3:::aws-glue*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:CreateTags",
```

```
        "ec2:DeleteTags"
      ],
      "Condition": {
        "ForAllValues:StringEquals": {
          "aws:TagKeys": [
            "aws-glue-service-resource"
          ]
        }
      },
      "Resource": [
        "arn:aws:ec2:*:*:network-interface/*",
        "arn:aws:ec2:*:*:security-group/*",
        "arn:aws:ec2:*:*:instance/*"
      ]
    }
  ]
}
```

La tabella seguente descrive le autorizzazioni concesse dalla policy.

| Azione | Risorsa | Descrizione |
| --- | --- | --- |
| `"glue:*"` | `"*"` | Concede l'autorizzazione per eseguire tutte le operazioni API AWS Glue. |
| `"s3:GetBucketLocation", "s3:ListBucket", "s3:ListAllMyBuckets", "s3:GetBucketAcl"` | `"*"` | Permette di elencare i bucket Amazon S3 dai server notebook. |

| Azione | Risorsa | Descrizione |
| --- | --- | --- |
| `"s3:GetObject"` | `"arn:aws:s3:::crawler-public*", "arn:aws:s3:::aws-glue-*"` | Permette di ottenere gli oggetti Amazon S3 usati da esempi e tutorial dai notebook.<br><br>Convenzione per la denominazione: i nomi di bucket Amazon S3 iniziano con crawler-public e aws-glue-. |
| `"s3:PutObject", "s3:DeleteObject"` | `"arn:aws:s3:::aws-glue*"` | Permette di inserire ed eliminare oggetti Amazon S3 nell'account dai notebook.<br><br>Convenzione per la denominazione: utilizza cartelle Amazon S3 denominate aws-glue. |
| `"ec2:CreateTags", "ec2:DeleteTags"` | `"arn:aws:ec2:*:*:network-interface/*", "arn:aws:ec2:*:*:security-group/*", "arn:aws:ec2:*:*:instance/*"` | Consente il tagging delle risorse Amazon EC2 create per i server notebook.<br><br>Convenzione di denominazione: contrassegna AWS Glue le istanze Amazon EC2 con. aws-glue-service-resource |

5. Nella schermata Verifica policy, inserisci il Nome policy, ad esempio GlueServiceNotebookPolicyDefault. Digita una descrizione facoltativa e, al termine, seleziona Create policy (Crea policy).

## Fase 5: creare un ruolo IAM per i server notebook

Se prevedi di utilizzare i notebook con gli endpoint di sviluppo, devi concedere le autorizzazioni per il ruolo IAM. Fornisci tali autorizzazioni utilizzando AWS Identity and Access Management IAM, tramite un ruolo IAM.

> **Note**
>
> Quando crei un ruolo IAM utilizzando la console IAM, la console crea automaticamente un profilo dell'istanza e le assegna lo stesso nome del ruolo a cui corrisponde.

Per creare un ruolo IAM per i notebook

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione a sinistra seleziona Ruoli.
3. Scegli Crea ruolo.
4. Per tipo di ruolo, scegli AWS Service (Servizio AWS ), trova e seleziona EC2, il caso d'uso EC2 e quindi Next: Permissions (Successivo: autorizzazioni).
5. Nella pagina Allega criteri di autorizzazione, scegli le politiche che contengono le autorizzazioni richieste; ad esempio, AWSGlueServiceNotebookRoleper le autorizzazioni generali AWS Glue e la politica AWS gestita AmazonS3 per l'accesso FullAccess alle risorse Amazon S3. Scegli quindi Next: Review (Avanti: Verifica).

   > **Note**
   >
   > Verifica che una delle policy in questo ruolo conceda le autorizzazioni per le origini e le destinazioni Amazon S3. Conferma che le tue policy concedano l'accesso completo al percorso in cui archivi i notebook al momento della creazione di un server notebook. Puoi fornire una policy personalizzata per l'accesso a risorse Amazon S3 specifiche. Per ulteriori informazioni sulla creazione di una policy Amazon S3 per le risorse, vedi Specificare le risorse in una policy.
   > Se prevedi di accedere a origini e destinazioni Amazon S3 crittografate con SSE-KMS, collega una policy che permetta ai notebook di decrittografare i dati. Per ulteriori informazioni, consulta Protezione dei dati utilizzando la crittografia lato server con chiavi gestite (SSE-KMS). AWS KMS

Di seguito è riportato un esempio di :

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:*:111122223333:key/key-id"
      ]
    }
  ]
}
```

6. In Nome ruolo, immetti un nome per il ruolo. Crea il ruolo usando come prefisso del nome la stringa `AWSGlueServiceNotebookRole` per permettere il passaggio del ruolo dagli utenti della console al server notebook. Le policy fornite da AWS Glue prevedono ruoli di servizio IAM che iniziano con `AWSGlueServiceNotebookRole`. In caso contrario, è necessario aggiungere una policy per concedere agli utenti l'autorizzazione `iam:PassRole` per i ruoli IAM in modo da soddisfare la convenzione per la denominazione. Ad esempio, specifica `AWSGlueServiceNotebookRoleDefault`. Quindi seleziona Create role (Crea ruolo).

## Fase 6: Creare una policy IAM per i notebook SageMaker AI

Se prevedi di utilizzare notebook SageMaker AI con endpoint di sviluppo, devi specificare le autorizzazioni quando crei il notebook. È possibile fornire queste autorizzazioni usando AWS Identity and Access Management (IAM).

Per creare una policy IAM per i notebook AI SageMaker

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo. https://console.aws.amazon.com/iam/
2. Nel riquadro di navigazione sinistro, scegli Policy.

3. Scegli Crea policy.
4. Nella pagina Create Policy (Crea policy), passa alla scheda per modificare JSON. Crea il tuo documento di policy con le istruzioni JSON seguenti. Modifica *bucket-name* e *region-code account-id* per il tuo ambiente.

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Action": [
                "s3:ListBucket"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket"
            ]
        },
        {
            "Action": [
                "s3:GetObject"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:s3:::amzn-s3-demo-bucket*"
            ]
        },
        {
            "Action": [
                "logs:CreateLogStream",
                "logs:DescribeLogStreams",
                "logs:PutLogEvents",
                "logs:CreateLogGroup"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:logs:us-east-1:111122223333:log-group:/aws/
sagemaker/*",
                "arn:aws:logs:us-east-1:111122223333:log-group:/aws/
sagemaker/*:log-stream:aws-glue-*"
            ]
```

```
        },
        {
            "Action": [
                "glue:UpdateDevEndpoint",
                "glue:GetDevEndpoint",
                "glue:GetDevEndpoints"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:glue:us-east-1:111122223333:devEndpoint/*"
            ]
        },
        {
            "Action": [
                "sagemaker:ListTags"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:sagemaker:us-east-1:111122223333:notebook-instance/
*"
            ]
        }
    ]
}
```

Selezionare Review policy (Esamina policy).

La tabella seguente descrive le autorizzazioni concesse dalla policy.

| Azione | Risorsa | Descrizione |
| --- | --- | --- |
| `"s3:ListBucket*"` | `"arn:aws:s3::: bucket-name "` | Concede l'autorizzazione per elencare i bucket Amazon S3. |
| `"s3:GetObject"` | `"arn:aws:s3::: bucket-name *"` | Concede l'autorizzazione per ottenere oggetti Amazon S3 utilizzati SageMaker dai notebook AI. |

| Azione | Risorsa | Descrizione |
| --- | --- | --- |
| `"logs:CreateLogStream", "logs:DescribeLogStreams", "logs:PutLogEvents", "logs:CreateLogGroup"` | `"arn:aws:logs:region-code:account-id:log-group:/aws/sagemaker/*", "arn:aws:logs:region-code:account-id:log-group:/aws/sagemaker/*:log-stream:aws-glue-*"` | Concede l'autorizzazione a scrivere log su Amazon CloudWatch Logs dai notebook.<br><br>Convenzione per la denominazione: scrive per registrare i gruppi i cui nomi iniziano con aws-glue. |
| `"glue:UpdateDevEndpoint", "glue:GetDevEndpoint", "glue:GetDevEndpoints"` | `"arn:aws:glue:region-code:account-id:devEndpoint/*"` | Concede l'autorizzazione a utilizzare un endpoint di sviluppo da notebook AI. SageMaker |
| `"sagemaker:ListTags"` | `"arn:aws:sagemaker:region-code:account-id:notebook-instance/*"` | Concede l'autorizzazione a restituire tag per una risorsa AI. SageMaker Il `aws-glue-dev-endpoint` tag è necessario sul notebook SageMaker AI per collegare il notebook a un endpoint di sviluppo. |

5. Nella schermata Verifica policy, inserisci il Nome policy, ad esempio `AWSGlueSageMakerNotebook`. Digita una descrizione facoltativa e, al termine, seleziona Create policy (Crea policy).

## Fase 7: Creare un ruolo IAM per i notebook SageMaker AI

Se prevedi di utilizzare notebook SageMaker AI con endpoint di sviluppo, devi concedere le autorizzazioni per il ruolo IAM. Fornisci tali autorizzazioni utilizzando AWS Identity and Access Management (IAM), tramite un ruolo IAM.

Per creare un ruolo IAM per i notebook SageMaker AI

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo. https://console.aws.amazon.com/iam/
2. Nel pannello di navigazione a sinistra seleziona Ruoli.
3. Scegli Crea ruolo.
4. Per il tipo di ruolo, scegli AWS Servizio, trova e scegli SageMaker, quindi scegli lo use case SageMaker - Execution. Quindi scegliere Next: Permissions (Successivo: Autorizzazioni).
5. Nella pagina Allega criteri di autorizzazione, scegli le politiche che contengono le autorizzazioni richieste, ad esempio. AmazonSageMakerFullAccess Scegli Prossimo: Rivedi.

   Se prevedi di accedere a origini e destinazioni Amazon S3 crittografate con SSE-KMS, collega una policy che permetta ai notebook di decrittografare i dati, come mostrato nell'esempio seguente. Per ulteriori informazioni, vedere Protezione dei dati mediante la crittografia lato server con AWS KMS-Managed Keys (SSE-KMS).

   JSON

   ```
   {
     "Version":"2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Action": [
           "kms:Decrypt"
         ],
         "Resource": [
           "arn:aws:kms:*:111122223333:key/key-id"
         ]
       }
     ]
   }
   ```

6. In Nome ruolo, immetti un nome per il ruolo. Per consentire il trasferimento del ruolo dagli utenti della console all' SageMaker intelligenza artificiale, usa un nome con il prefisso della stringa. `AWSGlueServiceSageMakerNotebookRole` AWS Gluea condizione che le politiche prevedano che i ruoli IAM comincino con`AWSGlueServiceSageMakerNotebookRole`. In caso contrario, è necessario aggiungere una policy per concedere agli utenti l'autorizzazione `iam:PassRole` per i ruoli IAM in modo da soddisfare la convenzione per la denominazione.

Per esempio, inserisci `AWSGlueServiceSageMakerNotebookRole-Default` e quindi seleziona Create role (Crea ruolo).

7. Dopo aver creato il ruolo, allega la policy che consente le autorizzazioni aggiuntive necessarie per creare notebook SageMaker AI. AWS Glue

   Apto il ruolo appena creato, `AWSGlueServiceSageMakerNotebookRole-Default` e scegli Attach policies (Collega policy). Collega la policy creata denominata `AWSGlueSageMakerNotebook` al ruolo.

## AWS Esempi di politiche di controllo degli accessi di Glue

Questa sezione contiene esempi di policy di controllo degli accessi basate sull'identità (IAM) e sulle risorse di AWS Glue.

Indice


- Esempi di policy basate sull'identità per Glue AWS
  - Best practice per le policy
  - Le autorizzazioni a livello di risorsa si applicano solo agli specifici oggetti di AWS Glue
  - Utilizzo della console AWS Glue
  - Consentire agli utenti di visualizzare le loro autorizzazioni
  - Concessione di autorizzazioni di sola lettura a una tabella
  - Filtra le tabelle per GetTables autorizzazione
  - Concessione di accesso completo a una tabella e a tutte le partizioni
  - Controllo degli accessi per nome prefisso e diniego esplicito
  - Autorizzazione dell'accesso utilizzando tag
  - Negazione dell'accesso utilizzando tag
  - Utilizzo di tag con operazioni API in elenco e in batch
  - Controllo delle impostazioni utilizzando le chiavi di condizione o di contesto
    - Policy di controllo che controllano le impostazioni utilizzando le chiavi di condizione
    - Policy di controllo che controllano le impostazioni utilizzando le chiavi di contesto
  - Negare a un'identità la possibilità di creare sessioni di anteprima dei dati
- Esempi di policy basate sulle risorse per Glue AWS

- Considerazioni sull'utilizzo di politiche basate sulle risorse con Glue AWS
- Utilizza una policy della risorsa per controllare gli accessi nello stesso account

## Esempi di policy basate sull'identità per Glue AWS

Per impostazione predefinita, gli utenti e i ruoli non sono autorizzati a creare o modificare le risorse AWS Glue. Per concedere agli utenti l'autorizzazione a eseguire azioni sulle risorse di cui hanno bisogno, un amministratore IAM può creare policy IAM.

Per informazioni su come creare una policy basata su identità IAM utilizzando questi documenti di policy JSON di esempio, consulta Creazione di policy IAM (console) nella Guida per l'utente di IAM.

Per i dettagli sulle azioni e sui tipi di risorse definiti da AWS Glue, incluso il formato di ARNs per ogni tipo di risorsa, vedere Azioni, risorse e chiavi di condizione per AWS Glue nel riferimento di autorizzazione del servizio.

> **Note**
>
> Gli esempi forniti in questa sezione utilizzano tutti la Regione `us-west-2`. Puoi sostituirlo con la AWS regione che desideri utilizzare.

Argomenti

- Best practice per le policy
- Le autorizzazioni a livello di risorsa si applicano solo agli specifici oggetti di AWS Glue
- Utilizzo della console AWS Glue
- Consentire agli utenti di visualizzare le loro autorizzazioni
- Concessione di autorizzazioni di sola lettura a una tabella
- Filtra le tabelle per GetTables autorizzazione
- Concessione di accesso completo a una tabella e a tutte le partizioni
- Controllo degli accessi per nome prefisso e diniego esplicito
- Autorizzazione dell'accesso utilizzando tag
- Negazione dell'accesso utilizzando tag
- Utilizzo di tag con operazioni API in elenco e in batch
- Controllo delle impostazioni utilizzando le chiavi di condizione o di contesto

- Negare a un'identità la possibilità di creare sessioni di anteprima dei dati

Best practice per le policy

Le politiche basate sull'identità determinano se qualcuno può creare, accedere o eliminare le risorse AWS Glue nel tuo account. Queste azioni possono comportare costi aggiuntivi per l' Account AWS. Quando si creano o modificano policy basate sull'identità, seguire queste linee guida e raccomandazioni:

- Inizia con le policy AWS gestite e passa alle autorizzazioni con privilegi minimi: per iniziare a concedere autorizzazioni a utenti e carichi di lavoro, utilizza le politiche gestite che concedono le autorizzazioni per molti casi d'uso comuni.AWS  Sono disponibili nel tuo. Account AWS Ti consigliamo di ridurre ulteriormente le autorizzazioni definendo politiche gestite dai AWS clienti specifiche per i tuoi casi d'uso. Per maggiori informazioni, consulta Policy gestite da AWS o Policy gestite da AWS per le funzioni dei processi nella Guida per l'utente di IAM.
- Applicazione delle autorizzazioni con privilegio minimo - Quando si impostano le autorizzazioni con le policy IAM, concedere solo le autorizzazioni richieste per eseguire un'attività. È possibile farlo definendo le azioni che possono essere intraprese su risorse specifiche in condizioni specifiche, note anche come autorizzazioni con privilegio minimo. Per maggiori informazioni sull'utilizzo di IAM per applicare le autorizzazioni, consulta Policy e autorizzazioni in IAM nella Guida per l'utente di IAM.
- Condizioni d'uso nelle policy IAM per limitare ulteriormente l'accesso - Per limitare l'accesso ad azioni e risorse è possibile aggiungere una condizione alle policy. Ad esempio, è possibile scrivere una condizione di policy per specificare che tutte le richieste devono essere inviate utilizzando SSL. Puoi anche utilizzare le condizioni per concedere l'accesso alle azioni del servizio se vengono utilizzate tramite uno specifico Servizio AWS, ad esempio CloudFormation. Per maggiori informazioni, consultare la sezione Elementi delle policy JSON di IAM: condizione nella Guida per l'utente di IAM.
- Utilizzo dello strumento di analisi degli accessi IAM per convalidare le policy IAM e garantire autorizzazioni sicure e funzionali - Lo strumento di analisi degli accessi IAM convalida le policy nuove ed esistenti in modo che aderiscano al linguaggio (JSON) della policy IAM e alle best practice di IAM. Lo strumento di analisi degli accessi IAM offre oltre 100 controlli delle policy e consigli utili per creare policy sicure e funzionali. Per maggiori informazioni, consultare Convalida delle policy per il Sistema di analisi degli accessi IAM nella Guida per l'utente di IAM.
- Richiedi l'autenticazione a più fattori (MFA): se hai uno scenario che richiede utenti IAM o un utente root nel Account AWS tuo, attiva l'MFA per una maggiore sicurezza. Per richiedere la MFA quando

vengono chiamate le operazioni API, aggiungere le condizioni MFA alle policy. Per maggiori informazioni, consultare Protezione dell'accesso API con MFA nella Guida per l'utente di IAM.

Per maggiori informazioni sulle best practice in IAM, consulta Best practice di sicurezza in IAM nella Guida per l'utente di IAM.

Le autorizzazioni a livello di risorsa si applicano solo agli specifici oggetti di AWS Glue

È possibile definire il controllo granulare solo per specifici oggetti di AWS Glue. Pertanto, devi scrivere la policy IAM del tuo cliente in modo che le operazioni API che consentono Amazon Resource Names (ARNs) per l'`Resource`istruzione non vengano mescolate con le operazioni API che non lo consentono ARNs.

Ad esempio, la seguente policy di IAM consente operazioni delle API per `GetClassifier` e `GetJobRun`. Definisce `Resource` come * perché AWS Glue non consente ARNs classificatori ed esecuzioni di job. Perché ARNs sono consentiti per operazioni API specifiche come `GetDatabase` e`GetTable`, ARNs possono essere specificati nella seconda metà della policy.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:GetClassifier*",
        "glue:GetJobRun*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "glue:Get*"
      ],
      "Resource": [
        "arn:aws:glue:us-east-1:111122223333:catalog",
        "arn:aws:glue:us-east-1:111122223333:database/default",
        "arn:aws:glue:us-east-1:111122223333:table/default/e*1*",
        "arn:aws:glue:us-east-1:111122223333:connection/connection2"
```

```
      ]
    }
  ]
}
```

Per un elenco di AWS Glue oggetti che lo consentono ARNs, vedereAWS GlueSpecificare la risorsa ARNs.

Utilizzo della console AWS Glue

Per accedere alla console AWS Glue, devi disporre di un set minimo di autorizzazioni. Queste autorizzazioni devono consentirti di elencare e visualizzare i dettagli sulle risorse AWS Glue presenti nel tuo Account AWS. Se crei una policy basata sull'identità più restrittiva rispetto alle autorizzazioni minime richieste, la console non funzionerà nel modo previsto per le entità (utenti o ruoli) associate a tale policy.

Non è necessario concedere autorizzazioni minime per la console agli utenti che effettuano chiamate solo verso AWS CLI o l' AWS API. Al contrario, è opportuno concedere l'accesso solo alle azioni che corrispondono all'operazione API che stanno cercando di eseguire.

Per garantire che utenti e ruoli possano ancora utilizzare la console AWS Glue, collega anche AWS Glue `ConsoleAccess` o la policy `ReadOnly` AWS gestita alle entità. Per maggiori informazioni, consulta Aggiunta di autorizzazioni a un utente nella Guida per l'utente di IAM.

Affinché un utente possa lavorare con la AWS Glue console, deve disporre di un set minimo di autorizzazioni che gli consentano di utilizzare le AWS Glue risorse del proprio AWS account. Oltre a queste autorizzazioni AWS Glue, la console richiede le autorizzazioni dei servizi seguenti:

- Autorizzazioni Amazon CloudWatch Logs per visualizzare i log.
- AWS Identity and Access Management (IAM) autorizzazioni per elencare e trasferire ruoli.
- AWS CloudFormation autorizzazioni per lavorare con gli stack.
- Autorizzazioni Amazon Elastic Compute Cloud (Amazon EC2) per VPCs elenchi, sottoreti, gruppi di sicurezza, istanze e altri oggetti.
- Autorizzazioni Amazon Simple Storage Service (Amazon S3) per elencare bucket e oggetti e per recuperare e salvare script.
- Autorizzazioni Amazon Redshift necessarie per l'utilizzo dei cluster.
- Autorizzazioni Amazon Relational Database Service (Amazon RDS) per elencare le istanze.

Per ulteriori informazioni sulle autorizzazioni necessarie agli utenti per visualizzare e usare la console di AWS Glue, consultare Fase 3: Collegamento di una policy agli utenti o ai gruppi che accedono a AWS Glue.

Se decidi di creare una policy IAM più restrittiva delle autorizzazioni minime richieste, la console non funzionerà come previsto per gli utenti con tale policy IAM. Per garantire che gli utenti possano continuare a usare la console AWS Glue, collega anche la policy gestita `AWSGlueConsoleFullAccess`, come descritto in AWS politiche gestite (predefinite) per AWS Glue.

Consentire agli utenti di visualizzare le loro autorizzazioni

Questo esempio mostra in che modo è possibile creare una policy che consente agli utenti IAM di visualizzare le policy inline e gestite che sono collegate alla relativa identità utente. Questa politica include le autorizzazioni per completare questa azione sulla console o utilizzando l'API o in modo programmatico. AWS CLI AWS

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
```

```
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Concessione di autorizzazioni di sola lettura a una tabella

La policy seguente consente di concedere autorizzazioni di sola lettura per una tabella `books` nel database `db1`. Per ulteriori informazioni sulla risorsa Amazon Resource Names (ARNs), consultaCatalogo dati ARNs.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "GetTablesActionOnBooks",
      "Effect": "Allow",
      "Action": [
        "glue:GetTables",
        "glue:GetTable"
      ],
      "Resource": [
        "arn:aws:glue:us-east-1:111122223333:catalog",
        "arn:aws:glue:us-east-1:111122223333:database/db1",
        "arn:aws:glue:us-east-1:111122223333:table/db1/books"
      ]
    }
  ]
}
```

Questa policy consente di concedere autorizzazioni di sola lettura per una tabella `books` nel database denominato `db1`. Per concedere l'autorizzazione `Get` a una tabella è richiesta anche l'autorizzazione alle risorse del database e del catalogo.

La policy seguente concede il livello minimo di autorizzazioni necessarie per creare una tabella `tb1` nel database `db1`:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:CreateTable"
      ],
      "Resource": [
        "arn:aws:glue:us-east-1:111122223333:table/db1/tbl1",
        "arn:aws:glue:us-east-1:111122223333:database/db1",
        "arn:aws:glue:us-east-1:111122223333:catalog"
      ]
    }
  ]
}
```

Filtra le tabelle per GetTables autorizzazione

Supponiamo che ci siano tre tabelle (`customers`, `stores` e `store_sales`) nel database `db1`. La policy seguente concede l'autorizzazione `GetTables` a `stores` e `store_sales`, ma non a `customers`. Quando chiami `GetTables` con questa policy, il risultato contiene solo le due tabelle autorizzate (la tabella `customers` non viene restituita).

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "GetTablesExample",
      "Effect": "Allow",
      "Action": [
        "glue:GetTables"
      ],
      "Resource": [
        "arn:aws:glue:us-east-1:111122223333:catalog",
        "arn:aws:glue:us-east-1:111122223333:database/db1",
```

```
        "arn:aws:glue:us-east-1:111122223333:table/db1/store_sales",
        "arn:aws:glue:us-east-1:111122223333:table/db1/stores"
      ]
    }
  ]
}
```

È possibile semplificare la policy precedente utilizzando `store*` per includere qualsiasi nome di tabella che inizi con `store`:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "GetTablesExample2",
      "Effect": "Allow",
      "Action": [
        "glue:GetTables"
      ],
      "Resource": [
        "arn:aws:glue:us-east-1:111122223333:catalog",
        "arn:aws:glue:us-east-1:111122223333:database/db1",
        "arn:aws:glue:us-east-1:111122223333:table/db1/store*"
      ]
    }
  ]
}
```

Analogamente, utilizzando `/db1/*` per includere tutte le tabelle incluse nella cartella `db1`, la policy seguente concede l'autorizzazione `GetTables` a tutte le tabelle presenti in `db1`.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
```

```
      "Sid": "GetTablesReturnAll",
      "Effect": "Allow",
      "Action": [
        "glue:GetTables"
      ],
      "Resource": [
        "arn:aws:glue:us-east-1:111122223333:catalog",
        "arn:aws:glue:us-east-1:111122223333:database/db1",
        "arn:aws:glue:us-east-1:111122223333:table/db1/*"
      ]
    }
  ]
}
```

Se non viene fornito nessun ARN di tabella, una chiamata a `GetTables` si conclude correttamente ma restituisce un elenco vuoto:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "GetTablesEmptyResults",
      "Effect": "Allow",
      "Action": [
        "glue:GetTables"
      ],
      "Resource": [
        "arn:aws:glue:us-east-1:111122223333:catalog",
        "arn:aws:glue:us-east-1:111122223333:database/db1"
      ]
    }
  ]
}
```

Se la policy non contiene l'ARN del database, una chiamata a `GetTables` ha esito negativo e restituisce `AccessDeniedException`:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "GetTablesAccessDeny",
      "Effect": "Allow",
      "Action": [
        "glue:GetTables"
      ],
      "Resource": [
        "arn:aws:glue:us-east-1:111122223333:catalog",
        "arn:aws:glue:us-east-1:111122223333:table/db1/*"
      ]
    }
  ]
}
```

Concessione di accesso completo a una tabella e a tutte le partizioni

La policy seguente concede tutte le autorizzazioni su una tabella denominata `books` nel database db1. Questo include le autorizzazioni di lettura e scrittura sulla tabella stessa, sulle versioni archiviate e su tutte le partizioni.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "FullAccessOnTable",
      "Effect": "Allow",
      "Action": [
        "glue:CreateTable",
        "glue:GetTable",
        "glue:GetTables",
        "glue:UpdateTable",
        "glue:DeleteTable",
        "glue:BatchDeleteTable",
```

```
        "glue:GetTableVersion",
        "glue:GetTableVersions",
        "glue:DeleteTableVersion",
        "glue:BatchDeleteTableVersion",
        "glue:CreatePartition",
        "glue:BatchCreatePartition",
        "glue:GetPartition",
        "glue:GetPartitions",
        "glue:BatchGetPartition",
        "glue:UpdatePartition",
        "glue:DeletePartition",
        "glue:BatchDeletePartition"
      ],
      "Resource": [
        "arn:aws:glue:us-east-1:111122223333:catalog",
        "arn:aws:glue:us-east-1:111122223333:database/db1",
        "arn:aws:glue:us-east-1:111122223333:table/db1/books"
      ]
    }
  ]
}
```

Nella pratica, la policy precedente può essere semplificata:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "FullAccessOnTable",
      "Effect": "Allow",
      "Action": [
        "glue:*Table*",
        "glue:*Partition*"
      ],
      "Resource": [
        "arn:aws:glue:us-east-1:111122223333:catalog",
        "arn:aws:glue:us-east-1:111122223333:database/db1",
        "arn:aws:glue:us-east-1:111122223333:table/db1/books"
      ]
    }
```

```
  ]
}
```

tieni presente che il livello minimo di granularità del controllo degli accessi è a livello di tabella. Questo significa che non è possibile concedere a un utente l'accesso ad alcune partizioni in una tabella, ma non ad altre, o ad alcune colonne ma non ad altre. Un utente ha accesso a tutte le parti di una tabella o a nessuna.

Controllo degli accessi per nome prefisso e diniego esplicito

In questo esempio, supponiamo che i database e le tabelle del tuo AWS Glue Data Catalog siano organizzati utilizzando prefissi di nome. I database nella fase di sviluppo hanno il nome prefisso `dev-` e quelli in produzione hanno il nome prefisso `prod-`. È possibile utilizzare la seguente politica per concedere agli sviluppatori l'accesso completo a tutti i database, le tabelle e così via che hanno il prefisso. UDFs `dev-` Tuttavia, puoi anche concedere l'accesso in sola lettura a tutti gli elementi con il prefisso `prod-`.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "DevAndProdFullAccess",
      "Effect": "Allow",
      "Action": [
        "glue:*Database*",
        "glue:*Table*",
        "glue:*Partition*",
        "glue:*UserDefinedFunction*",
        "glue:*Connection*"
      ],
      "Resource": [
        "arn:aws:glue:us-east-1:111122223333:catalog",
        "arn:aws:glue:us-east-1:111122223333:database/dev-*",
        "arn:aws:glue:us-east-1:111122223333:database/prod-*",
        "arn:aws:glue:us-east-1:111122223333:table/dev-*/*",
        "arn:aws:glue:us-east-1:111122223333:table/*/dev-*",
        "arn:aws:glue:us-east-1:111122223333:table/prod-*/*",
        "arn:aws:glue:us-east-1:111122223333:table/*/prod-*",
```

```
        "arn:aws:glue:us-east-1:111122223333:userDefinedFunction/dev-*/*",
        "arn:aws:glue:us-east-1:111122223333:userDefinedFunction/*/dev-*",
        "arn:aws:glue:us-east-1:111122223333:userDefinedFunction/prod-*/*",
        "arn:aws:glue:us-east-1:111122223333:userDefinedFunction/*/prod-*",
        "arn:aws:glue:us-east-1:111122223333:connection/dev-*",
        "arn:aws:glue:us-east-1:111122223333:connection/prod-*"
      ]
    },
    {
      "Sid": "ProdWriteDeny",
      "Effect": "Deny",
      "Action": [
        "glue:*Create*",
        "glue:*Update*",
        "glue:*Delete*"
      ],
      "Resource": [
        "arn:aws:glue:us-east-1:111122223333:database/prod-*",
        "arn:aws:glue:us-east-1:111122223333:table/prod-*/*",
        "arn:aws:glue:us-east-1:111122223333:table/*/prod-*",
        "arn:aws:glue:us-east-1:111122223333:userDefinedFunction/prod-*/*",
        "arn:aws:glue:us-east-1:111122223333:userDefinedFunction/*/prod-*",
        "arn:aws:glue:us-east-1:111122223333:connection/prod-*"
      ]
    }
  ]
}
```

La seconda istruzione nella policy precedente utilizza il codice esplicito `deny`. Puoi utilizzare il codice esplicito `deny` per sovrascrivere qualsiasi autorizzazione `allow` concessa al principale. Questo consente di bloccare l'accesso a risorse critiche e a impedire a un'altra policy di concedere accidentalmente l'accesso a esse.

Nell'esempio precedente, anche se la prima istruzione concede l'accesso completo alle risorse `prod-`, la seconda istruzione revoca esplicitamente l'accesso in scrittura, mantenendo solo l'accesso in lettura alle risorse `prod-`.

Autorizzazione dell'accesso utilizzando tag

Supporre ad esempio che si voglia limitare l'accesso al trigger `t2` a un utente specifico denominato `Tom` nel proprio account. Tutti gli altri utenti, tra cui `Sam`, hanno accesso al trigger `t1`. I trigger `t1` e `t2` hanno le seguenti proprietà.

```
aws glue get-triggers
{
    "Triggers": [
        {
            "State": "CREATED",
            "Type": "SCHEDULED",
            "Name": "t1",
            "Actions": [
                {
                    "JobName": "j1"
                }
            ],
            "Schedule": "cron(0 0/1 * * ? *)"
        },
        {
            "State": "CREATED",
            "Type": "SCHEDULED",
            "Name": "t2",
            "Actions": [
                {
                    "JobName": "j1"
                }
            ],
            "Schedule": "cron(0 0/1 * * ? *)"
        }
    ]
}
```

L'amministratore AWS Glue ha associato il valore di tag `Tom` (`aws:ResourceTag/Name": "Tom"`) al trigger `t2`. L'amministratore AWS Glue ha inoltre fornito a Tom una policy IAM con un'istruzione di condizione basata sul tag. Di conseguenza, Tom può utilizzare solo un'operazione AWS Glue che agisce sulle risorse con il valore di tag `Tom`.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "glue:*",
```

```
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Name": "Tom"
        }
      }
    }
  ]
}
```

Quando Tom cerca di accedere al trigger `t1` riceve un messaggio di accesso rifiutato. Allo stesso tempo, può recuperare regolarmente il trigger `t2`.

```
aws glue get-trigger --name t1

An error occurred (AccessDeniedException) when calling the GetTrigger operation:
 User: Tom is not authorized to perform: glue:GetTrigger on resource: arn:aws:glue:us-
east-1:123456789012:trigger/t1

aws glue get-trigger --name t2
{
    "Trigger": {
        "State": "CREATED",
        "Type": "SCHEDULED",
        "Name": "t2",
        "Actions": [
            {
                "JobName": "j1"
            }
        ],
        "Schedule": "cron(0 0/1 * * ? *)"
    }
}
```

Tom non può usare l'operazione dell'API `GetTriggers` plurale per elencare i trigger in quanto questa operazione non supporta il filtro sui tag.

Per concedere a Tom l'accesso a `GetTriggers`, l'amministratore di AWS Glue crea una policy che divide le autorizzazioni in due sezioni. Una sezione consente a Tom di accedere a tutti i trigger con l'operazione API `GetTriggers`. La seconda sezione consente a Tom di accedere alle operazioni

API che sono contrassegnate con il valore `Tom`. Con questa policy, a Tom è consentito l'accesso `GetTriggers` e `GetTrigger` al trigger `t2`.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "glue:GetTriggers",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "glue:*",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Name": "Tom"
        }
      }
    }
  ]
}
```

Negazione dell'accesso utilizzando tag

Un altro approccio per una policy delle risorse consiste nel negare in modo esplicito l'accesso alle risorse.

> **Important**
>
> Una policy di negazione esplicita non funziona per le operazioni dell'API plurali, come `GetTriggers`.

Nella seguente policy di esempio, sono consentite tutte le operazioni di processo AWS Glue Tuttavia, la seconda dichiarazione `Effect` nega esplicitamente l'accesso ai processi contrassegnati con la chiave `Team` e il valore `Special`.

Quando un amministratore collega le seguenti policy a un'identità, questa può accedere a tutti i processi tranne quelli contrassegnati con la chiave `Team` e il valore `Special`.

Utilizzo di tag con operazioni API in elenco e in batch

Un terzo approccio per scrivere una policy basata sulle risorse consiste nel consentire l'accesso alle risorse utilizzando un'operazione API `List` per elencare le risorse corrispondenti a un dato valore di tag. Quindi, si utilizza l'operazione API `Batch` corrispondente per consentire l'accesso ai dettagli delle risorse specifiche. Con questo approccio, l'amministratore non ha bisogno di consentire l'accesso alle operazioni API `GetCrawlers`, `GetDevEndpoints`, `GetJobs` o `GetTriggers` plurali. Puoi invece consentire la possibilità di elencare le risorse con le seguenti operazioni API:

- `ListCrawlers`
- `ListDevEndpoints`
- `ListJobs`
- `ListTriggers`

Puoi inoltre consentire la possibilità di ottenere i dettagli delle singole risorse con le seguenti operazioni API:

- `BatchGetCrawlers`
- `BatchGetDevEndpoints`
- `BatchGetJobs`
- `BatchGetTriggers`

In qualità di amministratore, per l'utilizzo di questo approccio, è possibile:

1. Aggiungere i tag a crawler, endpoint di sviluppo, processi e trigger.
2. Rifiutare l'accesso degli utenti alle operazioni API `Get`, ad esempio `GetCrawlers`, `GetDevEndponts`, `GetJobs` e `GetTriggers`.
3. Per permettere agli utenti di determinare a quali risorse con tag hanno accesso, consentire l'accesso degli utenti alle operazioni API `List`, ad esempio `ListCrawlers`, `ListDevEndponts`, `ListJobs` e `ListTriggers`.
4. Nega agli utenti l'accesso ai AWS Glue tag APIs, ad esempio e. `TagResource` `UntagResource`

5. Consentire l'accesso degli utenti ai dettagli delle risorse con le operazioni API `BatchGet`, ad esempio `BatchGetCrawlers`, `BatchGetDevEndponts`, `BatchGetJobs` e `BatchGetTriggers`.

Ad esempio, quando si richiama l'operazione `ListCrawlers`, fornire un valore di tag che corrisponda al nome dell'utente. Quindi il risultato è un elenco di crawler corrispondenti ai valori di tag forniti. Fornisci l'elenco dei nomi a `BatchGetCrawlers` per ottenere informazioni dettagliate su ogni crawler con il tag specificato.

Ad esempio, se Tom deve essere in grado di recuperare solo i dettagli dei trigger con il tag `Tom`, l'amministratore può aggiungere i tag ai trigger per `Tom`, rifiutare l'accesso all'operazione API `GetTriggers` a tutti gli utenti e consentire l'accesso di tutti gli utenti a `ListTriggers` e `BatchGetTriggers`.

Ecco la policy basata sulle risorse che l'amministratore di AWS Glue concede a Tom. Nella prima sezione della policy, le operazioni API AWS Glue sono rifiutate per `GetTriggers`. Nella seconda sezione della policy, `ListTriggers` è consentito per tutte le risorse. Tuttavia, nella terza sezione, tali risorse con il tag `Tom` possono eseguire l'accesso `BatchGetTriggers`.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "glue:GetTriggers",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "glue:ListTriggers"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
```

```
      "Action": [
        "glue:BatchGetTriggers"
      ],
      "Resource": [
        "*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Name": "Tom"
        }
      }
    }
  ]
}
```

Usando gli stessi trigger dell'esempio precedente, Tom può accedere al trigger `t2`, ma non al trigger `t1`. L'esempio seguente mostra i risultati quando Tom cerca di accedere a `t1` e `t2` con `BatchGetTriggers`.

```
aws glue batch-get-triggers --trigger-names t2
{
    "Triggers": {
        "State": "CREATED",
        "Type": "SCHEDULED",
        "Name": "t2",
        "Actions": [
            {
                "JobName": "j2"
            }
        ],
        "Schedule": "cron(0 0/1 * * ? *)"
    }
}

aws glue batch-get-triggers  --trigger-names t1

An error occurred (AccessDeniedException) when calling the BatchGetTriggers operation:
 No access to any requested resource.
```

L'esempio seguente mostra i risultati quando Tom cerca di accedere al trigger `t2` e `t3` (che non esiste) nella stessa chiamata `BatchGetTriggers`. Nota che poiché Tom ha accesso al trigger `t2`

esistente, viene restituito solo `t2`. Sebbene a Tom sia consentito accedere al trigger `t3`, il trigger `t3` non esiste e pertanto `t3` viene restituito nella risposta in un elenco di `"TriggersNotFound": []`.

```
aws glue batch-get-triggers --trigger-names t2 t3
{
    "Triggers": {
        "State": "CREATED",
        "Type": "SCHEDULED",
        "Name": "t2",
        "Actions": [
            {
                "JobName": "j2"
            }
        ],
        "TriggersNotFound": ["t3"],
        "Schedule": "cron(0 0/1 * * ? *)"
    }
}
```

Controllo delle impostazioni utilizzando le chiavi di condizione o di contesto

Quando si concedono le autorizzazioni per creare e aggiornare i processi, è possibile utilizzare le chiavi di condizione o le chiavi di contesto. Queste sezioni trattano le chiavi:

- Policy di controllo che controllano le impostazioni utilizzando le chiavi di condizione
- Policy di controllo che controllano le impostazioni utilizzando le chiavi di contesto

Policy di controllo che controllano le impostazioni utilizzando le chiavi di condizione

AWS Glue fornisce tre chiavi `glue:VpcIds` di condizione IAM e`glue:SecurityGroupIds`. `glue:SubnetIds` Quando si concedono le autorizzazioni per creare e aggiornare i processi, è possibile utilizzare le chiavi di condizione nelle policy IAM. È possibile utilizzare questa impostazione per garantire che i processi o le sessioni non vengano creati (o aggiornati) per essere eseguiti al di fuori dell'ambiente VPC desiderato. Le informazioni sull'impostazione del VPC non sono un input diretto dalla richiesta `CreateJob`, ma vengono inferite dal campo "connessioni" del processo che punta a una connessione AWS Glue.

Esempio di utilizzo

Crea un tipo di connessione di AWS Glue rete denominata "traffic-monitored-connection" con il VpcId «vpc-id1234" desiderato, e. SubnetIds SecurityGroupIds

Specifica la condizione delle chiavi di condizione per le operazioni `CreateJob` e `UpdateJob`nella policy IAM.

```
{
  "Effect": "Allow",
  "Action": [
    "glue:CreateJob",
    "glue:UpdateJob"
  ],
  "Resource": [
    "*"
  ],
  "Condition": {
    "ForAnyValue:StringLike": {
      "glue:VpcIds": [
        "vpc-id1234"
      ]
    }
  }
}
```

È possibile creare una policy IAM simile per vietare la creazione di un processo AWS Glue senza specificare le informazioni di connessione.

Limitazione delle sessioni su VPCs

Per imporre l'esecuzione delle sessioni create all'interno di un VPC specificato, puoi limitare l'autorizzazione del ruolo aggiungendo un effetto `Deny` all'operazione `glue:CreateSession`, a condizione che glue:vpc-id non sia uguale a vpc-<123>. Esempio:

```
"Effect": "Deny",
"Action": [
    "glue:CreateSession"
 ],
"Condition": {
    "StringNotEquals" : {"glue:VpcIds" : ["vpc-123"]}
}
```

È inoltre possibile imporre l'esecuzione delle sessioni create all'interno di un VPC aggiungendo un effetto `Deny` all'operazione `glue:CreateSession` a condizione che `glue:vpc-id` sia nullo. Esempio:

```
{
    "Effect": "Deny",
    "Action": [
        "glue:CreateSession"
    ],
      "Condition": {
        "Null": {"glue:VpcIds": true}
    }
},
{
    "Effect": "Allow",
    "Action": [
        "glue:CreateSession"
    ],
    "Resource": ["*"]
}
```

Policy di controllo che controllano le impostazioni utilizzando le chiavi di contesto

AWS Glue fornisce una chiave di contesto (`glue:CredentialIssuingService=glue.amazonaws.com`) per ogni sessione di ruolo che AWS Glue rende disponibile per l'endpoint job e developer. Ciò consente di implementare controlli di sicurezza per le azioni intraprese dagli AWS Glue script. AWS Gluefornisce un'altra chiave di contesto (`glue:RoleAssumedBy=glue.amazonaws.com`) per ogni sessione di ruolo in cui AWS Glue effettua una chiamata a un altro AWS servizio per conto del cliente (non tramite un job/dev endpoint, ma direttamente dal AWS Glue servizio).

Esempio di utilizzo

Specifica l'autorizzazione condizionale nella policy IAM e allegala al ruolo che deve essere utilizzato da un processo AWS Glue. Ciò garantisce che determinate azioni si allowed/denied basino sull'utilizzo della sessione di ruolo per un ambiente di AWS Glue job runtime.

```
{
    "Effect": "Allow",
    "Action": "s3:GetObject",
    "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
    "Condition": {
        "StringEquals": {
            "glue:CredentialIssuingService": "glue.amazonaws.com"
        }
```

```
    }
}
```

Negare a un'identità la possibilità di creare sessioni di anteprima dei dati

Questa sezione contiene un esempio di policy IAM utilizzato per negare a un'identità la possibilità di creare sessioni di anteprima dei dati. Collega questa policy all'identità, che è distinta dal ruolo utilizzato dalla sessione di anteprima dei dati durante la sua esecuzione.

```
{
    "Sid": "DatapreviewDeny",
    "Effect": "Deny",
     "Action": [
           "glue:CreateSession"
     ],
     "Resource": [
          "arn:aws:glue:*:*:session/glue-studio-datapreview*"
      ]
 }
```

## Esempi di policy basate sulle risorse per Glue AWS

Questa sezione contiene le policy di esempio basate su risorse, tra cui le policy che concedono l'accesso multi-account.

Gli esempi utilizzano il AWS Command Line Interface (AWS CLI) per interagire con le operazioni dell'API di AWS Glue servizio. È possibile eseguire le stesse operazioni sulla AWS Glue console o utilizzando una delle AWS SDKs.

> **Important**
>
> Modificando una policy della risorsa AWS Glue, potresti accidentalmente revocare le autorizzazioni per gli utenti AWS Glue esistenti nel tuo account e provocare interruzioni impreviste. Prova questi esempi solo con gli account di sviluppo o di test e verifica che non interrompano nessun flusso di lavoro esistente prima di apportare le modifiche.

Argomenti

- Considerazioni sull'utilizzo di politiche basate sulle risorse con Glue AWS
- Utilizza una policy della risorsa per controllare gli accessi nello stesso account

Considerazioni sull'utilizzo di politiche basate sulle risorse con Glue AWS

> **Note**
>
> Sia le policy IAM che una policy della risorsa AWS Glue richiedono pochi secondi, per la propagazione. Dopo aver collegato una nuova policy, potresti anche notare che la policy precedente è ancora in vigore finché la nuova policy non viene propagata attraverso il sistema.

È possibile utilizzare un documento di policy scritte in formato JSON per creare o modificare una policy della risorsa. La sintassi della policy è la stessa di una policy IAM basata sulle identità (consulta la documentazione di riferimento sulle policy JSON IAM), con le seguenti eccezioni:

- Un blocco `"Principal"` o `"NotPrincipal"` è obbligatorio per ogni istruzione di policy.
- Il `"Principal"` o il `"NotPrincipal"` deve identificare principali esistenti validi. I modelli dei caratteri jolly (ad esempio `arn:aws:iam::`*`account-id`*`:user/*`) non sono consentiti.
- Il `"Resource"` blocco della policy richiede che tutte le risorse corrispondano ARNs alla seguente sintassi delle espressioni regolari (dove la prima `%s` è la e la seconda è *`region`* la): `%s` *`account-id`*

  ```
  *arn:aws:glue:%s:%s:(\*|[a-zA-Z\*]+\/?.*)
  ```

  Ad esempio, sia `arn:aws:glue:us-west-2:`*`account-id`*`:*` che `arn:aws:glue:us-west-2:`*`account-id`*`:database/default` sono consentiti, ma non è consentito `*`.
- A differenza delle politiche basate sull'identità, una policy relativa alle AWS Glue risorse deve contenere solo Amazon Resource Names (ARNs) di risorse che appartengono al catalogo a cui è allegata la policy. Queste iniziano ARNs sempre con. `arn:aws:glue:`
- Una policy non può impedire l'ulteriore creazione o modifica dell'identità che la crea.
- La dimensione di un documento JSON di policy della risorsa non può superare 10 KB.

Utilizza una policy della risorsa per controllare gli accessi nello stesso account

In questo esempio, un utente admin nell'account A crea una policy della risorsa che concede all'utente IAM `Alice` dell'account A l'accesso completo al catalogo. Alice non ha alcuna policy IAM collegata.

Per fare ciò, l'utente amministratore esegue il seguente AWS CLI comando.

```
# Run as admin of Account A
$ aws glue put-resource-policy --profile administrator-name --region us-west-2 --
policy-in-json '{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Principal": {
        "AWS": [
          "arn:aws:iam::account-A-id:user/Alice"
        ]
      },
      "Effect": "Allow",
      "Action": [
        "glue:*"
      ],
      "Resource": [
        "arn:aws:glue:us-west-2:account-A-id:*"
      ]
    }
  ]
}'
```

Invece di inserire il documento di policy JSON come parte del AWS CLI comando, potete salvare un documento di policy in un file e fare riferimento al percorso del file nel AWS CLI comando, preceduto da. `file://` Di seguito è riportato un esempio di come svolgere questa operazione.

```
$ echo '{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Principal": {
        "AWS": [
          "arn:aws:iam::account-A-id:user/Alice"
        ]
      },
      "Effect": "Allow",
      "Action": [
        "glue:*"
      ],
      "Resource": [
        "arn:aws:glue:us-west-2:account-A-id:*"
```

```
      ]
    }
  ]
}' > /temp/policy.json

$ aws glue put-resource-policy --profile admin1 \
    --region us-west-2 --policy-in-json file:///temp/policy.json
```

Dopo la propagazione di questa policy basata sulle risorse, Alice può accedere a tutte le risorse di AWS Glue nell'account A, come segue.

```
# Run as user Alice
$ aws glue create-database --profile alice --region us-west-2 --database-input '{
    "Name": "new_database",
    "Description": "A new database created by Alice",
    "LocationUri": "s3://amzn-s3-demo-bucket"
}'

$ aws glue get-table --profile alice --region us-west-2 --database-name "default" --
table-name "tbl1"}
```

in risposta alla chiamata `get-table` di Alice, il servizio AWS Glue restituisce quanto segue.

```
{
  "Table": {
    "Name": "tbl1",
    "PartitionKeys": [],
    "StorageDescriptor": {
        ......
    },
    ......
  }
}
```

## Concessione di politiche AWS gestite per AWS Glue

Una politica AWS gestita è una politica autonoma creata e amministrata da AWS. AWS le politiche gestite sono progettate per fornire autorizzazioni per molti casi d'uso comuni, in modo da poter iniziare ad assegnare autorizzazioni a utenti, gruppi e ruoli.

Tieni presente che le policy AWS gestite potrebbero non concedere le autorizzazioni con il privilegio minimo per i tuoi casi d'uso specifici, poiché sono disponibili per tutti i clienti. AWS Si consiglia

pertanto di ridurre ulteriormente le autorizzazioni definendo policy gestite dal cliente specifiche per i propri casi d'uso.

Non è possibile modificare le autorizzazioni definite nelle politiche gestite. AWS Se AWS aggiorna le autorizzazioni definite in una politica AWS gestita, l'aggiornamento ha effetto su tutte le identità principali (utenti, gruppi e ruoli) a cui è associata la politica. AWS è più probabile che aggiorni una policy AWS gestita quando ne Servizio AWS viene lanciata una nuova o quando diventano disponibili nuove operazioni API per i servizi esistenti.

Per ulteriori informazioni, consultare Policy gestite da AWS nella Guida per l'utente di IAM.

## AWS politiche gestite (predefinite) per AWS Glue

AWS affronta molti casi d'uso comuni fornendo policy IAM autonome create e amministrate da. AWS Queste policy AWS gestite concedono le autorizzazioni necessarie per i casi d'uso comuni in modo da evitare di dover esaminare quali autorizzazioni sono necessarie. Per ulteriori informazioni, consultare Policy gestite da AWS nella Guida per l'utente di IAM.

Le seguenti politiche AWS gestite, che puoi allegare alle identità del tuo account, sono specifiche AWS Glue e raggruppate per scenario d'uso:

- AWSGlueConsoleFullAccess— Garantisce l'accesso completo alle AWS Glue risorse quando un'identità a cui è associata la politica utilizza il. Console di gestione AWS Se si segue la convenzione per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questa policy è in genere collegata agli utenti della console AWS Glue.
- AWSGlueServiceRole— Garantisce l'accesso alle risorse che AWS Glue i vari processi richiedono per l'esecuzione per conto dell'utente. Queste risorse includono Amazon S3AWS Glue, IAM, CloudWatch Logs e Amazon EC2. Se si segue la convenzione di denominazione per le risorse specificata in questa policy, i processi AWS Glue dispongono delle autorizzazioni richieste. Questa policy è in genere collegata ai ruoli specificati quando si definiscono crawler, processi ed endpoint di sviluppo.
- AwsGlueSessionUserRestrictedServiceRole— Fornisce l'accesso completo a tutte le AWS Glue risorse ad eccezione delle sessioni. Permette agli utenti di creare e utilizzare solo le sessioni interattive associate all'utente. Questa politica include altre autorizzazioni necessarie AWS Glue per gestire AWS Glue le risorse in altri AWS servizi. La politica consente inoltre di aggiungere tag alle AWS Glue risorse di altri AWS servizi.

> **Note**
>
> Per ottenere tutti i vantaggi della sicurezza, non concedere questa policy a un utente a cui sia stata assegnata la policy `AWSGlueServiceRole`, `AWSGlueConsoleFullAccess` o `AWSGlueConsoleSageMakerNotebookFullAccess`.

- AwsGlueSessionUserRestrictedPolicy— Fornisce l'accesso per creare sessioni AWS Glue interattive utilizzando l'operazione `CreateSession` API solo se vengono forniti una chiave di tag «proprietario» e un valore che corrispondono all'ID AWS utente dell'assegnatario. Questa policy di identità è collegata all'utente IAM che richiama l'operazione dell'API `CreateSession`. Questa politica consente inoltre all'assegnatario di interagire con le risorse della sessione AWS Glue interattiva create con un tag e un valore «proprietario» che corrispondono al proprio ID utente. AWS Questa policy nega l'autorizzazione di modificare o rimuovere i tag "proprietario" da una risorsa di sessione AWS Glue dopo la creazione della sessione.

  > **Note**
  >
  > Per ottenere tutti i vantaggi della sicurezza, non concedere questa policy a un utente a cui sia stata assegnata la policy `AWSGlueServiceRole`, `AWSGlueConsoleFullAccess` o `AWSGlueConsoleSageMakerNotebookFullAccess`.

- AwsGlueSessionUserRestrictedNotebookServiceRole— Fornisce un accesso sufficiente alla sessione del AWS Glue Studio notebook per interagire con risorse di sessione interattive specificheAWS Glue. Si tratta di risorse create con il valore del tag «owner» che corrisponde all'ID AWS utente del principale (utente o ruolo IAM) che crea il notebook. Per ulteriori informazioni su questi tag, consulta il grafico Valori della chiave dell'entità principale nella Guida per l'utente IAM.

  Questa policy del ruolo di servizio viene collegata al ruolo specificato con un comando magic all'interno del notebook o viene passata come ruolo all'operazione `CreateSession` dell'API. Questa politica consente inoltre al principale di creare una sessione AWS Glue interattiva dall'interfaccia del AWS Glue Studio notebook solo se la chiave del tag «proprietario» e il valore corrispondono all'ID AWS utente del principale. Questa policy nega l'autorizzazione di modificare o rimuovere i tag "proprietario" da una risorsa di sessione AWS Glue dopo la creazione della sessione. Questa policy include anche le autorizzazioni per la scrittura e la lettura da bucket Amazon S3, la CloudWatch scrittura di log e la creazione ed eliminazione di tag per le risorse Amazon EC2 utilizzate da. AWS Glue

> **Note**
>
> Per ottenere tutti i vantaggi della sicurezza, non concedere questa policy a un ruolo a cui sia stata assegnata la policy `AWSGlueServiceRole`, `AWSGlueConsoleFullAccess` o `AWSGlueConsoleSageMakerNotebookFullAccess`.

- AwsGlueSessionUserRestrictedNotebookPolicy— Fornisce l'accesso per creare una sessione AWS Glue interattiva dall'interfaccia del AWS Glue Studio notebook solo se sono presenti un tag, chiave («proprietario») e un valore che corrispondono all' AWS utente principale (utente o ruolo IAM) che crea IDof il notebook. Per ulteriori informazioni su questi tag, consulta il grafico Valori della chiave dell'entità principale nella Guida per l'utente IAM.

  Questa policy è associata al principale (utente o ruolo IAM) che crea sessioni dall'interfaccia AWS Glue Studio notebook. Inoltre, questa policy offre un accesso adeguato al notebook AWS Glue Studio per interagire con specifiche risorse della sessione interattiva AWS Glue. Si tratta di risorse create con il valore del tag «owner» che corrisponde all'ID AWS utente del principale. Questa policy nega l'autorizzazione di modificare o rimuovere i tag "proprietario" da una risorsa di sessione AWS Glue dopo la creazione della sessione.
- AWSGlueServiceNotebookRole— Concede l'accesso alle AWS Glue sessioni avviate su un AWS Glue Studio taccuino. Questa politica consente di elencare e ottenere informazioni sulla sessione per tutte le sessioni, ma consente solo agli utenti di creare e utilizzare le sessioni contrassegnate con il proprio ID AWS utente. Questa politica nega l'autorizzazione a modificare o rimuovere i tag «proprietario» dalle risorse AWS Glue della sessione contrassegnate con il loro AWS ID.

  Assegna questo criterio all' AWS utente che crea lavori utilizzando l'interfaccia del notebook in. AWS Glue Studio
- AWSGlueConsoleSageMakerNotebookFullAccess— Garantisce l'accesso completo alle risorse AWS Glue e SageMaker AI quando l'identità a cui è associata la policy utilizza il Console di gestione AWS. Se si segue la convenzione per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questa policy è in genere associata agli utenti della AWS Glue console che gestiscono notebook con SageMaker intelligenza artificiale.
- AWSGlueSchemaRegistryFullAccess— Garantisce l'accesso completo alle risorse del registro AWS Glue dello schema quando l'identità a cui è allegata la politica utilizza o. Console di gestione AWS AWS CLI Se si segue la convenzione per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questo criterio viene in genere associato agli utenti della AWS Glue console o a AWS CLI chi gestisce lo AWS Glue Schema Registry.

- AWSGlueSchemaRegistryReadonlyAccess— Concede l'accesso in sola lettura alle risorse del registro AWS Glue dello schema quando un'identità a cui è associata la politica utilizza o. Console di gestione AWS AWS CLI Se si segue la convenzione per la denominazione per le risorse specificate nella policy, gli utenti hanno la piena funzionalità della console. Questo criterio viene in genere associato agli utenti della AWS Glue console o AWS CLI che utilizzano lo Schema Registry. AWS Glue

> **Note**
>
> Per esaminare queste policy di autorizzazione, accedi alla console IAM ed esegui la ricerca delle policy specifiche.

Puoi anche creare policy IAM personalizzate per concedere le autorizzazioni per operazioni e risorse AWS Glue. Puoi associare queste policy personalizzate agli utenti o ai gruppi IAM che richiedono tali autorizzazioni.

Per creare una connessione con la configurazione VPC utilizzando un ruolo IAM personalizzato, deve disporre delle seguenti azioni di accesso VPC:

- gestore dei segreti: GetSecretValue
- gestore dei segreti: PutSecretValue
- gestore dei segreti: DescribeSecret
- ec2: CreateNetworkInterface
- ec2: DeleteNetworkInterface
- ec2: DescribeNetworkInterfaces
- ec2: DescribeSubnets

## AWS Glue gli aggiornamenti alle politiche AWS gestite

Visualizza i dettagli sugli aggiornamenti delle politiche AWS gestite per AWS Glue da quando questo servizio ha iniziato a tenere traccia di queste modifiche. Per ricevere avvisi automatici sulle modifiche a questa pagina, iscriviti al feed RSS nella pagina della cronologia di AWS Glue Document.

| Modifica | Descrizione | Data |
| --- | --- | --- |
| AwsGlueSessionUser RestrictedNotebookPolicy — Aggiornamento minore di una politica esistente. | Aggiungere l'azione consenti per `glue:TagResource` sulla chiave tag proprietario. Necessario per il supporto tag-on-create per le sessioni con la chiave del tag proprietario. | 30 agosto 2024 |
| AwsGlueSessionUserRestricted NotebookServiceRole — Aggiornamento minore di una politica esistente. | Aggiungere l'azione consenti per `glue:TagResource` sulla chiave tag proprietario. Necessario per il supporto tag-on-create per le sessioni con la chiave del tag proprietario. | 30 agosto 2024 |
| AwsGlueSessionUser RestrictedPolicy — Aggiornam ento minore di una politica esistente. | Aggiungere l'azione consenti per `glue:TagResource` sulla chiave tag proprietario. Necessario per il supporto tag-on-create per le sessioni con la chiave del tag proprietario. | 5 agosto 2024 |
| AwsGlueSessionUserRestricte dServiceRole — Aggiornam ento minore di una politica esistente. | Aggiungere l'azione consenti per `glue:TagResource` sulla chiave tag proprietario. Necessario per il supporto tag-on-create per le sessioni con la chiave del tag proprietario. | 5 agosto 2024 |
| AwsGlueSessionUser RestrictedPolicy — Aggiornam ento minore di una politica esistente. | Aggiungi `glue:StartCompletion` e `glue:GetCompletion` alla policy. Necessario per l'integrazione dei dati di Amazon Q in AWS Glue. | 30 aprile 2024 |

| Modifica | Descrizione | Data |
| --- | --- | --- |
| AwsGlueSessionUserRestrictedNotebookServiceRole — Aggiornamento minore di una politica esistente. | Aggiungi `glue:StartCompletion` e `glue:GetCompletion` alla policy. Necessario per l'integrazione dei dati di Amazon Q in AWS Glue. | 30 aprile 2024 |
| AwsGlueSessionUserRestrictedServiceRole — Aggiornamento minore di una politica esistente. | Aggiungi `glue:StartCompletion` e `glue:GetCompletion` alla policy. Necessario per l'integrazione dei dati di Amazon Q in AWS Glue. | 30 aprile 2024 |
| AWSGlueServiceNotebookRole — Aggiornamento minore di una politica esistente. | Aggiungi `glue:StartCompletion` e `glue:GetCompletion` alla policy. Necessario per l'integrazione dei dati di Amazon Q in AWS Glue. | 30 gennaio 2024 |
| AwsGlueSessionUserRestrictedNotebookPolicy — Aggiornamento minore di una politica esistente. | Aggiungi `glue:StartCompletion` e `glue:GetCompletion` alla policy. Necessario per l'integrazione dei dati di Amazon Q in AWS Glue. | 29 novembre 2023 |
| AWSGlueServiceNotebookRole — Aggiornamento minore di una politica esistente. | Aggiungi `codewhisperer:GenerateRecommendations` alla policy. Necessario per una nuova funzionalità in cui AWS Glue genera CodeWhisperer consigli. | 9 ottobre 2023 |

| Modifica | Descrizione | Data |
| --- | --- | --- |
| AWSGlueServiceRole — Aggiornamento minore di una politica esistente. | Restringi l'ambito delle CloudWatch autorizzazioni per riflettere meglio la registraz ione di AWS Glue. | 4 agosto 2023 |
| AWSGlueConsoleFullAccess — Aggiornamento minore di una politica esistente. | Aggiungi le autorizzazioni Elenca e Descrivi per le ricette `databrew` alla policy. Necessario per fornire l'accesso amministrativo completo alle nuove funzional ità in cui AWS Glue può accedere alle ricette. | 9 maggio 2023 |
| AWSGlueConsoleFullAccess — Aggiornamento minore di una politica esistente. | Aggiungi `cloudform ation:ListStacks` alla policy. Conserva le funzional ità esistenti dopo le modifiche ai requisiti di CloudFormation autorizzazione. | 28 marzo 2023 |
| Aggiunte nuove policy gestite per la funzionalità sessioni interattive:<br>• AwsGlueSessionUser RestrictedServiceRole<br>• AwsGlueSessionUser RestrictedPolicy<br>• AwsGlueSessionUser RestrictedNotebook ServiceRole<br>• AwsGlueSessionUser RestrictedNotebookPolicy | Queste policy sono state progettate per fornire ulteriore sicurezza per le sessioni interattive e i notebook in AWS Glue Studio. Le policy limitano l'accesso all'opera zione dell'API `CreateSes sion` , in modo che solo il proprietario abbia accesso. | 30 novembre 2021 |

| Modifica | Descrizione | Data |
| --- | --- | --- |
| AWSGlueConsoleSageMakerNotebookFullAccess — Aggiornamento a una politica esistente. | È stata rimossa una risorsa ridondante ARN (`arn:aws:s3:::aws-glue-*/*`) per l'azione che concede le autorizzazioni read/write sui bucket AWS Glue Amazon S3 utilizzati per archiviare script e file temporanei.<br><br>Risolto un problema di sintassi modificando `"StringEquals"` in `"ForAnyValue:StringLike"` e spostate le righe `"Effect": "Allow"` per precedere la riga `"Action":` in ogni luogo in cui erano fuori uso. | 15 luglio 2021 |
| AWSGlueConsoleFullAccess — Aggiornamento a una policy esistente. | È stata rimossa una risorsa ridondante ARN (`arn:aws:s3:::aws-glue-*/*`) per l'azione che concede le autorizzazioni read/write sui bucket AWS Glue Amazon S3 utilizzati per archiviare script e file temporanei. | 15 luglio 2021 |
| AWS Glue ha iniziato il rilevamento delle modifiche. | AWS Glueha iniziato a tenere traccia delle modifiche per le sue politiche gestite. AWS | 10 giugno 2021 |

# Concessione di policy con ambito dinamico per l'esecuzione dei processi

AWS Glue offre una nuova potente funzionalità: politiche di sessione dinamiche per l'esecuzione dei lavori. Questa funzionalità consente di specificare autorizzazioni personalizzate e granulari per ogni processo eseguito senza creare ruoli IAM multipli.

Quando avvii un processo Glue usando l'API `StartJobRun`, puoi includere una policy di sessione inline. Questa policy modifica temporaneamente le autorizzazioni del ruolo di esecuzione del processo per la durata dell'esecuzione di quello specifico processo. È simile all'utilizzo di credenziali temporanee con l'`AssumeRole`API in altri AWS servizi.

- Sicurezza avanzata: puoi limitare le autorizzazioni del processo al minimo necessario per ogni esecuzione.
- Gestione semplificata: elimina la necessità di creare e gestire numerosi ruoli IAM per diversi scenari.
- Flessibilità: puoi regolare le autorizzazioni in modo dinamico in base ai parametri di runtime o alle specifiche esigenze del tenant.
- Scalabilità: questo metodo è perfetto negli ambienti multi-tenant in cui è necessario isolare le risorse tra i tenant.

Esempi per concedere l'uso di policy con ambito dinamico:

Gli esempi seguenti mostrano come concedere ai processi l'accesso in sola lettura e scrittura a uno specifico percorso di bucket Amazon S3, dove il percorso è determinato dinamicamente dall'ID di esecuzione del processo. Illustra come implementare autorizzazioni granulari e specifiche per l'esecuzione di ogni processo.

Dalla CLI

```
aws glue start-job-run \
    --job-name "your-job-name" \
    --execution-role-session-policy '{
        "Version": "2012-10-17",
        "Statement": [
            {
                "Effect": "Allow",
                "Action": [
                    "s3:GetObject",
```

```
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::specific-bucket/${JobRunId}/*"
            ]
        }
    ]
}'
```

## Specificare la risorsa AWS Glue ARNs

InAWS Glue, puoi controllare l'accesso alle risorse utilizzando una policy AWS Identity and Access Management (IAM). In una policy, devi utilizzare un Amazon Resource Name (ARN) per identificare la risorsa a cui si applica la policy stessa. Non tutte le risorse sono AWS Glue supportate ARNs.

Argomenti

- Catalogo dati ARNs
- ARNs per oggetti non di catalogo in AWS Glue
- Controllo accessi per operazioni API singole non nel catalogo AWS Glue
- Controllo degli accessi per le operazioni API di AWS Glue per oggetti non inclusi nel catalogo che richiamano più elementi
- Controllo degli accessi per operazioni API AWS Glue non legate al catalogo BatchGet

### Catalogo dati ARNs

Le risorse del catalogo dati sono organizzate secondo una struttura gerarchica nella quale `catalog` funge da root.

```
arn:aws:glue:region:account-id:catalog
```

Ogni AWS account ha un unico catalogo dati in una AWS regione con l'ID dell'account a 12 cifre come ID del catalogo. Alle risorse sono ARNs associate risorse univoche, come illustrato nella tabella seguente.

| Tipo di risorsa | Formato ARN |
| --- | --- |
| Catalogo | `arn:aws:glue: region:account-id :catalog` |

| Tipo di risorsa | Formato ARN |
| --- | --- |
| | Ad esempio: `arn:aws:glue:us-east-1:123456789012:catalog` |
| Database | `arn:aws:glue:` *`region`*`:`*`account-id`* `:database/` *`database name`*<br><br>Ad esempio: `arn:aws:glue:us-east-1:123456789012:database/db1` |
| Tabella | `arn:aws:glue:` *`region`*`:`*`account-id`* `:table/`*`database name`*`/`*`table name`*<br><br>Ad esempio: `arn:aws:glue:us-east-1:123456789012:table/db1/tbl1` |
| Catalogo di tabelle S3 federate (tutti i bucket della tabella) | `arn:aws:glue:` *`region`*`:`*`account-id`* `:catalog/s3tablescatalog`<br><br>Ad esempio: `arn:aws:glue:us-east-1:123456789012:catalog/s3tablescatalog` |
| Catalogo bucket di tabelle S3 federate (catalogo figlio) | `arn:aws:glue:` *`region`*`:`*`account-id`* `:catalog/s3tablescatalog/` *`bucket name`*<br><br>Ad esempio: `arn:aws:glue:us-east-1:123456789012:catalog/s3tablescatalog/amzn-s3-demo-bucket1` |
| Database di tabelle S3 federate | `arn:aws:glue:` *`region`*`:`*`account-id`* `:database/s3tablescatalog/` *`child catalog name`*`/`*`database name`*<br><br>Ad esempio: `arn:aws:glue:us-east-1:123456789012:database/s3tablescatalog/amzn-s3-demo-bucket1/nsdb1` |

| Tipo di risorsa | Formato ARN |
| --- | --- |
| Tabella S3 federata | arn:aws:glue: *region*:*account-id* :table/s3tablescatalog/ *child catalog name*/*database name*/*table name*<br><br>Ad esempio: arn:aws:glue:us-east-1:123456789012:table/s3tablescatalog/amzn-s3-demo-bucket1/nsdb1/s3tbl1 |
| Catalogo di tabelle S3 federate (un singolo bucket della tabella registrato con Lake Formation) | arn:aws:glue: *region*:*account-id* :catalog/*catalog name*<br><br>Ad esempio: arn:aws:glue:us-east-1:123456789012:catalog/amzn-s3-demo-bucket1 |
| Database di tabelle S3 federate | arn:aws:glue: *region*:*account-id* :catalog/*catalog name*/*database name*<br><br>Ad esempio: arn:aws:glue:us-east-1:123456789012:database/amzn-s3-demo-bucket1/nsdb1 |
| Tabella S3 federata | arn:aws:glue: *region*:*account-id* :catalog/*catalog name*/*database name*/*table name*<br><br>Ad esempio: arn:aws:glue:us-east-1:123456789012:table/amzn-s3-demo-bucket1/ *nsdb1*/s3tbl1 |
| Catalogo federato o gestito (catalogo di primo livello in un multicatalogo) | arn:aws:glue: *region*:*account-id* :catalog/ *top-level catalog name*<br><br>Ad esempio: arn:aws:glue:us-east-1:123456789012:catalog/nscatalog<br><br>Il formato ARN per un catalogo gestito segue la stessa struttura. |

| Tipo di risorsa | Formato ARN |
| --- | --- |
| Catalogo federato a più livelli (catalogo figlio in un catalogo multilivello) | `arn:aws:glue:` *`region`*`:`*`account-id`* `:catalog/` *`top-level catalog name`*`/`*`child catalog name`*<br><br>Ad esempio: `arn:aws:glue:us-east-1:123456789012:catalog/nscatalog/dbcatalog` |
| Database federato | `arn:aws:glue:` *`region`*`:`*`account-id`* `:database/` *`name space catalog name`*`/`*`child catalog name`*`/`*`database name`*<br><br>Ad esempio: `arn:aws:glue:us-east-1:123456789012:database/nscatalog/dbcatalog/schemadb` |
| Tabella federata | `arn:aws:glue:` *`region`*`:`*`account-id`* `:table/`*`name space catalog name`*`/`*`child catalog name`*`/`*`database name`*`/`*`table name`*<br><br>Ad esempio: `arn:aws:glue:us-east-1:123456789012:table/nscatalog/dbcatalog/schemadb/rstbl1` |
| Container link al catalogo | `arn:aws:glue:` *`region`*`:`*`account-id`* `:catalog/`*`link container name`*<br><br>Ad esempio: `arn:aws:glue:glue:us-east-1:123456789012:catalog` /linkcontainer-example |
| Database | `arn:aws:glue:` *`region`*`:`*`account-id`* `:catalog/`*`link container name`*`/`*`database name`*<br><br>Ad esempio: `arn:aws:glue:glue:us-east-1:123456789012:database` /linkcontainer-example/link-db |
| Funzione definita dall'utente | `arn:aws:glue:` *`region`*`:`*`account-id`* `:userDefinedFunction/` *`database name`*`/`*`user-defined function name`*<br><br>Ad esempio: `arn:aws:glue:us-east-1:123456789012:userDefinedFunction/db1/func1` |

| Tipo di risorsa | Formato ARN |
| --- | --- |
| Connessione | `arn:aws:glue:` *`region`*`:`*`account-id`* `:connection/` *`connection name`*<br><br>Ad esempio: `arn:aws:glue:us-east-1:123456789012:connection/connection1` |
| Sessioni interattive | `arn:aws:glue:` *`region`*`:`*`account-id`* `:session/` *`interactive session id`*<br><br>Ad esempio: `arn:aws:glue:us-east-1:123456789012:session/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111` |

Per abilitare un controllo granulare degli accessi, puoi utilizzarli ARNs nelle policy IAM e nelle policy delle risorse per concedere e negare l'accesso a risorse specifiche. Nelle policy sono ammessi i caratteri jolly. Ad esempio, il seguente ARN corrisponde a tutte le tabelle nel database `default`.

```
arn:aws:glue:us-east-1:123456789012:table/default/*
```

**Important**

Tutte le operazioni eseguite su una risorsa del catalogo dati richiedono l'autorizzazione alla risorsa e tutti i predecessori di tale risorsa. Ad esempio, la creazione di una partizione per una tabella necessita dell'autorizzazione su tabella, database e catalogo in cui si trova la tabella stessa. L'esempio seguente mostra le autorizzazioni necessarie per creare le partizioni sulla tabella `PrivateTable` nel database `PrivateDatabase` nel catalogo dati.

```
{
   "Sid": "GrantCreatePartitions",
   "Effect": "Allow",
   "Action": [
       "glue:BatchCreatePartitions"
   ],
   "Resource": [
       "arn:aws:glue:us-east-1:123456789012:table/PrivateDatabase/PrivateTable",
       "arn:aws:glue:us-east-1:123456789012:database/PrivateDatabase",
       "arn:aws:glue:us-east-1:123456789012:catalog"
   ]
```

```
}
```

Oltre alle autorizzazioni per la risorsa e tutti i suoi predecessori, tutte le operazioni di eliminazione richiedono l'autorizzazione su tutti i figli di tale risorsa. Ad esempio, per eliminare un database è necessario disporre di autorizzazioni per tutte le tabelle e per le funzioni definite dall'utente del database, così come per il database e il catalogo in cui si trova il database. L'esempio seguente mostra le autorizzazioni necessarie per eliminare il database `PrivateDatabase` nel catalogo dati.

```
{
   "Sid": "GrantDeleteDatabase",
   "Effect": "Allow",
   "Action": [
       "glue:DeleteDatabase"
   ],
   "Resource": [
       "arn:aws:glue:us-east-1:123456789012:table/PrivateDatabase/*",
       "arn:aws:glue:us-east-1:123456789012:userDefinedFunction/PrivateDatabase/
*",
       "arn:aws:glue:us-east-1:123456789012:database/PrivateDatabase",
       "arn:aws:glue:us-east-1:123456789012:catalog"
   ]
}
```

Riepilogando, le operazioni sulle risorse del catalogo dati seguono queste regole di autorizzazione:

- Le operazioni sul catalogo richiedono solo l'autorizzazione per il catalogo.
- Le operazioni su un database richiedono l'autorizzazione su database e catalogo.
- Le operazioni di eliminazione su un database richiedono l'autorizzazione su database e catalogo, oltre che su tutte le tabelle e funzioni definite dall'utente del database.
- Le operazioni su una tabella, partizione o versione di una tabella richiedono l'autorizzazione su tabella, database e catalogo.
- Le operazioni su una funzione definita dall'utente richiedono l'autorizzazione su funzione definita dall'utente, catalogo e database.
- Le operazioni su una connessione richiedono l'autorizzazione su connessione e catalogo.

## ARNs per oggetti non di catalogo in AWS Glue

Alcune risorse AWS Glue consentono autorizzazioni a livello di risorsa per controllare l'accesso utilizzando un ARN. Puoi utilizzarli ARNs nelle tue policy IAM per abilitare un controllo granulare degli accessi. La tabella seguente elenca le risorse che possono contenere risorse. ARNs

| Tipo di risorsa | Formato ARN |
| --- | --- |
| Crawler | `arn:aws:glue:` *`region`*`:`*`account-id`* `:crawler/` *`crawler-name`*<br><br>Ad esempio: `arn:aws:glue:us-east-1:123456789012:crawler/mycrawler` |
| Processo | `arn:aws:glue:` *`region`*`:`*`account-id`* `:job/`*`job-name`*<br><br>Ad esempio: `arn:aws:glue:us-east-1:123456789012:job/testjob` |
| Trigger | `arn:aws:glue:` *`region`*`:`*`account-id`* `:trigger/` *`trigger-name`*<br><br>Ad esempio: `arn:aws:glue:us-east-1:123456789012:trigger/sampletrigger` |
| Endpoint di sviluppo | `arn:aws:glue:` *`region`*`:`*`account-id`* `:devEndpoint/` *`development-endpoint-name`*<br><br>Ad esempio: `arn:aws:glue:us-east-1:123456789012:devEndpoint/temporarydevendpoint` |
| Trasformazione basata su machine learning | `arn:aws:glue:` *`region`*`:`*`account-id`* `:mlTransform/` *`transform-id`*<br><br>Ad esempio: `arn:aws:glue:us-east-1:123456789012:mlTransform/tfm-1234567890` |

## Controllo accessi per operazioni API singole non nel catalogo AWS Glue

Le operazioni delle API singole non nel catalogo AWS Glue, agiscono su un singolo elemento (endpoint di sviluppo). Alcuni esempi sono `GetDevEndpoint`, `CreateUpdateDevEndpoint` e `UpdateDevEndpoint`. Per queste operazioni, una policy deve inserire il nome dell'API nel blocco `"action"` e la risorsa ARN nel blocco `"resource"`.

Supponiamo che si desideri consentire a un utente di chiamare l'operazione `GetDevEndpoint`. La policy seguente concede il livello minimo di autorizzazioni necessarie per un endpoint denominato `myDevEndpoint-1`:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "MinimumPermissions",
      "Effect": "Allow",
      "Action": "glue:GetDevEndpoint",
      "Resource": "arn:aws:glue:us-east-1:111122223333:devEndpoint/
myDevEndpoint-1"
    }
  ]
}
```

La policy seguente consente l'accesso di `UpdateDevEndpoint` alle risorse che corrispondono a `myDevEndpoint-` con un carattere jolly (*):

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "PermissionWithWildcard",
      "Effect": "Allow",
      "Action": "glue:UpdateDevEndpoint",
```

```
      "Resource": "arn:aws:glue:us-east-1:111122223333:devEndpoint/myDevEndpoint-
*"
    }
  ]
}
```

Puoi combinare le due policy come nell'esempio seguente. È possibile vedere `EntityNotFoundException` per ogni endpoint di sviluppo il cui nome inizia con `A`. Tuttavia, quando si cerca di accedere ad altri endpoint di sviluppo viene restituito un errore di accesso negato.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "CombinedPermissions",
      "Effect": "Allow",
      "Action": [
        "glue:UpdateDevEndpoint",
        "glue:GetDevEndpoint"
      ],
      "Resource": "arn:aws:glue:us-east-1:111122223333:devEndpoint/A*"
    }
  ]
}
```

## Controllo degli accessi per le operazioni API di AWS Glue per oggetti non inclusi nel catalogo che richiamano più elementi

Alcune operazioni API AWS Glue richiamano più elementi (più endpoint di sviluppo); ad esempio, `GetDevEndpoints`. Per questa operazione, è possibile specificare solo una risorsa con caratteri jolly (*) e non una risorsa specifica. ARNs

Ad esempio, per includere `GetDevEndpoints` nella policy, la risorsa deve rientrare tra quelle identificate dal carattere jolly (*). L'ambito delle operazioni singole (`GetDevEndpoint`, `CreateDevEndpoint` e `DeleteDevendpoint`) viene definito anche per tutte le risorse (*) nell'esempio.

```
{
            "Sid": "PluralAPIIncluded",
            "Effect": "Allow",
            "Action": [
                "glue:GetDevEndpoints",
                "glue:GetDevEndpoint",
                "glue:CreateDevEndpoint",
                "glue:UpdateDevEndpoint"
            ],
            "Resource": [
                "*"
            ]
}
```

## Controllo degli accessi per operazioni API AWS Glue non legate al catalogo BatchGet

Alcune operazioni API AWS Glue richiamano più elementi (più endpoint di sviluppo); ad esempio, `BatchGetDevEndpoints`. Per questa operazione, è possibile specificare un ARN per limitare l'ambito delle risorse accessibili.

Ad esempio, per consentire l'accesso a un determinato endpoint di sviluppo, includere `BatchGetDevEndpoints` nella policy con il relativo ARN di risorsa.

```
{
            "Sid": "BatchGetAPIIncluded",
            "Effect": "Allow",
            "Action": [
                "glue:BatchGetDevEndpoints"
            ],
            "Resource": [
                "arn:aws:glue:us-east-1:123456789012:devEndpoint/de1"
            ]
}
```

Con questa policy puoi accedere all'endpoint di sviluppo denominato `de1`. Tuttavia, se cerchi di accedere all'endpoint di sviluppo `de2`, viene restituito un errore.

```
An error occurred (AccessDeniedException) when calling the BatchGetDevEndpoints
 operation: No access to any requested resource.
```

> **Important**
>
> Per approcci alternativi alla configurazione delle policy IAM, ad esempio usando le operazioni API `List` e `BatchGet`, consultare Esempi di policy basate sull'identità per Glue AWS.

# Come concedere l'accesso multi-account

Autorizzazione dell'accesso alle risorse del catalogo dati tra account consente ai processi di estrazione, trasformazione e caricamento (ETL) di eseguire query e join di dati da account diversi.

Argomenti


- Metodi per concedere l'accesso multi-account in AWS Glue
- Aggiunta o aggiornamento della policy della risorsa di catalogo dati
- Come effettuare una chiamata API multi-account
- Come effettuare una chiamata ETL multi-account
- Registrazione su più account CloudTrail
- Proprietà e fatturazione delle risorse multi-account
- Restrizioni di accesso multi-account


## Metodi per concedere l'accesso multi-account in AWS Glue

Puoi concedere l'accesso ai tuoi dati ad AWS account esterni utilizzando AWS Glue metodi o utilizzando sovvenzioni AWS Lake Formation tra account. I AWS Glue metodi utilizzano policy AWS Identity and Access Management (IAM) per ottenere un controllo granulare degli accessi. Lake Formation utilizza un modello di autorizzazioni `GRANT/REVOKE` più semplice, simile ai comandi `GRANT/REVOKE` in un sistema di database relazionale.

In questa sezione viene descritto l'utilizzo dei metodi AWS Glue. Per ulteriori informazioni sull'utilizzo delle autorizzazioni multi-account Lake Formation, consulta Concedere autorizzazioni Lake Formation nella Guida per gli sviluppatori di AWS Lake Formation .

Sono disponibili due metodi AWS Glue, per concedere l'accesso multi-account a una risorsa:

- Utilizzare una policy della risorsa del catalogo dati
- Utilizzare un ruolo IAM

### Concedere l'accesso multi-account utilizzando una policy della risorsa

Di seguito sono riportati i passaggi generali per concedere l'accesso multi-account utilizzando una policy della risorsa del catalogo dati:

1. Un amministratore (o un altro tipo di identità autorizzato) nell'Account A collega una policy della risorsa del catalogo dati nell'account A. Questa policy concede all'account B autorizzazioni multi-account specifiche per eseguire operazioni su una risorsa in un catalogo dell'account A.
2. Un amministratore nell'account B collega una policy IAM a un'identità IAM nell'account B che delega le autorizzazioni ricevute dall'Account A.

   L'identità nell'account B ora ha accesso alla risorsa specificata nell'account A.

   Per poter accedere alla risorsa, l'identità necessita dell'autorizzazione sia da parte del proprietario della risorsa (Account A), sia del relativo account padre (Account B).

### Concedere l'accesso multi-account utilizzando un ruolo IAM

Di seguito sono riportati i passaggi generali per concedere l'accesso multi-account utilizzando un ruolo IAM:

1. Un amministratore (o un altro tipo di identità autorizzato) nell'account proprietario della risorsa (Account A) crea un ruolo IAM.
2. L'amministratore nell'Account A collega una policy al ruolo che concede autorizzazioni multi-account per accedere alla risorsa in questione.
3. L'amministratore dell'account A collega una policy di attendibilità al ruolo che identifica un'identità IAM in un account differente (Account B) come il principale che può assumere il ruolo.

   Il responsabile della politica di fiducia può anche essere un responsabile del AWS servizio se si desidera concedere a un AWS servizio l'autorizzazione ad assumere il ruolo.
4. Un amministratore nell'account B ora delega le autorizzazioni a una o più identità IAM nell'account B in modo che possano assumere quel ruolo. In questo modo, consente alle identità nell'account B di accedere alla risorsa nell'account A.

Per ulteriori informazioni sull'uso di IAM per delegare le autorizzazioni, consultare Gestione degli accessi nella Guida per l'utente di IAM. Per ulteriori informazioni su utenti, gruppi, ruoli e autorizzazioni, consultare Identità (utenti, gruppi e ruoli) nella Guida per l'utente di IAM.

Per confrontare questi due approcci, consulta In che modo i ruoli IAM differiscono dalle policy basate sulla risorsa nella Guida per l'utente IAM. AWS Glue supporta entrambe le opzioni, con la limitazione che una policy della risorsa può concedere l'accesso solo alle risorse di catalogo dati.

Ad esempio, per concedere al ruolo `Dev` nell'account B l'accesso al database `db1` nell'account A, collegare al catalogo nell'account A le seguenti policy sulle risorse.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:GetDatabase"
      ],
      "Principal": {
        "AWS": [
          "arn:aws:iam::111122223333:role/Dev"
        ]
      },
      "Resource": [
        "arn:aws:glue:us-east-1:111122223333:catalog",
        "arn:aws:glue:us-east-1:111122223333:database/db1"
      ]
    }
  ]
}
```

Inoltre, prima che B possa effettivamente accedere a `db1` nell'account A, l'account B dovrebbe collegare al ruolo `Dev` la seguente policy IAM.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
```

```
      "Action": [
        "glue:GetDatabase"
      ],
      "Resource": [
        "arn:aws:glue:us-east-1:111122223333:catalog",
        "arn:aws:glue:us-east-1:111122223333:database/db1"
      ]
    }
  ]
}
```

## Aggiunta o aggiornamento della policy della risorsa di catalogo dati

Puoi aggiungere o aggiornare la politica delle risorse di AWS Glue Data Catalog utilizzando la console, l'API o AWS Command Line Interface (AWS CLI).

> **Important**
>
> Se hai già concesso autorizzazioni multi-account dal tuo account con AWS Lake Formation, l'aggiunta o l'aggiornamento della policy della risorsa del catalogo dati richiede un passaggio aggiuntivo. Per ulteriori informazioni, consulta Gestire autorizzazioni multi-account tramite AWS Glue e Lake Formation nella Guida per gli sviluppatori di AWS Lake Formation .
> Per determinare se esistono concessioni multi-account di Lake Formation, utilizza l'operazione dell'API `glue:GetResourcePolicies` o la AWS CLI. Se `glue:GetResourcePolicies` restituisce policy diverse da una policy catalogo dati già esistente, allora esistono concessioni Lake Formation. Per ulteriori informazioni, consulta Visualizzazione di tutte le sovvenzioni tra account utilizzando il funzionamento dell' GetResourcePolicies API nella Guida per gli AWS Lake Formation sviluppatori.

Aggiunta o aggiornamento della policy della risorsa del catalogo dati (console)

1. Apri la console AWS Glue all'indirizzo https://console.aws.amazon.com/glue/.

   Accedi come utente amministrativo AWS Identity and Access Management (IAM) con l'`glue:PutResourcePolicy`autorizzazione.

2. Nel pannello di navigazione scegli Impostazioni.

3. Nella pagina Data catalog settings (Impostazioni del catalogo dati), sottoPermissions (Autorizzazioni) incolla una policy della risorsa nell'area di testo. Quindi scegli Save (Salva).

   Se nella console viene visualizzato un avviso che indica che le autorizzazioni della policy saranno in aggiunta alle autorizzazioni concesse tramite Lake Formation, scegli Proceed (Continua).

Per aggiungere o aggiornare la policy della risorsa del catalogo dati (AWS CLI)

- Invia un comando `aws glue put-resource-policy`. Se esistono già concessioni per la Lake Formation, assicurati di includere l'opzione `--enable-hybrid` con il valore `'TRUE'`.

  Per esempi di utilizzo di questo comando, consulta Esempi di policy basate sulle risorse per Glue AWS.

## Come effettuare una chiamata API multi-account

Tutte le operazioni AWS Glue Data Catalog dispongono di un campo `CatalogId`. Se sono state concesse le autorizzazioni necessarie per l'accesso a più account, un chiamante è in grado di effettuare chiamate API del catalogo dati su tutti gli account. Il chiamante passa l'ID dell'account AWS di destinazione nel `CatalogId` in modo da accedere alla risorsa in tale account di destinazione.

Se non viene specificato alcun valore `CatalogId`, AWS Glue utilizza il proprio ID account del chiamante per impostazione predefinita e la chiamata non è multi-account.

## Come effettuare una chiamata ETL multi-account

Alcuni AWS Glue PySpark e Scala APIs hanno un campo ID del catalogo. Se sono state PySpark concesse tutte le autorizzazioni necessarie per consentire l'accesso tra account diversi, un job ETL può effettuare chiamate Scala alle operazioni API tra più account inserendo l'ID dell'account di destinazione nel campo ID del catalogo per accedere alle risorse del Data Catalog in un AWS account di destinazione.

Se non viene specificato alcun valore ID catalogo, AWS Glue utilizza il proprio ID account del chiamante per impostazione predefinita e la chiamata non è multi-account.

Per PySpark APIs tale assistenza`catalog_id`, consulta. GlueContext classe Per il supporto APIs di Scala`catalogId`, vediAWS GlueScala GlueContext APIs.

L'esempio seguente mostra le autorizzazioni richieste dall'assegnatario per l'esecuzione di un processo ETL. In questo esempio, *grantee-account-id* è il `catalog-id` client che esegue il job ed *grantor-account-id* è il proprietario della risorsa. Questo esempio illustra come concedere le autorizzazioni per tutte le risorse del catalogo nell'account del concedente. Per limitare l'ambito delle risorse concesse, è possibile fornire informazioni specifiche ARNs per il catalogo, il database, la tabella e la connessione.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:GetConnection",
        "glue:GetDatabase",
        "glue:GetTable",
        "glue:GetPartition"
      ],
      "Principal": {
        "AWS": [
          "arn:aws:iam::111122223333:root"
        ]
      },
      "Resource": [
        "arn:aws:glue:us-east-1:111122223333:*"
      ]
    }
  ]
}
```

> **Note**
>
> Se una tabella nell'account del concedente punta a una posizione Amazon S3 che si trova anche nell'account del concedente, il ruolo IAM utilizzato per eseguire un processo ETL negli account dell'assegnatario deve avere le autorizzazioni per elencare e ottenere gli oggetti dall'account del concedente.

Poiché il client nell'Account A dispone già dell'autorizzazione per creare ed eseguire processi ETL, i passaggi di base per configurare un processo ETL per l'accesso multi-account sono i seguenti:

1. Consentire l'accesso ai dati multi-account (salta questo passaggio se l'accesso multi-account Amazon S3 è già configurato).
   a. Aggiornare la policy nel bucket Amazon S3 nell'account B per consentire l'accesso multi-account dall'Account A.
   b. Aggiornare la policy IAM nel bucket A per concedere l'accesso al bucket nell'account B.
2. Consentire l'accesso al catalogo dati multi-account
   a. Creare o aggiornare la policy della risorsa associata al catalogo dati nell'account B per consentire l'accesso da parte di un account A.
   b. Aggiornare la policy IAM nell'account A per consentire l'accesso al catalogo dati nell'account B.

## Registrazione su più account CloudTrail

Quando un job AWS Glue di estrazione, trasformazione e caricamento (ETL) accede ai dati sottostanti di una tabella Data Catalog condivisa tramite concessioni tra AWS Lake Formation account, si verifica un comportamento di registrazione aggiuntivo. AWS CloudTrail

Ai fini di questa discussione, l' AWS account che ha condiviso la tabella è l'account del proprietario e l'account con cui è stata condivisa la tabella è l'account del destinatario. Quando un processo ETL nell'account del destinatario accede ai dati nella tabella dell'account del proprietario, l' CloudTrail evento di accesso ai dati che viene aggiunto ai registri dell'account del destinatario viene copiato nei registri dell'account del proprietario. CloudTrail In questo modo gli account proprietari possono tenere traccia degli accessi ai dati da parte dei vari account destinatari. Per impostazione predefinita, gli CloudTrail eventi non includono un identificatore principale leggibile dall'uomo (ARN principale). Un amministratore nell'account destinatario può scegliere di includere l'ARN principale nei log.

Per ulteriori informazioni, consulta Registrazione tra account CloudTrail nella Guida per gli sviluppatori.AWS Lake Formation

> **Vedi anche**
>
> - the section called “Registrazione e monitoraggio”

### Proprietà e fatturazione delle risorse multi-account

Quando un utente di un AWS account (Account A) crea una nuova risorsa, ad esempio un database in un altro account (Account B), quella risorsa è quindi di proprietà dell'Account B, l'account in cui è stata creata. Un amministratore nell'account B ottiene automaticamente tutte le autorizzazioni per accedere alla nuova risorsa, tra cui la lettura, la scrittura e la concessione di autorizzazioni di accesso a un terzo account. L'utente nell'account A può accedere alla risorsa che ha creato solo se dispone delle autorizzazioni appropriate concesse dall'account B.

I costi di storage e gli altri costi direttamente associati alla nuova risorsa vengono fatturati all'account B, il proprietario della risorsa. Il costo delle richieste dall'utente che ha creato la risorsa viene fatturato all'account del richiedente, l'account A.

Per ulteriori informazioni sulla AWS Glue fatturazione e sui prezzi, consulta Come funzionano AWS i prezzi.

### Restrizioni di accesso multi-account

L'accesso multi-account AWS Glue presenta le restrizioni seguenti:

- L'accesso tra account AWS Glue non è permesso se hai creato database e tabelle utilizzando Amazon Athena o Amazon Redshift Spectrum prima del supporto di una Regione per AWS Glue e se l'account del proprietario della risorsa non ha migrato il catalogo dati di Amazon Athena verso AWS Glue. Puoi trovare l'attuale stato di migrazione utilizzando GetCatalogImportStatus (get_catalog_import_status). Per ulteriori dettagli su come migrare un catalogo Athena verso, consulta l'aggiornamento AWS Glue alla versione contenuta AWS Glue Data Catalog step-by-step nella Amazon Athena User Guide.
- L'accesso multi-account è supportato solo per le risorse del catalogo dati, tra cui database, tabelle, funzioni definite dall'utente e connessioni.
- L'accesso multi-account al catalogo dati da Athena richiede di registrare il catalogo come risorsa Athena `DataCatalog`. Per istruzioni, consulta Registrazione di un AWS Glue Data Catalog da un altro account nella Guida per l'utente di Amazon Athena.

## Risoluzione dei problemi relativi all'identità e all'accesso a AWS Glue

Utilizza le seguenti informazioni per aiutarti a diagnosticare e risolvere i problemi più comuni che potresti riscontrare quando lavori con AWS Glue e IAM.

Argomenti

- Non sono autorizzato a eseguire un'azione in AWS Glue
- Non sono autorizzato a eseguire iam: PassRole
- Voglio consentire a persone esterne a me di accedere Account AWS alle mie risorse AWS Glue

## Non sono autorizzato a eseguire un'azione in AWS Glue

Se ricevi un errore che indica che non sei autorizzato a eseguire un'operazione, le tue policy devono essere aggiornate per poter eseguire l'operazione.

L'errore di esempio seguente si verifica quando l'utente IAM `mateojackson` prova a utilizzare la console per visualizzare i dettagli relativi a una risorsa *`my-example-widget`* fittizia ma non dispone di autorizzazioni `glue:`*`GetWidget`* fittizie.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 glue:GetWidget on resource: my-example-widget
```

In questo caso, la policy per l'utente `mateojackson` deve essere aggiornata per consentire l'accesso alla risorsa *`my-example-widget`* utilizzando l'azione `glue:`*`GetWidget`*.

Se hai bisogno di aiuto, contatta il tuo AWS amministratore. L'amministratore è la persona che ti ha fornito le credenziali di accesso.

## Non sono autorizzato a eseguire iam: PassRole

Se ricevi un messaggio di errore indicante che non sei autorizzato a eseguire l'`iam:PassRole`azione, le tue politiche devono essere aggiornate per consentirti di trasferire un ruolo a AWS Glue.

Alcuni Servizi AWS consentono di passare un ruolo esistente a quel servizio invece di creare un nuovo ruolo di servizio o un ruolo collegato al servizio. Per eseguire questa operazione, è necessario disporre delle autorizzazioni per trasmettere il ruolo al servizio.

L'errore di esempio seguente si verifica quando un utente IAM denominato `marymajor` cerca di utilizzare la console per eseguire un'operazione in AWS Glue. Tuttavia, l'azione richiede che il servizio disponga delle autorizzazioni concesse da un ruolo di servizio. Mary non dispone delle autorizzazioni per trasmettere il ruolo al servizio.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In questo caso, le policy di Mary devono essere aggiornate per poter eseguire l'operazione `iam:PassRole`.

Se hai bisogno di aiuto, contatta il tuo AWS amministratore. L'amministratore è la persona che ti ha fornito le credenziali di accesso.

### Voglio consentire a persone esterne a me di accedere Account AWS alle mie risorse AWS Glue

È possibile creare un ruolo con il quale utenti in altri account o persone esterne all'organizzazione possono accedere alle tue risorse. È possibile specificare chi è attendibile per l'assunzione del ruolo. Per i servizi che supportano politiche basate sulle risorse o liste di controllo degli accessi (ACLs), puoi utilizzare tali politiche per consentire alle persone di accedere alle tue risorse.

Per maggiori informazioni, consulta gli argomenti seguenti:

- Per sapere se AWS Glue supporta queste funzionalità, consultaCome funziona AWS Glue con IAM.
- Per scoprire come fornire l'accesso alle tue risorse su Account AWS un sito di tua proprietà, consulta Fornire l'accesso a un utente IAM di un altro Account AWS utente di tua proprietà nella IAM User Guide.
- Per scoprire come fornire l'accesso alle tue risorse a terze parti Account AWS, consulta Fornire l'accesso a soggetti Account AWS di proprietà di terze parti nella Guida per l'utente IAM.
- Per informazioni su come fornire l'accesso tramite la federazione delle identità, consulta Fornire l'accesso a utenti autenticati esternamente (federazione delle identità) nella Guida per l'utente IAM.
- Per informazioni sulle differenze di utilizzo tra ruoli e policy basate su risorse per l'accesso multi-account, consulta Accesso a risorse multi-account in IAM nella Guida per l'utente IAM.

## AWS Lake Formation modelli di controllo degli accessi

AWS Glue 5.0 e versioni successive supportano due modelli per l'accesso ai dati tramite AWS Lake Formation:

Argomenti

- Utilizzo di AWS Glue with AWS Lake Formation per l'accesso completo alla tabella
- Utilizzo di AWS Glue with AWS Lake Formation per un controllo degli accessi a grana fine

# Utilizzo di AWS Glue with AWS Lake Formation per l'accesso completo alla tabella

## Introduzione all'accesso completo alla tabella

AWS Glue 5.0 e versioni successive supportano il controllo Full Table Access (FTA) in Apache Spark in base alle politiche definite in. AWS Lake Formation Questa funzionalità consente le operazioni di lettura e scrittura dei job AWS Glue Spark su tabelle AWS Lake Formation registrate quando il ruolo lavorativo ha accesso completo alla tabella. FTA è ideale per i casi d'uso che devono rispettare le normative di sicurezza a livello di tabella e supporta le funzionalità di Spark, tra cui Resilient Distributed Datasets (RDDs), librerie personalizzate e User Defined Functions () con tabelle. UDFs AWS Lake Formation

Quando un job AWS Glue Spark è configurato per Full Table Access (FTA), AWS Lake Formation le credenziali vengono utilizzate per i dati di read/write Amazon S3 per le tabelle AWS Lake Formation registrate, mentre le credenziali del ruolo di runtime del job verranno utilizzate per le tabelle non registrate con. read/write AWS Lake Formation Questa funzionalità abilita le operazioni di linguaggio di manipolazione dei dati (DML) tra cui le istruzioni CREATE, ALTER, DELETE, UPDATE e MERGE INTO sulle tabelle Apache Hive e Iceberg.

> **Note**
>
> Esaminare i requisiti e determinare se il controllo granulare degli accessi (FGAC) o l'accesso completo alla tabella (FTA) soddisfano le esigenze. È possibile abilitare un solo metodo di AWS Lake Formation autorizzazione per un determinato lavoro. AWS Glue Un processo non può eseguire contemporaneamente l'accesso completo alla tabella (FTA) e il controllo granulare degli accessi (FGAC).

## Come funziona Full-Table Access (FTA) su AWS Glue

AWS Lake Formation offre due approcci per il controllo dell'accesso ai dati: Fine-Grained Access Control (FGAC) e Full Table Access (FTA). FGAC offre una sicurezza avanzata attraverso il filtraggio a livello di colonna, riga e cella, ideale per scenari che richiedono autorizzazioni granulari. FTA è ideale per scenari di controllo degli accessi semplici in cui sono necessarie autorizzazioni a livello di tabella. Semplifica l'implementazione eliminando la necessità di abilitare una modalità di accesso dettagliata, migliora le prestazioni e riduce i costi evitando il driver di sistema e gli esecutori di

sistema e supporta operazioni di lettura e scrittura (inclusi i comandi CREATE, ALTER, DELETE, UPDATE e MERGE INTO).

Nella AWS Glue versione 4.0, l'accesso AWS Lake Formation basato sui dati funzionava tramite GlueContext class, la classe di utilità fornita da. AWS Glue Nella AWS Glue versione 5.0, l'accesso AWS Lake Formation basato ai dati è disponibile tramite Spark SQL nativo, Spark DataFrames e continua a essere supportato tramite la GlueContext classe.

## Implementazione dell'accesso completo alla tabella

Fase 1: Abilita l'accesso completo alla tabella in AWS Lake Formation

Per utilizzare la modalità di accesso completo alla tabella (FTA), è necessario consentire ai motori di query di terze parti di accedere ai dati senza la convalida dei tag di sessione IAM in AWS Lake Formation. Per abilitarli, seguire i passaggi descritti in Integrazione delle applicazioni per l'accesso completo alla tabella.

Fase 2: Configurazione delle autorizzazioni IAM per il ruolo di runtime dei processi

Per l'accesso in lettura o scrittura ai dati sottostanti, oltre alle AWS Lake Formation autorizzazioni, un ruolo di job runtime richiede l'autorizzazione `lakeformation:GetDataAccess` IAM. Con questa autorizzazione, AWS Lake Formation concede la richiesta di credenziali temporanee per accedere ai dati.

Di seguito è riportato una policy esemplificativa su come fornire le autorizzazioni IAM per accedere a uno script in Amazon S3, caricare i log su Amazon S3, le autorizzazioni dell'API AWS Glue e l'autorizzazione per accedere a AWS Lake Formation.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "ScriptAccess",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket/scripts/*"
```

```
      ]
    },
    {
      "Sid": "LoggingAccess",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket/logs/*"
      ]
    },
    {
      "Sid": "GlueCatalogAccess",
      "Effect": "Allow",
      "Action": [
        "glue:GetDatabase",
        "glue:GetDatabases",
        "glue:GetTable",
        "glue:GetTables",
        "glue:GetPartition",
        "glue:GetPartitions",
        "glue:CreateTable",
        "glue:UpdateTable"
      ],
      "Resource": [
        "arn:aws:glue:us-east-1:111122223333:catalog",
        "arn:aws:glue:us-east-1:111122223333:database/default",
        "arn:aws:glue:us-east-1:111122223333:table/default/*"
      ]
    },
    {
      "Sid": "LakeFormationAccess",
      "Effect": "Allow",
      "Action": [
        "lakeformation:GetDataAccess"
      ],
      "Resource": "*"
    }
  ]
}
```

Passaggio 2.1 Configurare le autorizzazioni AWS Lake Formation

AWS Glue I job Spark che leggono dati da Amazon S3 AWS Lake Formation richiedono l'autorizzazione SELECT.

AWS Glue Spark fa in modo che write/delete i dati in Amazon S3 AWS Lake Formation richiedano l'autorizzazione ALL.

AWS Glue I job Spark che interagiscono con AWS Glue Data catalog richiedono le autorizzazioni DESCRIBE, ALTER, DROP, a seconda dei casi.

Fase 3: Inizializza una sessione Spark per l'accesso completo alla tabella utilizzando AWS Lake Formation

Per accedere alle tabelle registrate con AWS Lake Formation, è necessario impostare le seguenti configurazioni durante l'inizializzazione di Spark per configurare Spark per l'utilizzo delle credenziali. AWS Lake Formation

Per accedere alle tabelle registrate con AWS Lake Formation, devi configurare in modo esplicito la sessione Spark per utilizzare le credenziali. AWS Lake Formation Aggiungere le seguenti configurazioni durante l'inizializzazione della sessione Spark:

```
from pyspark.sql import SparkSession

# Initialize Spark session with Lake Formation configurations
spark = SparkSession.builder \
    .appName("Lake Formation Full Table Access") \
    .config("spark.sql.catalog.glue_catalog",
 "org.apache.spark.sql.catalog.hive.GlueCatalog") \
    .config("spark.sql.catalog.glue_catalog.glue.lakeformation-enabled", "true") \
    .config("spark.sql.defaultCatalog", "glue_catalog") \
    .getOrCreate()
```

Configurazioni principali:

- `spark.sql.catalog.glue_catalog`: Registra un catalogo denominato «glue_catalog» che utilizza l'implementazione GlueCatalog
- `spark.sql.catalog.glue_catalog.glue.lakeformation-enabled`: AWS Lake Formation abilita esplicitamente l'integrazione per questo catalogo
- Il nome del catalogo ("glue_catalog" in questo esempio) può essere personalizzato, ma deve essere coerente in entrambe le impostazioni di configurazione

Hive

```
--conf spark.hadoop.fs.s3.credentialsResolverClass=com.amazonaws.AWS
 Glue.accesscontrol.AWSLakeFormationCredentialResolver
--conf spark.hadoop.fs.s3.useDirectoryHeaderAsFolderObject=true
--conf spark.hadoop.fs.s3.folderObject.autoAction.disabled=true
--conf spark.sql.catalog.skipLocationValidationOnCreateTable.enabled=true
--conf spark.sql.catalog.createDirectoryAfterTable.enabled=true
--conf spark.sql.catalog.dropDirectoryBeforeTable.enabled=true
```

Iceberg

```
--conf spark.hadoop.fs.s3.credentialsResolverClass=com.amazonaws.AWS
 Glue.accesscontrol.AWSLakeFormationCredentialResolver
--conf spark.hadoop.fs.s3.useDirectoryHeaderAsFolderObject=true
--conf spark.hadoop.fs.s3.folderObject.autoAction.disabled=true
--conf spark.sql.catalog.skipLocationValidationOnCreateTable.enabled=true
--conf spark.sql.catalog.createDirectoryAfterTable.enabled=true
--conf spark.sql.catalog.dropDirectoryBeforeTable.enabled=true
--conf spark.sql.catalog.<catalog>.AWS Glue.lakeformation-enabled=true
```

- `spark.hadoop.fs.s3.credentialsResolverClass=com.amazonaws.AWS Glue.accesscontrol.AWSLakeFormationCredentialResolver`: configura EMR Filesystem (EMRFS) per utilizzare le credenziali S3 per le tabelle registrate. AWS Lake Formation AWS Lake Formation Se la tabella non è registrata, utilizzare le credenziali del ruolo di runtime del processo.
- `spark.hadoop.fs.s3.useDirectoryHeaderAsFolderObject=true` e `spark.hadoop.fs.s3.folderObject.autoAction.disabled=true`: configurano EMRFS per utilizzare l'applicazione di intestazione di tipo di contenuto/directory x invece del suffisso $folder $ durante la creazione di cartelle S3. Ciò è necessario per la lettura delle tabelle, poiché le AWS Lake Formation credenziali non consentono la lettura di cartelle di AWS Lake Formation tabelle con il suffisso $folder$.
- `spark.sql.catalog.skipLocationValidationOnCreateTable.enabled=true`: configura Spark per saltare la convalida del vuoto della posizione della tabella prima della creazione. Ciò è necessario per le tabelle AWS Lake Formation registrate, poiché AWS Lake

Formation le credenziali per verificare la posizione vuota sono disponibili solo dopo la creazione della tabella Data Catalog. AWS Glue Senza questa configurazione, le credenziali del ruolo di runtime del processo convalideranno la posizione della tabella vuota.

- `spark.sql.catalog.createDirectoryAfterTable.enabled=true`: configura Spark per creare la cartella Amazon S3 dopo la creazione della tabella nel metastore Hive. Questo è necessario per le tabelle AWS Lake Formation registrate, poiché AWS Lake Formation le credenziali per creare la cartella Amazon S3 sono disponibili solo AWS Glue dopo la creazione della tabella Data Catalog.
- `spark.sql.catalog.dropDirectoryBeforeTable.enabled=true`: configura Spark per cancellare la cartella Amazon S3 prima dell'eliminazione della tabella nel metastore Hive. Ciò è necessario per le tabelle AWS Lake Formation registrate, poiché AWS Lake Formation le credenziali per eliminare la cartella S3 non sono disponibili dopo l'eliminazione della tabella dal Data Catalog. AWS Glue
- `spark.sql.catalog.<catalog>.AWS Glue.lakeformation-enabled=true`: configura il catalogo Iceberg per utilizzare le credenziali AWS Lake Formation AWS Lake Formation Amazon S3 per le tabelle registrate. Se la tabella non è registrata, usare le credenziali di ambiente predefinite.

## Modelli di utilizzo

### Utilizzo di FTA con DataFrames

Per gli utenti che hanno familiarità con Spark, DataFrames può essere utilizzato con AWS Lake Formation Full Table Access.

AWS Glue 5.0 aggiunge il supporto Spark nativo per Lake Formation Full Table Access, semplificando il modo in cui lavori con le tabelle protette. Questa funzionalità consente ai job AWS Glue 5.0 AWS Glue Spark di leggere e scrivere direttamente i dati quando viene concesso l'accesso completo alla tabella, eliminando le limitazioni che in precedenza limitavano determinate operazioni di estrazione, trasformazione e caricamento (ETL). Ora puoi sfruttare le funzionalità avanzate di Spark, tra cui Resilient Distributed Datasets (RDDs), librerie personalizzate e User Defined Functions () con tabelle. UDFs AWS Lake Formation

### Native Spark FTA nella versione 5.0 AWS Glue

AWS Glue 5.0 supporta il controllo dell'accesso completo alla tabella (FTA) in Apache Spark in base alle politiche definite in. AWS Lake Formation Questo livello di controllo è ideale per i casi d'uso che devono rispettare le norme di sicurezza a livello di tabella.

### Esempio di tabella di Apache Iceberg

```
from pyspark.sql import SparkSession

catalog_name = "spark_catalog"
aws_region = "us-east-1"
aws_account_id = "123456789012"
warehouse_path = "s3://amzn-s3-demo-bucket/warehouse/"

spark = SparkSession.builder \

 .config("spark.sql.extensions","org.apache.iceberg.spark.extensions.IcebergSparkSessionExtensi
 \
    .config(f"spark.sql.catalog.{catalog_name}",
 "org.apache.iceberg.spark.SparkSessionCatalog") \
    .config(f"spark.sql.catalog.{catalog_name}.warehouse", f"{warehouse_path}") \
    .config(f"spark.sql.catalog.{catalog_name}.client.region",f"{aws_region}") \
    .config(f"spark.sql.catalog.{catalog_name}.glue.account-id",f"{aws_account_id}") \
    .config(f"spark.sql.catalog.{catalog_name}.glue.lakeformation-enabled","true") \
    .config(f"spark.sql.catalog.dropDirectoryBeforeTable.enabled", "true") \
    .config(f"spark.sql.catalog.{catalog_name}.catalog-impl",
 "org.apache.iceberg.aws.glue.GlueCatalog") \
    .config(f"spark.sql.catalog.{catalog_name}.io-impl",
 "org.apache.iceberg.aws.s3.S3FileIO") \
    .config("spark.sql.defaultCatalog", catalog_name) \  # Add this line
    .getOrCreate()

database_name = "your_database"
table_name = "your_table"

df = spark.sql(f"select * from {database_name}.{table_name}")
df.show()
```

### Autorizzazioni IAM richieste

Il tuo AWS Glue ruolo di esecuzione del lavoro deve avere:

```
{
    "Action": "lakeformation:GetDataAccess",
    "Resource": "*",
    "Effect": "Allow"
}
```

Autorizzazioni di accesso S3 appropriate per le posizioni dei dati.

### Configurazione di Lake Formation

Prima di utilizzare Spark FTA nativo nella AWS Glue versione 5.0:

1. Consenti ai motori di query di terze parti di accedere ai dati senza la convalida dei tag di sessione IAM in AWS Lake Formation
2. Concedi le autorizzazioni di tabella appropriate al tuo ruolo di esecuzione del AWS Glue lavoro tramite console AWS Lake Formation
3. Configurare la sessione Spark con i parametri richiesti mostrati nell'esempio precedente

### Utilizzo di FTA con DynamicFrames

AWS Glue DynamicFrames può essere utilizzato con AWS Lake Formation Full Table Access per operazioni ETL ottimizzate. L'accesso completo alla tabella (FTA) fornisce un modello di sicurezza che concede autorizzazioni a livello di tabella, consentendo un'elaborazione dei dati più rapida rispetto al controllo granulare degli accessi (FGAC) poiché aggira il sovraccarico dei controlli delle autorizzazioni a livello di riga e colonna. Questo approccio è utile quando è necessario elaborare intere tabelle e le autorizzazioni a livello di tabella soddisfano i requisiti di sicurezza.

Nella AWS Glue versione 4.0, DynamicFrames con FTA richiedeva una configurazione specifica GlueContext . Sebbene il DynamicFrame codice AWS Glue 4.0 esistente con FTA continuerà a funzionare nella versione AWS Glue 5.0, la versione più recente offre anche il supporto nativo di Spark FTA con maggiore flessibilità. Per un nuovo sviluppo, prendi in considerazione l'utilizzo dell'approccio nativo Spark descritto nella DataFrames sezione, soprattutto se hai bisogno di funzionalità aggiuntive come Resilient Distributed Datasets (RDDs), librerie personalizzate e Funzioni definite dall'utente () con tabelle. UDFs AWS Lake Formation

### Autorizzazioni richieste

Il ruolo IAM che esegue il processo Glue deve avere:

- Autorizzazione `lakeformation:GetDataAccess`
- Autorizzazioni appropriate per la tabella di Lake Formation concesse tramite la console Lake Formation

Esempio DynamicFrame di implementazione in 5.0 AWS Glue

```
from awsglue.context import GlueContext
from pyspark.context import SparkContext

# Initialize Glue context
sc = SparkContext()
glueContext = GlueContext(sc)

# Configure catalog for Iceberg tables
catalog_name = "glue_catalog"
aws_region = "us-east-1"
aws_account_id = "123456789012"
warehouse_path = "s3://amzn-s3-demo-bucket/warehouse/"

spark = glueContext.spark_session
spark.conf.set(f"spark.sql.catalog.{catalog_name}",
 "org.apache.iceberg.spark.SparkCatalog")
spark.conf.set(f"spark.sql.catalog.{catalog_name}.warehouse", f"{warehouse_path}")
spark.conf.set(f"spark.sql.catalog.{catalog_name}.catalog-impl",
 "org.apache.iceberg.aws.glue.GlueCatalog")
spark.conf.set(f"spark.sql.catalog.{catalog_name}.io-impl",
 "org.apache.iceberg.aws.s3.S3FileIO")
spark.conf.set(f"spark.sql.catalog.{catalog_name}.glue.lakeformation-enabled","true")
spark.conf.set(f"spark.sql.catalog.{catalog_name}.client.region",f"{aws_region}")
spark.conf.set(f"spark.sql.catalog.{catalog_name}.glue.id", f"{aws_account_id}")

# Read Lake Formation-protected table with DynamicFrame
df = glueContext.create_data_frame.from_catalog(
    database="your_database",
    table_name="your_table"
)
```

## Configurazione aggiuntiva

Configura la modalità di accesso completo alla tabella nei AWS Glue Studio notebook

Per accedere alle tabelle AWS Lake Formation registrate dalle sessioni interattive di Spark nei AWS Glue Studio notebook, devi utilizzare la modalità di autorizzazione alla compatibilità. Usare il comando `%%configure` magic per impostare la configurazione di Spark prima di iniziare la sessione interattiva. Questa configurazione deve essere il primo comando del notebook, in quanto non può essere applicata dopo l'inizio della sessione. Scegliere la configurazione in base al tipo di tabella:

Per le tabelle Hive

```
%%configure
--conf
 spark.hadoop.fs.s3.credentialsResolverClass=com.amazonaws.glue.accesscontrol.AWSLakeFormationC
--conf spark.hadoop.fs.s3.useDirectoryHeaderAsFolderObject=true
--conf spark.hadoop.fs.s3.folderObject.autoAction.disabled=true
--conf spark.sql.catalog.skipLocationValidationOnCreateTable.enabled=true
--conf spark.sql.catalog.createDirectoryAfterTable.enabled=true
--conf spark.sql.catalog.dropDirectoryBeforeTable.enabled=true
```

Per le tabelle Iceberg

```
%%configure
--conf
 spark.hadoop.fs.s3.credentialsResolverClass=com.amazonaws.glue.accesscontrol.AWSLakeFormationC
--conf spark.hadoop.fs.s3.useDirectoryHeaderAsFolderObject=true
--conf spark.hadoop.fs.s3.folderObject.autoAction.disabled=true
--conf spark.sql.catalog.skipLocationValidationOnCreateTable.enabled=true
--conf spark.sql.catalog.createDirectoryAfterTable.enabled=true
--conf spark.sql.catalog.dropDirectoryBeforeTable.enabled=true
--conf spark.sql.catalog.glue_catalog.glue.lakeformation-enabled=true
--conf spark.sql.catalog.glue_catalog.warehouse=s3://example-s3-bucket_DATA_LOCATION
--conf spark.sql.catalog.glue_catalog.catalog-
impl=org.apache.iceberg.aws.glue.GlueCatalog
--conf spark.sql.catalog.glue_catalog.io-impl=org.apache.iceberg.aws.s3.S3FileIO
--conf spark.sql.catalog.glue_catalog.glue.account-id=ACCOUNT_ID
--conf spark.sql.catalog.glue_catalog.glue.region=REGION
```

Sostituire i segnaposto:

- S3_DATA_LOCATION: *s3://amzn-s3-demo-bucket*
- REGIONE: *AWS Region (e.g., us-east-1)*
- ID ACCOUNT: *Your AWS Account ID*

> **Note**
>
> È necessario impostare queste configurazioni prima di eseguire qualsiasi operazione Spark sul notebook.

Operazioni supportate

Queste operazioni utilizzeranno le AWS Lake Formation credenziali per accedere ai dati della tabella.

> **Note**
>
> Sull'abilitazione AWS Lake Formation:
>
> - Per FTA: abilitare la configurazione di Spark `spark.sql.catalog.{catalog_name}.glue.lakeformation-enabled`

- CREATE TABLE
- ALTER TABLE
- INSERT INTO
- INSERT OVERWRITE
- SELECT
- UPDATE
- MERGE INTO
- DELETE FROM
- ANALYZE TABLE
- REPAIR TABLE
- DROP TABLE
- Query su origini dati Spark
- Scritture di origini dati Spark

> **Note**
>
> Le operazioni non elencate sopra continueranno a utilizzare le autorizzazioni IAM per accedere ai dati delle tabelle.

## Migrazione da AWS Glue 4.0 a AWS Glue 5.0 FTA

Durante la migrazione da AWS Glue 4.0 GlueContext FTA a Spark FTA AWS Glue 5.0 nativo:

1. Consentire ai motori di query di terze parti di accedere ai dati senza la convalida dei tag di sessione IAM in AWS Lake Formation. Seguire Fase 1: Abilita l'accesso completo alla tabella in AWS Lake Formation.
2. Non è necessario cambiare il ruolo di runtime del processo. Tuttavia, verifica che il ruolo di esecuzione del AWS Glue lavoro disponga dell'autorizzazione lakeformation: IAM. GetDataAccess
3. Modificare le configurazioni delle sessioni Spark nello script. Assicurarsi che siano presenti le seguenti configurazioni di Spark:

```
--conf spark.sql.catalog.spark_catalog=org.apache.iceberg.spark.SparkSessionCatalog
--conf spark.sql.catalog.spark_catalog.warehouse=s3://<bucket-name>/warehouse/
--conf spark.sql.catalog.spark_catalog.client.region=<REGION>
--conf spark.sql.catalog.spark_catalog.glue.account-id=ACCOUNT_ID
--conf spark.sql.catalog.spark_catalog.glue.lakeformation-enabled=true
--conf spark.sql.catalog.dropDirectoryBeforeTable.enabled=true
```

4. Aggiorna lo script in modo che GlueContext DataFrames venga modificato in Spark nativo. DataFrames
5. Aggiorna il tuo AWS Glue job per usare 5.0 AWS Glue

## Considerazioni e limitazioni

- Se una tabella Hive viene creata utilizzando un processo per cui non è abilitato l'accesso completo alla tabella e non viene inserito alcun record, le letture o scritture successive da un processo con accesso completo alla tabella avranno esito negativo. Questo perché AWS Glue Spark senza accesso completo alla tabella aggiunge il suffisso $folder$ al nome della cartella della tabella. Per risolvere questo problema, è possibile procedere in questi modi:
  - Inserire almeno una riga nella tabella da un processo che non ha FTA abilitato.
  - Configurare il processo per cui FTA non è abilitato in modo che non utilizzi il suffisso $folder$ nel nome della cartella in S3. Ciò si può ottenere impostando la configurazione `spark.hadoop.fs.s3.useDirectoryHeaderAsFolderObject=true` di Spark.
  - Creare una cartella Amazon S3 nella posizione delle tabelle `s3://path/to/table/table_name` utilizzando la console Amazon S3 o la CLI di Amazon S3
- L'accesso completo alla tabella funziona esclusivamente con EMR Filesystem (EMRFS). Il file system S3A non è compatibile.
- L'accesso completo alla tabella è supportato per le tabelle Hive e Iceberg. Il supporto per le tabelle Hudi e Delta non è stato ancora aggiunto.

- I lavori che fanno riferimento a tabelle con regole FGAC ( AWS Lake Formation Fine-Grained Access Control) o Data Catalog Views avranno esito negativo. AWS Glue Per interrogare una tabella con regole FGAC o AWS Glue Data Catalog View, è necessario utilizzare la modalità FGAC. È possibile abilitare la modalità FGAC seguendo i passaggi descritti nella AWS documentazione: Utilizzo con per un controllo granulare degli accessi. AWS Glue AWS Lake Formation
- L'accesso completo alla tabella non supporta Spark Streaming.
- Non può essere utilizzato contemporaneamente a FGAC.

# Utilizzo di AWS Glue with AWS Lake Formation per un controllo degli accessi a grana fine

## Panoramica di

Con AWS la versione 5.0 e successive di Glue, puoi sfruttare AWS Lake Formation per applicare controlli di accesso granulari alle tabelle di Data Catalog supportate da S3. Questa capacità consente di configurare i controlli di accesso a livello di tabella, riga, colonna e cella per read query all'interno dei processi di AWS Glue per Apache Spark. Consulta le seguenti sezioni per saperne di più su Lake Formation e su come usarlo con AWS Glue.

`GlueContext`il controllo degli accessi a livello di tabella con AWS Lake Formation autorizzazioni supportate in Glue 4.0 o versioni precedenti non è supportato in Glue 5.0. Usare il nuovo controllo granulare degli accessi (FGAC) nativo di Spark in Glue 5.0. Si notino i seguenti dettagli:

- Se hai bisogno di un controllo degli accessi a grana fine (FGAC) per il controllo degli row/column/cell accessi, dovrai migrare da /Glue in Glue 4.0 e versioni precedenti al dataframe Spark `GlueContext` in DynamicFrame Glue 5.0. Per alcuni esempi, consultare Migrazione da GlueContext/Glue DynamicFrame Spark DataFrame.
- Se hai bisogno del controllo Full Table Access (FTA), puoi sfruttare FTA con DynamicFrames AWS Glue 5.0. Puoi anche migrare all'approccio Spark nativo per funzionalità aggiuntive come Resilient Distributed Datasets (RDDs), librerie personalizzate e funzioni definite dall'utente () con tabelle. UDFs AWS Lake Formation Per esempi, vedi Migrazione da AWS Glue 4.0 a AWS Glue 5.0.
- Se non è necessario il FGAC, non occorre eseguire la migrazione al dataframe Spark e le funzionalità di `GlueContext`, come i segnalibri di processo e i predicati push down, continueranno a funzionare.

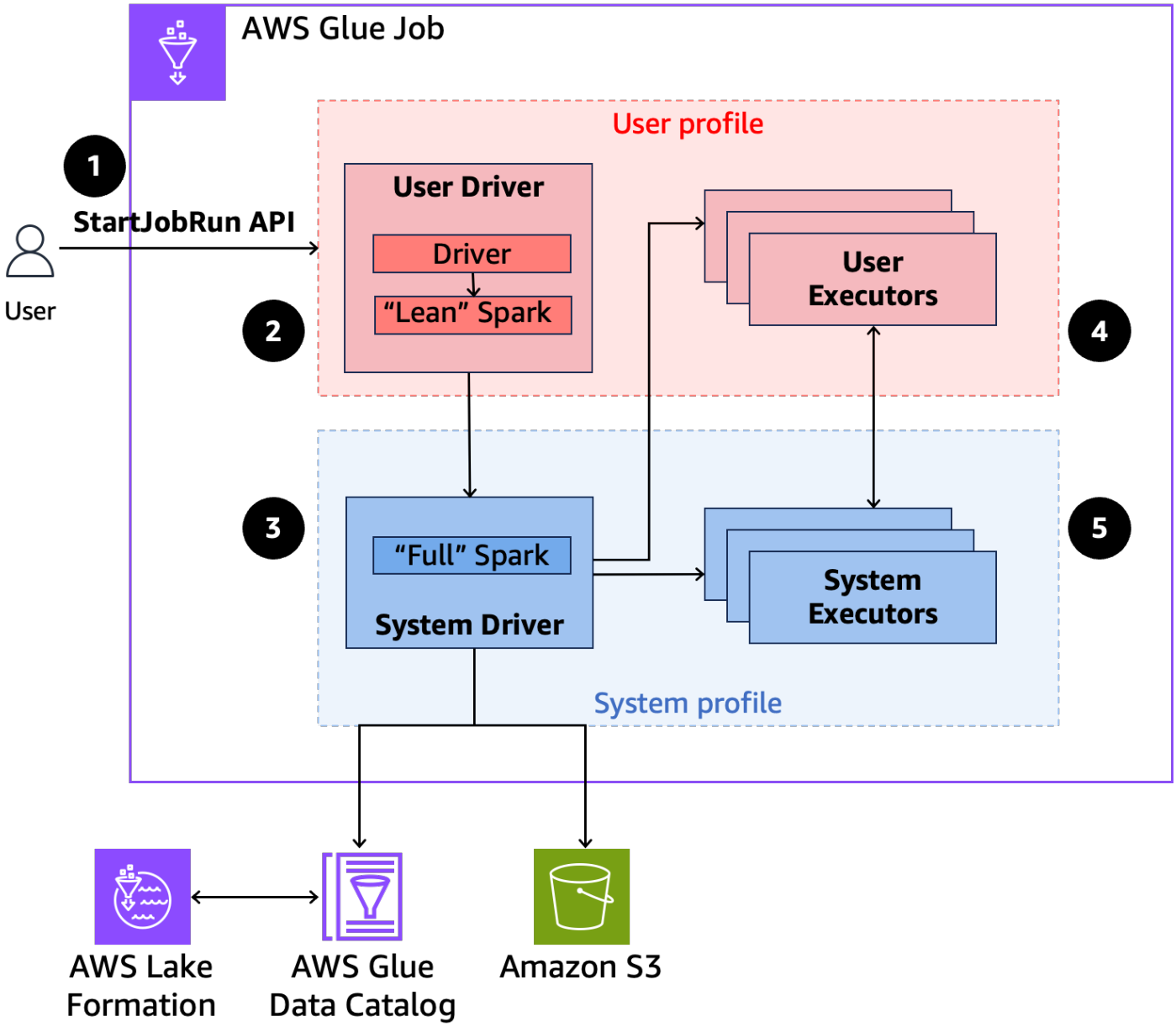
- I processi con FGAC un minimo di 4 worker: un driver utente, un driver di sistema, un esecutore di sistema e un esecutore utente in standby.

L'utilizzo di AWS AWS Lake Formation Glue with comporta costi aggiuntivi.

## Come funziona AWS Glue con AWS Lake Formation

L'utilizzo di AWS Glue with Lake Formation ti consente di applicare un livello di autorizzazioni su ogni lavoro Spark per applicare il controllo delle autorizzazioni di Lake Formation quando AWS Glue esegue i lavori. AWS Glue utilizza i profili di risorse Spark per creare due profili per eseguire i lavori in modo efficace. Il profilo utente esegue il codice fornito dall'utente, mentre il profilo di sistema applica le policy di Lake Formation. Per ulteriori informazioni, consultare Cos'è AWS Lake Formation e Considerazioni e limitazioni.

Di seguito è riportata una panoramica di alto livello su come AWS Glue ottiene l'accesso ai dati protetti dalle politiche di sicurezza di Lake Formation.




1. Un utente chiama l'`StartJobRun`API su un lavoro AWS Glue abilitato per AWS Lake Formation.
2. AWS Glue invia il lavoro a un driver utente ed esegue il lavoro nel profilo utente. Il driver utente esegue una versione snella di Spark che non è in grado di avviare attività, richiedere esecutori, accedere a S3 o al catalogo di Glue. Costruisce un piano per il processo.
3. AWS Glue imposta un secondo driver chiamato driver di sistema e lo esegue nel profilo di sistema (con un'identità privilegiata). AWS Glue imposta un canale TLS crittografato tra i due driver per la comunicazione. Il driver utente utilizza il canale per inviare i piani per il processo al driver di sistema. Il driver di sistema non esegue il codice inviato dall'utente. Esegue Spark

completo e comunica con S3 e il catalogo dati per l'accesso ai dati. Richiede esecutori e compila il piano per il processo in una sequenza di fasi di esecuzione.

4. AWS Glue esegue quindi le fasi sugli executor con il driver utente o il driver di sistema. Il codice utente in qualsiasi fase viene eseguito esclusivamente sugli esecutori dei profili utente.
5. Le fasi che leggono i dati dalle tabelle del Data Catalog protette da AWS Lake Formation o che applicano filtri di sicurezza vengono delegate agli esecutori di sistema.

## Requisito minimo dei worker

Un job abilitato per Lake Formation in AWS Glue richiede un minimo di 4 lavoratori: un driver utente, un driver di sistema, un esecutore di sistema e un esecutore utente in standby. Questo è un aumento rispetto al minimo di 2 lavoratori richiesti per i lavori standard di AWS Glue.

Un job abilitato per Lake Formation in AWS Glue utilizza due driver Spark, uno per il profilo di sistema e l'altro per il profilo utente. Analogamente, anche gli esecutori sono divisi in due profili:

- Esecutori di sistema: gestiscono le attività in cui vengono applicati i filtri dati di Lake Formation.
- Esecutori utente: vengono richiesti dal driver di sistema in base alle esigenze.

Poiché i lavori Spark sono di natura pigra, AWS Glue riserva il 10% del totale dei lavoratori (minimo 1), dopo aver detratto i due driver, agli esecutori degli utenti.

Tutti i processi abilitati da Lake Formation hanno il dimensionamento automatico abilitato, il che significa che gli esecutori utente verranno avviati solo quando necessario.

Per un esempio di configurazione, consultare Considerazioni e limitazioni.

## Autorizzazioni IAM per il ruolo di runtime del processo

Le autorizzazioni di Lake Formation controllano l'accesso alle risorse di AWS Glue Data Catalog, alle sedi Amazon S3 e ai dati sottostanti in tali sedi. Le autorizzazioni IAM controllano l'accesso a Lake Formation and AWS Glue APIs e alle risorse. Anche se disponi dell'autorizzazione di Lake Formation per accedere a una tabella nel Data Catalog (SELECT), l'operazione fallisce se non disponi dell'autorizzazione IAM per il funzionamento dell'API `glue:Get*`.

Di seguito è riportato un esempio di policy su come fornire le autorizzazioni IAM per accedere a uno script in S3, caricare i log su S3, le autorizzazioni dell'API AWS Glue e l'autorizzazione per accedere a Lake Formation.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "ScriptAccess",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::*.amzn-s3-demo-bucket/scripts",
        "arn:aws:s3:::*.amzn-s3-demo-bucket/*"
      ]
    },
    {
      "Sid": "LoggingAccess",
      "Effect": "Allow",
      "Action": [
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket/logs/*"
      ]
    },
    {
      "Sid": "GlueCatalogAccess",
      "Effect": "Allow",
      "Action": [
        "glue:Get*",
        "glue:Create*",
        "glue:Update*"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Sid": "LakeFormationAccess",
      "Effect": "Allow",
      "Action": [
```

```
        "lakeformation:GetDataAccess"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

## Configurazione delle autorizzazioni di Lake Formation per il ruolo di runtime del processo

Innanzitutto, registrare la posizione della tabella Hive con Lake Formation. Poi, creare le autorizzazioni per il ruolo di runtime del processo nella tabella desiderata. Per maggiori dettagli su Lake Formation, vedi What is AWS Lake Formation? nella Guida per gli AWS Lake Formation sviluppatori.

Dopo aver impostato le autorizzazioni di Lake Formation, è possibile inviare processi Spark su AWS Glue.

## Invio di un'esecuzione di processo

Dopo aver completato l'impostazione delle sovvenzioni Lake Formation, puoi inviare lavori Spark su AWS Glue. Per eseguire i processi Iceberg, è necessario fornire le seguenti configurazioni Spark. Per eseguire la configurazione tramite i parametri del processo Glue, inserire il seguente parametro:

- Chiave:

```
--conf
```

- Valore:

```
spark.sql.catalog.spark_catalog=org.apache.iceberg.spark.SparkSessionCatalog
       --conf spark.sql.catalog.spark_catalog.warehouse=<S3_DATA_LOCATION>
       --conf spark.sql.catalog.spark_catalog.glue.account-id=<ACCOUNT_ID>
       --conf spark.sql.catalog.spark_catalog.client.region=<REGION>
       --conf spark.sql.catalog.spark_catalog.glue.endpoint=https://
glue.<REGION>.amazonaws.com
```

## Utilizzo di una sessione interattiva

Dopo aver completato l'impostazione delle AWS Lake Formation sovvenzioni, puoi utilizzare Interactive Sessions on AWS Glue. È necessario fornire le seguenti configurazioni di Spark tramite il comando magic `%%configure` prima di eseguire il codice.

```
%%configure
{
    "--enable-lakeformation-fine-grained-access": "true",
    "--conf":
 "spark.sql.catalog.spark_catalog=org.apache.iceberg.spark.SparkSessionCatalog
 --conf spark.sql.catalog.spark_catalog.warehouse=<S3_DATA_LOCATION> --conf
 spark.sql.catalog.spark_catalog.catalog-impl=org.apache.iceberg.aws.glue.GlueCatalog
 --conf spark.sql.catalog.spark_catalog.io-impl=org.apache.iceberg.aws.s3.S3FileIO --
conf
 spark.sql.extensions=org.apache.iceberg.spark.extensions.IcebergSparkSessionExtensions
 --conf spark.sql.catalog.spark_catalog.client.region=<REGION> --conf
 spark.sql.catalog.spark_catalog.glue.account-id=<ACCOUNT_ID> --conf
 spark.sql.catalog.spark_catalog.glue.endpoint=https://glue.<REGION>.amazonaws.com"
}
```

## Notebook FGAC for AWS Glue 5.0 o sessioni interattive

Per abilitare Fine-Grained Access Control (FGAC) in AWS Glue devi specificare le configurazioni Spark richieste per Lake Formation come parte di %%configure magic prima di creare la prima cella.

Specificarlo in un secondo momento utilizzando le chiamate `SparkSession.builder().conf("").get()` o `SparkSession.builder().conf("").create()` non sarà sufficiente. Questa è una modifica rispetto al comportamento di AWS Glue 4.0.

## Supporto per il formato a tabella aperta

AWS La versione 5.0 o successiva di Glue include il supporto per il controllo granulare degli accessi basato su Lake Formation. AWS Glue supporta i tipi di tabelle Hive e Iceberg. Nella tabella seguente sono descritte tutte le operazioni supportate.

| Operazioni | Hive | Iceberg |
| --- | --- | --- |
| Comandi DDL | Solo con autorizzazioni del ruolo IAM | Solo con autorizzazioni del ruolo IAM |

| Operazioni | Hive | Iceberg |
| --- | --- | --- |
| Query incrementali | Non applicabile | Completamente supportata |
| Query temporali | Non applicabile a questo formato di tabella | Completamente supportata |
| Tabelle dei metadati | Non applicabile a questo formato di tabella | Supportata, ma alcune tabelle sono nascoste. Per ulteriori informazioni, consultare Considerazioni e limitazioni. |
| `DML INSERT` | Solo con autorizzazioni IAM | Solo con autorizzazioni IAM |
| AGGIORNAMENTO DML | Non applicabile a questo formato di tabella | Solo con autorizzazioni IAM |
| `DML DELETE` | Non applicabile a questo formato di tabella | Solo con autorizzazioni IAM |
| Operazioni di lettura | Completamente supportata | Completamente supportata |
| Stored procedure | Non applicabile | Supportata con le eccezioni di `register_table` e `migrate`. Per ulteriori informazioni, consultare Considerazioni e limitazioni. |

## Migrazione da GlueContext/Glue DynamicFrame Spark DataFrame

Di seguito sono riportati esempi in Python e Scala di migrazione di `GlueContext`/Glue `DynamicFrame` in Glue 4.0 a Spark `DataFrame` in Glue 5.0.

Python

Prima:

```
escaped_table_name= ``<dbname>`.`<table_name>``

additional_options = {
```

```
  "query": f'select * from {escaped_table_name} WHERE column1 = 1 AND column7 = 7'
}

# DynamicFrame example
dataset = glueContext.create_data_frame_from_catalog(
    database="<dbname>",
    table_name=escaped_table_name,
    additional_options=additional_options)
```

Dopo:

```
table_identifier= '`<catalogname>`.`<dbname>`.`<table_name>`' #catalogname is optional

# DataFrame example
dataset = spark.sql(f'select * from {table_identifier} WHERE column1 = 1 AND column7 =
 7')
```

Scala

Prima:

```
val escapedTableName = "`<dbname>`.`<table_name>`"

val additionalOptions = JsonOptions(Map(
    "query" -> s"select * from $escapedTableName WHERE column1 = 1 AND column7 = 7"
    )
)

# DynamicFrame example
val datasource0 = glueContext.getCatalogSource(
    database="<dbname>",
    tableName=escapedTableName,
    additionalOptions=additionalOptions).getDataFrame()
```

Dopo:

```
val tableIdentifier = "`<catalogname>`.`<dbname>`.`<table_name>`" //catalogname is
 optional

# DataFrame example
val datasource0 = spark.sql(s"select * from $tableIdentifier WHERE column1 = 1 AND
 column7 = 7")
```

## Considerazioni e limitazioni

Quando si utilizza Lake Formation con AWS Glue, tenere conto delle seguenti considerazioni e limitazioni.

- AWS Glue with Lake Formation è disponibile in tutte le regioni supportate.
- AWS Glue supporta il controllo granulare degli accessi tramite Lake Formation solo per le tabelle Apache Hive e Apache Iceberg. I formati Apache Hive includono Parquet, ORC e CSV.
- Si può usare Lake Formation solo con i processi Spark.
- AWS Glue with Lake Formation supporta solo una singola sessione Spark per tutta la durata di un lavoro.
- Quando Lake Formation è abilitato,AWS Glue richiede un numero maggiore di worker perché richiede un driver di sistema, esecutori di sistema, un driver utente e, facoltativamente, gli esecutori utente (necessari quando il lavoro ha UDF o). `spark.createDataFrame`
- AWS Glue with Lake Formation supporta solo le query tabellari tra account condivise tramite link alle risorse. Il link alla risorsa deve avere lo stesso nome della risorsa dell'account di origine.
- Per abilitare il controllo granulare degli accessi per i lavori AWS Glue, passate il `--enable-lakeformation-fine-grained-access` parametro job.
- Puoi configurare i tuoi lavori AWS Glue in modo che funzionino con la gerarchia multicatalogo di AWS Glue. Per informazioni sui parametri di configurazione da utilizzare con l'`StartJobRun`API AWS Glue, vedere Working with AWS Glue multi-catalog hierarchy on EMR Serverless.
- I seguenti elementi non sono supportati:
  - Resilient Distributed Dataset (RDD)
  - Streaming di Spark
  - Scrivere con le autorizzazioni concesse da Lake Formation
  - Controllo degli accessi per colonne annidate
- AWS Glue blocca le funzionalità che potrebbero compromettere il completo isolamento del driver di sistema, tra cui:
  - UDT, HiveUDF e qualsiasi funzione definita dall'utente che coinvolga classi personalizzate
  - Origini dati personalizzate
  - Fornitura di jar aggiuntivi per l'estensione, il connettore o il metastore di Spark
  - Comando `ANALYZE TABLE`

- Per applicare i controlli di accesso, `EXPLAIN PLAN` e le operazioni DDL, come `DESCRIBE TABLE` non espongono informazioni riservate.
- AWS Glue limita l'accesso ai registri Spark del driver di sistema sulle applicazioni Lake. Formation-enabled Poiché il driver di sistema viene eseguito con più accessi, gli eventi e i log generati da quest'ultimo possono includere informazioni riservate. Per impedire a utenti o codici non autorizzati di accedere a questi dati sensibili, AWS Glue ha disabilitato l'accesso ai registri dei driver di sistema. Per la risoluzione dei problemi, contatta l'assistenza AWS .
- Se è stata registrata una posizione di tabella con Lake Formation, il percorso di accesso ai dati passa attraverso le credenziali archiviate di Lake Formation indipendentemente dall'autorizzazione IAM per il ruolo di runtime del processo di AWS Glue. Se si configura erroneamente il ruolo registrato con la posizione della tabella, i processi inviati che utilizzano il ruolo con l'autorizzazione S3 IAM per la posizione della tabella avranno esito negativo.
- La scrittura su una tabella Lake Formation utilizza l'autorizzazione IAM anziché le autorizzazioni concesse da Lake Formation. Se il ruolo di runtime del processo dispone delle autorizzazioni S3 necessarie, è possibile utilizzarlo per eseguire operazioni di scrittura.

Di seguito sono riportate considerazioni e limitazioni per l'utilizzo di Apache Iceberg:

- È possibile utilizzare Apache Iceberg solo con il catalogo delle sessioni e non con i cataloghi con nomi arbitrari.
- Le tabelle Iceberg registrate in Lake Formation supportano solo le tabelle di metadati`history`,`metadata_log_entries`,, `snapshots` `filesmanifests`, e. `refs` AWS Glue nasconde le colonne che potrebbero contenere dati sensibili, ad esempio `partitionspath`, e`summaries`. Questa limitazione non si applica alle tabelle Iceberg che non sono registrate in Lake Formation.
- Le tabelle non registrate in Lake Formation supportano tutte le stored procedure di Iceberg. Le procedure di `register_table` e `migrate` non sono supportate per nessuna tabella.
- Ti consigliamo di utilizzare Iceberg DataFrameWriterV2 anziché V1.

Esempio di allocazione dei worker

Per un processo configurato con i seguenti parametri:

```
--enable-lakeformation-fine-grained-access=true
--number-of-workers=20
```

L'assegnazione di worker sarebbe:

- Un worker per il driver utente.
- Un worker per il driver di sistema.
- Il 10% dei restanti 18 worker (ovvero 2 worker) era riservato agli esecutori degli utenti.
- Fino a 16 worker assegnati agli esecutori di sistema.

Con il dimensionamento automatico abilitato, gli esecutori degli utenti possono utilizzare qualsiasi capacità non allocata dagli esecutori di sistema, se necessario.

Controllo dell'allocazione degli esecutori degli utenti

È possibile modificare la percentuale di prenotazione per gli esecutori degli utenti utilizzando la seguente configurazione:

```
--conf spark.dynamicAllocation.maxExecutorsRatio=<value between 0 and 1>
```

Questa configurazione consente un controllo ottimizzato sul numero di esecutori degli utenti riservati rispetto alla capacità totale disponibile.

## Risoluzione dei problemi

Consultare le sezioni seguenti per vedere le soluzioni di risoluzione dei problemi.

Registrazione dei log

AWS Glue utilizza i profili di risorse Spark per suddividere l'esecuzione del lavoro. AWS Glue utilizza il profilo utente per eseguire il codice fornito, mentre il profilo di sistema applica le politiche di Lake Formation. È possibile accedere ai registri per le attività eseguite come profilo utente.

Live UI e Spark History Server

Live UI e Spark History Server contengono tutti gli eventi Spark generati dal profilo utente e gli eventi oscurati generati dal driver di sistema.

È possibile vedere tutte le attività sia dall'utente che dal driver di sistema nella scheda Esecutori. Tuttavia, i link di registro sono disponibili solo per il profilo utente. Inoltre, alcune informazioni sono oscurate da Live UI, come il numero di record di output.

Il processo ha avuto esito negativo con autorizzazioni Lake Formation insufficienti

Assicurarsi che il ruolo di runtime del processo disponga delle autorizzazioni per eseguire SELECT e DESCRIBE sulla tabella a cui si sta accedendo.

Processo con esecuzione RDD non riuscita

AWS Glue attualmente non supporta le operazioni RDD (Resilient Distributed Dataset) sui lavori abilitati per Lake Formation.

Impossibile accedere ai file di dati in Amazon S3

Assicurarsi di aver registrato la posizione del data lake in Lake Formation.

Eccezione di convalida della sicurezza

AWS Glue ha rilevato un errore di convalida della sicurezza. Contatta l' AWS assistenza per ricevere assistenza.

Condivisione del catalogo dati e delle tabelle di AWS Glue tra account

È possibile condividere database e tabelle tra account e continuare a utilizzare Lake Formation. Per ulteriori informazioni, consulta Condivisione dei dati tra account in Lake Formation e Come posso condividere AWS Glue Data Catalog e tabelle utilizzando più account? .

La tabella seguente descrive un riepilogo di come scegliere tra Fine-grained Access Control (FGAC) e Full table access (FTA) per il carico di lavoro.

| Funzionalità | Controllo granulare degli accessi (FGAC) | Accesso completo alla tabella (FTA) |
|---|---|---|
| Livello di accesso | Livello di colonna/riga | Tabella completa |
| Caso d'uso | Interrogazioni ed ETL con autorizzazioni limitate | ETL |
| Impatto sulle prestazioni | Richiede transizioni system/user spaziali per la valutazione del controllo degli accessi, aggiungendo latenza | Prestazioni ottimizzate |

# Utilizzo di Amazon S3 Access Grants con AWS Glue

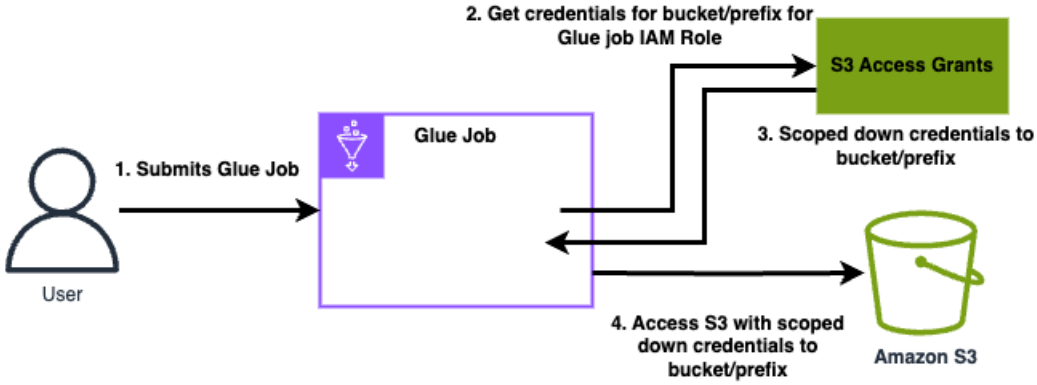
Con la versione 5.0 di Glue, Amazon S3 Access Grants fornisce una soluzione di controllo degli accessi scalabile che puoi utilizzare per aumentare l'accesso ai tuoi dati Amazon S3 da. AWS Glue Se si dispone di una configurazione di autorizzazioni complessa o di grandi dimensioni per i dati S3, si può utilizzare Access Grants S3 per scalare le autorizzazioni relative ai dati S3 per utenti e ruoli.

Usa S3 Access Grants per aumentare l'accesso ai dati di Amazon S3 oltre alle autorizzazioni concesse dal ruolo di runtime o ai ruoli IAM associati alle identità con accesso al tuo lavoro. AWS Glue Per ulteriori informazioni, consulta Gestione degli accessi con S3 Access Grants nella Guida per l'utente di Amazon S3.

## Come funziona con S3 Access Grants AWS Glue

AWS Glue le versioni 5.0 e successive forniscono un'integrazione nativa con S3 Access Grants. Puoi abilitare S3 Access Grants AWS Glue ed eseguire i job Spark. Quando il processo Spark effettua una richiesta di dati S3, Amazon S3 fornisce credenziali temporanee che rientrano nell'ambito del bucket, del prefisso o dell'oggetto specifico.

Di seguito è riportata una panoramica di alto livello su come AWS Glue ottenere l'accesso ai dati a cui S3 Access Grants gestisce l'accesso.




1. Un utente invia un job AWS Glue Spark che utilizza dati archiviati in Amazon S3.
2. AWS Glue richiede a S3 Access Grants di fornire credenziali temporanee per l'utente che dà accesso al bucket, al prefisso o all'oggetto.
3. AWS Glue restituisce credenziali temporanee sotto forma di token AWS Security Token Service (STS) per l'utente. Il token ha accesso al bucket, al prefisso o all'oggetto S3.
4. AWS Glue utilizza il token STS per recuperare i dati da S3.

5. AWS Glue riceve i dati da S3 e restituisce i risultati all'utente.

## S3 Access concede considerazioni con AWS Glue

Prendi nota dei seguenti comportamenti e limitazioni quando usi S3 Access Grants con. AWS Glue

Supporto funzionalità

- S3 Access Grants è supportato dalle versioni 5.0 e successive. AWS Glue
- Spark è l'unico tipo di lavoro supportato quando usi S3 Access Grants con. AWS Glue
- Delta Lake e Hudi sono gli unici formati a tabella aperta supportati quando usi S3 Access Grants con. AWS Glue
- Le seguenti funzionalità non sono supportate per l'utilizzo con S3 Access Grants:
  - Tabelle Apache Iceberg
  - AWS Richieste CLI ad Amazon S3 che utilizzano ruoli IAM
  - Accesso a S3 tramite il protocollo S3A open source

Considerazioni comportamentali

- AWS Glue fornisce una cache delle credenziali per garantire che un utente non debba fare richieste ripetute per le stesse credenziali all'interno di un job Spark. Pertanto, richiede AWS Glue sempre il privilegio di livello predefinito quando richiede le credenziali. Per ulteriori informazioni, consulta Richiedi l'accesso ai dati di S3 nella Guida per l'utente di Amazon S3.

## Configura S3 Access Grants con AWS Glue

Prerequisiti

Il chiamante o l'amministratore ha creato un'istanza S3 Access Grants.

Configura AWS Glue le politiche e la configurazione del lavoro

Per configurare S3 Access Grants con, AWS Glue devi configurare le policy di trust e IAM e passare la configurazione tramite i parametri del processo.

1. Configura le seguenti policy Minimal Trust e IAM sul ruolo utilizzato per le sovvenzioni (il AWS Glue ruolo che esegue sessioni o job).

Policy di trust:

```
{
            "Sid": "Stmt1234567891011",
            "Action": [
                "sts:AssumeRole",
                "sts:SetSourceIdentity",
                "sts:SetContext"
            ],
            "Effect": "Allow",
            "Principal": {
                "Service": "access-grants.s3.amazonaws.com"
            },
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "123456789012",
                    "aws:SourceArn": "arn:aws:s3:<region>:123456789012:access-grants/
default"
                }
            }
        }
```

Policy IAM:

```
{
            "Sid": "S3Grants",
            "Effect": "Allow",
            "Action": [
                "s3:GetDataAccess",
                "s3:GetAccessGrantsInstanceForPrefix"
            ],
            "Resource": "arn:aws:s3:<region>:123456789012:access-grants/default"
        },
        {
            "Sid": "BucketLevelReadPermissions",
            "Effect": "Allow",
            "Action": [
                "s3:ListBucket"
            ],
            "Resource": [
                "arn:aws:s3:::*"
            ],
```

```
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "123456789012"
                },
                "ArnEquals": {
                    "s3:AccessGrantsInstanceArn": [
                        "arn:aws:s3:<region>:123456789012:access-grants/default"
                    ]
                }
            }
        },
        {
            "Sid": "ObjectLevelReadPermissions",
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion",
                "s3:GetObjectAcl",
                "s3:GetObjectVersionAcl",
                "s3:ListMultipartUploadParts"
            ],
            "Resource": [
                "arn:aws:s3:::*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceAccount": "123456789012"
                },
                "ArnEquals": {
                    "s3:AccessGrantsInstanceArn": [
                        "arn:aws:s3:<region>:123456789012:access-grants/default"
                    ]
                }
            }
        }
```

2. Nel tuo AWS Glue lavoro, passa la seguente configurazione Spark tramite i parametri del AWS Glue lavoro o. `SparkConf`

```
--conf spark.hadoop.fs.s3.s3AccessGrants.enabled=true \
--conf spark.hadoop.fs.s3.s3AccessGrants.fallbackToIAM=false
```

# Propagazione affidabile delle identità con ETL AWS Glue

Con IAM Identity Center, puoi connetterti ai provider di identità (IdPs) e gestire centralmente l'accesso per utenti e gruppi attraverso AWS i servizi di analisi. È possibile integrare gestori di identità digitali come Okta, Ping e Microsoft Entra ID (in precedenza Azure Active Directory) con il Centro identità IAM per consentire agli utenti nell'organizzazione di accedere ai dati utilizzando un'esperienza di accesso singolo. Il Centro identità IAM supporta anche la connessione con altri gestori dell'identità digitali di terze parti.

Con la AWS Glue versione 5.0 e versioni successive, puoi propagare le identità degli utenti da IAM Identity Center a AWS Glue sessioni interattive. AWS Glue Le sessioni interattive propagheranno ulteriormente l'identità fornita a servizi downstream come Amazon S3 Access Grants AWS Lake Formation e Amazon Redshift, consentendo l'accesso sicuro ai dati tramite l'identità utente in questi servizi downstream.

## Panoramica di

Identity Center è l'approccio consigliato per l'autenticazione e l'autorizzazione della forza lavoro per organizzazioni di qualsiasi dimensione e tipo. AWS Con Identity Center, puoi creare e gestire le identità degli utenti o connettere la tua fonte di identità esistente, tra cui Microsoft Active Directory, Okta, Ping Identity JumpCloud, Google Workspace e Microsoft Entra ID (precedentemente Azure AD). AWS

La propagazione affidabile delle identità è una funzionalità di IAM Identity Center che gli amministratori dei AWS servizi connessi possono utilizzare per concedere e controllare l'accesso ai dati del servizio. L'accesso a questi dati si basa su attributi utente come le associazioni di gruppo. La configurazione di una propagazione affidabile delle identità richiede la collaborazione tra gli amministratori dei AWS servizi connessi e gli amministratori di IAM Identity Center.

## Funzionalità e vantaggi

L'integrazione delle sessioni AWS Glue interattive con IAM Identity Center Trusted Identity Propagation offre i seguenti vantaggi:

- La capacità di applicare l'autorizzazione a livello di tabella e il controllo granulare degli accessi con le identità del Centro identità sulle tabelle del catalogo dati gestite da AWS Glue di Lake Formation.
- La capacità di applicare l'autorizzazione con le identità del Centro identità sui cluster Amazon Redshift.

- Consente il monitoraggio completo delle azioni degli utenti per il controllo.
- La capacità di applicare l'autorizzazione a livello di prefisso di Amazon S3 con le identità del Centro identità sui prefissi Amazon S3 gestiti da Amazon S3 Access Grants.

## Casi d'uso

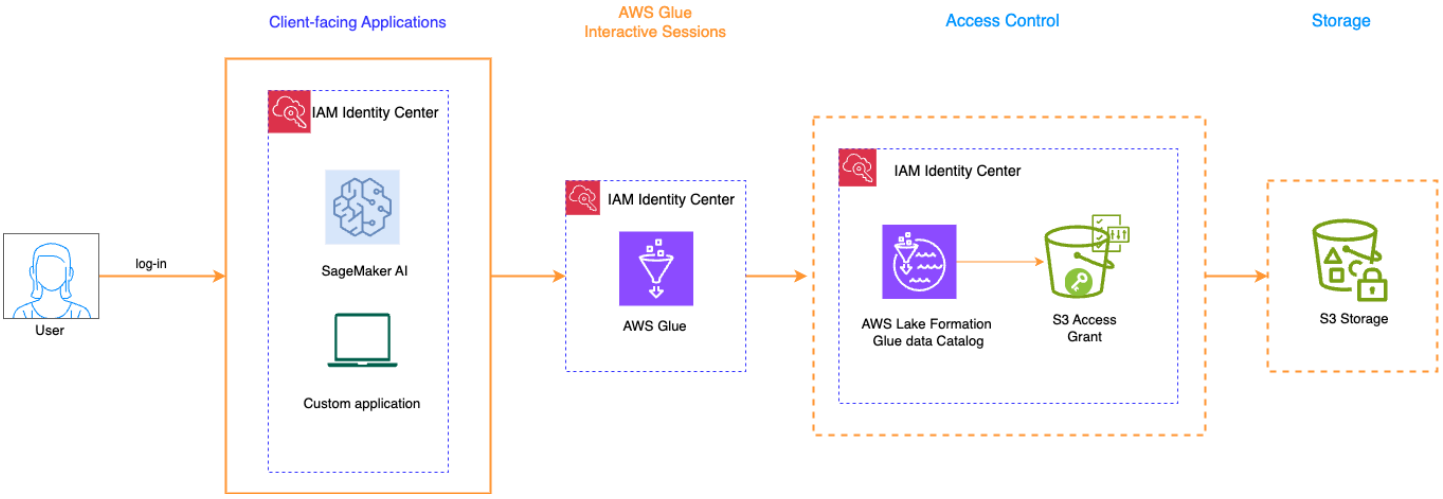
### Esplorazione e analisi interattive dei dati

I data engineer utilizzano le proprie identità aziendali per accedere e analizzare senza problemi i dati su più account. AWS Tramite SageMaker Studio, lanciano sessioni Spark interattive tramite AWS Glue ETL, collegandosi a varie fonti di dati tra cui Amazon S3 e AWS Glue Data Catalog. Mentre gli ingegneri esplorano i set di dati, Spark applica controlli granulari degli accessi definiti in Lake Formation in base alle loro identità, assicurando che possano visualizzare solo i dati autorizzati. Tutte le query e le trasformazioni dei dati vengono registrate con l'identità dell'utente, creando un audit trail chiaro. Questo approccio semplificato consente la prototipazione rapida di nuovi prodotti di analisi, mantenendo al contempo una rigorosa governance dei dati in tutti gli ambienti client.

### Preparazione dei dati e ingegneria delle caratteristiche

Gli scienziati dei dati di diversi team di ricerca collaborano su progetti complessi utilizzando una piattaforma di dati unificata. Accedono a SageMaker Studio con le proprie credenziali aziendali, accedendo immediatamente a un vasto data lake condiviso che si estende su più account. AWS Quando iniziano a progettare funzionalità per nuovi modelli di apprendimento automatico, le sessioni Spark lanciate tramite AWS Glue ETL applicano le politiche di sicurezza a livello di colonne e righe di Lake Formation basate sulle loro identità diffuse. Gli scienziati possono preparare in modo efficiente i dati e progettare le funzionalità utilizzando strumenti familiari, mentre i team addetti alla conformità hanno la certezza che ogni interazione con i dati venga tracciata e verificata automaticamente. Questo ambiente sicuro e collaborativo accelera le pipeline di ricerca mantenendo al contempo i rigorosi standard di protezione dei dati richiesti nei settori regolamentati.

# Come funziona




Un utente accede alle applicazioni rivolte ai clienti (SageMaker AI o applicazioni personalizzate) utilizzando la propria identità aziendale tramite IAM Identity Center. Questa identità viene quindi propagata attraverso l'intera pipeline di accesso ai dati.

L'utente autenticato avvia le sessioni AWS AWS Glue interattive, che fungono da motore di calcolo per l'elaborazione dei dati. Queste sessioni mantengono il contesto dell'identità dell'utente durante tutto il flusso di lavoro.

AWS Lake Formation e AWS Glue Data Catalog collaborano per applicare controlli di accesso granulari. Lake Formation applica policy di sicurezza basate sull'identità propagata dell'utente, mentre Amazon S3 Access Grants fornisce livelli di autorizzazione aggiuntivi, garantendo agli utenti di accedere solo ai dati che sono autorizzati a visualizzare.

Infine, il sistema si connette ad Amazon S3 Storage dove risiedono i dati effettivi. Tutti gli accessi sono regolati dalle policy di sicurezza combinate, che mantengono la governance dei dati e consentono l'esplorazione e l'analisi interattive dei dati. Questa architettura consente un accesso sicuro e basato sull'identità ai dati attraverso più AWS servizi, mantenendo al contempo un'esperienza utente senza interruzioni per i data scientist e gli ingegneri che lavorano con set di dati di grandi dimensioni.

# Integrazioni

## AWS ambiente di sviluppo gestito

Le seguenti applicazioni AWS gestite rivolte ai client supportano la propagazione affidabile delle identità con AWS Glue sessioni interattive:

- Sagemaker Unified Studio
- Amazon SageMaker AI

Sagemaker Unified Studio

Per utilizzare la propagazione attendibile delle identità con Sagemaker Unified Studio:

1. Configurare il progetto Sagemaker Unified Studio con la propagazione attendibile delle identità abilitata come ambiente di sviluppo rivolto al cliente.
2. Configura Lake Formation per abilitare il controllo granulare degli accessi per le AWS Glue tabelle in base all'utente o al gruppo in IAM Identity Center.
3. Configurare Amazon S3 Access Grants per consentire l'accesso temporaneo alle posizioni dei dati sottostanti in Amazon S3.
4. Apri lo spazio JupyterLab IDE di Sagemaker Unified Studio e AWS Glue selezionalo come elaborazione per l'esecuzione su notebook.

### Ambiente del notebook ospitato autonomamente e gestito dal cliente

Per abilitare la propagazione affidabile delle identità per gli utenti di applicazioni sviluppate su misura, consultate Access AWS services using trusted identity propagation nel Security Blog. AWS

## Guida introduttiva alla propagazione affidabile delle identità in ETL AWS Glue

Questa sezione aiuta a configurare AWS Glue l'applicazione con sessioni interattive per l'integrazione con IAM Identity Center e abilitare la propagazione delle identità affidabili.

### Prerequisiti

- Un'istanza di Identity Center nella AWS regione in cui si desidera creare sessioni AWS Glue interattive abilitate alla propagazione dell'identità affidabile. Un'istanza di Identity Center può esistere solo in una singola regione per un AWS account. Per ulteriori informazioni, consultare Abilita Centro identità IAM ed effettuare il provisioning di utenti e gruppi dall'origine delle identità nel Centro identità IAM.
- Abilitare la propagazione attendibile delle identità per servizi downstream come Lake Formation o Amazon S3 Access Grants o il cluster Amazon Redshift con cui il carico di lavoro interattivo interagisce per accedere ai dati.

## Autorizzazioni necessarie per connettere AWS Glue ETL con IAM Identity Center

### Creazione di un ruolo IAM

Il ruolo che crea la connessione al Centro identità IAM richiede le autorizzazioni per creare e modificare la configurazione dell'applicazione in AWS Glue e nel Centro identità IAM, come indicato nella seguente policy in linea.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "glue:CreateGlueIdentityCenterConfiguration",
                "sso:CreateApplication",
                "sso:PutApplicationAssignmentConfiguration",
                "sso:PutApplicationAuthenticationMethod",
                "sso:PutApplicationGrant",
                "sso:PutApplicationAccessScope",
                "sso:ListInstances"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Le seguenti policy in linea contengono autorizzazioni specifiche necessarie per visualizzare, aggiornare ed eliminare le proprietà di integrazione AWS Glue con il Centro identità IAM.

Utilizza la seguente policy in linea per consentire a un ruolo IAM di visualizzare un' AWS Glue integrazione con IAM Identity Center.

JSON

```
{
```

```
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "glue:GetGlueIdentityCenterConfiguration"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Utilizza la seguente policy in linea per consentire a un ruolo IAM di aggiornare AWS Glue l'integrazione con IAM Identity Center.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "glue:UpdateGlueIdentityCenterConfiguration",
                "sso:PutApplicationAccessScope",
                "sso:DeleteApplicationAccessScope"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Utilizza la seguente policy in linea per consentire a un ruolo IAM di eliminare un' AWS Glue integrazione con IAM Identity Center.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "glue:DeleteGlueIdentityCenterConfiguration",
                "sso:DeleteApplication"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Descrizione delle autorizzazioni

- `glue:CreateGlueIdentityCenterConfiguration`— Concede l'autorizzazione a creare la configurazione AWS Glue iDC.
- `glue:GetGlueIdentityCenterConfiguration`: concede l'autorizzazione per ottenere una configurazione IdC esistente.
- `glue:DeleteGlueIdentityCenterConfiguration`— Concede il permesso di eliminare una configurazione IdC esistente AWS Glue .
- `glue:UpdateGlueIdentityCenterConfiguration`— Concede il permesso di aggiornare una configurazione IdC esistente AWS Glue .
- `sso:CreateApplication`— Concede l'autorizzazione a creare un'applicazione IAM Identity AWS Glue Center gestita.
- `sso:DescribeApplication`- Concede l'autorizzazione a descrivere un'applicazione IAM Identity Center AWS Glue gestita.
- `sso:DeleteApplication`— Concede l'autorizzazione a eliminare un'applicazione IAM Identity Center AWS Glue gestita.
- `sso:UpdateApplication`— Concede l'autorizzazione ad aggiornare un'applicazione IAM Identity Center AWS Glue gestita.

- `sso:PutApplicationGrant`— Concede l'autorizzazione ad applicare token-exchange, IntrospectToken, RefreshToken e concessioni sull'applicazione iDC. RevokeToken
- `sso:PutApplicationAuthenticationMethod`— Concede l'autorizzazione a inserire AuthenticationMethod sull'applicazione iDC AWS Glue gestita che consente al responsabile del servizio di interagire con l'applicazione iDC. AWS Glue
- `sso:PutApplicationAccessScope`— Concede l'autorizzazione ad aggiungere o aggiornare l'elenco degli ambiti di servizio downstream autorizzati sull'applicazione iDC gestita. AWS Glue
- `sso:DeleteApplicationAccessScope`- Concede l'autorizzazione a eliminare gli ambiti a valle se viene rimosso un ambito per l'applicazione iDC gestita. AWS Glue
- `sso:PutApplicationAssignmentConfiguration`— Concede l'autorizzazione a impostare l'impostazione «User-assignment-not-required» sull'applicazione iDC.
- `sso:ListInstances`— Concede il permesso di elencare le istanze e convalidare l'iDC InstanceArn specificato nel parametro. identity-center-configuration

## Connessione con AWS Glue IAM Identity Center

Quando AWS Glue è connesso a IAM Identity Center, crea un'applicazione iDC gestita singleton per account. L'esempio seguente mostra come è possibile connettersi a IAM Identity AWS Glue Center:

```
aws glue create-glue-identity-center-configuration \
--instance-arn arn:aws:sso:::instance/ssoins-123456789 \
--scopes '["s3:access_grants:read_write", "redshift:connect","lakeformation:query"]'
```

Per aggiornare gli ambiti dell'applicazione gestita (in genere eseguita per propagarsi a più servizi downstream), è possibile utilizzare:

```
aws glue update-glue-identity-center-configuration \
--scopes '["s3:access_grants:read_write", "redshift:connect","lakeformation:query"]'
```

Il parametro Ambiti è facoltativo e tutti gli ambiti verranno aggiunti se non vengono forniti.
I valori supportati sono `s3:access_grants:read_write`, `redshift:connect` e `lakeformation:query`.

Per ottenere i dettagli della configurazione, è possibile usare:

```
aws glue get-glue-identity-center-configuration
```

Puoi eliminare la connessione tra AWS Glue e IAM Identity Center utilizzando il seguente comando:

```
aws glue delete-glue-identity-center-configuration
```

> **Note**
>
> AWS Glue crea un'applicazione Identity Center gestita dal servizio nel tuo account che il servizio sfrutta per la convalida dell'identità e la propagazione dell'identità ai servizi downstream. AWS Glue l'applicazione Identity Center gestita creata viene condivisa tra tutte le trusted-identity-propagation sessioni dell'account.
> Avviso: non modificare manualmente le impostazioni sull'applicazione gestita del Centro identità. Qualsiasi modifica potrebbe influire su tutte le sessioni AWS Glue interattive trusted-identity-propagation abilitate nel tuo account.

## Creazione di una sessione AWS Glue interattiva con Trusted Identity Propagation abilitata

Dopo esserti connesso a IAM Identity Center, puoi utilizzare le credenziali di ruolo AWS Glue con identità avanzata per creare una sessione interattiva. AWS Glue Non è necessario passare parametri aggiuntivi durante la creazione di una sessione 5.0. AWS Glue Poiché AWS Glue è connesso a IAM Identity Center, se viene AWS Glue rilevato identity-enhanced-role-credentials, propagherà automaticamente le informazioni sull'identità ai servizi a valle che vengono richiamati come parte delle istruzioni. Tuttavia, il ruolo di runtime per la sessione deve disporre dell'autorizzazione `sts:SetContext`, come illustrato di seguito.

Autorizzazioni di ruolo di runtime per propagare l'identità

Poiché AWS Glue le sessioni sfruttano le credenziali potenziate dall'identità per propagare l'identità ai AWS servizi downstream, la policy di fiducia del ruolo di runtime deve disporre di autorizzazioni `sts:SetContext` aggiuntive per consentire la propagazione dell'identità ai servizi downstream (Amazon S3 access-grant, Lake Formation, Amazon Redshift). Per ulteriori informazioni su come creare un ruolo di runtime, consulta Configurazione di un ruolo di runtime.

JSON

```
{
  "Version":"2012-10-17",
```

```
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "glue.amazonaws.com"
      },
      "Action": [
        "sts:AssumeRole",
        "sts:SetContext"
      ]
    }
  ]
}
```

Inoltre, il ruolo Runtime richiederebbe le autorizzazioni per i servizi downstream che job-run richiamerebbe per recuperare i dati utilizzando l'identità dell'utente. AWS Fare riferimento ai seguenti link per configurare Amazon S3 Access Grants e Lake Formation:

- Usare Lake Formation con AWS Glue
- Utilizzo di Amazon S3 Access Grants con AWS Glue

## Considerazioni e limitazioni per l'integrazione di AWS Glue ETL Trusted Identity Propagation

> **Important**
>
> Per impostazione predefinita, le sessioni non sono private, il che significa che un utente iDC può accedere alla sessione di un altro utente iDC. Puoi usarla tagOnCreateper rendere private le tue sessioni. Ad esempio, la sessione può essere etichettata con un tag owner e il relativo valore come ID utente IDC e quindi sulla policy, è possibile utilizzare una chiave di condizione globale per `identitystore:UserId`eseguire la convalida confrontandola con il tag owner nella politica del principal/runtime ruolo client per tutte le operazioni API di sessione per garantire che un utente iDC non sia in grado di accedere alla sessione di un altro utente IDC.

Considera i seguenti punti quando utilizzi IAM Identity Center Trusted Identity Identity Propagation with Application: AWS Glue

- La propagazione affidabile dell'identità tramite Identity Center è supportata nella AWS Glue versione 5.0 e versioni successive e solo con sessioni AWS Glue interattive.
- AWS Glue il catalogo dati è coperto dall'integrazione del centro di identità di Lake Formation.
- Trusted Identity Propagation è limitata alle sessioni interattive in AWS Glue, ad esclusione di altre entità di elaborazione dati come job, trigger, flussi di lavoro e attività di machine learning. Tutte AWS Glue APIs, tuttavia, registrano le identità degli utenti per il controllo. AWS CloudTrail
- AWS Glue attualmente supporta l'integrazione con IAM Identity Center esclusivamente tramite interfacce API e CLI, non tramite la console.
- Una volta abilitata un'applicazione sul AWS Glue lato, assicurati di creare 5.0 sessioni con credenziali iDC ma non creare una sessione 4.0 con credenziali iDC.
- Trusted Identity Propagation con AWS Glue è supportato nelle seguenti regioni: AWS
  - Africa (Città del Capo) – af-south-1
  - Asia Pacifico (Hong Kong) – ap-east-1
  - Asia Pacifico (Tokyo) – ap-northeast-1
  - Asia Pacifico (Seoul) – ap-northeast-2
  - Asia Pacifico (Osaka) – ap-northeast-3
  - Asia Pacifico (Mumbai) – ap-south-1
  - Asia Pacifico (Singapore) – ap-southeast-1
  - Asia Pacifico (Sydney) – ap-southeast-2
  - Asia Pacifico (Giacarta) – ap-southeast-3
  - Canada (Centrale) – ca-central-1
  - Europa (Francoforte) – eu-central-1
  - Europa (Stoccolma) – eu-nord-1
  - Europa (Milano) – eu-south-1
  - Europa (Irlanda) – eu-west-1
  - Europa (Londra) – eu-west-2
  - Europa (Parigi) – eu-ovest-3
  - Medio Oriente (Bahrein) – me-south-1
  - Sud America (San Paolo) – sa-east-1
  - Stati Uniti orientali (Virginia settentrionale) – us-est-1
  - Stati Uniti orientali (Ohio) – us-est-2

- Stati Uniti occidentali (California settentrionale) – us-west-1
- Stati Uniti occidentali (Oregon) – us-west-2

## Sessioni utente in background per AWS Glue ETL

Le sessioni utente in background consentono ai carichi di lavoro di analisi e machine learning di lunga durata di continuare anche dopo che l'utente si è disconnesso dall'interfaccia del notebook. Questa funzionalità è implementata tramite la funzionalità affidabile AWS Glue di propagazione delle identità. La pagina seguente spiega le opzioni e i comportamenti di configurazione per le sessioni in background degli utenti.

> **Note**
>
> Le sessioni utente in background si applicano alle sessioni AWS Glue interattive avviate tramite interfacce di notebook come SageMaker Unified Studio. L'attivazione o la disabilitazione di questa funzionalità ha effetto solo sulle nuove sessioni interattive; le sessioni attive esistenti non ne risentono.

### Configura le sessioni utente in background

Le sessioni utente in background devono essere abilitate a due livelli per una corretta funzionalità:

1. Livello di istanza IAM Identity Center (configurato dagli amministratori iDC)
2. AWS Glue Livello di configurazione di Identity Center (configurato dagli amministratori AWS Glue )

#### Abilita le sessioni utente in background per AWS Glue

Per abilitare le sessioni utente in background per AWS Glue, è necessario impostare il `userBackgroundSessionsEnabled` parametro su `true` nella configurazione di Identity Center durante la creazione o l'aggiornamento della configurazione.

Prerequisiti

- Il ruolo IAM utilizzato per create/update la configurazione di AWS Glue Identity Center deve disporre dell'`sso:PutApplicationSessionConfiguration`autorizzazione. Questa autorizzazione consente di AWS Glue abilitare le sessioni utente in background a livello AWS Glue di applicazione iDC gestita.

- Le sessioni AWS Glue interattive devono utilizzare la AWS Glue versione 5.0 o successiva e devono essere abilitate la Trusted Identity Propagation.

Per abilitare le sessioni utente in background utilizzando: AWS CLI

```
aws glue create-glue-identity-center-configuration \
    --instance-arn "arn:aws:sso:::instance/ssoins-1234567890abcdef" \
    --user-background-sessions-enabled
```

Per aggiornare una configurazione esistente:

```
aws glue update-glue-identity-center-configuration \
    --user-background-sessions-enabled
```

Matrice di configurazione

L'effettiva configurazione della sessione utente in background dipende sia dall'impostazione di AWS Glue configurazione che dalle impostazioni a livello di istanza di IAM Identity Center:

| IAM Identity Center è abilitato? userBackgroundSession | AWS Glue userBackgroundSessionsAbilitato | Comportamento |
|---|---|---|
| Sì | TRUE | Sessioni utente in background abilitate |
| Sì | FALSE | La sessione scade con il logout dell'utente |
| No | TRUE | La creazione della sessione fallisce con Exception |
| No | FALSE | La sessione scade con il logout dell'utente |

## Durata predefinita della sessione in background dell'utente

Per impostazione predefinita, tutte le sessioni utente in background hanno un limite di durata di 7 giorni in IAM Identity Center. Gli amministratori possono modificare questa durata nella console

del Centro identità IAM. Questa impostazione si applica a livello di istanza IAM Identity Center e interessa tutte le applicazioni IAM Identity Center supportate all'interno di quell'istanza.

- La durata può essere impostata su qualsiasi valore, da 15 minuti a 90 giorni
- Questa impostazione è configurata nella console IAM Identity Center in Impostazioni → Autenticazione → Configura (sezione Lavori non interattivi)

> **Note**
>
> AWS Glue per impostazione predefinita, le sessioni interattive hanno un limite di timeout di inattività separato di 48 ore. Le sessioni termineranno quando viene raggiunto il timeout di inattività della AWS Glue sessione o la durata della sessione in background dell'utente, a seconda dell'evento che si verifica per primo.

## Impatto della disabilitazione delle sessioni utente in background

Quando le sessioni utente in background sono disabilitate a livello di AWS Glue configurazione:

- Sessioni interattive esistenti: continuano a funzionare senza interruzioni se sono state avviate con le sessioni utente in background abilitate. Queste sessioni continueranno a utilizzare i token di sessione in background esistenti fino a quando non termineranno naturalmente o non verranno interrotte esplicitamente.
- Nuove sessioni interattive: utilizzeranno il flusso standard di propagazione delle identità affidabili e termineranno quando l'utente si disconnette o scade la sessione interattiva (ad esempio quando chiude un notebook Unified Studio). SageMaker JupyterLab

### Modifica della durata delle sessioni utente in background

Quando l'impostazione della durata per le sessioni utente in background viene modificata in IAM Identity Center:

- Sessioni interattive esistenti: continua a funzionare con la stessa durata della sessione in background con cui sono state avviate
- Nuove sessioni interattive: utilizzerà la nuova durata della sessione per le sessioni in background

## Considerazioni sul runtime

### Condizioni di terminazione della sessione

Quando si utilizzano sessioni utente in background, una sessione AWS Glue interattiva continuerà a funzionare fino a quando non si verifica una delle seguenti condizioni:

- La sessione utente in background scade (in base alla configurazione iDC, fino a 90 giorni)
- La sessione in background dell’utente viene revocata manualmente da un amministratore.
- La sessione AWS Glue interattiva raggiunge il timeout di inattività (impostazione predefinita: 48 ore dopo l'ultima istruzione eseguita)
- L'utente arresta o riavvia in modo esplicito il kernel del notebook

### Persistenza dei dati

Quando si utilizzano sessioni utente in background:

- Gli utenti non possono riconnettersi all'interfaccia del notebook per visualizzare i risultati dopo essersi disconnessi
- Configura le istruzioni Spark per scrivere i risultati sullo storage persistente (come Amazon S3) prima del completamento dell'esecuzione

### Implicazioni sui costi

- I lavori continueranno a essere eseguiti fino al completamento anche dopo che gli utenti avranno terminato la JupyterLab sessione di SageMaker Unified Studio e verranno addebitati costi per l'intera durata dell'esecuzione completata
- Monitora le sessioni attive in background per evitare costi inutili derivanti da sessioni dimenticate o abbandonate

### Disponibilità delle funzionalità

Le sessioni utente in background AWS Glue sono disponibili per:

- AWS Glue solo sessioni interattive (i AWS Glue lavori e i lavori di streaming non sono supportati)
- AWS Glue versione 5.0 e successive
- Solo configurazioni abilitate per Trusted Identity Propagation

# Registrazione e monitoraggio AWS Glue

È possibile automatizzare l'esecuzione dei processi ETL (Extract, Transform and Load, estrazione, trasformazione e caricamento). AWS Glue fornisce alcuni parametri di crawler e processi che è possibile monitorare. Dopo aver configurato il AWS Glue Data Catalog con i metadati necessari, AWS Glue fornisce le statistiche sull'integrità dell'ambiente. Puoi automatizzare la chiamata di crawler e processi con una pianificazione basata sul tempo tramite un cron. Puoi anche attivare i processi quando si attiva un trigger basato su evento.

AWS Glueè integrato con AWS CloudTrail, un servizio che fornisce una registrazione delle azioni intraprese da un utente, ruolo o AWS servizio inAWS Glue. Se crei un trail, puoi abilitare la distribuzione continua di CloudTrail eventi a un bucket Amazon Simple Storage Service (Amazon S3), Amazon CloudWatch Logs e Amazon Events. CloudWatch Ogni evento o voce di log contiene informazioni sull'utente che ha generato la richiesta. Con AWS Glue la versione 5.1, il contesto di controllo AWS Glue e le chiamate AWS Lake Formation API saranno disponibili nei log. AWS CloudTrail

Usa Amazon CloudWatch Events per automatizzare AWS i tuoi servizi e rispondere automaticamente a eventi di sistema come problemi di disponibilità delle applicazioni o modifiche delle risorse. Gli eventi derivanti dai AWS servizi vengono trasmessi a CloudWatch Events quasi in tempo reale. È possibile scrivere semplici regole che indichino quali eventi sono da considerare di interesse e quali operazioni automatizzate intraprendere quando si verifica un evento previsto da una regola.

Consulta anche

- Automatizzazione con AWS Glue EventBridge
- Registrazione su più account CloudTrail

Un aspetto importante della sicurezza nel cloud è la registrazione. È necessario configurare la registrazione in modo da non acquisire segreti e materiale riservato mentre si acquisiscono le informazioni necessarie per eseguire il debug e proteggere l'infrastruttura cloud. Assicurati di sapere con ciò che è stato registrato.

# Convalida della conformità per AWS Glue

Per sapere se un Servizio AWS programma rientra nell'ambito di specifici programmi di conformità, consulta Servizi AWS la sezione Scope by Compliance Program Servizi AWS e scegli il programma di conformità che ti interessa. Per informazioni generali, consulta Programmi di AWS conformità Programmi di .

È possibile scaricare report di audit di terze parti utilizzando AWS Artifact. Per ulteriori informazioni, consulta Scaricamento dei report in AWS Artifact .

La vostra responsabilità di conformità durante l'utilizzo Servizi AWS è determinata dalla sensibilità dei dati, dagli obiettivi di conformità dell'azienda e dalle leggi e dai regolamenti applicabili. Per ulteriori informazioni sulla responsabilità di conformità durante l'utilizzo Servizi AWS, consulta la Documentazione AWS sulla sicurezza.

# Resilienza in AWS Glue

L'infrastruttura AWS globale è costruita attorno a AWS regioni e zone di disponibilità. AWS Le regioni forniscono più zone di disponibilità fisicamente separate e isolate, collegate con reti a bassa latenza, ad alto throughput e altamente ridondanti. Con le zone di disponibilità, è possibile progettare e gestire applicazioni e database che eseguono il failover automatico tra zone di disponibilità senza interruzioni. Le zone di disponibilità sono più disponibili, tolleranti ai guasti e scalabili rispetto alle infrastrutture tradizionali a data center singolo o multiplo.

Per ulteriori informazioni su AWS regioni e zone di disponibilità, consulta Global Infrastructure.AWS

Per ulteriori informazioni sulla resilienza dei AWS Glue job, consulta Error: failover behavior between VPCs in. AWS Glue

# Sicurezza dell'infrastruttura nell' AWS Glue

In quanto servizio gestito, AWS Glue è protetto dalle procedure di sicurezza della rete AWS globale descritte nel white paper Amazon Web Services: Overview of Security Processes.

Utilizzi chiamate API AWS pubblicate per accedere AWS Glue attraverso la rete. I client devono supportare Transport Layer Security (TLS) 1.0 o versioni successive. È consigliabile TLS 1.2 o versioni successive. I client devono, inoltre, supportare le suite di cifratura con PFS (Perfect Forward Secrecy), ad esempio Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman

(ECDHE). La maggior parte dei sistemi moderni, come Java 7 e versioni successive, supporta tali modalità.

Inoltre, le richieste devono essere firmate utilizzando un ID chiave di accesso e una chiave di accesso segreta associata a un principale IAM. In alternativa, è possibile utilizzare AWS Security Token Service (AWS STS) per generare le credenziali di sicurezza temporanee per sottoscrivere le richieste.

Argomenti

- Configurazione degli endpoint VPC (AWS PrivateLink) per AWS Glue (AWS PrivateLink)
- Configurazione di Amazon condiviso VPCs

# Configurazione degli endpoint VPC (AWS PrivateLink) per AWS Glue (AWS PrivateLink)

È possibile stabilire una connessione privata tra VPC e AWS Glue creando un endpoint VPC dell'interfaccia. Gli endpoint di interfaccia sono alimentati da AWS PrivateLink, una tecnologia che consente l'accesso privato AWS Glue APIs senza un gateway Internet, un dispositivo NAT, una connessione VPN o una connessione AWS Direct Connect. Le istanze presenti nel VPC non richiedono indirizzi IP pubblici per comunicare con AWS Glue APIs. Il traffico tra il tuo VPC e l'AWS Glue non esce dalla rete Amazon.

Ogni endpoint dell'interfaccia è rappresentato da una o più interfacce di rete elastiche nelle sottoreti.

Per ulteriori informazioni, consultare Endpoint VPC di interfaccia (AWS PrivateLink) nella Guida per l'utente di Amazon VPC.

## Considerazioni sugli endpoint VPC dell'AWS Glue

Prima di impostare un endpoint VPC dell'interfaccia per l'AWS Glue, assicurati di leggere le Proprietà e le limitazioni degli endpoint dell'interfaccia nella Guida per l'utente di Amazon VPC.

L'AWS Glue supporta l'esecuzione di chiamate per tutte le sue operazioni API all'interno del VPC.

## Creazione di un endpoint VPC interfaccia per l'AWS Glue

Puoi creare un endpoint VPC per il AWS Glue servizio utilizzando la console Amazon VPC o (). AWS Command Line Interface AWS CLI Per ulteriori informazioni, consultare Creazione di un endpoint di interfaccia nella Guida per l'utente di Amazon VPC.

Crea un endpoint VPC per AWS Glue, utilizzando il seguente nome di servizio:

- com.amazonaws. *region*.colla

Se si abilita il DNS privato per l'endpoint, è possibile effettuare richieste API verso AWS Glueutilizzando il nome DNS predefinito per la regione, ad esempio `glue.us-east-1.amazonaws.com`.

Per ulteriori informazioni, consultare Accesso a un servizio tramite un endpoint di interfaccia in Guida per l'utente di Amazon VPC.

## Creazione di una policy di endpoint VPC per l'AWS Glue

È possibile allegare un criterio all'endpoint VPC che controlla l'accesso all'AWS Glue. La policy specifica le informazioni riportate di seguito:

- Il principale che può eseguire operazioni.
- Le azioni che possono essere eseguite.
- Le risorse sui cui si possono eseguire operazioni.

Per ulteriori informazioni, consulta Controllo degli accessi ai servizi con endpoint VPC in Guida per l'utente di Amazon VPC.

Ad esempio, policy di endpoint VPC affinché AWS Glue permetta la creazione e l'aggiornamento del processo

Di seguito è riportato un esempio di una policy endpoint per l'AWS Glue. Se collegato a un endpoint, questo criterio concede l'accesso alle operazioni elencate AWS Glue per tutti i principali su tutte le risorse.

```
{
  "Statement": [
    {
      "Sid": "RestrictPassRole",
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::123456789012:role/GlueServiceRole*",
      "Condition": {
        "StringEquals": {
          "iam:PassedToService": "glue.amazonaws.com"
```

```
        }
      }
    }
  ]
}
```

Esempio: policy di endpoint VPC per permettere l'accesso in sola lettura al catalogo dati

Di seguito è riportato un esempio di una policy endpoint per l'AWS Glue. Se collegato a un endpoint, questo criterio concede l'accesso alle operazioni dell'AWS Glueelencate per tutti gli account principali su tutte le risorse.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "glue:GetDatabase",
        "glue:GetDatabases",
        "glue:GetTable",
        "glue:GetTables",
        "glue:GetTableVersion",
        "glue:GetTableVersions",
        "glue:GetPartition",
        "glue:GetPartitions",
        "glue:BatchGetPartition",
        "glue:SearchTables"
      ],
      "Resource": "*"
    }
  ]
}
```

## Configurazione di Amazon condiviso VPCs

AWS Gluesupporta cloud privati virtuali condivisi (VPCs) in Amazon Virtual Private Cloud. La condivisione di Amazon VPC consente a più AWS account di creare le proprie risorse applicative, come EC2 istanze Amazon e database ( Amazon Relational Database Service Amazon RDS), in Amazon condiviso e gestito centralmente. VPCs In questo modello, l'account proprietario del VPC (proprietario) condivide una o più sottoreti con altri account (partecipanti) che appartengono alla stessa organizzazione di. AWS Organizations Una volta condivisa una sottorete, i partecipanti

possono visualizzare, creare, modificare ed eliminare le proprie risorse delle applicazioni nelle sottoreti che sono condivise con loro stessi.

In AWS Glue, per creare una connessione con una sottorete condivisa, è necessario creare un gruppo di sicurezza all'interno del proprio account e collegare il gruppo di sicurezza alla sottorete condivisa.

Per ulteriori informazioni, consulta i seguenti argomenti:

- Utilizzo di Shared VPCs nella Guida per l'utente di Amazon VPC
- Che cos'è AWS Organizations? nella Guida per l'AWS Organizations utente

# Risoluzione dei problemi relativi a AWS Glue

Se riscontri problemi durante l'utilizzo AWS Glue, consulta gli argomenti di questa sezione.

Per la risoluzione dei problemi di Apache Spark in AWS Glue, consulta Risoluzione dei problemi di intelligenza artificiale generativa per Apache Spark, in AWS Glue cui fornisce assistenza basata sull'intelligenza artificiale per la diagnosi e la risoluzione dei problemi relativi ai job di Spark.

Trasforma la risoluzione dei problemi di Apache Spark con il nostro agente di risoluzione dei problemi basato sull'intelligenza artificiale, che ora supporta tutte le principali modalità di implementazione, tra cui AWS Glue Amazon EMR-EC2, Amazon EMR-Serverless e Amazon AI Notebooks. SageMaker Questo potente agente elimina i complessi processi di debug combinand o interazioni in linguaggio naturale, analisi del carico di lavoro in tempo reale e consigli sul codice intelligente in un'esperienza senza interruzioni. Per i dettagli sull'implementazione, consulta Cos'è l'agente di risoluzione dei problemi di Apache Spark per Amazon EMR.

Argomenti


- Raccolta di informazioni AWS Glue sulla risoluzione dei problemi
- Risoluzione degli errori di configurazione comuni di Glue
- Errori del crawler quando il crawler usa le autorizzazioni di Lake Formation
- Risoluzione dei problemi relativi AWS Glue agli errori di Ray nei log
- Eccezioni di machine learning AWS Glue
- AWS Glue quote


## Raccolta di informazioni AWS Glue sulla risoluzione dei problemi

Se si verificano errori o comportamenti imprevisti AWS Glue e si desidera contattare Supporto AWS, è innanzitutto necessario raccogliere informazioni sui nomi e i registri associati all'azione non riuscita. IDs La disponibilità di queste informazioni consente di aiutarti Supporto a risolvere i problemi che stai riscontrando.

Oltre all'ID account, raccogli le seguenti informazioni per ognuno di questi tipi di errori:

Quando un crawler ha esito negativo, raccogli le informazioni riportate di seguito.

- Il nome del crawler

  I log delle esecuzioni del crawler si trovano nella CloudWatch sezione Registri sotto. `/aws-glue/crawlers`

Quando una connessione di prova ha esito negativo, raccogli le informazioni riportate di seguito.

- Nome della connessione
- ID connessione
- Stringa di connessione JDBC nel modulo `jdbc:protocol://host:port/database-name`.

  I log delle connessioni di test si trovano in Logs in. CloudWatch `/aws-glue/testconnection`

Quando un processo ha esito negativo, raccogli le informazioni riportate di seguito.

- Nome del lavoro
- ID di esecuzione del processo nel formato `jr_xxxxx`.

  I log delle esecuzioni dei job si trovano nella sezione CloudWatch Registri sotto. `/aws-glue/jobs`

# Risoluzione degli errori di configurazione comuni di Glue

Se riscontri errori in AWS Glue, usa le seguenti informazioni per aiutarti a trovare l'origine dei problemi e risolverli.

> **Note**
>
> L' AWS Glue GitHub archivio contiene ulteriori istruzioni per la risoluzione dei problemi nelle AWS Glue Domande frequenti.

Argomenti


- Errore: risorsa non disponibile
- Errore: impossibile trovare l'endpoint S3 o il gateway NAT per il subnetId in VPC
- Errore: regola in entrata obbligatoria nel gruppo di sicurezza
- Errore: regola in uscita obbligatoria nel gruppo di sicurezza

- Errore: esecuzione del Job non riuscita perché al ruolo passato devono essere assegnate (presupponiamo le autorizzazioni di ruolo per il AWS Glue servizio)
- Errore: DescribeVpcEndpoints azione non autorizzata. impossibile convalidare l'ID VPC vpc-id
- Errore: DescribeRouteTables azione non autorizzata. impossibile convalidare l'id di sottorete: Subnet-ID in VPC id: vpc-id
- Errore: chiamata a ec2 non riuscita: DescribeSubnets
- Errore: chiamata a ec2 non riuscita: DescribeSecurityGroups
- Errore: impossibile trovare la sottorete per la zona di disponibilità
- Errore: eccezione dell'esecuzione del processo durante la scrittura in una destinazione JDBC
- Errore: Amazon S3: l'operazione non è valida per la classe di archiviazione dell'oggetto
- Errore: timeout di Amazon S3
- Errore: accesso ad Amazon S3 negato
- Errore: l'ID chiave di accesso Amazon S3 non esiste
- Errore: l'esecuzione del processo restituisce un errore durante l'accesso ad Amazon S3 con un URI s3a://
- Errore: token di servizio Amazon S3 scaduto
- Errore: non è stato trovato alcun DNS privato per l'interfaccia di rete
- Errore: provisioning dell'endpoint di sviluppo non riuscito
- Errore: server notebook CREATE_FAILED
- Errore: impossibile avviare il notebook locale
- Errore: esecuzione del crawler non riuscita
- Errore: le partizioni non sono state aggiornate
- Errore: aggiornamento del segnalibro del processo non riuscito a causa della mancata corrispondenza delle versioni
- Errore: un processo sta rielaborando i dati mentre i segnalibri del processo sono abilitati
- Errore: comportamento di failover tra VPCs AWS Glue


## Errore: risorsa non disponibile

Se AWS Glue restituisce un messaggio relativo alla risorsa non disponibile, puoi visualizzare i messaggi di errore o i log per ottenere ulteriori informazioni sul problema. Le attività seguenti descrivono i metodi generali per la risoluzione dei problemi.

- Per le connessioni e gli endpoint di sviluppo in uso, controlla che il cluster non abbia esaurito le interfacce di rete elastiche.

## Errore: impossibile trovare l'endpoint S3 o il gateway NAT per il subnetId in VPC

Controlla l'ID sottorete e l'ID VPC nel messaggio per diagnosticare il problema.

- Verifica che sia configurato un endpoint VPC Amazon S3, che è necessario con AWS Glue. Quindi, controlla se il gateway NAT fa parte della configurazione. Per ulteriori informazioni, consulta Endpoint Amazon VPC per Amazon S3.

## Errore: regola in entrata obbligatoria nel gruppo di sicurezza

Almeno un gruppo di sicurezza deve aprire tutte le porte di ingresso. Per limitare il traffico, è possibile limitare il gruppo di sicurezza di origine in una regola in entrata allo stesso gruppo di sicurezza.

- Per le connessioni in uso, verifica nel tuo gruppo di sicurezza la presenza di una regola in entrata autoreferenziale. Per ulteriori informazioni, consulta Impostazione dell'accesso di rete agli archivi di dati.
- Quando usi un endpoint di sviluppo, verifica nel gruppo di sicurezza la presenza di una regola in entrata autoreferenziale. Per ulteriori informazioni, consulta Impostazione dell'accesso di rete agli archivi di dati.

## Errore: regola in uscita obbligatoria nel gruppo di sicurezza

Almeno un gruppo di sicurezza deve aprire tutte le porte di uscita. Per limitare il traffico, è possibile limitare il gruppo di sicurezza di origine in una regola in uscita allo stesso gruppo di sicurezza.

- Per le connessioni in uso, verifica nel tuo gruppo di sicurezza la presenza di una regola in uscita autoreferenziale. Per ulteriori informazioni, consulta Impostazione dell'accesso di rete agli archivi di dati.
- Quando usi un endpoint di sviluppo, verifica nel gruppo di sicurezza la presenza di una regola in uscita autoreferenziale. Per ulteriori informazioni, consulta Impostazione dell'accesso di rete agli archivi di dati.

## Errore: esecuzione del Job non riuscita perché al ruolo passato devono essere assegnate (presupponiamo le autorizzazioni di ruolo per il AWS Glue servizio)

L'utente che definisce un processo deve avere l'autorizzazione `iam:PassRole` per AWS Glue.

- Quando un utente crea un AWS Glue lavoro, conferma che il ruolo dell'utente contenga una politica che contenga `iam:PassRole` for AWS Glue. Per ulteriori informazioni, consulta Fase 3: Collegamento di una policy agli utenti o ai gruppi che accedono a AWS Glue.

## Errore: DescribeVpcEndpoints azione non autorizzata. impossibile convalidare l'ID VPC vpc-id

- Controlla la policy a cui è stata assegnata l'autorizzazione. AWS Glue `ec2:DescribeVpcEndpoints`

## Errore: DescribeRouteTables azione non autorizzata. impossibile convalidare l'id di sottorete: Subnet-ID in VPC id: vpc-id

- Controlla la policy AWS Glue a `ec2:DescribeRouteTables` cui è stata passata l'autorizzazione.

## Errore: chiamata a ec2 non riuscita: DescribeSubnets

- Controlla la policy a cui è stata passata AWS Glue l'`ec2:DescribeSubnets`autorizzazione.

## Errore: chiamata a ec2 non riuscita: DescribeSecurityGroups

- Controlla la policy a cui è stata passata AWS Glue l' `ec2:DescribeSecurityGroups`autorizzazione.

## Errore: impossibile trovare la sottorete per la zona di disponibilità

- La zona di disponibilità potrebbe non essere disponibile per AWS Glue. Crea e utilizza una nuova sottorete in una zona di disponibilità diversa da quella specificata nel messaggio.

## Errore: eccezione dell'esecuzione del processo durante la scrittura in una destinazione JDBC

Quando esegui un processo che scrive in una destinazione JDBC, il processo potrebbe riscontrare errori nei seguenti scenari:

- Se il processo scrive in una tabella di Microsoft SQL Server e la tabella ha colonne definite di tipo `Boolean`, la tabella deve essere predefinita nel database SQL Server. Quando si definisce il processo sulla AWS Glue console utilizzando una destinazione SQL Server con l'opzione Crea tabelle nella destinazione dei dati, non mappare alcuna colonna di origine su una colonna di destinazione con tipo di dati`Boolean`. Potresti riscontrare un errore durante l'esecuzione del processo.

  Puoi evitare gli errori seguendo questa procedura:

  - Scegli una tabella esistente con la colonna Boolean (Booleano).
  - Modifica la trasformazione `ApplyMapping` e mappa la colonna Boolean (Booleano) nell'origine a un numero o una stringa della destinazione.
  - Modifica la trasformazione `ApplyMapping` per rimuovere la colonna Boolean (Booleano) dall'origine.
- Se il processo scrive in una tabella Oracle, potrebbe essere necessario regolare la lunghezza dei nomi degli oggetti Oracle. In alcune versioni di Oracle, la lunghezza massima degli identificatori è limitata a 30 byte o 128 byte. Questo limite riguarda i nomi di tabella e di colonna dei datastore di destinazione di Oracle.

  Puoi evitare gli errori seguendo questa procedura:

  - Denomina le tabelle di destinazione di Oracle entro il limite della tua versione.
  - I nomi di colonna predefiniti sono generati dai nomi dei campi nei dati. Per gestire il caso in cui i nomi di colonna sono più lunghi del limite, utilizza le trasformazioni `ApplyMapping` o `RenameField` per modificare il nome della colonna in modo che sia entro il limite.

## Errore: Amazon S3: l'operazione non è valida per la classe di archiviazione dell'oggetto

Se AWS Glue restituisce questo errore, è possibile che il tuo AWS Glue lavoro stia leggendo dati da tabelle con partizioni tra livelli di classi di storage Amazon S3.

- Utilizzando le esclusioni delle classi di storage, puoi assicurarti che i tuoi AWS Glue job funzionino su tabelle con partizioni tra questi livelli di classi di storage. Senza esclusioni, i processi che leggono i dati da questi livelli falliscono con il seguente errore: `AmazonS3Exception: The operation is not valid for the object's storage class`.

  Per ulteriori informazioni, consulta Esclusione delle classi di storage Amazon S3.

## Errore: timeout di Amazon S3

Se AWS Glue restituisce un errore di timeout della connessione, potrebbe essere perché sta tentando di accedere a un bucket Amazon S3 in un'altra regione. AWS

- Un endpoint VPC Amazon S3 può indirizzare il traffico solo verso i bucket all'interno di una regione. AWS Se hai bisogno di connetterti ai bucket di altre regioni, una possibile soluzione alternativa è quella di utilizzare un gateway NAT. Per ulteriori informazioni, consulta Gateway NAT.

## Errore: accesso ad Amazon S3 negato

Se AWS Glue restituisce un errore di accesso negato a un bucket o oggetto Amazon S3, è possibile che il ruolo IAM fornito non disponga di una policy che autorizzi il tuo data store.

- Un processo ETL deve avere accesso a un datastore Amazon S3 usato come origine o destinazione. Un crawler deve avere accesso a un datastore Amazon S3 in cui viene eseguito il crawling. Per ulteriori informazioni, consulta Fase 2: creare un ruolo IAM per AWS Glue.

## Errore: l'ID chiave di accesso Amazon S3 non esiste

Se durante l'esecuzione di un processo viene AWS Glue restituito un errore con l'ID della chiave di accesso ID not exist, potrebbe essere dovuto a uno dei seguenti motivi:

- Un processo ETL usa un ruolo IAM per accedere ai datastore. Verifica che il ruolo IAM per il processo non sia stato eliminato prima dell'inizio del processo.
- Un ruolo IAM contiene le autorizzazioni per accedere ai tuoi datastore. Verifica che qualsiasi policy Amazon S3 collegata contenente `s3:ListBucket` sia corretta.

# Errore: l'esecuzione del processo restituisce un errore durante l'accesso ad Amazon S3 con un URI **s3a://**

Se un processo restituisce un errore, ad esempio Failed to parse XML document with handler class (Impossibile analizzare il documento XML con classe del gestore), potrebbe essere a causa di un errore nel tentativo di elencare centinaia di file utilizzando un URI `s3a://`. Accedi al datastore utilizzando un URI `s3://`. La traccia di eccezione seguente evidenzia gli errori da cercare:

```
1. com.amazonaws.SdkClientException: Failed to parse XML document with handler class
 com.amazonaws.services.s3.model.transform.XmlResponsesSaxParser$ListBucketHandler
2. at
 com.amazonaws.services.s3.model.transform.XmlResponsesSaxParser.parseXmlInputStream(XmlRespons
3. at
 com.amazonaws.services.s3.model.transform.XmlResponsesSaxParser.parseListBucketObjectsResponse
4. at com.amazonaws.services.s3.model.transform.Unmarshallers
$ListObjectsUnmarshaller.unmarshall(Unmarshallers.java:70)
5. at com.amazonaws.services.s3.model.transform.Unmarshallers
$ListObjectsUnmarshaller.unmarshall(Unmarshallers.java:59)
6. at
 com.amazonaws.services.s3.internal.S3XmlResponseHandler.handle(S3XmlResponseHandler.java:62)
7. at
 com.amazonaws.services.s3.internal.S3XmlResponseHandler.handle(S3XmlResponseHandler.java:31)
8. at
 com.amazonaws.http.response.AwsResponseHandlerAdapter.handle(AwsResponseHandlerAdapter.java:70
9. at com.amazonaws.http.AmazonHttpClient
$RequestExecutor.handleResponse(AmazonHttpClient.java:1554)
10. at com.amazonaws.http.AmazonHttpClient
$RequestExecutor.executeOneRequest(AmazonHttpClient.java:1272)
11. at com.amazonaws.http.AmazonHttpClient
$RequestExecutor.executeHelper(AmazonHttpClient.java:1056)
12. at com.amazonaws.http.AmazonHttpClient
$RequestExecutor.doExecute(AmazonHttpClient.java:743)
13. at com.amazonaws.http.AmazonHttpClient
$RequestExecutor.executeWithTimer(AmazonHttpClient.java:717)
14. at com.amazonaws.http.AmazonHttpClient
$RequestExecutor.execute(AmazonHttpClient.java:699)
15. at com.amazonaws.http.AmazonHttpClient$RequestExecutor.access
$500(AmazonHttpClient.java:667)
16. at com.amazonaws.http.AmazonHttpClient
$RequestExecutionBuilderImpl.execute(AmazonHttpClient.java:649)
17. at com.amazonaws.http.AmazonHttpClient.execute(AmazonHttpClient.java:513)
```

```
18. at com.amazonaws.services.s3.AmazonS3Client.invoke(AmazonS3Client.java:4325)
19. at com.amazonaws.services.s3.AmazonS3Client.invoke(AmazonS3Client.java:4272)
20. at com.amazonaws.services.s3.AmazonS3Client.invoke(AmazonS3Client.java:4266)
21. at com.amazonaws.services.s3.AmazonS3Client.listObjects(AmazonS3Client.java:834)
22. at org.apache.hadoop.fs.s3a.S3AFileSystem.getFileStatus(S3AFileSystem.java:971)
23. at
 org.apache.hadoop.fs.s3a.S3AFileSystem.deleteUnnecessaryFakeDirectories(S3AFileSystem.java:115
24. at org.apache.hadoop.fs.s3a.S3AFileSystem.finishedWrite(S3AFileSystem.java:1144)
25. at org.apache.hadoop.fs.s3a.S3AOutputStream.close(S3AOutputStream.java:142)
26. at org.apache.hadoop.fs.FSDataOutputStream
$PositionCache.close(FSDataOutputStream.java:74)
27. at org.apache.hadoop.fs.FSDataOutputStream.close(FSDataOutputStream.java:108)
28. at org.apache.parquet.hadoop.ParquetFileWriter.end(ParquetFileWriter.java:467)
29. at
 org.apache.parquet.hadoop.InternalParquetRecordWriter.close(InternalParquetRecordWriter.java:1
30. at
 org.apache.parquet.hadoop.ParquetRecordWriter.close(ParquetRecordWriter.java:112)
31. at
 org.apache.spark.sql.execution.datasources.parquet.ParquetOutputWriter.close(ParquetOutputWrit
32. at org.apache.spark.sql.execution.datasources.FileFormatWriter
$SingleDirectoryWriteTask.releaseResources(FileFormatWriter.scala:252)
33. at org.apache.spark.sql.execution.datasources.FileFormatWriter$$anonfun
$org$apache$spark$sql$execution$datasources$FileFormatWriter$$executeTask
$3.apply(FileFormatWriter.scala:191)
34. at org.apache.spark.sql.execution.datasources.FileFormatWriter$$anonfun
$org$apache$spark$sql$execution$datasources$FileFormatWriter$$executeTask
$3.apply(FileFormatWriter.scala:188)
35. at org.apache.spark.util.Utils
$.tryWithSafeFinallyAndFailureCallbacks(Utils.scala:1341)
36. at org.apache.spark.sql.execution.datasources.FileFormatWriter$.org$apache$spark
$sql$execution$datasources$FileFormatWriter$$executeTask(FileFormatWriter.scala:193)
37. at org.apache.spark.sql.execution.datasources.FileFormatWriter$$anonfun$write$1$
$anonfun$3.apply(FileFormatWriter.scala:129)
38. at org.apache.spark.sql.execution.datasources.FileFormatWriter$$anonfun$write$1$
$anonfun$3.apply(FileFormatWriter.scala:128)
39. at org.apache.spark.scheduler.ResultTask.runTask(ResultTask.scala:87)
40. at org.apache.spark.scheduler.Task.run(Task.scala:99)
41. at org.apache.spark.executor.Executor$TaskRunner.run(Executor.scala:282)
42. at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
43. at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
44. at java.lang.Thread.run(Thread.java:748)
```

## Errore: token di servizio Amazon S3 scaduto

Durante il trasferimento di dati da e verso Amazon Redshift, vengono utilizzate le credenziali Amazon S3 temporanee che scadono dopo un'ora. Se stai eseguendo un processo lungo, potrebbe non riuscire. Per ulteriori informazioni su come configurare i processi lunghi per trasferire i dati da e verso Amazon Redshift, consulta aws-glue-programming-etl-connect-redshift-home.

## Errore: non è stato trovato alcun DNS privato per l'interfaccia di rete

Se un processo ha esito negativo o un endpoint di sviluppo non è in grado di effettuare il provisioning, potrebbe essere a causa di un problema della configurazione della rete.

- Se utilizzi il DNS fornito da Amazon, il valore di `enableDnsHostnames` deve essere impostato su "true". Per ulteriori informazioni, consulta DNS.

## Errore: provisioning dell'endpoint di sviluppo non riuscito

Se AWS Glue non riesce a fornire correttamente un endpoint di sviluppo, potrebbe essere a causa di un problema nella configurazione della rete.

- Quando definisci un endpoint di sviluppo, il VPC, la sottorete e i gruppi di sicurezza vengono convalidati per confermare che soddisfano determinati requisiti.
- Se hai fornito la chiave pubblica SSH opzionale, controlla che si tratti di una chiave pubblica SSH valida.
- Controlla nella console VPC che il tuo VPC utilizzi un set di opzioni DHCP valido. Per maggiori informazioni, consulta Set di opzioni DHCP.
- Se il cluster rimane nello stato PROVISIONING, contatta Supporto AWS.

## Errore: server notebook CREATE_FAILED

Se AWS Glue non riesce a creare il server notebook per un endpoint di sviluppo, potrebbe essere dovuto a uno dei seguenti problemi:

- AWS Glue passa un ruolo IAM ad Amazon EC2 durante la configurazione del server notebook. Il ruolo IAM deve avere una relazione di trust per Amazon EC2.
- Il ruolo IAM deve disporre di un profilo dell'istanza con lo stesso nome. Quando crei il ruolo per Amazon EC2 con la console IAM, il profilo dell'istanza con lo stesso nome viene creato

automaticamente. Verifica nel log la presenza di un errore relativo al nome del profilo dell'istanza `iamInstanceProfile.name` che non è valido. Per ulteriori informazioni, consulta Utilizzo dei profili dell'istanza .

- Verifica che il tuo ruolo sia autorizzato ad accedere ai bucket `aws-glue*` nella policy che passi per creare il server notebook.

## Errore: impossibile avviare il notebook locale

Se il notebook locale non si avvia e segnala errori indicanti che una directory o una cartella non viene trovata, potrebbe essere a causa di uno dei seguenti problemi:

- Se utilizzi Microsoft Windows, accertati che la variabile di ambiente `JAVA_HOME` punti alla directory di Java corretta. È possibile aggiornare Java senza aggiornare questa variabile; se fanno riferimento a una cartella che non esiste più, i notebook Jupyter non riescono ad avviarsi.

## Errore: esecuzione del crawler non riuscita

Se AWS Glue non riesce a eseguire correttamente un crawler per catalogare i dati, potrebbe essere per uno dei seguenti motivi. Per prima cosa controlla se è presente un errore nell'elenco dei crawler della console AWS Glue . Controlla se è presente un'icona con il punto esclamativo accanto al nome del crawler e passa il puntatore sopra l'icona per visualizzare i messaggi associati.

- Controlla i log del crawler eseguito nella sezione Logs in. CloudWatch `/aws-glue/crawlers`

## Errore: le partizioni non sono state aggiornate

Nel caso in cui le partizioni non siano state aggiornate nel Data Catalog durante l'esecuzione di un job ETL, queste istruzioni di registro della `DataSink` classe nei log possono essere utili: CloudWatch

- `"Attempting to fast-forward updates to the Catalog - nameSpace:"` — Mostra quale database, tabella e catalogId il processo sta tentando di modificare. Se questa istruzione non è presente qui, verifica se `enableUpdateCatalog` è impostato su true e se è stato correttamente passato come parametro `getSink()` o in `additional_options`.
- `"Schema change policy behavior:"` — Mostra il valore `updateBehavior` dello schema passato.
- “`Schemas qualify (schema compare):"` — Sarà vero o falso.

- "`Schemas qualify (case-insensitive compare):`" — Sarà vero o falso.
- Se entrambi sono falsi e il tuo non `updateBehavior` è impostato su `UPDATE_IN_DATABASE`, lo DynamicFrame schema deve essere identico o contenere un sottoinsieme delle colonne visualizzate nello schema della tabella Data Catalog.

Per ulteriori informazioni sull'aggiornamento delle partizioni, consulta la pagina Aggiornamento dello schema e aggiunta di nuove partizioni nel catalogo dati usando processi ETL AWS Glue.

## Errore: aggiornamento del segnalibro del processo non riuscito a causa della mancata corrispondenza delle versioni

È possibile che tu stia tentando di parametrizzare i AWS Glue job per applicarli a transformation/logic set di dati diversi in Amazon S3. Desideri tenere traccia dei file elaborati nelle posizioni fornite. Quando esegui lo stesso processo sullo stesso bucket di origine e scrivi contemporaneamente sulla same/different destinazione (concorrenza >1), il processo fallisce con questo errore:

```
py4j.protocol.Py4JJavaError: An error occurred while
 callingz:com.amazonaws.services.glue.util.Job.commit.:com.amazonaws.services.gluejobexecutor.m
 Continuation update failed due to version mismatch. Expected version 2 but found
 version 3
```

Soluzione: imposta la simultaneità su 1 o non eseguire il processo contemporaneamente.

Attualmente AWS Glue i segnalibri non supportano le esecuzioni di job simultanee e i commit avranno esito negativo.

## Errore: un processo sta rielaborando i dati mentre i segnalibri del processo sono abilitati

In alcuni casi, è possibile che abbiate abilitato i segnalibri dei AWS Glue job, ma il job ETL stia rielaborando dati che erano già stati elaborati in un'esecuzione precedente. Controlla queste cause comuni di questo errore:

Max concorrenza

L'impostazione del numero massimo di esecuzioni simultanee per il processo maggiore del valore predefinito di 1 può interferire con i segnalibri dei lavori. Ciò può accadere quando i segnalibri del

processo verificano l'ora dell'ultima modifica degli oggetti per individuare gli oggetti da rielaborare. Per ulteriori informazioni, consulta la discussione sulla concorrenza massima in Configurazione delle proprietà dei job per i job Spark in AWS Glue.

Oggetto del processo mancante

Assicurati che lo script di esecuzione del lavoro termini con il seguente commit:

```
job.commit()
```

Quando si include questo oggetto, AWS Glue registra il timestamp e il percorso dell'esecuzione del processo. Se si esegue nuovamente il processo con lo stesso percorso, AWS Glue elabora solo i nuovi file. Se non includi questo oggetto e i segnalibri del processo sono abilitati, il processo rielabora i file già elaborati con i nuovi file e crea la ridondanza nel datastore di destinazione del processo.

Parametro del contesto di trasformazione mancante

Il contesto di trasformazione è un parametro facoltativo nella classe `GlueContext`, ma i segnalibri non funzionano se non lo includi. Per risolvere questo errore, aggiungete il parametro del contesto di trasformazione quando create il DynamicFrame, come illustrato di seguito:

```
sample_dynF=create_dynamic_frame_from_catalog(database,
 table_name,transformation_ctx="sample_dynF")
```

Origine input

Se si sta usando un database relazionale (una connessione JDBC) per l'origine input, i segnalibri del processo funzionano solo se le chiavi primarie della tabella sono in ordine sequenziale. I segnalibri del processo funzionano per le nuove righe, ma non per le righe aggiornate. Questo perché i segnalibri processo cercano le chiavi primarie, già esistenti. Questo non si applica se l'origine di input è Amazon Simple Storage Service (Amazon S3).

Ora ultima modifica

Per origini di input Amazon S3, i segnalibri del processo controllano l'ora dell'ultima modifica piuttosto che i nomi dei file, per verificare gli oggetti da rielaborare. Se i dati dell'origine di input sono stati modificati dall'ultima esecuzione del processo, i file vengono rielaborati alla nuova esecuzione del processo.

## Errore: comportamento di failover tra VPCs AWS Glue

Il seguente processo viene utilizzato per il failover dei lavori nella AWS Glue versione 5.1 e nelle versioni precedenti.

Riepilogo: viene selezionata una AWS Glue connessione al momento dell'invio di un processo. Se l'esecuzione del processo riscontra dei problemi (mancanza di indirizzi IP, connettività all'origine, problema di instradamento), essa avrà esito negativo. Se sono configurati nuovi tentativi, AWS Glue riproverà con la stessa connessione.

1. Per ogni tentativo di esecuzione, AWS Glue controllerà lo stato delle connessioni nell'ordine indicato nella configurazione del processo, finché non ne troverà una da utilizzare. In caso di errore nella zona di disponibilità (AZ), le connessioni da quella zona non supereranno il controllo e verranno ignorate.
2. AWS Glue convalida la connessione con quanto segue:
   - Verifica la presenza di ID e sottorete Amazon VPC validi.
   - Verifica l'esistenza di un gateway NAT o di un endpoint Amazon VPC.
   - Verifica che la sottorete abbia più di 0 indirizzi IP allocati.
   - Verifica che l'AZ sia integra.

   AWS Glue non è in grado di verificare la connettività al momento dell'invio dell'esecuzione del processo.
3. Per i processi che utilizzano Amazon VPC, tutti i driver e gli esecutori verranno creati nella stessa AZ con la connessione selezionata al momento dell'invio dell'esecuzione del processo.
4. Se sono configurati nuovi tentativi, AWS Glue riproverà con la stessa connessione. Questo perché non è detto che i problemi con questa connessione durino a lungo. Se un'AZ riscontra un errore, le esecuzioni di processo esistenti (a seconda della fase in cui si trovano) in tale AZ possono riscontrare a loro volta un errore. Un nuovo tentativo dovrebbe rilevare un errore di AZ e scegliere un'altra AZ per la nuova esecuzione.

# Errori del crawler quando il crawler usa le autorizzazioni di Lake Formation

Utilizza le informazioni seguenti per diagnosticare e risolvere vari problemi durante la configurazione del crawler che utilizza le credenziali di Lake Formation.

## Errore: la posizione S3: s3://examplepath non è registrata

Affinché un crawler possa funzionare utilizzando le credenziali di Lake Formation, devi prima configurare le autorizzazioni di Lake Formation. Per risolvere questo errore, registra la posizione Amazon S3 di destinazione con Lake Formation. Per ulteriori informazioni, consulta la pagina Registrazione di una posizione Amazon S3.

## Errore: non User/Role è autorizzato a eseguire: lakeformation: on resource GetDataAccess

Aggiungi l'autorizzazione `lakeformation:GetDataAccess` al ruolo del crawler utilizzando la console IAM o AWS CLI. Con questa autorizzazione, Lake Formation concede la richiesta di credenziali temporanee per accedere ai dati. Vedi la politicy di seguito:

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "lakeformation:GetDataAccess"
    ],
    "Resource": "*"
  }
}
```

## Errore: autorizzazioni Lake Formation insufficienti su (nome del database: exampleDatabase, nome tabella: exampleTable)

Nella console Lake Formation (https://console.aws.amazon.com/lakeformation/), concedi al ruolo crawler i permessi di accesso ( `Create`,`Describe`,`Alter`) sul database, che è specificato come database di output. Puoi concedere le autorizzazioni anche sulla tabella. Per ulteriori informazioni, consulta Concessione delle autorizzazioni al database tramite il metodo delle risorse denominate.

## Errore: autorizzazioni di Lake Formation insufficienti su s3://examplepath

1. Crawling tra più account

a. Accedi alla console Lake Formation (https://console.aws.amazon.com/lakeformation/) utilizzando l'account in cui è registrato il bucket Amazon S3 (account B). Concedi le autorizzazioni per la posizione dei dati all'account in cui verrà eseguito il crawler. Ciò consentirà al crawler di leggere i dati dalla posizione Amazon S3 di destinazione.

b. Nell'account in cui viene creato il crawler (account A), concedi le autorizzazioni relative alla posizione dei dati nella posizione Amazon S3 di destinazione al ruolo IAM utilizzato per l'esecuzione del crawler, in modo che quest'ultimo possa leggere i dati dalla destinazione in Lake Formation. Per ulteriori informazioni, consulta Concessione delle autorizzazioni per la posizione dei dati (account esterno).

2. Nel crawling dell'account (il crawler e la posizione Amazon S3 sono nello stesso account): Concedi le autorizzazioni relative alla posizione dei dati al ruolo IAM utilizzato per l'esecuzione del crawler sulla posizione Amazon S3, in modo che il crawler possa leggere i dati dalla destinazione in Lake Formation. Per ulteriori informazioni, consulta la pagina Concessione delle autorizzazioni per la posizione dei dati (stesso account).

## Domande frequenti sulla configurazione del crawler utilizzando le credenziali di Lake Formation

1. Come posso configurare un crawler per l'esecuzione utilizzando le credenziali di Lake Formation tramite la console AWS ?

   Nella AWS Glue console (https://console.aws.amazon.com/glue/), durante la configurazione del crawler, seleziona l'opzione Usa le credenziali di Lake Formation per la scansione dell'origine dati Amazon S3. Per la scansione tra più account, specifica l' Account AWS ID in cui è registrata la sede Amazon S3 di destinazione con Lake Formation. Per effettuare il crawling all'interno dell'account, il campo accountId è facoltativo.

2. Come posso configurare un crawler per l'esecuzione utilizzando le credenziali di Lake Formation tramite AWS CLI?

   Durante la chiamata API `CreateCrawler`, aggiungi `LakeFormationConfiguration`:

```
"LakeFormationConfiguration": {
    "UseLakeFormationCredentials": true,
    "AccountId": "111111111111" (AWS account ID where the target Amazon S3 location
 is registered with Lake Formation)
  }
```

3. Quali sono le destinazioni supportate per un crawler che utilizza le credenziali di Lake Formation?

   Un crawler che utilizza le credenziali Lake Formation è supportato solo per Amazon S3 (crawling in un account e tra più account), per le destinazioni Catalogo dati in un account (dove la posizione sottostante è Amazon S3) e per le destinazioni Apache Iceberg.

4. Posso eseguire il crawling di più bucket Amazon S3 come parte di un singolo crawler utilizzando le credenziali di Lake Formation?

   No, per le destinazioni del crawling che utilizzano la distribuzione delle credenziali Lake Formation, le posizioni Amazon S3 sottostanti devono appartenere allo stesso bucket. Ad esempio, i clienti possono utilizzare più posizioni di destinazione (`s3://bucket1/folder1, s3://bucket1/folder2`) se sono sotto lo stesso bucket (bucket1). Specificare bucket diversi (s3://bucket1/folder1, s3://bucket2/folder2) non è supportato.

# Risoluzione dei problemi relativi AWS Glue agli errori di Ray nei log

> **Important**
>
> AWS Glue for Ray non è più aperto a nuovi clienti. I clienti esistenti possono continuare a utilizzare il servizio normalmente. Per ulteriori informazioni, consulta AWS Glue Ray End of Support.

AWS Glue fornisce l'accesso ai log emessi dai processi Ray durante l'esecuzione del lavoro. Se si riscontrano errori o comportamenti imprevisti nei processi Ray, raccogliere innanzitutto le informazioni dai log per determinare la causa dell'errore. Forniamo log simili anche per le sessioni interattive. I log delle sessioni sono forniti con il prefisso `/aws-glue/ray/sessions`.

Le righe di registro vengono inviate CloudWatch in tempo reale durante l'esecuzione del lavoro. Le istruzioni di stampa vengono aggiunte ai CloudWatch log al termine dell'esecuzione. I log vengono conservati per due settimane dopo l'esecuzione di un processo.

## Ispezione dei log dei processi Ray

Quando un processo non riesce, raccogli il nome e l'ID di esecuzione del processo. Puoi trovarle nella console. AWS Glue Accedi alla pagina del processo, quindi passa alla scheda Runs (Esecuzioni). I log di Ray Job sono archiviati nei seguenti gruppi di CloudWatch log dedicati.

- `/aws-glue/ray/jobs/script-log/`: archivia i log emessi dallo script principale di Ray.
- `/aws-glue/ray/jobs/ray-monitor-log/`: archivia i log emessi dal processo autoscaler di Ray. Questi log vengono generati per il nodo principale e non per altri nodi worker.
- `/aws-glue/ray/jobs/ray-gcs-logs/`: archivia i log emessi dal processo GCS (global control store). Questi log vengono generati per il nodo principale e non per altri nodi worker.
- `/aws-glue/ray/jobs/ray-process-logs/`: archivia i log emessi da altri processi Ray (principalmente l'agente del pannello di controllo) in esecuzione sul nodo principale. Questi log vengono generati per il nodo principale e non per altri nodi worker.
- `/aws-glue/ray/jobs/ray-raylet-logs/`: archivia i log emessi da ciascun processo raylet. Questi log vengono raccolti in un unico flusso per ogni nodo worker, incluso il nodo principale.
- `/aws-glue/ray/jobs/ray-worker-out-logs/`: archivia i log `stdout` per ogni worker nel cluster. Questi log vengono generati per ogni nodo worker, incluso il nodo principale.
- `/aws-glue/ray/jobs/ray-worker-err-logs/`: archivia i log `stderr` per ogni worker nel cluster. Questi log vengono generati per ogni nodo worker, incluso il nodo principale.
- `/aws-glue/ray/jobs/ray-runtime-env-log/`: archivia i log relativi al processo di configurazione di Ray. Questi log vengono generati per ogni nodo worker, incluso il nodo principale.

## Risoluzione degli errori relativi ai processi Ray

Per comprendere l'organizzazione dei gruppi di log di Ray e per individuare i gruppi di log che consentiranno di risolvere gli errori, è utile disporre di informazioni di base sull'architettura Ray.

In AWS Glue ETL, un worker corrisponde a un'istanza. Quando configuri i lavoratori per un AWS Glue lavoro, imposti il tipo e la quantità di istanze dedicate al lavoro. Ray usa il termine worker in diversi modi.

Ray utilizza i termini nodo principale e nodo worker per distinguere le responsabilità di un'istanza all'interno di un cluster Ray. Un nodo worker Ray può ospitare processi con più attori che eseguono calcoli per ottenere il risultato del calcolo distribuito. Gli attori che eseguono una replica di una funzione sono chiamati repliche. Gli attori di replica possono anche essere chiamati processi worker. Le repliche possono essere eseguite anche sul nodo principale, così chiamato perché esegue processi aggiuntivi per coordinare il cluster.

Ogni attore che contribuisce al calcolo genera il proprio flusso di log. Questo ci fornisce alcune informazioni dettagliate:

- Il numero minimo di processi che emettono i log potrebbe essere maggiore del numero di worker assegnati al processo. Spesso, ogni core di ogni istanza ha un attore.
- I nodi Ray principali emettono log di avvio e di gestione dei cluster. Al contrario, i nodi worker Ray emettono solo i log relativi al lavoro svolto su di essi.

Per ulteriori informazioni sull'architettura di Ray, consulta Whitepaper sull'architettura nella documentazione di Ray.

### Area problematica: accesso ad Amazon S3

Controlla il messaggio di errore dell'esecuzione del processo. Se ciò non fornisce informazioni sufficienti, controlla `/aws-glue/ray/jobs/script-log/`.

### Area problematica: gestione delle dipendenze PIP

Controlla `/aws-glue/ray/jobs/ray-runtime-env-log/`.

### Area problematica: ispezione dei valori intermedi nel processo principale

Scrivi su `stderr` o `stdout` dal tuo script principale e recupera i log da `/aws-glue/ray/jobs/script-log/`.

### Area problematica: ispezione dei valori intermedi in un processo secondario

Scrivi su `stderr` o `stdout` dalla funzione `remote`. Quindi, recupera i log da `/aws-glue/ray/jobs/ray-worker-out-logs/` o `/aws-glue/ray/jobs/ray-worker-err-logs/`. La funzione potrebbe essere stata eseguita su qualsiasi replica, quindi potrebbe essere necessario esaminare più log per trovare l'output previsto.

### Area problematica: interpretazione degli indirizzi IP nei messaggi di errore

In alcune situazioni di errore, il processo potrebbe emettere un messaggio di errore contenente un indirizzo IP. Questi indirizzi IP sono temporanei e vengono utilizzati dal cluster per identificare i nodi e per la comunicazione tra essi. I log di un nodo vengono pubblicati in un flusso di log con un suffisso univoco basato sull'indirizzo IP.

Inoltre CloudWatch, puoi filtrare i log per controllare quelli specifici di questo indirizzo IP identificando questo suffisso. Ad esempio, given *`FAILED_IP`* and*`JOB_RUN_ID`*, puoi identificare il suffisso con:

```
filter @logStream like /JOB_RUN_ID/
| filter @message like /IP-/
```

```
| parse @message "IP-[*]" as ip
| filter ip like /FAILED_IP/
| fields replace(ip, ":", "_") as uIP
| stats count_distinct by uIP as logStreamSuffix
| display logStreamSuffix
```

# Eccezioni di machine learning AWS Glue

In questo argomento vengono descritti i codici di errore HTTP e le stringhe per le eccezioni AWS Glue correlate al machine learning. I codici di errore e le stringhe di errore vengono forniti per ogni attività di machine learning che può verificarsi quando si esegue un'operazione. Inoltre, è possibile verificare se è possibile riprovare l'operazione che ha provocato l'errore.

## Annulla MLTask RunActivity

Questa attività presenta le seguenti eccezioni:

- EntityNotFoundException (400)
  - «Impossibile trovare MLTransform nell'account [AccountID] con handle [transformName].»
  - «Nessuna esecuzione ML Task Run trovata per [taskRunId]: nell'account [AccountId] per transform [transformName]».

  OK per riprovare: No.

## Crea MLTask RunActivity

Questa attività presenta le seguenti eccezioni:

- InvalidInputException (400)
  - “Errore del servizio interno a causa di un input imprevisto.”
  - “Una sorgente di input della tabella AWS Glue deve essere specificata nella trasformazione.”
  - “La colonna di origine di input [columnName] ha un tipo di dati non valido definito nel catalogo.”
  - “Deve essere fornita esattamente una tabella di record di input.”
  - “Dovrebbe specificare il nome del database.”
  - “Dovrebbe specificare il nome della tabella.”
  - “Lo schema non è definito nella trasformazione.”

- "Lo schema dovrebbe contenere una chiave primaria data: [primaryKey]."
- "Problema durante il recupero dello schema del catalogo dati: [message]."
- «Impossibile impostare la capacità massima e il lavoratore Num/Type contemporaneamente».
- «Entrambi WorkerType NumberOfWorkers devono essere impostati».
- «MaxCapacity dovrebbe essere >= [maxCapacity].»
- «NumberOfWorkers dovrebbe essere >= [maxCapacity].»
- "I tentativi massimi dovrebbero essere non negativi."
- "I parametri della ricerca di corrispondenze non sono stati impostati."
- "È necessario specificare una chiave primaria nei parametri della ricerca di corrispondenze."

OK per riprovare: No.

- AlreadyExistsException (400)
  - "La trasformazione con nome [transformName] esiste già."

  OK per riprovare: No.
- IdempotentParameterMismatchException (400)
  - "La richiesta di creazione idempotent per la trasformazione [transformName] aveva parametri non corrispondenti."

  OK per riprovare: No.
- InternalServiceException (500)
  - "Fallimento delle dipendenze."

  OK per riprovare: Sì.
- ResourceNumberLimitExceededException (400)
  - "Il numero di trasformazioni ML ([count]) ha superato il limite di [limit] trasformazioni."

  OK per riprovare: Sì, una volta eliminata una trasformazione per fare spazio a questa nuova.

## Elimina MLTransform attività

Questa attività presenta le seguenti eccezioni:

- EntityNotFoundException (400)
  - «Impossibile trovare MLTransform nell'account [AccountId] con handle [transformName]»

OK per riprovare: No.

## Ottieni MLTask RunActivity

Questa attività presenta le seguenti eccezioni:

- EntityNotFoundException (400)
  - «Impossibile trovare MLTransform nell'account [AccountID] con handle [transformName].»
  - «Nessuna esecuzione ML Task Run trovata per [taskRunId]: nell'account [AccountId] per transform [transformName]».

  OK per riprovare: No.

## Ottieni MLTask RunsActivity

Questa attività presenta le seguenti eccezioni:

- EntityNotFoundException (400)
  - «Impossibile trovare MLTransform nell'account [AccountID] con handle [transformName].»
  - «Nessuna esecuzione ML Task Run trovata per [taskRunId]: nell'account [AccountId] per transform [transformName]».

  OK per riprovare: No.

## Ottieni attività MLTransform

Questa attività presenta le seguenti eccezioni:

- EntityNotFoundException (400)
  - «Impossibile trovare MLTransform nell'account [AccountID] con handle [transformName].»

  OK per riprovare: No.

## Ottieni attività MLTransforms

Questa attività presenta le seguenti eccezioni:

- EntityNotFoundException (400)
  - «Impossibile trovare MLTransform nell'account [AccountID] con handle [transformName].»

  OK per riprovare: No.
- InvalidInputException (400)
  - "L'ID dell'account non può essere vuoto."
  - "Ordinamento non supportato per la colonna [column]."
  - "[column] non può essere vuota."
  - "Errore del servizio interno a causa di un input imprevisto."

  OK per riprovare: No.

## GetSaveLocationForTransformArtifactActivity

Questa attività presenta le seguenti eccezioni:

- EntityNotFoundException (400)
  - «Impossibile trovare MLTransform nell'account [AccountID] con handle [transformName].»

  OK per riprovare: No.
- InvalidInputException (400)
  - "Tipo di artefatto non supportato [ArtifactType]."
  - "Errore del servizio interno a causa di un input imprevisto."

  OK per riprovare: No.

## GetTaskRunArtifactActivity

Questa attività presenta le seguenti eccezioni:

- EntityNotFoundException (400)
  - «Impossibile trovare MLTransform nell'account [AccountID] con handle [transformName].»
  - «Nessuna esecuzione ML Task Run trovata per [taskRunId]: nell'account [AccountId] per transform [transformName]».

  OK per riprovare: No.

- InvalidInputException (400)
  - "Nome file '[fileName]' non valido per la pubblicazione."
  - "Impossibile recuperare artefatto per il tipo di attività [taskType]."
  - "Impossibile recuperare artefatto per [artifactType]."
  - "Errore del servizio interno a causa di un input imprevisto."

  OK per riprovare: No.

## Pubblica MLTransform ModelActivity

Questa attività presenta le seguenti eccezioni:

- EntityNotFoundException (400)
  - «Impossibile trovare MLTransform nell'account [AccountID] con handle [transformName].»
  - "Un modello esistente con versione - [version] non può essere trovato per ID account - [accountId] - e id di trasformazione - [transformId]."

  OK per riprovare: No.
- InvalidInputException (400)
  - "Nome file '[fileName]' non valido per la pubblicazione."
  - "Segno meno iniziale non valido sulla stringa non firmata [string]."
  - "Cifra non valida alla fine di [string]."
  - "Il valore della stringa [string] supera l'intervallo di unsigned long."
  - "Errore del servizio interno a causa di un input imprevisto."

  OK per riprovare: No.

## PullLatestMLTransformModelActivity

Questa attività presenta le seguenti eccezioni:

- EntityNotFoundException (400)
  - «Impossibile trovare MLTransform nell'account [AccountID] con handle [transformName].»

  OK per riprovare: No.

- InvalidInputException (400)
  - "Errore del servizio interno a causa di un input imprevisto."

  OK per riprovare: No.
- ConcurrentModificationException (400)
  - "Impossibile creare la versione del modello da addestrare a causa di inserti racing con parametri non corrispondenti."
  - "Il modello di trasformazione ML per id di trasformazione [transformId] è obsoleto o aggiornato da un altro processo; Riprovare."

  OK per riprovare: Sì.

## PutJobMetadataForMLTransformAttività

Questa attività presenta le seguenti eccezioni:

- EntityNotFoundException (400)
  - «Impossibile trovare MLTransform nell'account [AccountID] con handle [transformName].»
  - «Nessuna esecuzione ML Task Run trovata per [taskRunId]: nell'account [AccountId] per transform [transformName]».

  OK per riprovare: No.
- InvalidInputException (400)
  - "Errore del servizio interno a causa di un input imprevisto."
  - "Tipo di metadati processo sconosciuto [jobType]."
  - "È necessario fornire un ID di esecuzione attività per aggiornare."

  OK per riprovare: No.

## StartExportLabelsTaskRunActivity

Questa attività presenta le seguenti eccezioni:

- EntityNotFoundException (400)
  - «Impossibile trovare MLTransform nell'account [AccountID] con handle [transformName].»
  - "Nessun set di etichette esiste per transformId [transformId] nell'ID account [accountId]."

OK per riprovare: No.

- InvalidInputException (400)
  - "[message]."
  - "Il percorso S3 fornito non si trova nella stessa regione della trasformazione. Regione prevista - [region], ma ottenuta - [region]."

  OK per riprovare: No.

## StartImportLabelsTaskRunActivity

Questa attività presenta le seguenti eccezioni:

- EntityNotFoundException (400)
  - «Impossibile trovare MLTransform nell'account [AccountID] con handle [transformName].»

  OK per riprovare: No.
- InvalidInputException (400)
  - "[message]."
  - "Percorso del file di etichetta non valido."
  - "Impossibile accedere al file di etichetta in [labelPath]. [message]."
  - "Impossibile utilizzare il ruolo IAM fornito nella trasformazione. Ruolo: [role]."
  - "File etichetta di dimensione 0 non valido."
  - "Il percorso S3 fornito non si trova nella stessa regione della trasformazione. Regione prevista - [region], ma ottenuta - [region]."

  OK per riprovare: No.
- ResourceNumberLimitExceededException (400)
  - "Il file di etichetta ha superato il limite di [limit] MB."

  OK per riprovare: No. Considerare la possibilità di suddividere il file di etichetta in diversi file più piccoli.

## Inizio MLEvaluation TaskRunActivity

Questa attività presenta le seguenti eccezioni:

- EntityNotFoundException (400)
  - «Impossibile trovare MLTransform nell'account [AccountID] con handle [transformName].»

  OK per riprovare: No.
- InvalidInputException (400)
  - "Deve essere fornita esattamente una tabella di record di input."
  - "Dovrebbe specificare il nome del database."
  - "Dovrebbe specificare il nome della tabella."
  - "I parametri della ricerca di corrispondenze non sono stati impostati."
  - "È necessario specificare una chiave primaria nei parametri della ricerca di corrispondenze."

  OK per riprovare: No.
- MLTransformNotReadyException (400)
  - "Questa operazione può essere applicata solo a una trasformazione che si trova in uno stato READY."

  OK per riprovare: No.
- InternalServiceException (500)
  - "Fallimento delle dipendenze."

  OK per riprovare: Sì.
- ConcurrentRunsExceededException (400)
  - "Il numero di esecuzioni attività ML [count] ha superato il limite di trasformazione delle esecuzioni di attività [limit]."
  - "Il numero di esecuzioni attività ML [count] ha superato il limite di [limit] esecuzioni di attività."

  OK per riprovare: Sì, dopo aver atteso il completamento dell'esecuzione dell'attività.

## Inizio MLLabeling SetGenerationTaskRunActivity

Questa attività presenta le seguenti eccezioni:

- EntityNotFoundException (400)
  - «Impossibile trovare MLTransform nell'account [AccountID] con handle [transformName].»

  OK per riprovare: No.
- InvalidInputException (400)
  - “Deve essere fornita esattamente una tabella di record di input.”
  - “Dovrebbe specificare il nome del database.”
  - “Dovrebbe specificare il nome della tabella.”
  - “I parametri della ricerca di corrispondenze non sono stati impostati.”
  - “È necessario specificare una chiave primaria nei parametri della ricerca di corrispondenze.”

  OK per riprovare: No.
- InternalServiceException (500)
  - “Fallimento delle dipendenze.”

  OK per riprovare: Sì.
- ConcurrentRunsExceededException (400)
  - “Il numero di esecuzioni attività ML [count] ha superato il limite di trasformazione delle esecuzioni di attività [limit].”

  OK per riprovare: Sì, una volta completata l'esecuzione dell'attività.

## MLTransformAttività di aggiornamento

Questa attività presenta le seguenti eccezioni:

- EntityNotFoundException (400)
  - «Impossibile trovare MLTransform nell'account [AccountId] con handle [transformName].»

  OK per riprovare: No.
- InvalidInputException (400)
  - «Esiste già un'altra trasformazione con nome [TransformName].”
  - “[message].”
  - “Il nome della trasformazione non può essere vuoto.”
  - «Impossibile impostare la capacità massima e il lavoratore Num/Type contemporaneamente».

- «Entrambi WorkerType NumberOfWorkers devono essere impostati».
- «MaxCapacity dovrebbe essere >= [minMaxCapacity].»
- «NumberOfWorkers dovrebbe essere >= [minNumWorkers].»
- "I tentativi massimi dovrebbero essere non negativi."
- "Errore del servizio interno a causa di un input imprevisto."
- "I parametri della ricerca di corrispondenze non sono stati impostati."
- "È necessario specificare una chiave primaria nei parametri della ricerca di corrispondenze."

OK per riprovare: No.

- AlreadyExistsException (400)
  - "La trasformazione con nome [transformName] esiste già."

  OK per riprovare: No.
- IdempotentParameterMismatchException (400)
  - "La richiesta di creazione idempotent per la trasformazione [transformName] aveva parametri non corrispondenti."

  OK per riprovare: No.

# AWS Glue quote

È possibile contattare Supporto AWS per richiedere un aumento della quota per le quote di servizio elencate in. Riferimenti generali di AWS Salvo diversa indicazione, ogni quota si applica a una Regione specifica. Per ulteriori informazioni, consulta Endpoint e quote per AWS Glue.

# Migliorare AWS Glue le prestazioni

Strategia di base per l'ottimizzazione delle prestazioni

Per migliorare AWS Glue le prestazioni, potresti prendere in considerazione l'aggiornamento di alcuni parametri relativi alle AWS Glue prestazioni. Quando ci si prepara all'ottimizzazione dei parametri, utilizza la seguenti best practice:

- Determinare gli obiettivi di prestazioni prima di iniziare a identificare i problemi.
- Prima di provare di modificare i parametri di ottimizzazione, utilizza i parametri per identificare i problemi.

Per ottenere risultati più coerenti durante l'ottimizzazione di un processo, sviluppa una strategia di base per il lavoro di ottimizzazione.

In genere, l'ottimizzazione delle prestazioni viene eseguita nel seguente flusso di lavoro:

1. Determinazione degli obiettivi delle prestazioni.
2. Misurazione dei parametri.
3. Identificazione dei colli di bottiglia.
4. Riduzione dell'impatto dei colli di bottiglia.
5. Ripeti i passaggi da 2 a 4 fino a raggiungere l'obiettivo prefissato.

## Strategie di ottimizzazione per il tipo di processo

Lavori Spark: segui le indicazioni contenute nelle migliori pratiche per l'ottimizzazione delle prestazioni per i lavori di Apache Spark su AWS Glue Prescriptive Guidance. AWS

Altri lavori: è possibile eseguire l'ottimizzazione AWS Glue per i job della shell Ray e AWS Glue Python adattando le strategie disponibili in altri ambienti di runtime.

## Miglioramento delle prestazioni AWS Glue per i job di Apache Spark

AWS Glue Per migliorare le prestazioni di Spark, potresti prendere in considerazione l'aggiornamento di alcuni parametri relativi alle prestazioni AWS Glue e di Spark.

Per ulteriori informazioni sulle strategie specifiche per identificare gli ostacoli attraverso le metriche e ridurne l'impatto, consulta le migliori pratiche per l'ottimizzazione delle prestazioni per i lavori di Apache Spark su Prescriptive AWS Glue Guidance. AWS Questa guida introduce gli argomenti chiave applicabili ad Apache Spark in tutti gli ambienti di runtime, come l'architettura Spark e i Resilient Distributed Datasets. Utilizzando questi argomenti, la guida favorisce l'implementazione di strategie specifiche di ottimizzazione delle prestazioni, come l'ottimizzazione degli shuffle e la parallelizzazione delle attività.

Puoi identificare i punti deboli configurando la visualizzazione dell'interfaccia utente di Spark. AWS Glue Per ulteriori informazioni, consulta the section called “Monitoraggio con l'interfaccia utente di Spark”.

Inoltre, AWS Glue offre funzionalità prestazionali che possono essere applicabili al tipo specifico di archivio dati a cui si connette il job. Le informazioni di riferimento sui parametri prestazionali per gli archivi dati sono disponibili in the section called “Parametri di connessione”.

# Ottimizzazione delle letture con pushdown in Glue ETL AWS

Il pushdown è una tecnica di ottimizzazione che avvicina la logica di recupero dei dati all'origine dati. L'origine potrebbe essere un database o un file system come Amazon S3. Quando si eseguono determinate operazioni direttamente sulla sorgente, è possibile risparmiare tempo e potenza di elaborazione non trasferendo tutti i dati della rete al motore Spark gestito da AWS Glue.

Un altro modo per dire ciò è che il pushdown riduce la scansione dei dati. Per ulteriori informazioni sul processo di identificazione di quando questa tecnica è appropriata, consulta Reduce the amount of data scan nella guida Best practices for performance tuning AWS Glue for Apache Spark jobs su AWS Prescriptive Guidance.

## Il predicato pushdown sui file archiviati su Amazon S3

Quando lavori su Amazon S3 con file organizzati per prefisso, puoi filtrare i percorsi Amazon S3 di destinazione definendo un predicato pushdown. Invece di leggere il set di dati completo e applicare filtri all'interno di un `DynamicFrame`, è possibile applicare il filtro direttamente ai metadati della partizione archiviati in Catalogo dati AWS Glue. Questo approccio consente di elencare e leggere in modo selettivo solo i dati necessari. Per ulteriori informazioni su questo processo, inclusa la scrittura per partizioni in un bucket, consultare la pagina the section called “Gestione delle partizioni”.

È possibile ottenere il pushdown dei predicati in Amazon S3 utilizzando il parametro `push_down_predicate`. Prendere in considerazione un bucket in Amazon S3 partizionato per

anno, mese e giorno. Se desideri recuperare i dati dei clienti relativi a giugno 2022, puoi indicare a AWS Glue di leggere solo i percorsi Amazon S3 pertinenti. Il `push_down_predicate` in questo caso è `year='2022' and month='06'`. Mettendo tutto insieme, l'operazione di lettura può essere eseguita come segue:

Python

```
customer_records = glueContext.create_dynamic_frame.from_catalog(
    database = "customer_db",
    table_name = "customer_tbl",
    push_down_predicate = "year='2022' and month='06'"
)
```

Scala

```
val customer_records = glueContext.getCatalogSource(
database="customer_db",
tableName="customer_tbl",
pushDownPredicate="year='2022' and month='06'"
).getDynamicFrame()
```

Nello scenario precedente, `push_down_predicate` recupera un elenco di tutte le partizioni da Catalogo dati AWS Glue e le filtra prima di leggere i file Amazon S3 sottostanti. Mentre elencare le partizioni è utile nella maggior parte dei casi, questo processo può richiedere molto tempo quando si lavora con set di dati che contengono milioni di partizioni. Per risolvere questo problema e migliorare le prestazioni è possibile utilizzare l'eliminazione delle partizioni lato server. Questo viene fatto creando un indice di partizione per i dati nel AWS Glue Data Catalog. Per ulteriori informazioni sugli indici di partizione, consultare la pagina the section called "Creazione di indici di partizione". È quindi possibile utilizzare l'opzione `catalogPartitionPredicate` per fare riferimento all'indice. Per un esempio di recupero di partizioni con `catalogPartitionPredicate`, consultare la pagina the section called "Predicati di partizione di catalogo".

## Esecuzione del pushdown quando si utilizzano origini JDBC

Il lettore JDBC AWS Glue utilizzato nel pushdown `GlueContext` supporta i database supportati fornendo query SQL personalizzate che possono essere eseguite direttamente sul sorgente. Ciò può essere ottenuto impostando il parametro `sampleQuery`. La tua query di esempio può specificare quali colonne selezionare e fornire un predicato di pushdown per limitare i dati trasferiti al motore Spark.

Per impostazione predefinita, le query di esempio funzionano su un singolo nodo, il che può causare errori di processo quando si gestiscono grandi volumi di dati. Per utilizzare questa funzionalità per eseguire query sui dati su larga scala, è necessario configurare il partizionamento delle query impostando `enablePartitioningForSampleQuery` su true, che distribuirà la query su più nodi attraverso una chiave a scelta. Il partizionamento delle query richiede anche alcuni altri parametri di configurazione necessari. Per ulteriori informazioni sul partizionamento delle query, consultare la pagina the section called "Lettura in parallelo da JDBC".

Durante l'impostazione`enablePartitioningForSampleQuery`, AWS Glue combinerà il predicato pushdown con un predicato di partizionamento durante l'interrogazione del database. `sampleQuery`È necessario terminare con un `AND` for AWS Glue per aggiungere le condizioni di partizionamento. (Se non fornisci un predicato pushdown, `sampleQuery` deve terminare con un `WHERE`). Di seguito è possibile trovare un esempio in cui effettuiamo il pushdown di un predicato per recuperare solo le righe il cui `id` è maggiore di 1.000. Questa `sampleQuery` restituirà solo le colonne del nome e della posizione per le righe il cui `id` è maggiore del valore specificato:

Python

```
sample_query = "select name, location from customer_tbl WHERE id>=1000 AND"
customer_records = glueContext.create_dynamic_frame.from_catalog(
    database="customer_db",
    table_name="customer_tbl",
    sample_query = "select name, location from customer_tbl WHERE id>=1000 AND",

    additional_options = {
                           "hashpartitions": 36 ,
                           "hashfield":"id",
                           "enablePartitioningForSampleQuery":True,
                           "sampleQuery":sample_query
                          }
)
```

Scala

```
val additionalOptions = Map(
        "hashpartitions" -> "36",
        "hashfield" -> "id",
        "enablePartitioningForSampleQuery" -> "true",
        "sampleQuery" -> "select name, location from customer_tbl WHERE id >= 1000
 AND"
        )
```

```
    val customer_records = glueContext.getCatalogSource(
        database="customer_db",
        tableName="customer_tbl").getDynamicFrame()
```

> **Note**
>
> Se `customer_tbl` ha un nome diverso nel Catalogo dati e nel datastore sottostante, è necessario fornire il nome della tabella sottostante in sample_query, poiché la query viene passata al datastore sottostante.

Puoi anche eseguire query su tabelle JDBC senza integrarti con AWS Glue Data Catalog. Invece di fornire nome utente e password come parametri del metodo, è possibile riutilizzare le credenziali di una connessione preesistente fornendo `useConnectionProperties` e `connectionName`. In questo esempio, recuperiamo le credenziali da una connessione chiamata `my_postgre_connection`.

Python

```
connection_options_dict = {
    "useConnectionProperties": True,
    "connectionName": "my_postgre_connection",
    "dbtable":"customer_tbl",
    "sampleQuery":"select name, location from customer_tbl WHERE id>=1000 AND",
    "enablePartitioningForSampleQuery":True,
    "hashfield":"id",
    "hashpartitions":36
    }

customer_records = glueContext.create_dynamic_frame.from_options(
    connection_type="postgresql",
    connection_options=connection_options_dict
    )
```

Scala

```
val connectionOptionsJson = """
      {
```

```
        "useConnectionProperties": true,
        "connectionName": "my_postgre_connection",
        "dbtable": "customer_tbl",
        "sampleQuery": "select name, location from customer_tbl WHERE id>=1000 AND",
        "enablePartitioningForSampleQuery" : true,
        "hashfield" : "id",
        "hashpartitions" : 36
      }
    """

    val connectionOptions = new JsonOptions(connectionOptionsJson)

    val dyf = glueContext.getSource("postgresql",
connectionOptions).getDynamicFrame()
```

## Note e limitazioni per il pushdown in AWS Glue

Il pushdown, come concetto, è applicabile alla lettura da fonti non in streaming. AWS Glue supporta una varietà di fonti: la capacità di premere dipende dalla sorgente e dal connettore.

- Quando ci si connette a Snowflake, è possibile utilizzare l'opzione `query`. Funzionalità simili esistono nel connettore Redshift in AWS Glue 4.0 e versioni successive. Per ulteriori informazioni sulla lettura da Snowflake con `query`, consultare la pagina the section called "Lettura da Snowflake".
- Il lettore DynamoDB ETL non supporta filtri o predicati pushdown. Inoltre, MongoDB e DocumentDB non supportano questo tipo di funzionalità.
- Quando si leggono dati in formati di tabelle aperte archiviati in Amazon S3, il metodo di partizionamento dei file in Amazon S3 da solo non è più sufficiente. Per leggere e scrivere da partizioni utilizzando formati di tabelle aperte, consultare la documentazione relativa al formato.
- DynamicFrame i metodi non eseguono il pushdown di proiezione di Amazon S3. Tutte le colonne verranno lette dai file che soddisfanno il filtro dei predicati.
- Quando si lavora con i `custom.jdbc` connettori in AWS Glue, la capacità di premere dipende dalla sorgente e dal connettore. Consulta la documentazione del connettore appropriata per confermare se e come supporta il pushdown in AWS Glue.

# Utilizzo di Auto Scaling per AWS Glue

Il dimensionamento automatico è disponibile per il tuo ETL di AWS Glue, le sessioni interattive e i processi di streaming con AWS Glue versione 3.0 o successive.

Con Auto Scaling abilitato, otterrai i seguenti vantaggi:

- AWS Glue aggiunge automaticamente e rimuove i lavoratori del cluster a seconda del parallelismo in ogni fase o microbatch del processo in esecuzione.
- Elimina la necessità di sperimentare e decidere il numero di dipendenti da assegnare per i tuoi processi ETL di AWS Glue.
- Con il numero massimo di worker dato, AWS Glue sceglierà le risorse della dimensione giusta per il carico di lavoro.
- È possibile vedere come cambia la dimensione del cluster durante l'esecuzione del processo esaminando le CloudWatch metriche nella pagina dei dettagli dell'esecuzione del lavoro in AWS Glue Studio.

Il dimensionamento automatico per i processi ETL di AWS Glue e i processi di streaming consentono l'aumento e la riduzione orizzontali on demand delle risorse di calcolo dei tuoi processi AWS Glue. Il dimensionamento verticale on demand consente di allocare solo le risorse di calcolo richieste inizialmente all'avvio dell'esecuzione del processo e anche di effettuare il provisioning delle risorse richieste in base alla domanda durante il processo.

Il dimensionamento automatico supporta anche la riduzione dinamica orizzontale delle risorse dei processi AWS Glue nel corso di un processo. Durante l'esecuzione di un processo, quando vengono richiesti più esecutori dall'applicazione Spark, verrà aggiunto al cluster un numero maggiore di dipendenti. Quando l'esecutore sarà inattivo e non avrà attività di calcolo in corso, l'esecutore e il dipendente corrispondente verranno rimossi.

Gli scenari comuni in cui il dimensionamento automatico aiuta a ridurre i costi e l'utilizzo delle applicazioni Spark includono:

- un driver Spark che elenca un gran numero di file in Amazon S3 o esegue un caricamento mentre gli esecutori sono inattivi;
- le fasi Spark che vengono eseguite con pochi esecutori a causa dell'eccesso di provisioning;
- il disallineamento dei dati o la domanda di calcolo non uniforme tra le varie fasi di Spark.

## Requisiti

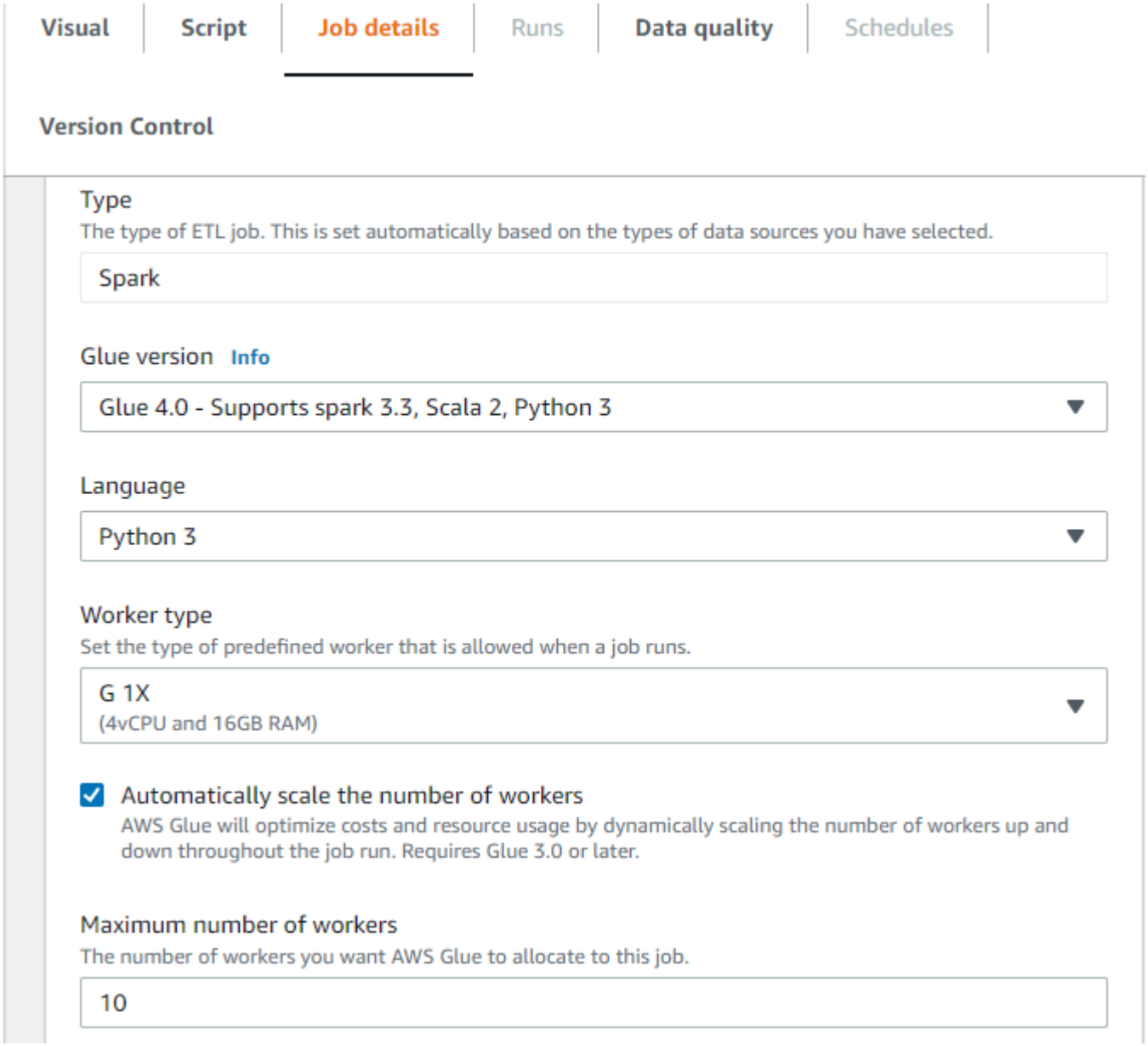
Dimensionamento automatico è disponibile solo per AWS Glue versione 3.0 o successiva. Per utilizzare Dimensionamento automatico, consulta la Guida alla migrazione per migrare i processi esistenti a AWS Glue 3.0 o versioni successive oppure creare nuovi processi con AWS Glue 3.0 o versioni successive.

Il dimensionamento automatico è disponibile per i processi AWS Glue con i tipi di worker `G.1X`, `G.2X`, `G.4X`, `G.8X`, `G.12X`, `G.16X`, `R.1X`, `R.2X`, `R.4X`, `R.8X` o `G.025X` (solo per i processi di streaming). DPUs Gli standard non sono supportati per Auto Scaling.

## Abilitazione dell'Auto Scaling in AWS Glue Studio

Nella scheda Dettagli processo in AWS Glue Studio, scegli il tipo Spark o Spark Streaming e Versione di Glue **`Glue 3.0`** o successive. In seguito, una casella di controllo verrà visualizzata sotto Tipo di worker.

- Seleziona l'opzione Dimensiona automaticamente il numero di worker.
- Imposta la proprietà Numero massimo di dipendenti per definire il numero massimo di dipendenti che possono essere ceduti all'esecuzione del processo.




## Abilitazione dell'Auto Scaling con CLI o SDK AWS

Per abilitare l'Auto Scaling dalla AWS CLI per l'esecuzione del processo, esegui `start-job-run` con la seguente configurazione:

```
{
    "JobName": "<your job name>",
    "Arguments": {
        "--enable-auto-scaling": "true"
    },
```

```
    "WorkerType": "G.2X", // G.1X, G.2X, G.4X, G.8X, G.12X, G.16X, R.1X, R.2X, R.4X,
 and R.8X are supported for Auto Scaling Jobs
    "NumberOfWorkers": 20, // represents Maximum number of workers
    ...other job run configurations...
}
```

Una volta terminata l'esecuzione del processo ETL, puoi anche chiamare`get-job-run` per verificare l'effettivo utilizzo delle risorse del processo eseguito in secondi DPU. Nota: il nuovo campo DPUSecondsverrà visualizzato solo per i lavori in batch nella AWS Glue versione 4.0 o successiva abilitata con Auto Scaling. Questo campo non è supportato per i processi di streaming.

```
$ aws glue get-job-run --job-name your-job-name --run-id jr_xx --endpoint https://
glue.us-east-1.amazonaws.com --region us-east-1
{
    "JobRun": {
        ...
        "GlueVersion": "3.0",
        "DPUSeconds": 386.0
    }
}
```

È inoltre possibile configurare le esecuzioni dei processi con Auto Scaling utilizzando l'SDK AWS Glue con la stessa configurazione.

## Abilitazione del dimensionamento automatico con sessioni interattive

Per abilitare l'Auto Scaling durante la creazione di AWS Glue lavori con sessioni interattive, consulta Configurazione delle AWS Glue sessioni interattive.

## Suggerimenti e considerazioni

Suggerimenti e considerazioni per ottimizzare l'Auto Scaling AWS Glue :

- Se non hai idea del valore iniziale del numero massimo di lavoratori, puoi iniziare dal calcolo approssimativo spiegato in Estimate DPU. AWS Glue Non è necessario configurare un valore estremamente elevato nel numero massimo di worker per dati di volume molto basso.
- AWS Glue Auto Scaling si configura `spark.sql.shuffle.partitions` e `spark.default.parallelism` si basa sul numero massimo di DPU (calcolato con il numero massimo di lavoratori e il tipo di lavoratore) configurato sul lavoro. Nel caso in cui preferisci il

valore fisso su tali configurazioni, puoi sovrascrivere questi parametri con i seguenti parametri del processo:

- Chiave: `--conf`
- Value (Valore): `spark.sql.shuffle.partitions=200 --conf spark.default.parallelism=200`

- Per i lavori di streaming, per impostazione predefinita, AWS Glue non esegue la scalabilità automatica all'interno di microbatch e richiede diversi micro batch per avviare la scalabilità automatica. Nel caso in cui desideri abilitare il dimensionamento automatico all'interno di micro batch, fornisci `--auto-scale-within-microbatch`. Per ulteriori informazioni, consulta Riferimento ai parametri di processo.

## Monitoraggio dell'Auto Scaling con i parametri di Amazon CloudWatch

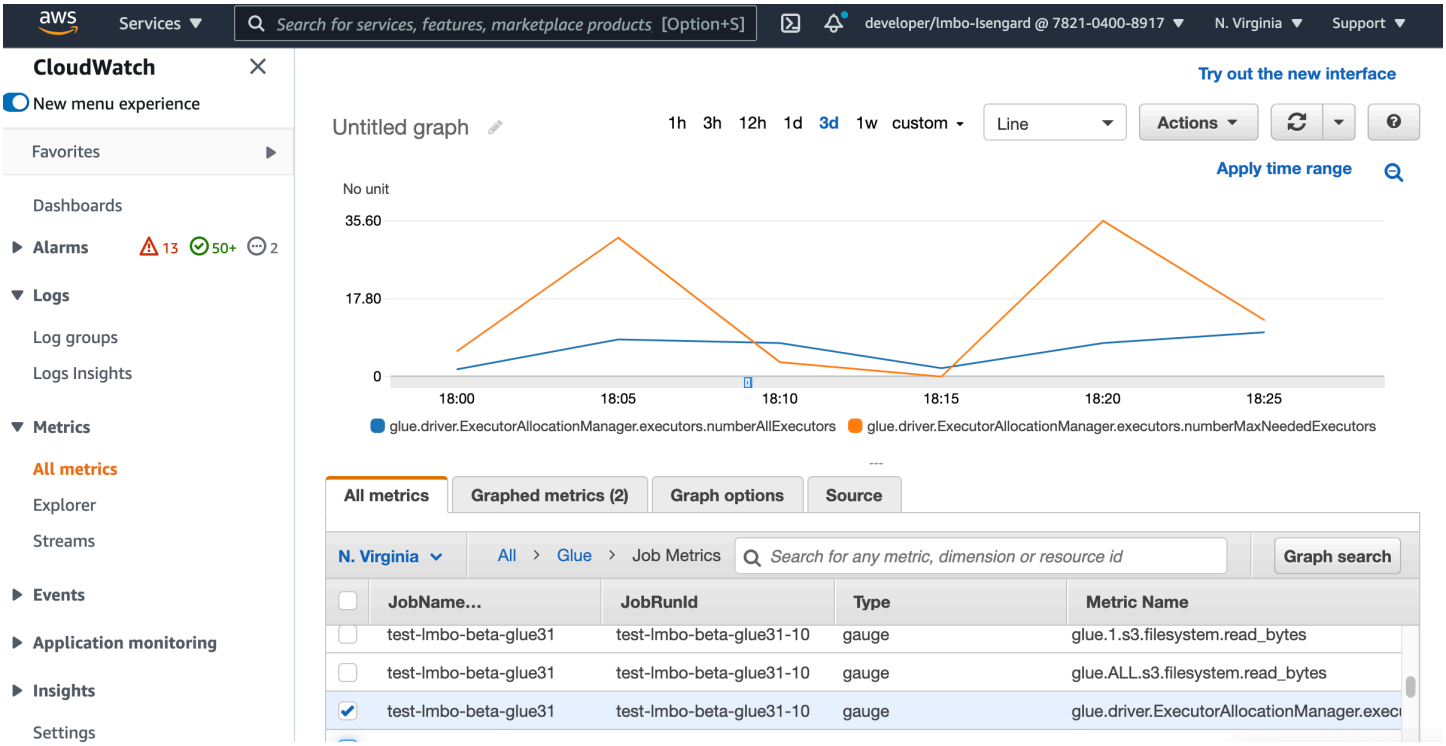
Le metriche dell' CloudWatch executor sono disponibili per i job AWS Glue 3.0 o versioni successive se abiliti Auto Scaling. I parametri possono essere impiegati per monitorare la domanda e l'utilizzo ottimizzato degli esecutori nelle applicazioni Spark abilitate con Auto Scaling. Per ulteriori informazioni, consulta Monitoraggio AWS Glue tramite CloudWatch parametri Amazon.

Puoi anche utilizzare metriche di AWS Glue osservabilità per ottenere informazioni sull'utilizzo delle risorse. Ad esempio, tramite il monitoraggio di `glue.driver.workerUtilization`, puoi tracciare la quantità di risorse effettivamente utilizzate con e senza il dimensionamento automatico. Un altro esempio: monitorando `glue.driver.skewness.job` e `glue.driver.skewness.stage`, puoi osservare come i dati sono disallineati. Queste informazioni ti aiuteranno a decidere di abilitare il dimensionamento automatico ed eseguire il fine-tuning delle configurazioni. Per ulteriori informazioni, consulta Monitoring with Monitoraggio con parametri AWS Glue di osservabilità.

- colla.driver. ExecutorAllocationManager.esecutori. numberAllExecutors
- colla.driver. ExecutorAllocationManager.esecutori. numberMaxNeededEsecutori

Per ulteriori dettagli su questi parametri, consulta Monitoraggio per la pianificazione della capacità DPU.

> **Note**
>
> CloudWatch le metriche degli esecutori non sono disponibili per le sessioni interattive.

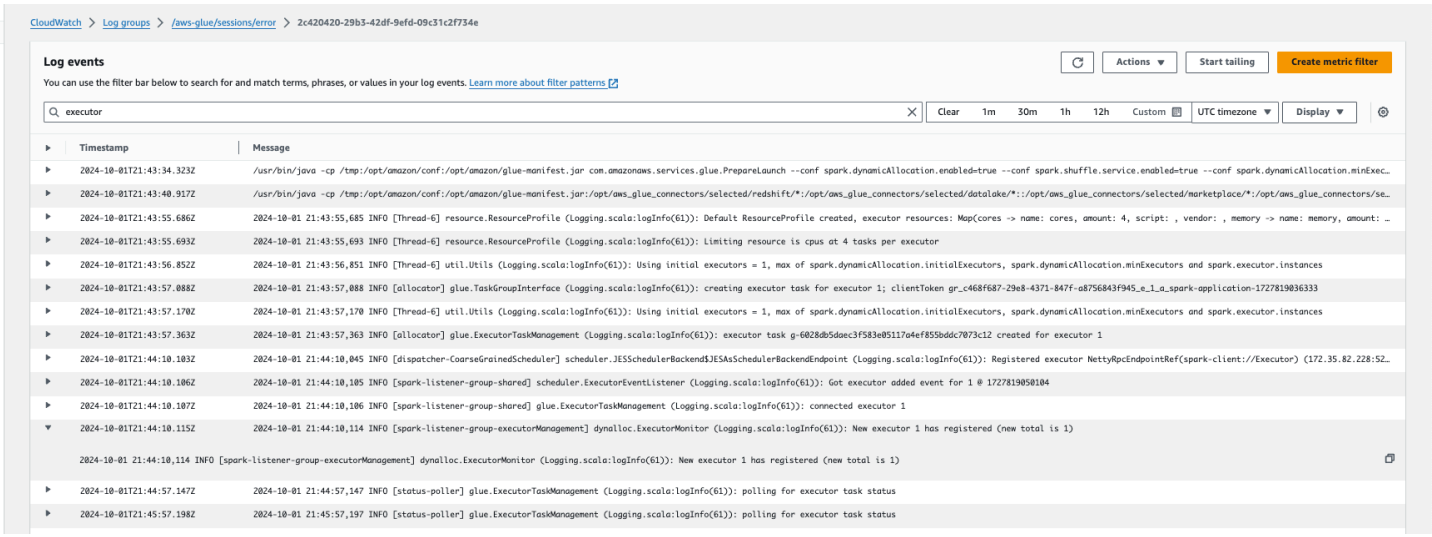



# Monitoraggio dell'Auto Scaling con Amazon Logs CloudWatch

Se utilizzi sessioni interattive, puoi monitorare il numero di executor abilitando Amazon CloudWatch Logs continui e cercando «executor» nei log o utilizzando l'interfaccia utente Spark. Per fare ciò, usa il comando magico `%%configure` per abilitare la registrazione continua insieme a `enable auto scaling`.

```
%%configure{
    "--enable-continuous-cloudwatch-log": "true",
    "--enable-auto-scaling": "true"
}
```

Negli CloudWatch eventi di Amazon Logs, cerca «executor» nei log:

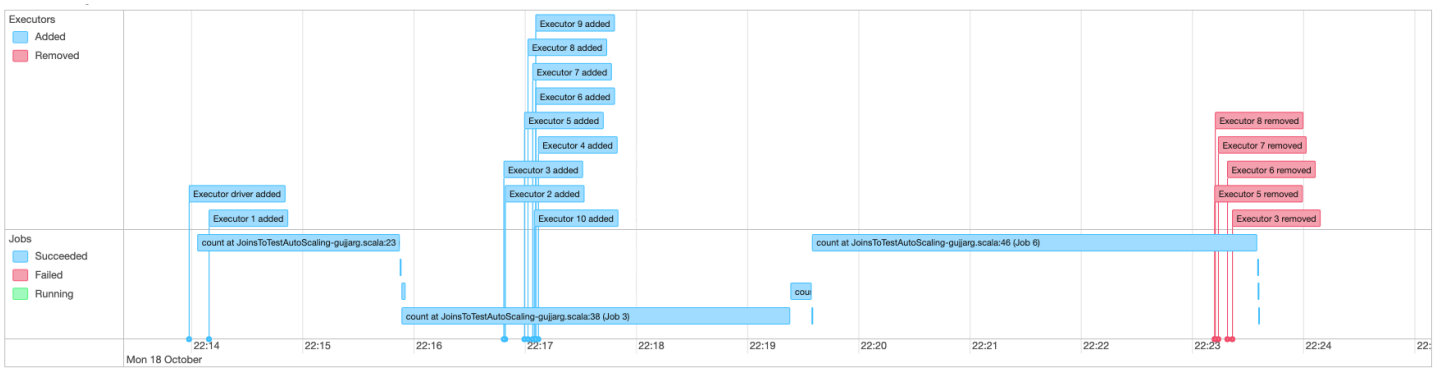



# Monitoraggio di Auto Scaling con interfaccia utente di Spark

Con Auto Scaling abilitato, puoi anche monitorare gli esecutori aggiunti e rimossi con il dimensionamento automatico verso l'alto e la riduzione verticale in base alla domanda nei tuoi processi AWS Glue l'interfaccia utente di Spark Glue. Per ulteriori informazioni, consulta Abilitazione dell'interfaccia utente Web di Apache Spark per processi AWS Glue.

Quando utilizzi sessioni interattive dal notebook Jupyter, puoi eseguire il comando magico per abilitare il dimensionamento automatico insieme all'interfaccia utente Spark:

```
%%configure{
    "--enable-auto-scaling": "true",
    "--enable-continuous-cloudwatch-log": "true"
}
```



## Monitoraggio dell'utilizzo della DPU di esecuzione del processo Auto Scaling

È possibile utilizzare la Vista esecuzione dei processi AWS Glue Studio per controllare l'utilizzo della DPU da parte dei processi di dimensionamento automatico.

1. Scegli Monitoraggio dal pannello di AWS Glue Studio navigazione. Viene visualizzata la pagina Monitoraggio.
2. Scorri in basso fino all'elenco Job runs (Esecuzioni processo).
3. Passa all'esecuzione del processo che ti interessa e scorri fino alla colonna ore DPU per verificare l'utilizzo per l'esecuzione specifica del processo.

### Limitazioni

AWS GlueLo streaming Auto Scaling attualmente non supporta un DataFrame join in streaming con un elemento statico DataFrame creato all'esterno di. `ForEachBatch` Una statica DataFrame creata all'interno di `ForEachBatch` funzionerà come previsto.

## Partizionamento del carico di lavoro con esecuzione delimitata

Gli errori nelle applicazioni Spark derivano in genere da script Spark inefficienti, esecuzione in memoria di trasformazioni su larga scala e anomalie del set di dati. Esistono molte ragioni che possono causare problemi di memoria del driver o dell'executor, ad esempio una differenza di dati, un elenco di troppi oggetti o una riproduzione casuale di dati di grandi dimensioni. Questi spesso si verificano nell'elaborazione di enormi quantità di dati backlog con Spark.

AWS Glue consente di risolvere i problemi di eccezione di memoria e semplificare l'elaborazione ETL con il partizionamento del carico di lavoro. Quando il partizionamento del carico di lavoro è abilitato, ogni esecuzione del processo ETL seleziona solo i dati non elaborati, con il limite superiore impostato sulla dimensione del set di dati o sul numero di file da elaborare nell'esecuzione del processo. Le future esecuzioni dei processi elaboreranno i dati rimanenti. Ad esempio, se sono presenti 1000 file da elaborare, è possibile impostare il numero di file su 500 e dividerli in due esecuzioni del processo.

Il partizionamento del carico di lavoro è supportato solo per le origini dati di Amazon S3.

# Abilitazione del partizionamento del carico di lavoro

È possibile abilitare l'esecuzione delimitata impostando manualmente le opzioni nello script o aggiungendo le proprietà della tabella catalogo.

Per abilitare il partizionamento del carico di lavoro con esecuzione delimitata nello script:

1. Per evitare di rielaborare i dati, abilitare i segnalibri di processo nel nuovo processo o nel processo esistente. Per ulteriori informazioni, consulta Monitoraggio dei dati elaborati mediante segnalibri di processo.
2. Modifica lo script e imposta il limite nelle opzioni aggiuntive nella finestra API `getSource` AWS Glue. Devi inoltre impostare il contesto di trasformazione per il segnalibro di processo affinché memorizzi l'elemento `state`. Ad esempio:

   Python

   ```
   glueContext.create_dynamic_frame.from_catalog(
       database = "database",
       table_name = "table_name",
       redshift_tmp_dir = "",
       transformation_ctx = "datasource0",
       additional_options = {
           "boundedFiles" : "500", # need to be string
         # "boundedSize" : "1000000000" unit is byte
       }
   )
   ```

   Scala

   ```
   val datasource0 = glueContext.getCatalogSource(
       database = "database", tableName = "table_name", redshiftTmpDir = "",
       transformationContext = "datasource0",
       additionalOptions = JsonOptions(
           Map("boundedFiles" -> "500") // need to be string
         //"boundedSize" -> "1000000000" unit is byte
       )
   ).getDynamicFrame()
   ```

   ```
   val connectionOptions = JsonOptions(
       Map("paths" -> List(baseLocation), "boundedFiles" -> "30")
   ```

```
)
val source = glueContext.getSource("s3", connectionOptions, "datasource0", "")
```

Per abilitare il partizionamento del carico di lavoro con esecuzione delimitata nella tabella del catalogo dati:

1. Imposta le coppie di chiave-valore nel campo `parameters` della struttura della tabella nel catalogo dati. Per ulteriori informazioni, consulta Visualizzazione e modifica dei dettagli tabella.
2. Imposta il limite superiore per la dimensione del set di dati o il numero di file elaborati:
   - Impostaboundedsize alla dimensione target del set di dati in byte. L'esecuzione del processo si interromperà dopo aver raggiunto la dimensione target dalla tabella.
   - Imposta `boundedFiles` al numero target dei file. L'esecuzione del processo si interromperà dopo aver elaborato il numero target dei file.

> **Note**
>
> Devi impostare solo uno tra `boundedSize` e `boundedFiles`, in quanto è supportato un solo limite.

## Configurazione di un trigger AWS Glue per l'esecuzione automatica del processo

Una volta abilitata l'esecuzione delimitata, è possibile impostare un trigger AWS Glue per eseguire automaticamente il processo e caricare in modo incrementale i dati in esecuzioni sequenziali. Accedi alla console AWS Glue e crea un trigger, imposta l'orario di pianificazione e allegalo al tuo lavoro. Attiverà automaticamente la successiva esecuzione del processo ed elaborerà il nuovo batch di dati.

È possibile utilizzare i flussi di lavoro AWS Glue anche per orchestrare più processi allo scopo di elaborare dati da partizioni diverse in parallelo. Per ulteriori informazioni, consulta Trigger AWS Glue e Flussi di lavoro AWS Glue.

Per ulteriori informazioni sui casi d'uso e sulle opzioni, consulta il blog Optimizing Spark applications with workload partitioning in AWS Glue.

# Problemi noti per AWS Glue

Prendi nota dei seguenti problemi noti per AWS Glue.

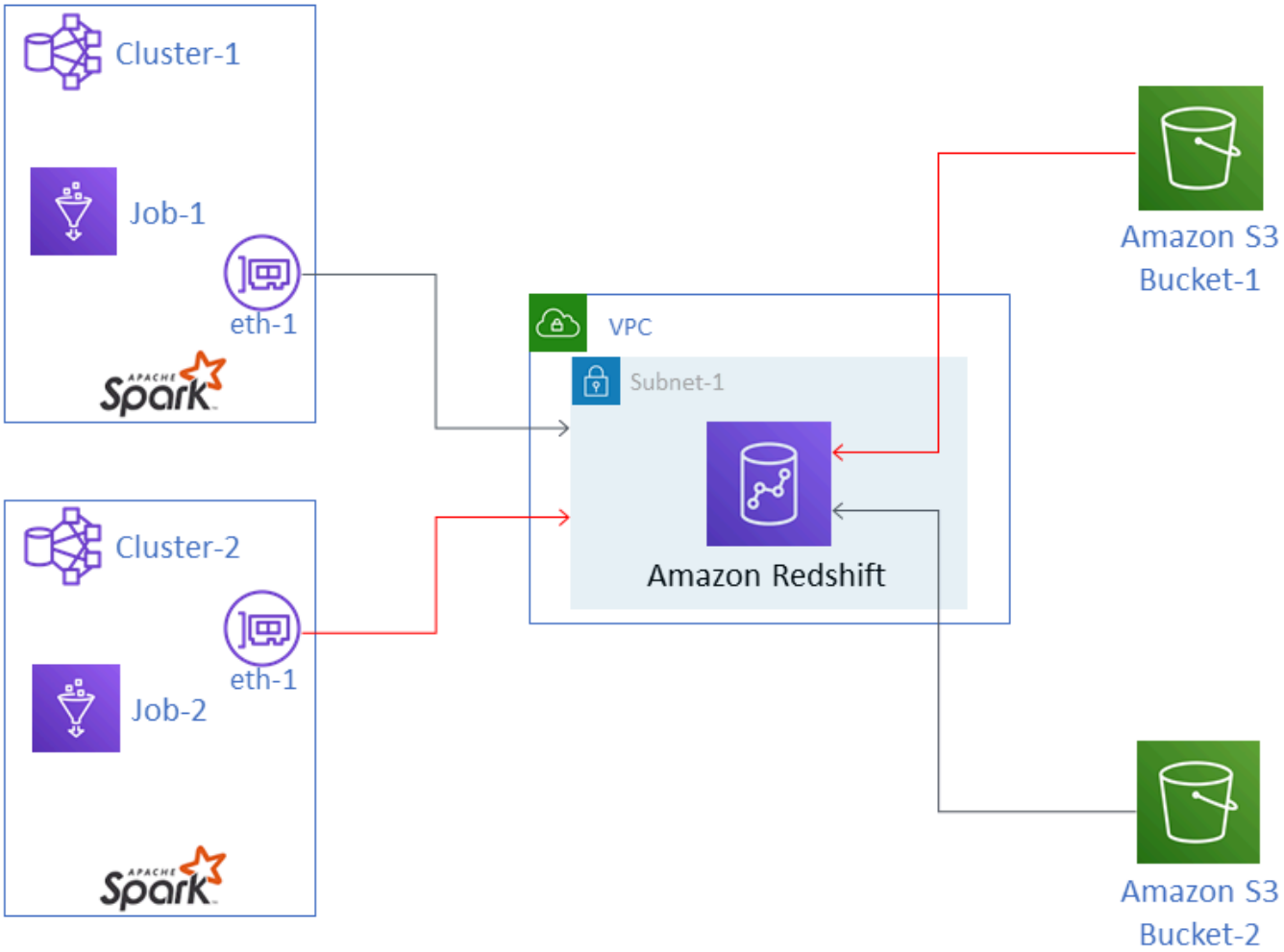
Argomenti


- Prevenzione dell'accesso ai dati tra processi


## Prevenzione dell'accesso ai dati tra processi

Considera la situazione in cui hai due job AWS Glue Spark in un unico AWS account, ciascuno in esecuzione in un cluster AWS Glue Spark separato. I processi utilizzano le connessioni AWS Glue per accedere alle risorse nello stesso cloud privato virtuale (VPC, Virtual Private Cloud). In questo caso, un processo in esecuzione in un cluster potrebbe essere in grado di accedere ai dati dal processo in esecuzione nell'altro cluster.

Il seguente diagramma illustra un esempio di questa situazione.




Nel diagramma, AWS Glue `Job-1` è in esecuzione in `Cluster-1` e Job-2 è in esecuzione in `Cluster-2`. Entrambi i processi funzionano con la stessa istanza di Amazon Redshift, che si trova in `Subnet-1` di un VPC. `Subnet-1` potrebbe essere una sottorete pubblica o privata.

`Job-1`sta trasformando i dati da Amazon Simple Storage Service (Amazon S3) `Bucket-1` e scrivendo i dati su Amazon Redshift. `Job-2`sta facendo lo stesso con i dati in. `Bucket-2` `Job-1`utilizza il ruolo AWS Identity and Access Management (IAM) `Role-1` (non mostrato), che dà accesso a`Bucket-1`. `Job-2`usa `Role-2` (non mostrato), che dà accesso a`Bucket-2`.

Questi processi dispongono di percorsi di rete che consentono la comunicazione con i cluster reciproci e quindi di accedere ai dati reciproci. Ad esempio, `Job-2` può accedere ai dati in `Bucket-1`. Nel diagramma, questo viene mostrato come il percorso in rosso.

Per evitare questa situazione, ti consigliamo di collegare diverse configurazioni di sicurezza a `Job-1` e `Job-2`. Collegando le configurazioni di sicurezza, l'accesso ai dati tra processi viene bloccato in virtù dei certificati creati da AWS Glue. Le configurazioni di sicurezza possono essere configurazioni fittizie . In altre parole, puoi creare le configurazioni di sicurezza senza abilitare la crittografia dei dati di Amazon S3, dei dati CloudWatch Amazon o dei segnalibri di lavoro. Tutte e tre le opzioni di crittografia possono essere disabilitate.

Per ulteriori informazioni sulle configurazioni di sicurezza, consulta the section called "Crittografia dei dati scritti da AWS Glue".

Per collegare una configurazione di sicurezza a un processo

1. Apri la console all' AWS Glue indirizzo. https://console.aws.amazon.com/glue/
2. Nella pagina Configure the job properties (Configura le proprietà del processo) per il processo, espandere la sezione Security configuration, script libraries, and job parameters (Configurazione di sicurezza, librerie di script e parametri di processi).
3. Selezionare una configurazione di sicurezza nell'elenco.

# Cronologia della documentazione per AWS Glue

| Modifica | Descrizione | Data |
| --- | --- | --- |
| AWS Glue for Ray non è più aperto a nuovi clienti. | AWS Glue for Ray non è più aperto a nuovi clienti. I clienti esistenti possono continuare a utilizzare il servizio normalmente. Per ulteriori informazioni, consulta AWS Glue Ray End of Support. | 30 aprile 2026 |
| AWS Glue for Ray non sarà più aperto a nuovi clienti a partire dal 30 aprile 2026. | AWS Glue for Ray non sarà più aperto a nuovi clienti a partire dal 30 aprile 2026. Se desideri utilizzarlo AWS Glue per Ray, registrati prima di tale data. I clienti esistenti possono continuare a utilizzare il servizio normalmente. Per ulteriori informazioni, consulta AWS Glue Ray End of Support. | 31 marzo 2026 |
| Support per la AWS Glue versione 5.1 | Sono state aggiunte informazioni sul supporto per la AWS Glue versione 5.1. Le funzionalità includono l'aggiornamento di Apache Spark alla 3.5.6, gli aggiornamenti del formato open table, le viste materializzate Iceberg, la versione 3 del formato Iceberg, letture e scritture con accesso completo alla tabella Hudi, il controllo DDL/ | 26 novembre 2025 |

| | | |
|---|---|---|
| | DML granulare degli accessi nativo di Spark sulle tabelle Hive, Iceberg e Delta Lake e altro ancora. AWS Glue Per ulteriori informazioni AWS Glue , AWS Glue consulta le note di rilascio e la migrazione dei lavori alla versione 5.1. | |
| AWS Glue Data Quality ora supporta l'etichettatura delle regole per una reportistica avanzata | AWS Glue Data Quality ora supporta l'etichettatura delle regole, che consente di organizzare e analizzar e i risultati sulla qualità dei dati in modo più efficace. Puoi interrogare i risultati in base a etichette specifiche per identificare le regole non valide all'interno di particola ri categorie, contare i risultati delle regole per team o dominio e creare report mirati per le diverse parti interessa te. Per ulteriori informazioni, consulta Etichette. | 21 novembre 2025 |
| AWS Glue Data Quality ora supporta le interrogazioni di pre-elaborazione | AWS Glue Data Quality ora supporta le costanti in DQDL, consentendoti di definire valori costanti e di farvi riferimen to in tutto lo script. Questo aiuta a prevenire problemi relativi ai limiti delle dimension i delle query quando si lavora con istruzioni SQL di grandi dimensioni. Per ulteriori informazioni, vedere Costanti. | 21 novembre 2025 |

| | | |
|---|---|---|
| Support per i nuovi tipi di worker G.12X, G.16X e i tipi di worker R ottimizzati per la memoria | È stato aggiunto il supporto per i nuovi tipi di lavorator i G.12X, G.16X e i tipi di lavoratori R.1X, R.2X, R.4X, R.8X ottimizzati per la memoria per i lavori. AWS Glue Questi nuovi tipi di worker forniscono ulteriori opzioni di elaborazione e memoria per diversi requisiti di carico di lavoro. Per ulteriori AWS Glue informazioni, vedere Aggiungere lavori in. | 30 gennaio 2025 |
| Support per altri 14 nuovi connettori nativi SaaS per AWS Glue | Sono stati aggiunti altri quattordici connettori nativi SaaS. AWS Glue Per ulteriori informazioni, vedere Aggiungere una AWS Glue connessione. | 30 gennaio 2025 |
| Support per altri 16 nuovi connettori nativi SaaS per AWS Glue | Sono stati aggiunti altri sedici connettori nativi SaaS. AWS Glue Per ulteriori informazioni, vedere Aggiungere una AWS Glue connessione. | 17 dicembre 2024 |
| Generazione automatica delle statistiche delle colonne | AWS Glue Data Quality ora supporta Amazon SageMaker AI LakeHouse tabelle e tabelle Iceberg, Delta e HUDI AWS Lake Formation gestite in Data Catalog ed ETL. Per ulteriori informazioni, consulta Qualità dei dati di AWS Glue. | 6 dicembre 2024 |

| | | |
|---|---|---|
| Supporto per le integrazioni Zero-ETL | Zero-ETL è un insieme di integrazioni completamente gestite AWS che riduce al minimo la necessità di creare pipeline di dati ETL. Per ulteriori informazioni, consulta Integrazioni Zero-ETL. | 3 dicembre 2024 |
| Supporto per le connessioni riutilizzabili | Un nuovo schema di AWS Glue connessioni offre un modo unificato per gestire le connessioni dati tra AWS servizi e applicazioni AWS Glue, come Amazon Athena e SageMaker Amazon Unified Studio. Per ulteriori informazioni consulta Connessione ai dati. | 3 dicembre 2024 |

| | | |
|---|---|---|
| Support per AWS Glue la versione 5.0. | Sono state aggiunte informazioni sul supporto per la AWS Glue versione 5.0. Le funzionalità includono l'aggiornamento di Apache Spark alla versione 3.52, l'aggiornamento di Java alla versione 17, aggiornamenti del formato a tabella aperta, il controllo degli accessi nativo Spark a livello granulare, l'integrazione di Sagemaker Lakehouse e l'astrazione del data warehouse, il supporto in Sagemaker Unified Studio e altro ancora. Per ulteriori informazioni, consulta le note di AWS Glue rilascio e la migrazione dei AWS Glue lavori alla AWS Glue versione 5.0. | 3 dicembre 2024 |
| Connessione all' AWS Glue Data Catalog utilizzo dell'endpoint AWS Glue REST Iceberg | L'endpoint AWS Glue Iceberg Rest supporta le operazioni API specificate nella specifica di Apache Iceberg REST. Utilizzando un client Iceberg REST, è possibile collegare l'applicazione in esecuzione e su un motore di analisi al catalogo REST ospitato nel Catalogo dati. Per ulteriori informazioni, consulta Accesso al Catalogo dati. | 3 dicembre 2024 |

| | | |
|---|---|---|
| Generazione automatica delle statistiche delle colonne | Generare automaticamente statistiche sulle colonne per nuove tabelle nel AWS Glue Data Catalog. Per ulteriori informazioni, consulta Generare automaticamente statistiche sulle colonne. | 3 dicembre 2024 |
| Support per gli aggiornamenti generativi dell'intelligenza artificiale per Apache Spark in AWS Glue | Spark Upgrades in AWS Glue consente ai data engineer e agli sviluppatori di aggiornare e migrare, aggiornare e migrare i job Spark esistenti alle ultime versioni di AWS Glue Spark utilizzando l'intelligenza artificiale generativa. Per ulteriori informazioni, consulta Analisi degli aggiornamenti con l'IA. | 22 novembre 2024 |
| Support per la risoluzione dei problemi di intelligenza artificiale generativa per Apache Spark in AWS Glue | Generative AI Troubleshooting for Apache Spark jobs in AWS Glue aiuta i data engineer e gli scienziati a diagnosticare e risolvere i problemi nelle loro applicazioni Spark con facilità. Per ulteriori informazioni, consulta Risoluzione dei problemi dei processi Spark con l'IA. | 22 novembre 2024 |

| | | |
|---|---|---|
| Supporto per gli ottimizzatori Iceberg per accedere ai bucket Amazon S3 in un VPC | AWS Glue Data Catalog supporta gli ottimizzatori di tabelle Iceberg per accedere ai bucket Amazon S3 da uno specifico Virtual Private Cloud (VPC) utilizzando una connessione di rete. AWS Glue Per ulteriori informazioni, consulta Ottimizzazione delle tabelle Iceberg. | 20 novembre 2024 |
| Support per altri nove nuovi connettori nativi SaaS per AWS Glue | Sono stati aggiunti altri nove connettori nativi SaaS. AWS Glue Per ulteriori informazioni, vedere Aggiungere una AWS Glue connessione. | 19 novembre 2024 |
| Support per dieci nuovi connettori nativi SaaS per AWS Glue | Sono stati aggiunti dieci connettori nativi SaaS. AWS Glue Per ulteriori informazioni, vedere Aggiungere una AWS Glue connessione. | 15 novembre 2024 |
| Support per l'esecuzione dei job, l'accodamento per AWS Glue i job ETL | Puoi attivare la coda per l'esecuzione dei processi per eseguire i processi in un secondo momento, quando non possono essere eseguiti immediatamente a causa delle Service Quotas. Per maggiori informazioni, consulta Configurazione delle proprietà dei job per i job Spark in. AWS Glue | 3 settembre 2024 |

| | | |
|---|---|---|
| Modifiche delle policy aggiornate | Modifiche documentate alle AwsGlueSessionUser RestrictedNotebookServiceRolepolitiche AwsGlueSessionUserRestrictedNotebookPolicyand, necessarie per il supporto delle sessioni con tag-on-create la chiave del tag owner. Per ulteriori informazioni, consulta AWS Glue gli aggiornamenti alle politiche AWS gestite. | 30 agosto 2024 |
| Il Rilevamento delle anomalie e le Regole dinamiche sono ora disponibili a livello generale | AWS Glue Data Quality utilizza un algoritmo di apprendimento automatico per apprendere dalle tendenze passate e quindi prevedere i valori futuri per rilevare le anomalie. Le regole dinamiche consentono di fornire soglie dinamiche. Per ulteriori informazioni, consulta Ottimizzazione delle prestazioni delle query per le tabelle Iceberg. | 7 agosto 2024 |

| | | |
|---|---|---|
| Modifiche delle policy aggiornate | Modifiche documentate alle AwsGlueSessionUser RestrictedServiceRolepolitiche AwsGlueSessionUserRestricte dPolicye, necessarie per il supporto delle sessioni con tag-on-create la chiave del tag proprietario. Per ulteriori informazioni, consulta AWS Glue gli aggiornamenti alle politiche AWS gestite. | 5 agosto 2024 |
| La generazione delle statistic he delle colonne per le tabelle Iceberg è ora generalmente disponibile | AWS Glue supporta il calcolo e l'aggiornamento del numero di valori distinti (NDVs) per ogni colonna nelle tabelle Iceberg. Per ulteriori informazi oni, consulta Rilevamento delle anomalie nelle regole dinamiche e di qualità AWS Glue dei dati. | 9 luglio 2024 |
| Support per i profili AWS Glue di utilizzo | Gli amministratori possono creare profili di AWS Glue utilizzo per varie classi di utenti all'interno dell'acco unt, come sviluppatori, tester e team di prodotto. Questa flessibilità consente agli amministratori di applicare diversi controlli di utilizzo e di costo per ciascuna classe di utenti. Per ulteriori informazi oni, consulta AWS Glue Configurazione dei profili di utilizzo. | 18 giugno 2024 |

| | | |
|---|---|---|
| Support per un connettore Salesforce per AWS Glue Spark | Sono state aggiunte informazioni su un nuovo AWS Glue connettore per Salesforce. Questa funzionalità consente di utilizzare Spark AWS Glue per leggere e scrivere su Salesforce nella AWS Glue versione 4.0 e successive. Per ulteriori informazioni consulta Connessione a Salesforce. | 22 maggio 2024 |

| | | |
|---|---|---|
| Integrazione dei dati di Amazon Q in AWS Glue (GA) | L'integrazione dei dati di Amazon Q AWS Glue è una nuova funzionalità di intelligenza artificiale generativa AWS Glue che consente ai data engineer e agli sviluppatori ETL di creare lavori di integrazione dei dati utilizzando il linguaggio naturale. Gli ingegneri e gli sviluppatori possono chiedere a Q di creare lavori, risolvere problemi e rispondere a domande sull' AWS Glue integrazione dei dati. Per ulteriori informazioni, consulta Integrazione dei dati di Amazon Q in AWS Glue. Questa funzionalità include un aggiornamento e una `AwsGlueSessionUserRestrictedServiceRole` AWS gestione delle `AwsGlueSessionUserRestrictedPolicy` politiche. `AwsGlueSessionUserRestrictedNotebookServiceRole` Per ulteriori informazioni, consulta AWS Glue Aggiornamenti alle politiche AWS gestite. | 30 aprile 2024 |

| | | |
|---|---|---|
| Integrazione dei dati di Amazon Q in AWS Glue (anteprima) | L'integrazione dei dati di Amazon Q AWS Glue è una nuova funzionalità di intelligenza artificiale generativa AWS Glue che consente ai data engineer e agli sviluppatori ETL di creare lavori di integrazione dei dati utilizzando il linguaggio naturale. Gli ingegneri e gli sviluppatori possono chiedere a Q di creare lavori, risolvere problemi e rispondere a domande sull' AWS Glue integrazione dei dati. Per ulteriori informazioni, consulta Integrazione dei dati di Amazon Q in AWS Glue. Questa funzionalità include un aggiornamento della politica `AwsGlueSessionUserRestrictedNotebookPolicy` AWS gestita. Per ulteriori informazioni, consulta AWS Glue gli aggiornamenti delle politiche AWS gestite. | 30 gennaio 2024 |

| | | |
|---|---|---|
| Aggiornamento della documentazione per AWS Glue lo streaming | Aggiunto un nuovo capitolo con contenuti nuovi e riorganizzati per AWS Glue lo streaming. Questo contenuto descrive come funziona lo streaming AWS Glue, le caratteristiche dell'elab orazione dei dati in tempo reale e come monitorar e i processi di streaming. Per ulteriori informazioni, consulta la pagina AWS Glue Streaming. | 27 dicembre 2023 |
| Supporto per l'utilizzo del rilevamento dei dati sensibili granulari | La trasformazione relativa al rilevamento dei dati sensibili fornisce la possibili tà di rilevare, mascherare o rimuovere le entità che hai definito o che sono predefinite da AWS Glue. Le azioni granulari consentono inoltre di applicare un'azione specifica per entità. Per ulteriori informazioni, consulta Utilizzo del rilevamento dei dati sensibili granulari. | 26 novembre 2023 |

| | | |
|---|---|---|
| Support per il monitoraggio dei lavori con le metriche AWS Glue di Observability | Usa le metriche AWS Glue di Observability per generare approfondimenti su ciò che accade all'interno dei tuoi lavori di Apache Spark AWS Glue per migliorare la classific azione e l'analisi dei problemi. Per ulteriori informazioni, consulta Monitoraggio con metriche di osservabilità. AWS Glue | 26 novembre 2023 |
| Support per il rilevamento delle anomalie in AWS Glue Data Quality | AWS Glue Il rilevamento delle anomalie relative alla qualità dei dati applica nel tempo algoritmi di machine learning (ML) alle statistiche dei dati per rilevare modelli anomali e problemi nascosti di qualità dei dati che sono difficili da rilevare attraverso le regole. Per ulteriori informazioni, consulta Rilevamento delle anomalie nella qualità dei dati. AWS Glue | 26 novembre 2023 |

| | | |
|---|---|---|
| Aggiornamento al comportamento di registrazione predefinito dell'interfaccia utente di Spark | I job Spark che generano i log dell'interfaccia utente Spark ora verranno scritti con un modello di nome file diverso per supportare l'interfaccia utente Spark nella console. AWS Glue Ciò non modifica il comportamento dei log. CloudWatch È possibile ripristinare il comportamento legacy aggiornando la configurazione del processo. Per ulteriori informazioni, consulta Monitoraggio dei processi tramite l'interfaccia utente Web di Apache Spark. | 17 novembre 2023 |

| | | |
|---|---|---|
| Support per nuove fonti di dati in AWS Glue for Spark | Le connessioni ad Amazon OpenSearch Service, Azure SQL, Azure Cosmos for NoSQL, SAP HANA Teradata Vantage e Vertica sono ora supportate nativamente all'interno. AWS Glue Inoltre, le connessioni a queste fonti di dati, oltre a MongoDB, sono ora disponibili per l'uso nell' AWS Glue editor visivo di Studio. Per ulteriori informazi oni, consulta Tipi di connessio ne e opzioni per ETL in AWS Glue for Spark AWS Glue per informazioni sul supporto di Spark e Aggiungere una AWS Glue connessione per informazioni sull'uso nell'editor visivo di Studio. AWS Glue | 17 novembre 2023 |
| Supporto per generare le statistiche delle colonne | Puoi calcolare statistiche a livello di colonna per AWS Glue Data Catalog tabelle in formati di dati come Parquet, ORC, JSON, ION, CSV e XML senza configurare pipeline di dati aggiuntive. Per ulteriori informazioni, consulta Utilizzo delle statistiche delle colonne. | 16 novembre 2023 |

| | | |
|---|---|---|
| Supporto per la compattazione dei dati per le tabelle Iceberg | Per migliorare le prestazioni di lettura da parte di servizi di AWS analisi come Amazon Athena e Amazon EMR e i processi AWS Glue ETL, Data Catalog offre la compattazione gestita (un processo che compatta piccoli oggetti Amazon S3 in oggetti più grandi) per le tabelle Iceberg in Data Catalog. Per ulteriori informazioni, consulta Ottimizzazione delle tabelle Iceberg. | 13 novembre 2023 |
| Aggiornamento al comportamento di attesa dell'esecuzione del processo | Le esecuzioni del processo standard di shell (interprete di comandi) Spark e Python ora passeranno a `WAITING` in determinate situazioni, anziché passare immediatamente a `FAILED`. Per ulteriori informazioni, consulta Stati di esecuzione dei processi AWS Glue. | 8 novembre 2023 |
| AWS Glue StudioAWS Glue guida per l'utente consolidata nella guida per sviluppatori | La guida per AWS Glue Studio l'utente è stata spostata nella guida per sviluppatori per creare un'unica guida utente unificata per AWS Glue Studio la AWS Glue console e l'accesso AWS Glue Studio programmatico. | 25 ottobre 2023 |

| | | |
|---|---|---|
| Aggiornamento della policy gestita AWSGlue ServiceNotebookRole AWS | Sono state aggiunte informazioni su un aggiornamento minore alla politica AWSGlue ServiceNotebookRole AWS gestita. Per ulteriori informazioni, vedere AWS Glue Aggiornamenti alle politiche AWS gestite. | 9 ottobre 2023 |
| AWS Glue Studio supporta cinque nuove trasformazioni integrate | AWS Glue Studio supporta le seguenti cinque nuove trasformazioni integrate: Record matching, Remove null rows, Parse JSON column, Extract JSON path e Regex extractor. Per ulteriori informazioni, consulta Modifica dei nodi di trasformazione dei dati gestiti. AWS Glue | 11 agosto 2023 |
| Aggiornamento della politica AWSGlue ServiceRole AWS gestita | Sono state aggiunte informazioni su un aggiornamento minore alla politica AWSGlue ServiceRole AWS gestita. Per ulteriori informazioni, vedere AWS Glue Aggiornamenti alle politiche AWS gestite. | 4 agosto 2023 |

| | | |
|---|---|---|
| Supporto per il crawling delle tabelle Apache Hudi | Sono state aggiunte informazioni sull'utilizzo AWS Glue per eseguire la scansione delle tabelle Hudi nei bucket Amazon S3 e sulla registrazione delle tabelle Hudi in. AWS Glue Data Catalog Per ulteriori informazioni, consulta le pagine Which data stores can I crawl? e Crawler properties. | 21 luglio 2023 |
| Aggiornamento della politica gestita AWSGlue ConsoleFullAccess AWS | Sono state aggiunte informazioni su un aggiornamento minore alla politica AWSGlue ConsoleFullAccess AWS gestita. Per ulteriori informazioni, vedere AWS Glue Aggiornamenti alle politiche AWS gestite. | 14 luglio 2023 |
| Supporto per il crawling delle tabelle Apache Iceberg | Sono state aggiunte informazioni sull'utilizzo AWS Glue per eseguire la scansione delle tabelle Iceberg nei bucket Amazon S3 e sulla registrazione delle tabelle Iceberg in. AWS Glue Data Catalog Per ulteriori informazioni, consulta le pagine Which data stores can I crawl? e Crawler properties. | 7 luglio 2023 |

| | | |
|---|---|---|
| Support per AWS Glue with Ray | Sono state aggiunte informazioni su AWS Glue with Ray, un nuovo motore in grado di supportare i AWS Glue lavori. Riorganizzato il contenuto esistente AWS Glue con Spark per chiarire le ambiguità. | 30 maggio 2023 |
| Support per la qualità AWS Glue dei dati (GA) | AWS Glue La qualità dei dati è ora disponibile a livello generale. AWS Glue Data Quality ti aiuta a valutare e monitorare la qualità dei tuoi dati. Per informazioni su come utilizzare AWS Glue Data Quality con Data Catalog, consulta AWS Glue Data Quality. Per ulteriori informazioni sulla qualità AWS Glue dei dati per AWS Glue Studio, consulta Valutazione della qualità dei dati con AWS Glue Studio. | 24 maggio 2023 |
| Supporto per tipi di worker di grandi dimensioni per i processi Apache Spark | È ora disponibile il supporto per l'uso dei tipi di worker `G.4X` e `G.8X` per i processi Apache Spark. Questi tipi di worker sono adatti per i processi i cui carichi di lavoro contengono trasformazioni, aggregazioni, join e query con i maggiori requisiti. Per ulteriori informazioni, consulta Aggiungere lavori in AWS Glue. | 08 maggio 2023 |

| | | |
|---|---|---|
| Supporto per la creazione di indici di partizione durante il crawling delle tabelle | Sono state aggiunte informazioni sul modo in cui i crawler supportano la creazione di indici di partizione per le tabelle rilevate dal crawler. Per ulteriori informazioni, consulta la pagina Setting the partition index crawler configuration option. | 24 aprile 2023 |
| Supporto per i parametri di utilizzo delle risorse | Sono state aggiunte informazioni sulla visualizzazione dell'utilizzo delle risorse del servizio e sulla configurazione degli allarmi in Amazon. CloudWatch Per ulteriori informazioni, consulta la pagina AWS Glue resource monitoring. | 7 aprile 2023 |
| Aggiornamento della politica gestita AWSGlue ConsoleFullAccess AWS | Sono state aggiunte informazioni su un aggiornamento minore alla politica AWSGlue ConsoleFullAccess AWS gestita. Per ulteriori informazioni, vedere AWS Glue Aggiornamenti alle politiche AWS gestite. | 28 marzo 2023 |

| | | |
|---|---|---|
| Sono state aggiunte linee guida per l'utilizzo AWS Glue con un AWS SDK con esempi | La Guida per gli AWS Glue sviluppatori contiene due nuove sezioni che forniscon o informazioni per aiutarti a utilizzare AWS Glue un AWS SDK. Per ulteriori informazioni, consulta Utilizzo AWS Glue con un AWS SDK e Esempi di codice per l' AWS Glueutilizzo. AWS SDKs | 23 febbraio 2023 |
| Aggiornamento della documentazione per IAM con AWS Glue | Informazioni riorganizzate e aggiunte sull'utilizzo di IAM con AWS Glue. Per ulteriori informazioni, consulta Identity and Access Management per AWS Glue. | 15 febbraio 2023 |
| Support per l'esecuzione di job ETL in streaming nella AWS Glue versione 4.0 | Sono state aggiunte informazi oni sul supporto per l'esecuzi one di processi ETL di streaming in Glue versione 4.0 e nuove opzioni per la connessione a un cluster Kafka o a un cluster Amazon Managed Streaming per Apache Kafka e flussi di dati Amazon Kinesis. Per ulteriori informazioni, consulta Aggiunta di processi ETL di streaming in AWS Glue e Tipi di connessione e opzioni per ETL in AWS Glue. | 8 febbraio 2023 |

| | | |
|---|---|---|
| Supporto per il crawling delle origini dati MongoDB Atlas | Sono state aggiunte informazioni sull'utilizzo AWS Glue per la scansione delle fonti di dati MongoDB Atlas. Per ulteriori informazioni, consulta le seguenti sezioni: Per quali archivi di dati posso eseguire il crawling?, Proprietà di connessione MongoDB e MongoDB Atlas e Utilizzo di una connessione MongoDB o MongoDB Atlas. | 6 febbraio 2023 |
| Supporto per il crawling delle tabelle Delta Lake con un connettore Delta Lake nativo | Sono state aggiunte informazioni sull'utilizzo AWS Glue per la scansione delle tabelle Delta Lake utilizzando un connettore Delta Lake nativo. Questa funzionalità ti consente di utilizzare i motori di AWS query per interrogare direttamente il registro delle transazioni Delta e utilizzare funzionalità come i viaggi nel tempo e le garanzie ACID, e di sincronizzare i metadati Delta Lake dai file di transazione di Amazon S3 nel Data Catalog per abilitare le autorizzazioni alle colonne sulle tue query in Lake Formation. Per ulteriori informazioni, consulta Come specificare le opzioni di configurazione per un archivio di dati Delta Lake e Interrogazione delle tabelle Delta Lake. | 15 dicembre 2022 |

| | | |
|---|---|---|
| Support for AWS Glue Data Quality (anteprima) | Il supporto è ora disponibile per AWS Glue Data Quality (anteprima). AWS Glue Data Quality consente di valutare e monitorare la qualità dei dati quando si utilizza la AWS Glue versione 3.0. Per informazioni su come utilizzare AWS Glue Data Quality con Data Catalog, vedi AWS Glue Data Quality (anteprima). Per ulteriori informazioni sulla qualità AWS Glue dei dati per AWS Glue Studio, consulta Valutazione della qualità dei dati con AWS Glue Studio. | 30 novembre 2022 |
| Supporto per un nuovo connettore Amazon Redshift Spark con nuove funzionalità e miglioramenti delle prestazioni | È ora disponibile il supporto per un nuovo connettore Amazon Redshift Spark con un nuovo driver JDBC da utilizzare con i processi AWS Glue ETL per creare applicazioni Apache Spark in grado di leggere e scrivere dati in Amazon Redshift come parte delle pipeline di acquisizione e trasformazione dei dati. Per ulteriori informazioni, consulta Spostamento di dati da e verso Amazon Redshift. | 29 novembre 2022 |

| | | |
|---|---|---|
| Support per AWS Glue la versione 4.0. | Sono state aggiunte informazioni sul supporto per la AWS Glue versione 4.0. Le funzionalità includono il supporto nativo per i framework data lake aperti con Apache Hudi, Delta Lake e Apache Iceberg e il supporto nativo per il plug-in di archiviazione cloud shuffle basato su Amazon S3 (un plug-in Apache Spark) per utilizzare Amazon S3 per la capacità di archiviazione shuffle ed elastica. Per ulteriori informazioni, vedere Note di AWS Glue rilascio e Migrazione dei AWS Glue lavori alla AWS Glue versione 4.0. | 28 novembre 2022 |
| AWS Glue Studio ora offre trasformazioni visive personalizzate | Le trasformazioni visive personalizzate consentono ai clienti di definire, riutilizzare e condividere la logica ETL specifica dell'azienda tra i propri team. Per ulteriori informazioni, consulta Trasformazioni visive personalizzate. | 28 novembre 2022 |

| | | |
|---|---|---|
| Support per l'utilizzo del AWS Glue crawler per pubblicare metadati per archivi dati JDBC | È ora disponibile il supporto per l'utilizzo del AWS Glue crawler per pubblicare metadati come commenti e tipi rawtype nel Data Catalog for JDBC data store. Per ulteriori informazioni, consulta Parametri impostati nelle tabelle del catalogo dati per crawler, proprietà del crawler e struttura. JdbcTarget | 18 novembre 2022 |
| Supporto per il crawling di datastore Snowflake | È ora disponibile il supporto per AWS Glue eseguire la scansione delle tabelle e delle viste Snowflake e per pubblicare i metadati nel Data Catalog come voce di tabella. Per le tabelle esterne Snowflake in Amazon S3, il crawler esegue il crawling anche della posizione Amazon S3 e del tipo di formato di file della tabella esterna e lo compila come parametri della tabella. Per ulteriori informazioni, consulta Quali datastore posso sottoporre a crawling? , Proprietà della connessione a AWS Glue e Parametri impostati nelle tabelle del catalogo di dati dal crawler. | 18 novembre 2022 |

| | | |
|---|---|---|
| Supporto per una migliore gestione dello shuffle delle applicazioni Spark | È ora disponibile il supporto per un nuovo plug-in di archiviazione cloud shuffle per Apache Spark. Per ulteriori informazioni, consulta Plug-in shuffle di AWS Glue Spark con Amazon S3 e Plug-in di archiviazione cloud shuffle per Apache Spark. | 15 novembre 2022 |
| È stato aggiunto il supporto per le destinazioni catalogo dati quando si accelera il crawling per le notifiche di eventi Amazon S | Oltre al supporto esistente per le destinazioni Amazon S3, ora è disponibile il supporto per accelerare il crawling per le destinazioni catalogo dati tramite notifiche di eventi Amazon S3. Per ulteriori informazioni, consulta Accelerazione della ricerca per indicizzazione usando le notifiche eventi Amazon S3. | 13 ottobre 2022 |
| Supporto per specificare il numero massimo di tabelle che un crawler può creare | È ora disponibile il supporto per specificare il numero massimo di tabelle che il crawler può creare. Per ulteriori informazioni, consulta la pagina Come specificare il numero massimo di tabelle che il crawler può creare. | 6 settembre 2022 |

| | | |
|---|---|---|
| Supporto per Python 3.9 nei job della shell Python in AWS Glue | Il supporto è ora disponibile per l'esecuzione di script compatibili con Python 3.9 nei AWS Glue job della shell Python in e per la scelta di utilizzare set di librerie preconfezionate. Per ulteriori informazioni, consulta Processi della shell Python in AWS Glue. | 11 agosto 2022 |
| Support per l'esecuzione di AWS Glue lavori non urgenti o non urgenti utilizzando capacità inutilizzata | È ora disponibile il supporto per la configurazione di esecuzioni flessibili per processi non urgenti come processi di pre-produzione, test e caricamenti di dati una tantum. Per ulteriori informazioni, consulta Aggiungere lavori in. AWS Glue | 9 agosto 2022 |
| Il supporto per un nuovo tipo di worker per i processi di streaming | Il supporto di questo servizio di Support per l'uso del tipo di worker `G.025X` per processi di streaming a basso volume. Per ulteriori informazioni, consulta Aggiungere lavori in AWS Glue. | 14 luglio 2022 |
| Support per l'uso di Kafka SASL nelle connessioni AWS Glue | Il supporto è ora disponibile per l'uso di Kafka SASL nelle connessioni. AWS Glue Per ulteriori informazioni, consulta AWS Glue Proprietà di connessione Kafka per l'autenticazione client. | 5 luglio 2022 |

| | | |
|---|---|---|
| Supporto per il connettore Apache Kafka per gli schemi protobuf | Il supporto di Apache Kafka Connector è attualmente disponibile per gli schemi Protobuf. Per ulteriori informazioni, consulta Registro degli schemi di AWS Glue. | 09 giugno 2022 |
| Support per Auto Scaling for AWS Glue jobs (GA) | Sono state aggiunte informazioni sull'utilizzo di Auto Scaling for jobs nella AWS Glue versione 3.0 per scalare dinamicamente le risorse di elaborazione. Per ulteriori informazioni, consulta Utilizzo di Auto Scaling per AWS Glue. | 14 aprile 2022 |
| Aggiornamento della documentazione per lo AWS Glue sviluppo e il test degli script di lavoro AWS Glue | Informazioni riorganizzate e aggiunte sui metodi di sviluppo e test disponibili per AWS Glue, comprese le istruzioni per lo sviluppo con Docker. Per ulteriori informazioni, consulta Sviluppo e test di script di AWS Glue lavoro. | 14 marzo 2022 |
| Aggiunta di buffer di protocollo (protobuf) come formato di dati supportato per il registro degli schemi AWS Glue | Aggiunte informazioni su Protobuf come formato dati supportato (oltre ad AVRO e JSON). Per ulteriori informazioni, consulta Registro degli schemi di AWS Glue. | 25 febbraio 2022 |

| | | |
|---|---|---|
| Supporto per il crawling delle tabelle Delta Lake | Sono state aggiunte informazioni sull'utilizzo AWS Glue per la scansione delle tabelle Delta Lake. Per ulteriori informazioni, consulta Come specificare le opzioni di configurazione per un archivio dati Delta Lake. | 24 febbraio 2022 |
| Support per approfondimenti sul AWS Glue lavoro | Sono state aggiunte informazioni sull'utilizzo di AWS Glue Job Insights per semplificare il debug e l'ottimizzazione dei job. AWS Glue Per ulteriori informazioni, consulta Monitoraggio con AWS Glue Job Insights. | 8 febbraio 2022 |
| Supporto per il crawling di tabelle Catalogo dati supportate da Amazon S3 utilizzando un endpoint VPC | Oltre all'archivio dati di Amazon S3, si possono configurare le tabelle Catalogo dati supportate da Amazon S3 per consentire l'accesso solo a un ambiente Amazon Virtual Private Cloud (Amazon VPC) per motivi di sicurezza , audit o controllo. Per ulteriori informazioni, consulta Crawling di un datastore Amazon S3 o di tabelle Catalogo dati supportate da Amazon S3 utilizzando un endpoint VPC. | 3 febbraio 2022 |

| | | |
|---|---|---|
| Supporto per le tavole governate dalla Lake Formation | Sono state aggiunte informazioni sul AWS Glue supporto per le tabelle governate da Lake Formation, che supportano le transazioni ACID, la compattazione automatica dei dati e le query sui viaggi nel tempo. Per ulteriori informazioni, consulta API AWS Glue, e Guida per gli sviluppatori di AWS Lake Formation. | 30 novembre 2021 |
| Nuove politiche AWS gestite aggiunte per sessioni e notebook interattivi | Le nuove policy gestite per IAM hanno fornito una maggiore sicurezza per l'utilizzo AWS Glue con sessioni e notebook interattivi. Per ulteriori informazioni, consulta la sezione Policy gestite da AWS per AWS Glue. | 30 novembre 2021 |
| Il registro dello schema Glue ora supportato con i processi di streaming | È possibile creare processi di streaming che accedono alle tabelle che fanno parte di Glue Schema Registry. Per ulteriori informazioni, consulta AWS Glue Schema Registry e Aggiunta di processi di streaming ETL in AWS Glue. | 15 novembre 2021 |

| | | |
|---|---|---|
| Supporto per nuove caratteristiche di machine learning | Aggiunte informazioni sulle nuove funzionalità per la trasformazione di machine learning Ricerca corrispondenze, tra cui la corrispondenza incrementale e il punteggio di corrispondenza. Per ulteriori informazioni, consulta Ricerca di corrispondenze incrementali e Stima della qualità delle corrispondenze utilizzando i punteggi di confidenza delle corrispondenze. | 31 ottobre 2021 |
| (Anteprima privata) Support per AWS Glue lavori flessibili | Sono state aggiunte informazioni sulla configurazione dei job AWS Glue Spark con una classe di esecuzione flessibile, adatta per lavori che non richiedono tempo e i cui tempi di inizio e completamento possono variare. Per ulteriori informazioni, consulta Aggiungere lavori in. AWS Glue | 29 ottobre 2021 |
| Supporto per accelerare la ricerca per indicizzazione usando le notifiche eventi Amazon S3 | Sono state aggiunte informazioni sull'accelerazione della ricerca per indicizzazione utilizzando le notifiche degli eventi Amazon S3. Per ulteriori informazioni, consulta Accelerazione della ricerca per indicizzazione usando le notifiche eventi Amazon S3. | 15 ottobre 2021 |

| | | |
|---|---|---|
| Opzioni di configurazione di sicurezza aggiuntive relative al controllo degli accessi e VPCs | Sono state aggiunte informazioni su come configurare nuove autorizzazioni di controllo degli accessi AWS Glue e sulla configurazione di. VPCs Per ulteriori informazioni, consulta AWS Tags in AWS Glue, Identity-Based Policies (IAM Policies) che controllano le impostazioni utilizzando chiavi di condizione o chiavi contestuali e Configurazione di tutte le AWS chiamate in modo che passino attraverso il tuo VPC. | 13 ottobre 2021 |
| Supporto per le policy di endpoint VPC | Aggiunte informazioni sul supporto per policy endpoint Virtual Private Cloud (VPC) in AWS Glue. Per ulteriori informazioni consulta AWS Glue ed endpoint VPC di interfaccia (AWS PrivateLink). | 11 ottobre 2021 |
| Glue Studio è ora disponibile in Cina | AWS Glue Studio è ora disponibile nelle regioni Cina, Pechino e Ningxia. | 11 ottobre 2021 |

| | | |
|---|---|---|
| AWS Glue Studio offre la creazione di notebook, per la modifica interattiva dei lavori | I notebook consentono di scrivere ed eseguire codice, visualizzare i risultati e condividere informazioni. In genere, i data scientist utilizzano i notebook per esperimenti e attività di esplorazione dei dati. Per ulteriori informazioni, consulta Utilizzo di notebook. | 1° ottobre 2021 |
| L'accesso diretto alle fonti di streaming ora disponibile | Quando si aggiungono origini dati al processo ETL nell'edit or visivo, è possibile fornire informazioni per accedere al flusso di dati, anziché utilizzar e un database e una tabella di Data Catalog. | 30 settembre 2021 |
| Ha documentato la politica di supporto delle AWS Glue versioni | Sono state aggiunte informazi oni sulla politica di supporto delle AWS Glue versioni e sulle fasi di fine vita per alcune AWS Glue versioni. Per ulteriori informazioni, consulta Policy di supporto versione AWS Glue. | 24 settembre 2021 |
| I connettori personalizzati possono ora essere utilizzati con le anteprime dei dati | Quando modifichi il nodo dell'origine dati utilizzando un connettore personalizzato, puoi visualizzare in anteprima il set di dati scegliendo la scheda Anteprima dati. Per ulteriori informazioni, consulta Connettori personalizzati. | 24 settembre 2021 |

| | | |
|---|---|---|
| Support per sessioni AWS Glue interattive (anteprima privata) | (Anteprima privata) Sono state aggiunte informazioni sull'utilizzo di sessioni AWS Glue interattive per eseguire carichi di lavoro Spark nel cloud da qualsiasi notebook Jupyter. Le sessioni interattive sono il metodo preferito per sviluppare il codice ETL ( AWS Glue Extract, Transform, Load) quando si utilizza la versione 2.0 o successiva. AWS Glue Per ulteriori informazioni, vedere Configurazione ed esecuzione di sessioni AWS Glue interattive per Jupyter Notebook. | 24 agosto 2021 |
| Supporto per la creazione di flussi di lavoro dai progetti (GA) | (Anteprima pubblica) Sono state aggiunte informazioni sulla codifica dei casi d'uso comuni di estrazione, trasformazione e caricamento (ETL) nei piani e sulla creazione di flussi di lavoro dai piani. Consente agli analisti di dati di creare ed eseguire facilmente processi ETL complessi. Per ulteriori informazioni, consulta Esecuzione di attività ETL complesse utilizzando gli schemi e i flussi di lavoro in AWS Glue. | 23 agosto 2021 |

| | | |
|---|---|---|
| Support per AWS Glue la versione 3.0. | Sono state aggiunte informazioni sul supporto per la AWS Glue versione 3.0 che supporta l'aggiornamento del motore Apache Spark 3.0 per l'esecuzione dei job ETL di Apache Spark e altre ottimizzazioni e aggiornamenti. Per ulteriori informazioni, consulta le note di AWS Glue rilascio e la migrazione e dei lavori alla versione 3.0. AWS GlueAWS Glue Altre funzionalità di questa versione includono lo AWS Glue shuffle manager, un lettore CSV vettorializzato SIMD e i predicati delle partizioni del catalogo. Per ulteriori informazioni, consulta AWS Glue Spark shuffle manager con Amazon S3, Opzioni di formato per ingressi e uscite ETL e Filtraggio lato server utilizzando i predicati delle partizioni del catalogo AWS Glue. | 18 agosto 2021 |
| AWS GovCloud (US) Region | AWS Glue Studio è ora disponibile in AWS GovCloud (US) Region | 18 agosto 2021 |

| | | |
|---|---|---|
| Creazione di shell Python disponibile in AWS Glue Studio | Quando si crea un nuovo processo, è ora possibile scegliere di creare un processo di shell Python. Per ulteriori informazioni, consulta Avvio della creazione del processo e Modifica di processi shell Python in AWS Glue Studio. | 13 agosto 2021 |
| Support per l'avvio di un flusso di lavoro con un EventBridge evento Amazon | Sono state aggiunte informazioni su come AWS Glue può essere un consumatore di eventi in un'architettura basata sugli eventi. Per ulteriori informazioni, consulta Avvio di un AWS Glue flusso di lavoro con un EventBridge evento Amazon e Visualizzazione degli EventBridge eventi che hanno avviato un flusso di lavoro. | 14 luglio 2021 |
| Aggiunta di JSON come formato di dati supportato per il registro degli AWS Glue schemi | Aggiunte informazioni su JSON come formato dati supportato (oltre ad AVRO). Per ulteriori informazioni, consulta Registro degli schemi di AWS Glue. | 30 giugno 2021 |

| | | |
|---|---|---|
| Crea lavori di AWS Glue streaming senza una tabella Data Catalog | La funzione Python `create_data_frame_from_options` o `getSource` per gli script Scala supportano la creazione di processi ETL di streaming che fanno riferimento direttamente ai flussi di dati anziché richiedere una tabella del catalogo dati. | 15 giugno 2021 |
| AWS Glue le trasformazioni di apprendimento automatico ora supportano AWS Key Management Service le chiavi | È possibile specificare una configurazione o una AWS KMS chiave di sicurezza quando si configurano le trasformazioni di AWS Glue Machine Learning con la console, la CLI o il. AWS Glue APIs Per ulteriori informazioni, consulta Utilizzo della crittografia dati con le trasformazioni basate su machine learning e API di Machine Learning AWS Glue. | 15 giugno 2021 |
| Aggiornamento alla politica gestita AWSGlue ConsoleFullAccess AWS | Sono state aggiunte informazioni su un aggiornamento minore alla politica AWSGlue ConsoleFullAccess AWS gestita. Per ulteriori informazioni, vedere AWS Glue Aggiornamenti alle politiche AWS gestite. | 10 giugno 2021 |

| | | |
|---|---|---|
| Visualizzare il set di dati del processo durante la creazione e la modifica dei processi | È possibile utilizzare la nuova scheda di anteprima dati per un nodo nel diagramma del processo per visualizzare un esempio dei dati elaborati da tale nodo. Per ulteriori informazioni, consulta Utilizzo delle anteprime dei dati nell'editor visivo dei processi. | 7 giugno 2021 |
| Supporto per specificare un valore che indica la posizione della tabella per l'output del crawler. | Sono state aggiunte informazioni su come specificare un valore che indica la posizione della tabella durante la configurazione dell'output del crawler. Per ulteriori informazioni, consulta Come specificare la posizione della tabella. | 4 giugno 2021 |
| Supporto per il crawling di un campione di file in un set di dati durante il crawling di un archivio dati Amazon S3 | Sono state aggiunte informazioni su come eseguire il crawling di un campione di file durante il crawling di Amazon S3. Per ulteriori informazioni, consulta Proprietà del crawler. | 10 maggio 2021 |

| | | |
|---|---|---|
| Support per la scrittura AWS Glue ottimizzata del parquet | Sono state aggiunte informazioni sull'utilizzo del parquet writer AWS Glue ottimizzato DynamicFrames per creare o aggiornare tabelle con la `parquet` classificazione. Per ulteriori informazioni, consultate Creazione di tabelle, aggiornamento dello schema e aggiunta di nuove partizioni nel catalogo dati da processi AWS Glue ETL e Opzioni di formato per ingressi e uscite ETL in. AWS Glue | 4 maggio 2021 |
| Supporto per le password di autenticazione client Kafka | Sono state aggiunte informazioni su come i job ETL in streaming AWS Glue supportano l'autenticazione dei certificati client SSL con i produttori di stream Apache Kafka. Ora puoi fornire un certificato personalizzato durante la definizione di una AWS Glue connessione a un cluster Apache Kafka, che verrà utilizzato per l'autenticazione con esso. AWS Glue Per ulteriori informazioni, consulta Proprietà della connessione AWS Glue e API di connessione. | 28 Aprile 2021 |

| | | |
|---|---|---|
| Supporto per l'utilizzo di dati da Amazon Kinesis Data Streams in un altro account nei processi ETL di streaming | Sono state aggiunte informazioni su come creare un processo ETL di streaming per utilizzare i dati da Amazon Kinesis Data Streams in un altro account. Per ulteriori informazioni, consulta Aggiungere lavori ETL in streaming in. AWS Glue | 30 marzo 2021 |
| Trasformazione SQL disponibile | Puoi utilizzare un nodo di trasformazione SQL per scrivere la tua trasformazione sotto forma di query SQL. Per ulteriori informazioni, consulta Utilizzo di una query SQL per trasformare i dati. | 23 marzo 2021 |
| Supporto per la creazione di flussi di lavoro dagli schemi (anteprima pubblica) | (Anteprima pubblica) Sono state aggiunte informazioni sulla codifica dei casi d'uso comuni di estrazione, trasformazione e caricamento (ETL) nei piani e sulla creazione di flussi di lavoro dai piani. Consente agli analisti di dati di creare ed eseguire facilmente processi ETL complessi. Per ulteriori informazioni, consulta Esecuzione di attività ETL complesse utilizzando gli schemi e i flussi di lavoro in AWS Glue. | 22 marzo 2021 |

| | | |
|---|---|---|
| I connettori possono essere utilizzati per le destinazioni dati | L'utilizzo di un Marketplace AWS connettore o personalizzato per la destinazione dei dati è ora supportato. Per ulteriori informazioni, consulta Creazione di processi con connettori personalizzati. | 15 marzo 2021 |
| Support per le metriche di importanza delle colonne per le trasformazioni dell'apprendimento AWS Glue automatico | Sono state aggiunte informazioni sulla visualizzazione delle metriche di importanza delle colonne quando si lavora con le trasformazioni di apprendimento AWS Glue automatico. Per ulteriori informazioni, consulta Working with Machine Learning Transforms sulla AWS Glue console. | 5 febbraio 2021 |
| Job scheduling ora disponibile in AWS Glue Studio | È possibile definire una pianificazione basata sul tempo per le esecuzioni del processo in AWS Glue Studio. È possibile utilizzare la console per creare una pianificazione di base o definire una pianificazione più complessa utilizzando la sintassi cron di tipo Unix. Per ulteriori informazioni, consulta Pianificazione delle esecuzioni. | 21 dicembre 2020 |

| | | |
|---|---|---|
| AWS Glue Connettori personalizzati rilasciati | AWS Glue I connettori personalizzati consentono di scoprire e abbonarsi ai connettori in Marketplace AWS. Abbiamo anche rilasciat o le interfacce di runtime AWS Glue Spark per collegare connettori creati per Apache Spark Datasource, Athena federated query e JDBC. APIs Per ulteriori informazioni, consulta Utilizzo di connettori e connessioni con. AWS Glue Studio | 21 dicembre 2020 |
| Support per l'esecuzione di job ETL in streaming nella AWS Glue versione 2.0 | Aggiunte informazioni sull'esec uzione di processi ETL di streaming in Glue versione 2.0. Per ulteriori informazioni, consulta Aggiungere lavori ETL in streaming in. AWS Glue | 18 dicembre 2020 |
| Supporto per il partizion amento del carico di lavoro con esecuzione limitata | Aggiunte informazioni sull'abil itazione del partizionamento del carico di lavoro per configurare i limiti superiori della dimensione del set di dati o il numero di file elaborati nelle esecuzioni dei processi ETL. Per ulteriori informazioni, consulta Partizionamento del carico di lavoro con esecuzion e limitata. | 23 novembre 2020 |

| | | |
|---|---|---|
| Supporto per una gestione avanzata delle partizioni | Sono state aggiunte informazioni su come utilizzare new APIs per aggiungere o eliminare un indice di partizione in to/from una tabella esistente. Per ulteriori informazioni, consulta Utilizzo degli indici delle partizioni. | 23 novembre 2020 |
| Support per il registro AWS Glue degli schemi | Sono state aggiunte informazioni sull'utilizzo dello AWS Glue Schema Registry per individuare, controllare ed evolvere centralmente gli schemi. Per ulteriori informazioni, vedere AWS Glue Schema Registry. | 19 novembre 2020 |
| Supporto per il formato di input Grok nei processi ETL di streaming | Aggiunte informazioni sull'applicazione dei pattern Grok alle origini di streaming, ad esempio i file di log. Per ulteriori informazioni, consulta Applicazione di pattern Grok alle sorgenti di streaming. | 17 novembre 2020 |
| Support per l'aggiunta di tag ai flussi di lavoro sulla console AWS Glue | Sono state aggiunte informazioni sull'aggiunta di tag durante la creazione di un flusso di lavoro utilizzando la console AWS Glue . Per ulteriori informazioni, consulta Creazione e creazione di un flusso di lavoro utilizzando la AWS Glue console. | 27 ottobre 2020 |

| | | |
|---|---|---|
| Supporto per le esecuzione incrementale del crawler | Aggiunte informazioni sul supporto per le esecuzioni di crawler incrementali, che eseguono il crawling solo delle cartelle Amazon S3 aggiunte dall'ultima esecuzione. Per ulteriori informazioni, consulta Crawling incrementale. | 21 ottobre 2020 |
| Supporto per il rilevamento dello schema per le origini dati ETL di streaming. supporto per le origini dei dati ETL di streaming Avro e Kafka autogestito | I job di estrazione, trasforma zione e caricamento (ETL) in streaming AWS Glue possono ora rilevare automaticamente lo schema dei record in entrata e gestire le modifiche allo schema per record. Sono ora supportate le origini di dati Kafka autogestite. I processi ETL di streaming ora supportano il formato Avro nelle origini dati. Per ulteriori informazioni, vedere Streaming ETL in AWS Glue, Definizione delle proprietà del lavoro per un lavoro ETL di streaming e Note e restrizio ni per le sorgenti di streaming Avro. | 7 ottobre 2020 |
| Supporto per il crawling delle origini dei dati MongoDB e DocumentDB | Aggiunte informazioni sul supporto per il crawling delle origini dati MongoDB e Amazon DocumentDB (con compatibilità MongoDB). Per ulteriori informazioni, consulta Definizione di crawler. | 5 ottobre 2020 |

| | | |
|---|---|---|
| Supporto per la conformità a FIPS | Aggiunte informazioni sugli endpoint FIPS per i clienti che necessitano di moduli crittografici convalidati FIPS 140-2 quando accedono ai dati con AWS Glue. Per ulteriori informazioni, consulta la pagina Conformità FIPS. | 23 settembre 2020 |
| AWS Glue Studio fornisce un'interfaccia visiva facile da usare per la creazione e il monitoraggio dei lavori | Ora è possibile utilizzare una semplice interfaccia grafica per comporre lavori che spostano e trasformano i dati ed eseguirli su AWS Glue. È quindi possibile utilizzar e il pannello di controllo di esecuzione dei processi in AWS Glue Studio per monitorare l'esecuzione di ETL e garantire che i processi funzionino come previsto. Per ulteriori informazioni, consulta la Guida per l'utente di AWS Glue Studio. | 23 settembre 2020 |
| Supporto per la creazione di indici di tabella per migliorare le prestazioni delle query | Aggiunte informazioni sulla creazione di indici di tabella per consentire il recupero di un sottoinsieme di partizion i da una tabella. Per ulteriori informazioni, consulta Utilizzo degli indici delle partizioni. | 09 settembre 2020 |

| | | |
|---|---|---|
| Support per tempi di avvio ridotti durante l'esecuzione di job Apache Spark ETL nella AWS Glue versione 2.0. | Sono state aggiunte informazioni sul supporto per la AWS Glue versione 2.0 che fornisce un'infrastruttura aggiornata per l'esecuzione dei job ETL di Apache Spark con tempi di avvio ridotti, modifiche nella registrazione e supporto per specificare moduli Python aggiuntivi a livello di job. Per ulteriori informazioni, consulta Note di rilascio di AWS Glue ed Esecuzione di processi ETL Spark con tempi di avvio ridotti. | 10 agosto 2020 |
| Supporto per limitare il numero di esecuzioni simultanee del flusso di lavoro. | Aggiunte informazioni su come limitare il numero di esecuzioni simultanee per un determinato flusso di lavoro. Per ulteriori informazioni, consulta Creazione e creazione di un flusso di lavoro utilizzando la console. AWS Glue | 10 agosto 2020 |

| | | |
|---|---|---|
| Supporto per il crawling di un datastore Amazon S3 utilizzando un endpoint VPC | Aggiunte informazioni sulla configurazione dell'archivio dati Amazon S3 per consentire l'accesso solo a un ambiente Amazon Virtual Private Cloud (Amazon VPC) per motivi di sicurezza, audit o controllo. Per ulteriori informazioni, consulta Crawling di un datastore Amazon S3 utilizzando un endpoint VPC. | 7 agosto 2020 |
| Supporto per la ripresa delle esecuzioni del flusso di lavoro | Aggiunte informazioni su come riprendere le esecuzioni del flusso di lavoro completate solo parzialmente perché uno o più nodi (processi o crawler) non sono stati completati correttamente. Per ulteriori informazioni, consulta Ripresa e ripristino dell'esecuzione di un flusso di lavoro. | 27 luglio 2020 |
| Supporto per l'abilitazione di certificati emessi da CA privati nelle connessioni Kafka in AWS Glue. | Aggiunte informazioni sulle nuove opzioni di connessione che supportano l'abilitazione dei certificati emessi da una CA privati per le connessioni Kafka in AWS Glue. Per ulteriori informazioni, consulta Tipi e opzioni di connessione per ETL in AWS Glue e Parametri speciali utilizzati da AWS Glue. | 20 luglio 2020 |

| | | |
|---|---|---|
| Supporto per la lettura dei dati DynamoDB in un altro account | Sono state aggiunte informazioni sul AWS Glue supporto per la lettura dei dati dalla tabella DynamoDB di un altro AWS account Per ulteriori informazioni, vedere Lettura da dati DynamoDB in un altro account. | 17 luglio 2020 |
| Support per una connessione writer DynamoDB AWS Glue nella versione 1.0 o successiva | Aggiunte informazioni sul supporto per il writer DynamoDB e opzioni di connessione nuove o aggiornate per la lettura o la scrittura di DynamoDB. Per ulteriori informazioni, consulta Tipi di connessione e opzioni per ETL in AWS Glue. | 17 luglio 2020 |
| Support per i collegamenti alle risorse e per il controllo degli accessi tra account utilizzando entrambi AWS Glue e Lake Formation | Sono stati aggiunti contenuti sui nuovi oggetti di Data Catalog denominati link alle risorse e su come gestire la condivisione delle risorse di Data Catalog tra account con entrambi AWS Glue e AWS Lake Formation. Per ulteriori informazioni, consulta Concedere l'accesso multi-account e Link alle risorse della tabella. | 7 luglio 2020 |

| | | |
|---|---|---|
| Supporto per il campionamento dei registri durante il crawling dei datastore DynamoDB | Sono state aggiunte informazioni sulle nuove proprietà che puoi configurare durante il crawling di un datastore DynamoDB. Per ulteriori informazioni, consulta Proprietà del crawler. | 12 giugno 2020 |
| Supporto per l'arresto di un'esecuzione del flusso di lavoro. | Sono state aggiunte informazioni su come interrompere l'esecuzione di un flusso di lavoro per un determinato flusso di lavoro. Per ulteriori informazioni, vedere Arresto di un'esecuzione del flusso di lavoro. | 14 maggio 2020 |
| Supporto per i processi ETL di streaming Spark | Sono state aggiunte informazioni sulla creazione di processi ETL (Extract, Transform and Load) con origini dati in streaming. Per ulteriori informazioni, consulta Aggiunta di processi di streaming ETL in AWS Glue. | 27 aprile 2020 |

| | | |
|---|---|---|
| Supporto per la creazione di tabelle, l'aggiornamento dello schema e l'aggiunta di nuove partizioni nel catalogo dati dopo l'esecuzione di un processo ETL | Sono state aggiunte informazioni su come abilitare la creazione di tabelle, l'aggiornamento dello schema e l'aggiunta di nuove partizioni per visualizzare i risultati del processo ETL nel catalogo dati. Per ulteriori informazioni, consulta Creazione di tabelle, aggiornamento dello schema e aggiunta di nuove partizioni nel catalogo dati da AWS Glue ETL Jobs. | 2 aprile 2020 |
| Support per specificare una versione per il formato di dati Apache Avro come input e output ETL in AWS Glue | Aggiunte informazioni su come specificare una versione per il formato dati Apache Avro come input e output ETL in AWS Glue. La versione predefinita 1.7. Puoi utilizzare l'opzione del formato `version` per specificare Avro versione 1.8 per abilitare la lettura/scrittura logica. Per ulteriori informazioni, consulta Opzioni di formato per ingressi e uscite ETL in. AWS Glue | 31 marzo 2020 |

| | | |
|---|---|---|
| Supporto per il committer ottimizzato EMRFS S3 per la scrittura di dati Parquet in Amazon S3 | Sono state aggiunte informazioni su come impostare un nuovo flag per abilitare il committer ottimizzato EMRFR S3 per la scrittura dei dati Parquet in Amazon S3 durante la creazione o l'aggiornamento di un processo AWS Glue . Per ulteriori informazioni, vedere Parametri speciali utilizzati da. AWS Glue | 30 marzo 2020 |
| Il supporto per l'apprendimento automatico si trasforma in una risorsa gestita da tag di AWS risorse | Sono state aggiunte informazioni sull'utilizzo dei tag AWS delle risorse per gestire e controllare l'accesso alle trasformazioni del machine learning. AWS Glue Puoi assegnare tag di AWS risorsa a job, trigger, endpoint, crawler e trasformarsi in machine learning. AWS GluePer ulteriori informazioni, consulta Tag in.AWSAWS Glue | 2 marzo 2020 |
| Supporto per argomenti di lavoro non sovrascrivibili | Aggiunte informazioni sul supporto per parametri di lavoro speciali che non possono essere sovrascritti nei trigger o quando si esegue il processo. Per ulteriori informazioni, consulta Aggiunta di processi in AWS Glue. | 12 febbraio 2020 |

| | | |
|---|---|---|
| Supporto per nuove trasformazioni per l'utilizzo con set di dati in Amazon S3 | Sono state aggiunte informazioni sulle nuove trasformazioni (Merge, Purge e Transition) ed esclusioni delle classi di storage Amazon S3 per applicazioni Apache Spark per l'utilizzo con set di dati in Amazon S3. Per ulteriori informazioni sul supporto per queste trasformazioni per Python, mergeDynamicFrameconsulta Working with Datasets in Amazon S3. Per Scala, vedi e Scala. mergeDynamicFramesAWS Glue GlueContext APIs | 16 gennaio 2020 |
| Supporto per l'aggiornamento del Catalogo Dati con nuove informazioni di partizione da un processo ETL | Sono state aggiunte informazioni su come codificare uno script di estrazione, trasformazione e caricamento (ETL) per aggiornarlo AWS Glue Data Catalog con nuove informazioni sulla partizione. Con questa caratteristica, non è più necessario eseguire nuovamente il crawler al termine del processo per visualizzare le nuove partizioni. Per ulteriori informazioni, consulta Aggiornamento del catalogo dati con nuove partizioni. | 15 gennaio 2020 |

| | | |
|---|---|---|
| Nuovo tutorial: utilizzo di un SageMaker notebook AI | È stato aggiunto un tutorial che dimostra come usare un SageMaker notebook Amazon per aiutarti a sviluppare i tuoi script ETL e di machine learning. Vedi il tutorial: Usa un Amazon SageMaker Notebook con il tuo endpoint di sviluppo. | 3 gennaio 2020 |
| Supporto per la lettura da MongoDB e Amazon DocumentDB (compatibile con MongoDB) | Aggiunte informazioni sui nuovi tipi di connessione e opzioni di connessio ne per leggere e scrivere su MongoDB e Amazon DocumentDB (con compatibi lità MongoDB). Per ulteriori informazioni, consulta Tipi di connessione e opzioni per ETL in AWS Glue. | 17 dicembre 2019 |
| Varie correzioni e chiarimenti | Sono state aggiunte diverse correzioni e chiarimenti. Sono state rimosse delle voci dal capitolo Problemi noti. Sono stati aggiunti avvisi che AWS Glue supportano solo le chiavi master simmetriche del cliente (CMKs) quando si specificano le impostazi oni di crittografia del Data Catalog e si creano configura zioni di sicurezza. È stata aggiunta una nota che AWS Glue non supporta la scrittura su Amazon DynamoDB. | 9 dicembre 2019 |

| | | |
|---|---|---|
| Supporto per driver JDBC personalizzati | Sono state aggiunte informazioni sulla connessione a sorgenti e destinazioni di dati con driver JDBC che AWS Glue non supportano nativamente, come MySQL versione 8 e Oracle Database versione 18. Per ulteriori informazioni, vedere Valori JDBC ConnectionType. | 25 novembre 2019 |
| Support per il collegamento di notebook SageMaker AI a diversi endpoint di sviluppo | Sono state aggiunte informazioni su come collegare un notebook SageMaker AI a diversi endpoint di sviluppo. Aggiornamenti per descrivere la nuova azione della console per il passaggio a un nuovo endpoint di sviluppo e la nuova policy SageMaker AI IAM. Per ulteriori informazioni, consulta Working with Notebooks on the AWS Glue Console e Creazione di una policy IAM per Amazon SageMaker AI Notebooks. | 21 novembre 2019 |

| | | |
|---|---|---|
| Support per la AWS Glue versione nelle trasformazioni di apprendimento automatico | Sono state aggiunte informazioni sulla definizione della AWS Glue versione in una trasformazione di apprendimento automatico per indicare con quale versione di AWS Glue una trasformazione di apprendimento automatico è compatibile. Per ulteriori informazioni, consulta Working with Machine Learning Transforms sulla AWS Glue console. | 21 novembre 2019 |
| Supporto per il riavvolgimento dei segnalibri di processo | Sono state aggiunte informazioni sul riavvolgimento dei segnalibri di processo per qualsiasi esecuzione precedente, con conseguente rielaborazione dei dati dell'esecuzione del processo successivo solo dall'esecuzione del processo con il segnalibro. Sono descritte due nuove opzioni secondarie per l'opzione `job-bookmark-pause` che consentono di eseguire un processo tra due segnalibri. Per ulteriori informazioni, vedere Tracciamento dei dati elaborati utilizzando i segnalibri di lavoro e i parametri speciali utilizzati da AWS Glue. | 22 ottobre 2019 |

| | | |
|---|---|---|
| Supporto per certificati JDBC personalizzati per la connessione a un archivio dati | Sono state aggiunte informazioni sul AWS Glue supporto dei certificati JDBC personalizzati per le connessioni SSL a sorgenti o destinazioni di AWS Glue dati. Per ulteriori informazioni, consulta Uso di connessioni nella console AWS Glue. | 10 ottobre 2019 |
| Supporto per Python wheel | Sono state aggiunte informazioni sul AWS Glue supporto dei file wheel (insieme ai file egg) come dipendenze per i lavori della shell Python. Per ulteriori informazioni, consulta Fornire la propria libreria Python. | 26 settembre 2019 |
| Support per il controllo delle versioni degli endpoint di sviluppo in AWS Glue | Sono state aggiunte informazioni sulla definizione degli endpoint `Glue version` in fase di sviluppo. `Glue version`determina le versioni di Apache Spark e Python supportate. AWS Glue Per ulteriori informazioni, consulta Aggiunta di un endpoint di sviluppo. | 19 settembre 2019 |

| | | |
|---|---|---|
| Support per il monitoraggio AWS Glue tramite Spark UI | Sono state aggiunte informazioni sull'utilizzo dell'interfaccia utente di Apache Spark per monitorare ed eseguire il debug dei job AWS Glue ETL in esecuzione sul sistema dei AWS Glue job e delle applicazioni Spark sugli endpoint di sviluppo. AWS Glue Per ulteriori informazioni, consulta Monitoraggio tramite l'interfaccia utente di Spark. AWS Glue | 19 settembre 2019 |
| Miglioramento del supporto per lo sviluppo di script ETL locali utilizzando la libreria ETL pubblica AWS Glue | È stato aggiornato il contenuto della libreria AWS Glue ETL per riflettere il fatto che la AWS Glue versione 1.0 è ora supportata. Per ulteriori informazioni, vedete Sviluppo e test di script ETL a livello locale utilizzando la AWS Glue libreria ETL. | 18 settembre 2019 |
| Supporto per l'esclusione delle classi di archiviazione Amazon S3 durante l'esecuzione di processi | Sono state aggiunte informazioni sull'esclusione delle classi di storage Amazon S3 durante l'esecuzione di processi AWS Glue ETL che leggono file o partizioni da Amazon S3. Per ulteriori informazioni, consulta Esclusione delle classi di storage Amazon S3. | 29 agosto 2019 |

| | | |
|---|---|---|
| Support per lo sviluppo di script ETL locali utilizzando la libreria AWS Glue ETL pubblica | Aggiunte informazioni su come sviluppare e testare script ETL Python e Scala in locale senza la necessità di una connessione di rete. Per ulteriori informazioni, vedete Sviluppo e test di script ETL a livello locale utilizzando la libreria ETL. AWS Glue | 28 agosto 2019 |
| Problemi noti | Sono state aggiunte informazioni sui problemi noti in AWS Glue. Per ulteriori informazioni, consulta Problemi noti per AWS Glue. | 28 agosto 2019 |
| Il supporto per l'apprendimento automatico si trasforma in AWS Glue | Sono state aggiunte informazioni sulle funzionalità di apprendimento automatico fornite da AWS Glue per creare trasformazioni personalizzate. È possibile creare queste trasformazioni al momento della creazione di un processo. Per ulteriori informazioni, consulta Machine Learning Transforms in AWS Glue. | 8 agosto 2019 |
| Supporto per Amazon Virtual Private Cloud condiviso | Sono state aggiunte informazioni sul AWS Glue supporto per Amazon Virtual Private Cloud condiviso. Per ulteriori informazioni, consulta Shared Amazon VPCs. | 6 agosto 2019 |

| | | |
|---|---|---|
| Support per il controllo delle versioni in AWS Glue | Sono state aggiunte informazioni sulla definizione delle `Glue version` proprietà del lavoro. AWS Glue version determina le versioni di Apache Spark e Python supportate. AWS Glue Per ulteriori informazioni, consulta Aggiungere lavori in. AWS Glue | 24 luglio 2019 |
| Supporto per le opzioni di configurazione aggiuntive per gli endpoint di sviluppo | Sono state aggiunte informazioni sulle opzioni di configurazione per gli endpoint di sviluppo con carichi di lavoro intensi in termini di memoria. È possibile scegliere tra due nuove configurazioni che offrono maggiore quantità di memoria per esecutore. Per ulteriori informazioni, consulta Lavorare con gli endpoint di sviluppo sulla AWS Glue console. | 24 luglio 2019 |

| | | |
|---|---|---|
| Supporto per l'esecuzione di attività di estrazione, trasformazione e caricamento (ETL) utilizzando i flussi di lavoro | Sono state aggiunte informazioni sull'utilizzo di un nuovo costrutto chiamato workflow per progettare un'attività ETL (Extract, Transform and Load) complessa che AWS Glue può essere eseguita e monitorata come un'unica entità. Per ulteriori informazioni, vedere Esecuzione di attività ETL complesse utilizzando flussi di lavoro in. AWS Glue | 20 giugno 2019 |
| Supporto per Python 3.6 in processi shell di Python | Sono state aggiunte informazioni sul supporto di Python 3.6 in processi shell di Python. Puoi specificare Python 2.7 o Python 3.6 come proprietà di un processo. Per ulteriori informazioni, consulta Aggiunta di processi shell di Python in AWS Glue. | 5 giugno 2019 |
| Supporto di endpoint di cloud privato virtuale (VPC, Virtual Private Cloud) | Sono state aggiunte informazioni sulla connessione diretta AWS Glue tramite un endpoint di interfaccia nel VPC. Quando utilizzi un endpoint di interfaccia VPC, la comunicazione tra il tuo VPC e il tuo VPC AWS Glue viene condotta in modo completo e sicuro all'interno della rete. AWS Per ulteriori informazioni, consulta Utilizzo AWS Glue con endpoint VPC. | 4 giugno 2019 |

| | | |
|---|---|---|
| Support per la registrazione continua e in tempo reale dei AWS Glue lavori. | Sono state aggiunte informazioni sull'attivazione e la visualizzazione dei log dei job di Apache Spark in tempo reale, CloudWatch inclusi i registri dei driver, i log di ciascun executor e una barra di avanzamento dei job Spark. Per ulteriori informazioni, consulta l'articolo relativo alla registrazione continua dei processi AWS Glue. | 28 maggio 2019 |
| Supporto per le tabelle del Catalogo Dati esistenti come origini crawler | Sono state aggiunte informazioni su come specificare un elenco di tabelle del catalogo dati esistenti come origini crawler. I crawler possono quindi rilevare le modifiche agli schemi di tabella, aggiornare le definizioni di tabella e registrare nuove partizioni quando i nuovi dati diventano disponibili. Per ulteriori informazioni, consulta Proprietà dei crawler. | 10 maggio 2019 |

| | | |
|---|---|---|
| Supporto per le opzioni di configurazione aggiuntive per i processi con elevati requisiti di memoria | Sono state aggiunte informazioni sulle opzioni di configurazione per i processi Apache Spark con carichi di lavoro con elevati requisiti di memoria. È possibile scegliere tra due nuove configurazioni che offrono maggiore quantità di memoria per esecutore. Per ulteriori informazioni, consulta Aggiungere lavori in. AWS Glue | 5 aprile 2019 |
| Supporto per classificatori CSV personalizzati | Sono state aggiunte informazioni sull'utilizzo di un classificatore CSV personalizzato per dedurre lo schema di vari tipi di dati CSV. Per ulteriori informazioni, consulta Scrittura di classificatori personalizzati. | 26 marzo 2019 |
| Support per i tag AWS delle risorse | Sono state aggiunte informazioni sull'utilizzo dei tag AWS delle risorse per aiutarti a gestire e controllare l'accesso alle tue AWS Glue risorse. Puoi assegnare tag di AWS risorsa a job, trigger, endpoint e crawler in. AWS GluePer ulteriori informazioni, consulta Tag in.AWSAWS Glue | 20 marzo 2019 |

| | | |
|---|---|---|
| Supporto del Catalogo Dati per i processi Spark SQL | Sono state aggiunte informazioni sulla configurazione dei AWS Glue job e degli endpoint di sviluppo da utilizzare AWS Glue Data Catalog come Apache Hive Metastore esterno. In questo modo i processi e gli endpoint di sviluppo eseguono le query Apache Spark SQL direttamente sulle tabelle archiviate in AWS Glue Data Catalog. Per ulteriori informazioni, consulta l'argomento relativo al AWS Glue Data Catalog supporto di per i processi Spark SQL. | 14 marzo 2019 |
| Supporto per processi shell di Python | Aggiunte informazioni sui processi shell di Python e il nuovo campo Maximum capacity (Capacità massima). Per ulteriori informazioni, consulta l'argomento relativo all'aggiunta di processi shell di Python in AWS Glue. | 18 gennaio 2019 |

| | | |
|---|---|---|
| Supporto per le notifiche quando sono presenti modifiche di database e di tabelle | Aggiunte informazioni sugli eventi generati a causa di modifiche al database, alla tabella e alle chiamate all'API della partizione. È possibile configurare le azioni in CloudWatch Events per rispondere a questi eventi. Per ulteriori informazioni, consulta Automazione AWS Glue con CloudWatch eventi. | 16 gennaio 2019 |
| Supporto per la crittografia delle password di connessione | Aggiunte informazioni sulla crittografia di password utilizzate in oggetti di connessione. Per ulteriori informazioni, consulta Crittografia delle password di connessione. | 11 dicembre 2018 |
| Supporto per le autorizzazioni a livello della risorsa e per le policy basate sulla risorsa | Sono state aggiunte informazioni sull'utilizzo delle autorizzazioni a livello di risorsa e delle politiche basate sulle risorse con. AWS Glue Per ulteriori informazioni, consulta gli argomenti indicati in Sicurezza in AWS Glue. | 15 ottobre 2018 |
| Support per notebook con SageMaker intelligenza artificiale | Sono state aggiunte informazioni sull'utilizzo dei notebook SageMaker AI con endpoint di sviluppo. AWS Glue Per ulteriori informazioni, consulta Gestione di notebook. | 5 ottobre 2018 |

| | | |
|---|---|---|
| Supporto per la crittografia | Sono state aggiunte informazioni sull'utilizzo della crittografia con. AWS Glue Per ulteriori informazioni, consulta Crittografia dei dati inattivi, Crittografia dei dati in transito e Configurazione della crittografia in AWS Glue. | 24 agosto 2018 |
| Supporto per i parametri di processo Apache Spark | Aggiunta di informazioni sull'uso dei parametri Apache Spark per migliorare il debug e la profilatura dei processi ETL. È possibile tenere facilmente traccia delle metriche di runtime, ad esempio i byte letti e scritti, l'utilizzo della memoria e il carico sulla CPU del driver e degli esecutori e lo spostamento dei dati tra gli esecutori dalla console. AWS Glue Per ulteriori informazioni, consulta Monitoring AWS Glue Using CloudWatch Metrics, Job Monitoring and Debugging e Working with Jobs on the Console. AWS Glue | 13 luglio 2018 |
| Supporto di DynamoDB come origine dati | Aggiunta di informazioni sul crawling di DynamoDB e su come usarlo come origine dati dei processi ETL. Per ulteriori informazioni, consulta Catalogazione di tabelle con un crawler e Parametri di connessione. | 10 luglio 2018 |

| | | |
|---|---|---|
| Aggiornamenti alla procedura di creazione di un server notebook | Aggiornamento delle informazioni su come creare un server notebook in un'istanza a Amazon EC2 associata a un endpoint di sviluppo. Per ulteriori informazioni, consulta Creazione di un server notebook associato a un endpoint di sviluppo. | 9 luglio 2018 |
| Aggiornamenti ora disponibili tramite RSS | È ora possibile abbonarsi a un feed RSS per ricevere notifiche sugli aggiornamenti alla Guida per gli sviluppatori di AWS Glue . | 25 giugno 2018 |
| Supporto delle notifiche di ritardo per i processi | Aggiunte informazioni sulla configurazione di una soglia di ritardo durante l'esecuzione di un processo. Per ulteriori informazioni, consulta Aggiunta di processi in AWS Glue. | 25 maggio 2018 |
| Configurazione di un crawler per aggiungere nuove colonne | Sono state aggiunte informazioni sulla nuova opzione di configurazione per i crawler,. MergeNewColumns Per maggiori informazioni, consulta Configurazione di un crawler. | 7 maggio 2018 |

| | | |
|---|---|---|
| Supporto del timeout dei processi | Aggiunte informazioni sull'impostazione di una soglia di timeout durante l'esecuzione di un processo. Per ulteriori informazioni, consulta Aggiunta di processi in AWS Glue. | 10 aprile 2018 |
| Supporto script Scala ETL e processi trigger basati su stati di esecuzione aggiuntivi | Informazioni aggiunte sull'utilizzo di Scala come linguaggio di programmazione ETL. Ora l'API trigger supporta anche l'attivazione se viene soddisfatta una qualsiasi delle condizioni (in aggiunta a tutte le condizioni). Inoltre, i processi possono essere attivati sulla base di un'esecuzione processo "non riuscita" o "arrestata" (in aggiunta a un'esecuzione processo "riuscita"). | 12 gennaio 2018 |

# Aggiornamenti precedenti

La tabella seguente descrive le modifiche importanti apportate in ogni versione della Guida per sviluppatori AWS Glue  prima di gennaio 2018.

| Modifica | Descrizione | Data |
|---|---|---|
| Supporto origini dati XML e nuova opzione di configurazione crawler | Informazioni aggiunte sulla classificazione di origini dati XML e nuova opzione crawler per modifiche della partizione. | 16 novembre 2017 |

| Modifica | Descrizione | Data |
| --- | --- | --- |
| Nuove trasformazioni, supporto per motori di database Amazon RDS aggiuntivi e miglioramenti degli endpoint di sviluppo | Informazioni aggiunte sulle trasformazioni di filtraggio e mappatura, supporto per Amazon RDS Microsoft SQL Server e Amazon RDS Oracle e nuove caratteristiche per gli endpoint di sviluppo. | 29 settembre 2017 |
| AWS Glue versione iniziale | Questa è la versione iniziale della Guida per gli sviluppatori di AWS Glue . | 14 agosto 2017 |

# AWS Glossario

Per la AWS terminologia più recente, consultate il AWS glossario nella sezione Reference.Glossario AWS

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.