



Guida per gli sviluppatori

Amazon Keyspaces (per Apache Cassandra)



Amazon Keyspaces (per Apache Cassandra): Guida per gli sviluppatori

Copyright © 2025 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 Amazon Keyspaces?	1
Come funziona	2
Architettura di alto livello	2
Modello di dati Cassandra	5
Accesso ad Amazon Keyspaces	6
Casi d'uso	7
Che cos'è CQL?	7
Confronta Amazon Keyspaces con Cassandra	9
Differenze funzionali con Apache Cassandra	10
Apache Cassandra APIs, operazioni e tipi di dati	11
Creazione ed eliminazione asincrone di spazi chiave e tabelle	11
Autenticazione e autorizzazione	11
Archiviazione	11
Change Data Capture (CDC)	11
Configurazione del cluster	12
Connessioni	12
INparola chiave	13
FROZENcollezioni	13
Transazioni leggere	14
Sistema di bilanciamento del carico	14
Paginazione	14
Partizionatori	15
Dichiarazioni preparate	15
Eliminazione dell'intervallo	15
Tabelle di sistema	15
Timestamp	16
Tipi UDTs definiti dall'utente ()	16
Cassandra APIs, operazioni, funzioni e tipi di dati supportati	16
Supporto per l'API Cassandra	17
Supporto dell'API Cassandra Control Plane	18
Supporto API Cassandra Data Plane	19
Supporto della funzione Cassandra	19
Supporto per i tipi di dati Cassandra	20
Livelli di coerenza di Cassandra supportati	22

Livelli di coerenza di scrittura	22
Leggi i livelli di coerenza	23
Livelli di coerenza non supportati	24
Migrazione ad Amazon Keyspaces	25
Migrazione da Cassandra	26
Compatibilità	27
Stima dei prezzi	28
Strategia di migrazione	37
Migrazione online	38
Migrazione offline	50
Migrazione ibrida	52
Strumenti di migrazione	56
Caricamento dei dati utilizzando cqlsh	58
Caricamento dei dati utilizzando DSBulk	71
Accesso ad Amazon Keyspaces	88
Configurazione AWS Identity and Access Management	88
Registrati per un Account AWS	88
Crea un utente con accesso amministrativo	88
Configurazione di Amazon Keyspaces	90
Utilizzo della console	91
Usando AWS CloudShell	91
Ottenere le autorizzazioni IAM per AWS CloudShell	92
Interazione con Amazon Keyspaces tramite AWS CloudShell	93
Crea credenziali di accesso programmatico	94
Crea credenziali specifiche per il servizio	95
Crea credenziali IAM per l'autenticazione AWS	97
Endpoint di servizio	106
Porte e protocolli	106
Endpoint globali	107
AWS GovCloud (US) Region Endpoint FIPS	112
Endpoint delle regioni della Cina	113
Endpoint Streams	113
Connessione a endpoint dual-stack	116
IPv6 supporto in Amazon Keyspaces	117
Risoluzione DNS in Amazon Keyspaces	117
Uso di cqlsh	118

Utilizzo di <code>cqlsh-expansion</code>	119
Come configurare manualmente le <code>cqlsh</code> connessioni per TLS	124
Usando il AWS CLI	126
Download e configurazione della AWS CLI	126
Utilizzo di AWS CLI con Amazon Keyspaces	126
Utilizzo dell'API	131
Utilizzo di un driver client Cassandra	131
Utilizzo di un driver client Java Cassandra	132
Utilizzo di un driver client Cassandra Python	149
Utilizzo di un driver client Cassandra Node.js	154
Utilizzo di un driver client Cassandra.NET Core	158
Utilizzo di un driver client Cassandra Go	160
Utilizzo di un driver client Cassandra Perl	166
Configura l'accesso tra più account	167
Configura l'accesso tra account in un VPC condiviso	168
Configura l'accesso tra account diversi senza un VPC condiviso	172
Nozioni di base	177
Prerequisiti	178
Crea uno spazio chiave	178
Controlla lo stato di creazione del keyspace	182
Creare una tabella	182
Controlla lo stato di creazione della tabella	191
Operazioni CRUD	192
Crea	192
Lettura	196
Aggiornamento	200
Eliminazione	202
Eliminazione di una tabella	204
Eliminare uno spazio chiave	206
Tutorial e soluzioni	210
Connessione con endpoint VPC	210
Prerequisiti	211
Passaggio 1: avviare un' EC2istanza Amazon	211
Passaggio 2: configura la tua EC2 istanza Amazon	213
Fase 3: creare un endpoint VPC per Amazon Keyspaces	216
Passaggio 4: Configurare le autorizzazioni per la connessione agli endpoint VPC	222

Fase 5: Configurare il monitoraggio	226
Fase 6: (Facoltativo) Procedure consigliate per le connessioni	226
Fase 7: (Facoltativo) Pulizia	229
Integrazione con Apache Spark	231
Prerequisiti	232
Fase 1: Configurazione di Amazon Keyspaces	232
Passaggio 2: configurare il connettore Apache Cassandra Spark	234
Passaggio 3: Creare il file di configurazione dell'app	236
Fase 4: Preparare i dati di origine e la tabella di destinazione	238
Fase 5: Scrivere e leggere i dati di Amazon Keyspaces	240
Risoluzione dei problemi	243
Connessione da Amazon EKS	245
Prerequisiti	245
Fase 1: configurare il cluster Amazon EKS	248
Fase 2: Configurare l'applicazione	253
Fase 3: Creare l'immagine dell'applicazione	256
Fase 4: Implementazione dell'applicazione su Amazon EKS	257
Fase 5: Pulizia (facoltativa)	263
Esportazione di dati su Amazon S3	265
Prerequisiti	266
Fase 1: creare il bucket Amazon S3, scaricare gli strumenti e configurare l'ambiente	267
Fase 2: Configurare il lavoro AWS Glue	270
Fase 3: Eseguì il processo di esportazione AWS Glue dal AWS CLI	273
Fase 4: (Facoltativo) Pianificare il processo di esportazione	274
Fase 5: Pulizia (facoltativa)	275
Gestione delle risorse serverless	278
Stima della dimensione delle righe	279
Stima la dimensione codificata delle colonne	280
Stima la dimensione codificata dei valori dei dati in base al tipo di dati	281
Considera l'impatto delle funzionalità di Amazon Keyspaces sulla dimensione delle righe	282
Scegliete la formula giusta per calcolare la dimensione codificata di una riga	283
Esempio di calcolo della dimensione delle righe	284
Stima del consumo di capacità	286
Stima il consumo di capacità delle query di intervallo	286
Stima il consumo di capacità di lettura delle query limite	287
Stima il consumo di capacità di lettura delle scansioni delle tabelle	288

Stima del consumo di capacità di LWT	289
Stima del consumo di capacità delle colonne statiche	290
Stima della capacità per una tabella multiregionale	294
Stima del consumo di capacità con CloudWatch	296
Configura le modalità di capacità di lettura/scrittura	296
Configura la modalità di capacità su richiesta	297
Configura la modalità di capacità di throughput assegnata	299
Visualizza la modalità di capacità di una tabella	301
Cambia la modalità di capacità	303
Preriscalda un nuovo tavolo per sfruttare la capacità su richiesta	306
Preriscalda una tabella esistente per aumentarne la capacità su richiesta	309
Gestisci la capacità di throughput con la scalabilità automatica	312
Come funziona il ridimensionamento automatico di Amazon Keyspaces	313
Come funziona la scalabilità automatica per le tabelle multiregionali	314
Note per l'utilizzo	315
Configurazione e aggiornamento delle politiche di scalabilità automatica	316
Usa la capacità burst	330
Utilizzo delle funzionalità di Amazon Keyspaces	332
Spazi chiave di sistema	333
system	334
system_schema	335
system_schema_mcs	336
system_multiregion_info	339
Tipi definiti dall'utente () UDTs	342
Configurare le autorizzazioni	343
Crea un UDT	346
Visualizza UDTs	350
Elimina un UDT	353
Lavorare con le query CQL	355
Usa IN SELECT	355
Usa istruzioni batch	359
Ordina i risultati	365
Impagina i risultati	366
Lavorare con i flussi CDC	367
Come funzionano gli stream CDC	368
Usa i flussi CDC	372

Lavorare con i partizionatori	406
Cambia il partizionatore	407
Timestamp lato client	408
AWS Integrazione con i servizi	410
Crea una tabella con timestamp lato client	410
Configura i timestamp lato client	413
Utilizza i timestamp lato client nelle query	416
Replica multi regione	417
Vantaggi	418
Modalità di capacità e prezzi	419
Come funziona	419
Note per l'utilizzo	425
Configurare la replica multi regione	427
Backup e ripristino con point-in-time ripristino	456
Come funziona	456
Usa il ripristino point-in-time	462
Fai scadere i dati con Time to Live	475
Integrazione con i servizi AWS	476
Crea una tabella con un valore TTL predefinito	477
Aggiorna il valore TTL predefinito della tabella	481
Crea una tabella con TTL personalizzato	484
Aggiorna il TTL personalizzato della tabella	486
Si usa INSERT per impostare un TTL personalizzato per le nuove righe	489
Si usa UPDATE per impostare un TTL personalizzato per righe e colonne	490
Lavorare con AWS SDKs	491
Lavorare con i tag	492
Restrizioni di tagging	493
Aggiungi tag a keyspace, tabelle e stream	494
Crea report sull'allocazione dei costi	512
Crea risorse CloudFormation	513
Amazon Keyspaces e modelli CloudFormation	514
Scopri di più su CloudFormation	514
NoSQL Workbench	514
Scarica	515
Nozioni di base	516
Visualizza un modello di dati	518

Crea un modello di dati	521
Modifica un modello di dati	524
Esegui il commit di un modello di dati	526
Modelli di dati di esempio	536
Cronologia delle versioni	537
Esempi di codice	539
Nozioni di base	544
Salve Amazon Keyspaces	545
Informazioni di base	550
Azioni	612
Librerie e strumenti	657
Librerie ed esempi	657
Toolkit per sviluppatori Amazon Keyspaces (per Apache Cassandra)	657
Esempi di Amazon Keyspaces (per Apache Cassandra)	657
AWS Plugin di autenticazione Signature Version 4 (SigV4)	657
Esempi evidenziati e repository degli strumenti di sviluppo	658
Buffer del protocollo Amazon Keyspaces	658
CloudFormation modello per creare CloudWatch dashboard Amazon per i parametri di Amazon Keyspaces (per Apache Cassandra)	658
Utilizzo di Amazon Keyspaces (per Apache Cassandra) con AWS Lambda	658
Utilizzo di Amazon Keyspaces (per Apache Cassandra) con Spring	659
Utilizzo di Amazon Keyspaces (per Apache Cassandra) con Scala	659
Utilizzo di Amazon Keyspaces (per Apache Cassandra) con AWS Glue	659
Amazon Keyspaces (per Apache Cassandra) da Cassandra Query Language (CQL) a convertitore CloudFormation	659
Amazon Keyspaces (per Apache Cassandra), aiutanti per il driver Apache Cassandra per Java	660
Dimostrazione di compressione istantanea di Amazon Keyspaces (per Apache Cassandra)	660
Demo di Amazon Keyspaces (per Apache Cassandra) e del codec Amazon S3	660
Best practice	661
Progettazione NoSQL	662
NoSQL e RDBMS	663
Due concetti chiave	663
Approccio generale	664
Connessioni	665

Come funzionano	665
Come configurare le connessioni	666
Come configurare le politiche di nuovo tentativo	668
Connessioni endpoint VPC	669
Come monitorare le connessioni	670
Come gestire gli errori di connessione	671
Modellazione dei dati	671
Progettazione delle chiavi di partizione	672
Ottimizzazione dei costi	675
Valutazione dei costi a livello di tabella	675
Valuta la modalità di capacità del tuo tavolo	677
Valuta le impostazioni di Application Auto Scaling della tua tabella	682
Identificazione di risorse inutilizzate	689
Valutazione dei modelli di utilizzo delle tabelle	695
Valutare la capacità fornita per un provisioning di dimensioni adeguate	696
Risoluzione dei problemi	706
Errori generali	707
Errori generali	707
Errori di connessione	709
Errori di connessione a un endpoint Amazon Keyspaces	709
Errori di gestione della capacità	721
Errori di capacità serverless	721
Errori del linguaggio di definizione dei dati	727
Errori del linguaggio di definizione dei dati	727
Monitoraggio di Amazon Keyspaces	733
Monitoraggio con CloudWatch	734
Utilizzo delle metriche	735
Parametri e dimensioni	736
Visualizzazione dei parametri di	760
Creazione di allarmi	761
Registrazione con CloudTrail	762
Configurazione delle voci dei file di registro in CloudTrail	762
Informazioni DDL in CloudTrail	763
Informazioni DML in CloudTrail	764
Comprensione delle voci dei file di log di	765
Sicurezza	777

Protezione dei dati	778
Crittografia a riposo	779
Crittografia in transito	800
Riservatezza del traffico Internet	801
AWS Identity and Access Management	802
Destinatari	803
Autenticazione con identità	803
Gestione dell'accesso tramite policy	804
Come funziona Amazon Keyspaces con IAM	806
Esempi di policy basate sull'identità	811
AWS politiche gestite	819
Risoluzione dei problemi	832
Uso di ruoli collegati ai servizi	836
Convalida della conformità	845
Resilienza	846
Sicurezza dell'infrastruttura	846
Utilizzo di endpoint VPC dell'interfaccia	848
Utilizzo degli endpoint VPC dell'interfaccia con flussi	854
Analisi della configurazione e delle vulnerabilità per Amazon Keyspaces	858
Best practice di sicurezza	858
Best practice relative alla sicurezza di prevenzione per	859
Best practice relative alla sicurezza di rilevamento	860
Riferimento al linguaggio CQL	863
Elementi del linguaggio	864
Identificatori	864
Costanti	864
Termini	865
Tipi di dati	865
Codifica JSON dei tipi di dati Amazon Keyspaces	869
Istruzioni DDL	873
Keyspaces	874
Tabelle	878
Tipi	892
Istruzioni DML	894
SELECT	895
INSERT	897

UPDATE	899
DELETE	901
Funzioni integrate	901
Funzioni scalari	902
Quote	904
Quote di servizio Amazon Keyspaces	904
Aumento o diminuzione della velocità di trasmissione effettiva (per tabelle assegnate)	911
Aumento della velocità di trasmissione effettiva assegnata	911
Riduzione della velocità di trasmissione effettiva assegnata	912
Crittografia Amazon Keyspaces a riposo	912
Quote e valori predefiniti per i tipi definiti dall'utente (UDTs) in Amazon Keyspaces	913
Quote e valori predefiniti di Amazon Keyspaces UDT	913
Cronologia dei documenti	915
.....	cmxxviii

Che cos'è Amazon Keyspaces (per Apache Cassandra)?

Amazon Keyspaces (per Apache Cassandra) è un servizio di database scalabile, altamente disponibile e gestito compatibile con Apache Cassandra. Con Amazon Keyspaces, non è necessario effettuare il provisioning, applicare patch o gestire server e non è necessario installare, mantenere o utilizzare software.

Amazon Keyspaces è serverless, quindi paghi solo per le risorse che usi e il servizio ridimensiona automaticamente le tabelle verso l'alto e verso il basso in risposta al traffico delle applicazioni. Puoi creare applicazioni che soddisfano migliaia di richieste al secondo con throughput e storage praticamente illimitati.

Note

Apache Cassandra è un datastore open source a colonne larghe progettato per gestire grandi quantità di dati. [Per ulteriori informazioni, consulta Apache Cassandra.](#)

Amazon Keyspaces semplifica la migrazione, l'esecuzione e la scalabilità dei carichi di lavoro di Cassandra in Cloud AWS. Con pochi clic sulla console di AWS gestione o poche righe di codice, puoi creare keyspaces e tabelle in Amazon Keyspaces, senza distribuire alcuna infrastruttura o installare software.

Con Amazon Keyspaces, puoi eseguire i carichi di lavoro Cassandra esistenti AWS utilizzando lo stesso codice applicativo Cassandra e gli stessi strumenti di sviluppo che usi oggi.

Con il [calcolatore dei prezzi per Amazon Keyspaces \(per Apache Cassandra\)](#) disponibile su Github, puoi stimare i costi mensili per Amazon Keyspaces in base al tuo carico di lavoro Apache Cassandra esistente. Inserisci i parametri dello stato dello strumento Cassandra e della configurazione serverless prevista per Amazon Keyspaces per confrontare i costi diretti tra le due soluzioni. Tieni presente che questo calcolatore si concentra solo sui costi operativi di Amazon Keyspaces rispetto alla distribuzione Cassandra esistente. Non include fattori del costo totale di proprietà (TCO) come la manutenzione dell'infrastruttura, le spese generali operative o i costi di supporto per Cassandra.

Per un elenco degli endpoint Regioni AWS e degli endpoint disponibili, consulta [Service endpoints for Amazon Keyspaces](#).

Ti consigliamo di iniziare leggendo le seguenti sezioni:

Argomenti

- [Amazon Keyspaces: come funziona](#)
- [Casi d'uso di Amazon Keyspaces](#)
- [Cos'è Cassandra Query Language \(CQL\)?](#)

Amazon Keyspaces: come funziona

Amazon Keyspaces rimuove il sovraccarico amministrativo legato alla gestione di Cassandra. Per capire perché, è utile iniziare con l'architettura Cassandra e poi confrontarla con Amazon Keyspaces.

Argomenti

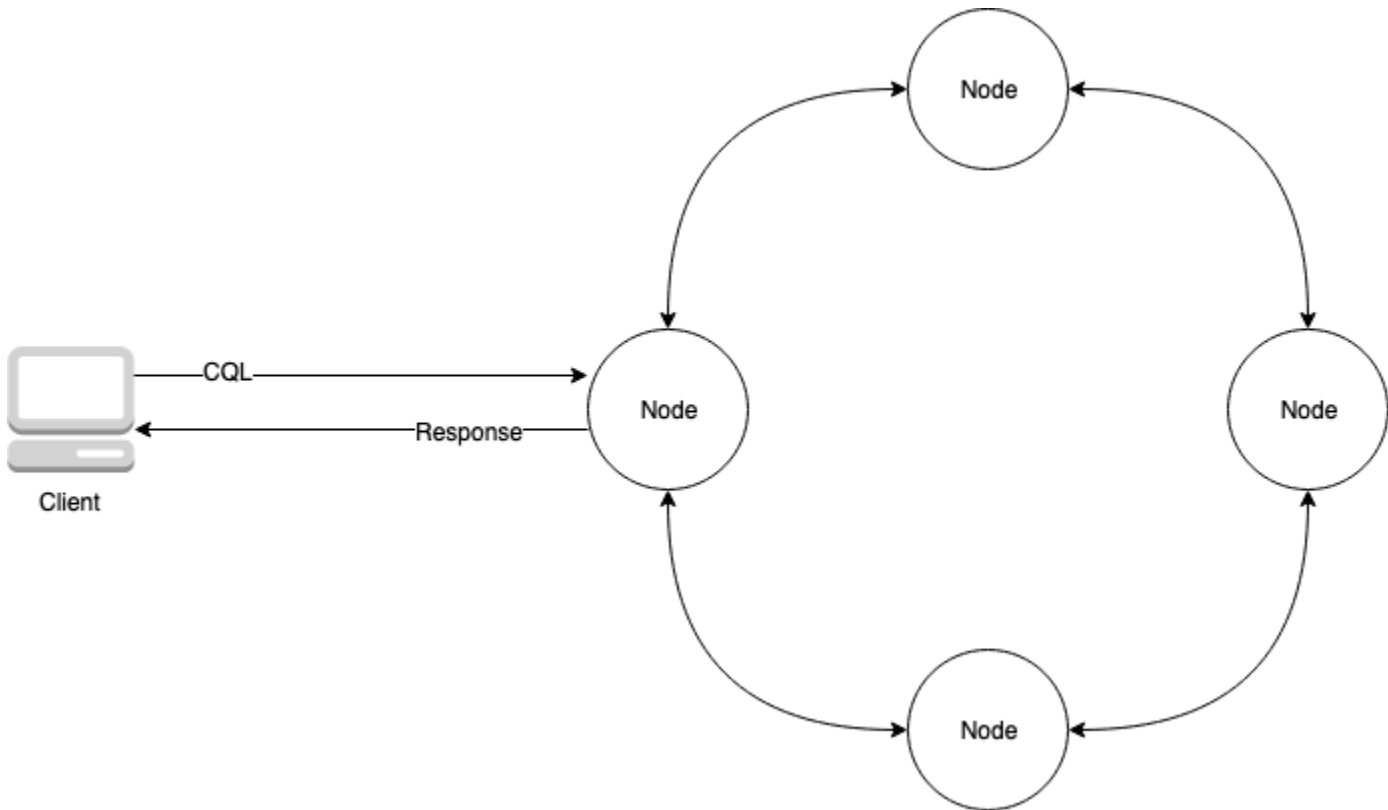
- [Architettura di alto livello: Apache Cassandra vs. Amazon Keyspaces](#)
- [Modello di dati Cassandra](#)
- [Accesso ad Amazon Keyspaces da un'applicazione](#)

Architettura di alto livello: Apache Cassandra vs. Amazon Keyspaces

Apache Cassandra tradizionale viene distribuito in un cluster composto da uno o più nodi. Sei responsabile della gestione di ogni nodo e dell'aggiunta e rimozione dei nodi man mano che il cluster cresce.

Un programma client accede a Cassandra connettendosi a uno dei nodi ed emettendo istruzioni Cassandra Query Language (CQL). CQL è simile a SQL, il linguaggio popolare utilizzato nei database relazionali. Anche se Cassandra non è un database relazionale, CQL fornisce un'interfaccia familiare per interrogare e manipolare i dati in Cassandra.

Il diagramma seguente mostra un semplice cluster Apache Cassandra, composto da quattro nodi.



Un'implementazione di Cassandra di produzione potrebbe consistere in centinaia di nodi, in esecuzione su centinaia di computer fisici in uno o più data center fisici. Ciò può comportare un onere operativo per gli sviluppatori di applicazioni che devono fornire, applicare patch e gestire i server oltre all'installazione, alla manutenzione e al funzionamento del software.

Con Amazon Keyspaces (per Apache Cassandra), non è necessario effettuare il provisioning, applicare patch o gestire i server, quindi puoi concentrarti sulla creazione di applicazioni migliori. Amazon Keyspaces offre due modalità di capacità di throughput per le letture e le scritture: on-demand e provisioned. Puoi scegliere la modalità di capacità di throughput della tabella per ottimizzare il prezzo delle letture e delle scritture in base alla prevedibilità e alla variabilità del carico di lavoro.

Con la modalità on-demand, paghi solo per le operazioni di lettura e scrittura effettivamente eseguite dall'applicazione. Non è necessario specificare in anticipo la capacità di throughput della tabella. Amazon Keyspaces gestisce il traffico delle applicazioni quasi istantaneamente man mano che aumenta o diminuisce, rendendolo una buona opzione per le applicazioni con traffico imprevedibile.

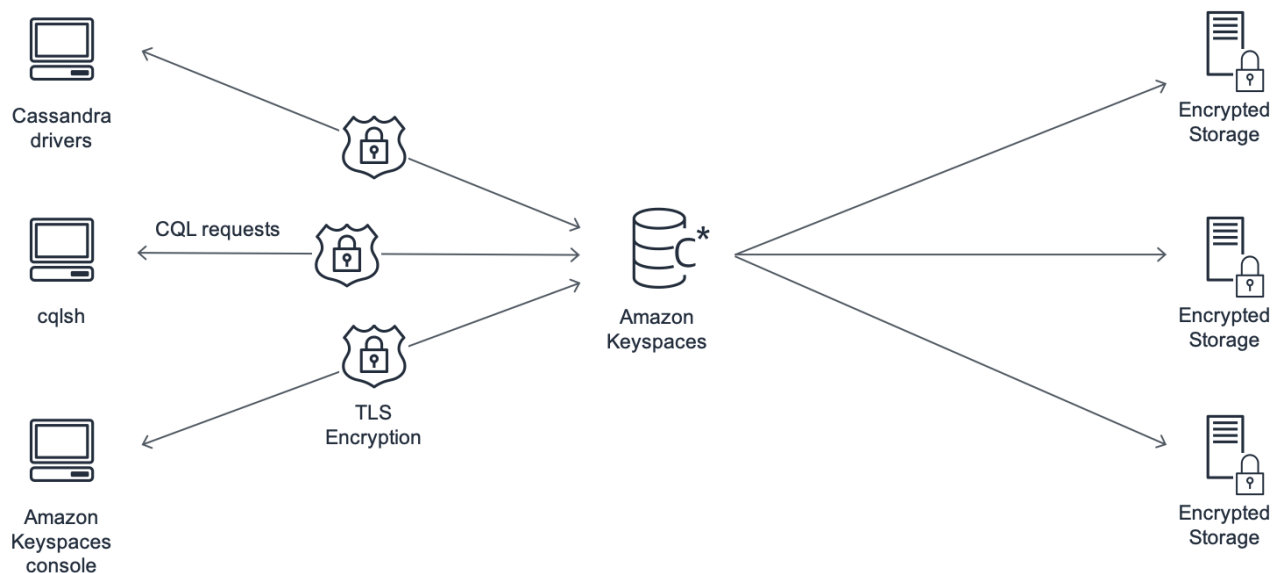
La modalità di capacità fornita ti aiuta a ottimizzare il prezzo del throughput se hai un traffico applicativo prevedibile e puoi prevedere in anticipo i requisiti di capacità della tua tabella. Con la modalità di capacità fornita, è possibile specificare il numero di letture e scritture al secondo che si

prevede che l'applicazione esegua. [È possibile aumentare e diminuire automaticamente la capacità assegnata per la tabella abilitando il ridimensionamento automatico.](#)

Puoi modificare la modalità di capacità della tabella una volta al giorno man mano che acquisisci ulteriori informazioni sui modelli di traffico del tuo carico di lavoro o se prevedi che si verifichi un forte aumento di traffico, ad esempio a causa di un evento importante che prevedi genererà molto traffico sulle tabelle. Per ulteriori informazioni sul provisioning della capacità di lettura e scrittura, consulta [the section called “Configura le modalità di capacità di lettura/scrittura”](#)

Amazon Keyspaces (per Apache Cassandra) archivia tre copie dei dati in più [zone di disponibilità per garantire durabilità e disponibilità](#) elevata. Inoltre, puoi beneficiare di un data center e di un'architettura di rete progettati per soddisfare i requisiti delle organizzazioni più sensibili alla sicurezza. La crittografia a riposo viene abilitata automaticamente quando crei una nuova tabella Amazon Keyspaces e tutte le connessioni client richiedono Transport Layer Security (TLS). Le funzionalità AWS di sicurezza aggiuntive includono [il monitoraggio](#) e [AWS Identity and Access Management](#) gli [endpoint del cloud privato virtuale \(VPC\)](#). Per una panoramica di tutte le funzionalità di sicurezza disponibili, consulta [Sicurezza](#)

Il diagramma seguente mostra l'architettura di Amazon Keyspaces.



Un programma client accede ad Amazon Keyspaces connettendosi a un endpoint predeterminato (nome host e numero di porta) ed emettendo istruzioni CQL. Per un elenco degli endpoint disponibili, consulta [the section called “Endpoint di servizio”](#)

Modello di dati Cassandra

Il modo in cui modelli i dati per il tuo business case è fondamentale per ottenere prestazioni ottimali da Amazon Keyspaces. Un modello di dati scadente può ridurre in modo significativo le prestazioni.

Anche se CQL è simile a SQL, i backend di Cassandra e i database relazionali sono molto diversi e devono essere affrontati in modo diverso. Di seguito sono riportate alcune delle questioni più importanti da considerare:

Storage

È possibile visualizzare i dati di Cassandra in tabelle, in cui ogni riga rappresenta un record e ogni colonna un campo all'interno di quel record.

Progettazione della tabella: interroga prima

Non ci sono messaggi in JOIN CQL. Pertanto, è necessario progettare le tabelle in base alla forma dei dati e al modo in cui è necessario accedervi per i casi d'uso aziendali. Ciò potrebbe comportare una denormalizzazione con dati duplicati. È necessario progettare ciascuna tabella in modo specifico per un particolare modello di accesso.

Partizioni

I dati vengono archiviati in partizioni su disco. Il numero di partizioni in cui sono archiviati i dati e il modo in cui vengono distribuiti tra le partizioni sono determinati dalla chiave di partizione. Il modo in cui si definisce la chiave di partizione può avere un impatto significativo sulle prestazioni delle query. Per le best practice, consulta [the section called “Progettazione delle chiavi di partizione”](#).

Chiave primaria

In Cassandra, i dati vengono archiviati come coppia chiave-valore. Ogni tabella Cassandra deve avere una chiave primaria, che è la chiave univoca per ogni riga della tabella. La chiave primaria è composta da una chiave di partizione richiesta e da colonne di clustering opzionali. I dati che compongono la chiave primaria devono essere univoci in tutti i record di una tabella.

- **Chiave di partizione:** la parte relativa alla chiave di partizione della chiave primaria è obbligatoria e determina in quale partizione del cluster sono archiviati i dati. La chiave di partizione può essere una singola colonna o un valore composto da due o più colonne. È consigliabile utilizzare una chiave di partizione composta se una chiave di partizione a colonna singola comporta che una singola partizione o pochissime partizioni contengano la maggior parte dei dati e quindi supportino la maggior parte delle operazioni su disco. I/O

- **Colonna di clustering:** la parte opzionale della colonna di clustering della chiave primaria determina il modo in cui i dati vengono raggruppati e ordinati all'interno di ciascuna partizione. Se includi una colonna di clustering nella chiave primaria, la colonna di clustering può avere una o più colonne. Se nella colonna di raggruppamento sono presenti più colonne, l'ordinamento è determinato dall'ordine in cui le colonne sono elencate nella colonna di raggruppamento, da sinistra a destra.

Per ulteriori informazioni sulla progettazione NoSQL e Amazon Keyspaces, consulta [the section called “Progettazione NoSQL”](#). Per ulteriori informazioni su Amazon Keyspaces e sulla modellazione dei dati, consulta [the section called “Modellazione dei dati”](#).

Accesso ad Amazon Keyspaces da un'applicazione

Amazon Keyspaces (per Apache Cassandra) implementa l'API Apache Cassandra Query Language (CQL), in modo da poter utilizzare i driver CQL e Cassandra che già utilizzi. Aggiornare l'applicazione è semplice: basta aggiornare il driver o la `cqlsh` configurazione Cassandra in modo che punti all'endpoint del servizio Amazon Keyspaces. Per ulteriori informazioni sulle credenziali richieste, consulta [the section called “Crea credenziali IAM per l'autenticazione AWS”](#).

Note

Per aiutarti a iniziare, puoi trovare esempi di end-to-end codice per la connessione ad Amazon Keyspaces utilizzando vari driver client Cassandra nell'archivio di esempi di codice Amazon Keyspaces su [GitHub](#).

Considerate il seguente programma Python, che si connette a un cluster Cassandra e interroga una tabella.

```
from cassandra.cluster import Cluster
#TLS/SSL configuration goes here

ksp = 'MyKeyspace'
tbl = 'WeatherData'

cluster = Cluster(['NNN.NNN.NNN.NNN'], port=NNNN)
session = cluster.connect(ksp)

session.execute('USE ' + ksp)
```

```
rows = session.execute('SELECT * FROM ' + tbl)
for row in rows:
    print(row)
```

Per eseguire lo stesso programma su Amazon Keyspaces, devi:

- Aggiungere l'endpoint e la porta del cluster: ad esempio, l'host può essere sostituito con un endpoint di servizio, ad esempio `cassandra.us-east-1.amazonaws.com` e il numero di porta con: `9142`
- Aggiungere la TLS/SSL configurazione: per ulteriori informazioni sull'aggiunta della TLS/SSL configurazione per connettersi ad Amazon Keyspaces utilizzando un driver Python del client Cassandra, consulta [Utilizzo di un driver client Cassandra Python per accedere ad Amazon Keyspaces a livello di codice](#)

Casi d'uso di Amazon Keyspaces

Di seguito sono riportati solo alcuni dei modi in cui è possibile utilizzare Amazon Keyspaces:

- Crea applicazioni che richiedono una bassa latenza: elabora i dati ad alta velocità per applicazioni che richiedono single-digit-millisecond latenza, come la manutenzione delle apparecchiature industriali, il monitoraggio del commercio, la gestione della flotta e l'ottimizzazione dei percorsi.
- Crea applicazioni utilizzando tecnologie open source: crea applicazioni AWS utilizzando Cassandra APIs e driver open source disponibili per un'ampia gamma di linguaggi di programmazione, come Java, Python, Ruby, Microsoft .NET, Node.js, PHP, C++, Perl e Go. Per alcuni esempi di codice, consulta [Librerie e strumenti](#).
- Sposta i carichi di lavoro di Cassandra sul cloud: la gestione autonoma delle tabelle Cassandra è dispendiosa in termini di tempo e denaro. Con Amazon Keyspaces, puoi configurare, proteggere e scalare le tabelle Cassandra Cloud AWS senza dover gestire l'infrastruttura. Per ulteriori informazioni, consulta [Gestione delle risorse serverless](#).

Cos'è Cassandra Query Language (CQL)?

Cassandra Query Language (CQL) è il linguaggio principale per comunicare con Apache Cassandra. Amazon Keyspaces (per Apache Cassandra) è compatibile con l'API CQL 3.x (retrocompatibile con la versione 2.x).

In CQL, i dati vengono archiviati in tabelle, colonne e righe. In questo senso CQL è simile a Structured Query Language (SQL). Questi sono i concetti chiave di CQL.

- Elementi CQL — Gli elementi fondamentali di CQL sono identificatori, costanti, termini e tipi di dati.
- Data Definition Language (DDL): le istruzioni DDL vengono utilizzate per gestire strutture di dati come keyspace e tabelle, che sono risorse AWS in Amazon Keyspaces. Le istruzioni DDL sono operazioni del piano di controllo in. AWS
- Data Manipulation Language (DML): le istruzioni DML vengono utilizzate per gestire i dati all'interno delle tabelle. Le istruzioni DML vengono utilizzate per selezionare, inserire, aggiornare ed eliminare i dati. Si tratta di operazioni sul piano dati in. AWS
- Funzioni integrate: Amazon Keyspaces supporta una varietà di funzioni scalari integrate che puoi utilizzare nelle istruzioni CQL.

Per ulteriori informazioni su CQL, consulta. [Riferimento al linguaggio CQL per Amazon Keyspaces \(per Apache Cassandra\)](#) Per le differenze funzionali con Apache Cassandra, vedere. [the section called “Differenze funzionali con Apache Cassandra”](#)

Per eseguire le query CQL, puoi effettuare una delle seguenti operazioni:

- Utilizzate l'editor CQL in. Console di gestione AWS
- Use AWS CloudShell e l'espansione [cqlsh](#).
- Usa un client. `cqlsh`
- Usa un driver client Cassandra con licenza Apache 2.0.

Oltre a CQL, puoi eseguire operazioni DDL (Data Definition Language) in Amazon Keyspaces utilizzando and the. AWS SDKs AWS Command Line Interface

Per ulteriori informazioni sull'utilizzo di questi metodi per accedere ad Amazon Keyspaces, consulta. [Accesso ad Amazon Keyspaces \(per Apache Cassandra\)](#)

Qual è la differenza tra Amazon Keyspaces (per Apache Cassandra) e Apache Cassandra?

[Per stabilire una connessione ad Amazon Keyspaces, puoi utilizzare un endpoint di AWS servizio pubblico o un endpoint privato utilizzando Interface VPC endpoints \(\) in Amazon AWS PrivateLink Virtual Private Cloud.](#) A seconda dell'endpoint utilizzato, Amazon Keyspaces può apparire al client in uno dei seguenti modi.

AWS connessione all'endpoint del servizio

Si tratta di una connessione stabilita su qualsiasi [endpoint pubblico](#). In questo caso, Amazon Keyspaces appare al client come un cluster Apache Cassandra 3.11.2 a nove nodi.

Interfaccia: connessione endpoint VPC

Si tratta di una connessione privata stabilita utilizzando un endpoint [VPC di interfaccia](#). In questo caso, Amazon Keyspaces appare al client come un cluster Apache Cassandra 3.11.2 a tre nodi.

Indipendentemente dal tipo di connessione e dal numero di nodi visibili al client, Amazon Keyspaces offre throughput e storage praticamente illimitati. A tale scopo, Amazon Keyspaces mappa i nodi su sistemi di bilanciamento del carico che indirizzano le query verso una delle tante partizioni di storage sottostanti. Per ulteriori informazioni sulle connessioni, consulta [the section called “Come funzionano”](#).

Amazon Keyspaces archivia i dati in partizioni. Una partizione è un'allocazione di storage per una tabella, supportata da unità a stato solido (). SSDs Amazon Keyspaces replica automaticamente i dati su più [zone di disponibilità](#) all'interno di una rete Regione AWS per garantire durabilità e disponibilità elevata. Man mano che le tue esigenze di throughput o storage aumentano, Amazon Keyspaces gestisce per te la gestione delle partizioni ed effettua automaticamente il provisioning delle partizioni aggiuntive necessarie.

Amazon Keyspaces supporta tutte le operazioni sul piano dati Cassandra di uso comune, come la creazione di keyspace e tabelle, la lettura e la scrittura di dati. Amazon Keyspaces è [serverless](#), quindi non è necessario effettuare il provisioning, applicare patch o gestire i server. Inoltre, non è necessario installare, mantenere o utilizzare software. Di conseguenza, in Amazon Keyspaces non è necessario utilizzare le operazioni dell'API del piano di controllo Cassandra per gestire le impostazioni di cluster e nodi.

Amazon Keyspaces configura automaticamente impostazioni come il fattore di replica e il livello di coerenza per offrirti disponibilità, durabilità e prestazioni elevate. single-digit-millisecond [Per una resilienza ancora maggiore e letture locali a bassa latenza, Amazon Keyspaces offre la replica in più regioni.](#)

Argomenti

- [Differenze funzionali: Amazon Keyspaces e Apache Cassandra](#)
- [Cassandra APIs, operazioni, funzioni e tipi di dati supportati](#)
- [Livelli di coerenza di lettura e scrittura supportati da Apache Cassandra e costi associati](#)

Differenze funzionali: Amazon Keyspaces e Apache Cassandra

Di seguito sono riportate le differenze funzionali tra Amazon Keyspaces e Apache Cassandra.

Argomenti

- [Apache Cassandra APIs, operazioni e tipi di dati](#)
- [Creazione ed eliminazione asincrone di spazi chiave e tabelle](#)
- [Autenticazione e autorizzazione](#)
- [Archiviazione](#)
- [Change Data Capture \(CDC\)](#)
- [Configurazione del cluster](#)
- [Connessioni](#)
- [INparola chiave](#)
- [FROZENcollezioni](#)
- [Transazioni leggere](#)
- [Sistema di bilanciamento del carico](#)
- [Paginazione](#)
- [Partizionatori](#)
- [Dichiarazioni preparate](#)
- [Eliminazione dell'intervallo](#)
- [Tabelle di sistema](#)
- [Timestamp](#)

- [Tipi UDTs definiti dall'utente \(\)](#)

Apache Cassandra APIs, operazioni e tipi di dati

Amazon Keyspaces supporta tutte le operazioni sul piano dati Cassandra di uso comune, come la creazione di keyspace e tabelle, la lettura e la scrittura di dati. Per vedere cosa è attualmente supportato, consulta [Cassandra APIs, operazioni, funzioni e tipi di dati supportati](#).

Creazione ed eliminazione asincrone di spazi chiave e tabelle

Amazon Keyspaces esegue operazioni DDL (Data Definition Language), come la creazione e l'eliminazione di keyspace, tabelle e tipi in modo asincrono. Per scoprire come monitorare lo stato di creazione delle risorse, consulta e. [the section called “Controlla lo stato di creazione del keyspace”](#) [the section called “Controlla lo stato di creazione della tabella”](#) Per un elenco delle istruzioni DDL nel riferimento al linguaggio CQL, vedere. [the section called “Istruzioni DDL”](#)

Autenticazione e autorizzazione

Amazon Keyspaces (per Apache Cassandra) utilizza AWS Identity and Access Management (IAM) per l'autenticazione e l'autorizzazione degli utenti e supporta le politiche di autorizzazione equivalenti a quelle di Apache Cassandra. Pertanto, Amazon Keyspaces non supporta i comandi di configurazione di sicurezza di Apache Cassandra.

Archiviazione

Amazon Keyspaces supporta i seguenti comandi batch:

- Batch registrato: supporta fino a 100 comandi per batch.
- Batch non registrato: supporta fino a 30 comandi per batch.

INSERTIn UPDATE un batch sono consentiti solo DELETE i comandi o.

Change Data Capture (CDC)

In Apache Cassandra, Change Data Capture (CDC) copia i segmenti di log di commit che acquisiscono gli aggiornamenti delle tabelle in una directory specificata sul nodo. Ogni record di modifica nel registro CDC può includere aggiornamenti su più keyspace e tabelle. Quando una riga viene aggiornata o eliminata, il registro CDC di Apache Cassandra mostra solo le colonne modificate.

Il registro non include lo stato precedente dei dati prima della modifica. Per comprendere tutti i dettagli delle modifiche ai dati, è necessario deserializzare questi log utilizzando strumenti Cassandra specifici.

Con gli stream CDC di Amazon Keyspaces, puoi selezionare il tipo di informazioni che lo stream CDC raccoglie per ogni riga. Per impostazione predefinita, gli stream CDC di Amazon Keyspaces acquisiscono la versione della riga prima e dopo la modifica. Amazon Keyspaces CDC fornisce record di modifiche deduplicati e ordinati a livello di tabella.

[In Amazon Keyspaces, ogni flusso CDC è una AWS risorsa con un Amazon Resource Name \(ARN\) e l'applicazione può utilizzare i flussi CDC di Amazon Keyspaces utilizzando l'API Amazon Keyspaces CDC utilizzando uno degli endpoint disponibili.](#)

Per ulteriori informazioni su Amazon Keyspaces CDC, consulta [the section called “Lavorare con i flussi CDC”](#)

Configurazione del cluster

Amazon Keyspaces è serverless, quindi non ci sono cluster, host o macchine virtuali Java () JVMs da configurare. Le impostazioni di Cassandra per la compattazione, la compressione, la memorizzazione nella cache, la raccolta dei rifiuti e il filtro bloom non sono applicabili ad Amazon Keyspaces e vengono ignorate se specificate.

Connessioni

Puoi utilizzare i driver Cassandra esistenti per comunicare con Amazon Keyspaces, ma devi configurare i driver in modo diverso. Amazon Keyspaces supporta fino a 3.000 query CQL per connessione TCP al secondo, ma non c'è limite al numero di connessioni che un driver può stabilire.

La maggior parte dei driver Cassandra open source stabilisce un pool di connessioni con Cassandra e bilancia il carico delle query su quel pool di connessioni. Amazon Keyspaces espone 9 indirizzi IP peer ai driver e il comportamento predefinito della maggior parte dei driver consiste nello stabilire una singola connessione a ciascun indirizzo IP peer. Pertanto, la velocità massima di trasmissione delle query CQL di un driver che utilizza le impostazioni predefinite è di 27.000 query CQL al secondo.

Per aumentare questo numero, si consiglia di aumentare il numero di connessioni per indirizzo IP che il driver mantiene nel proprio pool di connessioni. Ad esempio, impostando il numero massimo di connessioni per indirizzo IP su 2, la velocità effettiva massima del driver viene raddoppiata, portandola a 54.000 query CQL al secondo.

Come procedura ottimale, consigliamo di configurare i driver in modo che utilizzino 500 query CQL al secondo per connessione per ridurre il sovraccarico e migliorare la distribuzione. In questo scenario, la pianificazione di 18.000 query CQL al secondo richiede 36 connessioni. La configurazione del driver per 4 connessioni su 9 endpoint prevede 36 connessioni che eseguono 500 richieste al secondo. Per ulteriori informazioni sulle best practice per le connessioni, vedere [the section called “Connessioni”](#)

Quando ci si connette con gli endpoint VPC, potrebbero esserci meno endpoint disponibili. Ciò significa che è necessario aumentare il numero di connessioni nella configurazione del driver. Per ulteriori informazioni sulle best practice per le connessioni VPC, vedere [the section called “Connessioni endpoint VPC”](#)

INparola chiave

Amazon Keyspaces supporta la IN parola chiave nella SELECT dichiarazione. IN non è supportato con UPDATE and DELETE. Quando si utilizza la IN parola chiave nell'SELECT istruzione, i risultati della query vengono restituiti nell'ordine in cui le chiavi sono presentate nell'SELECT istruzione. In Cassandra, i risultati sono ordinati lessicograficamente.

Quando si utilizza ORDER BY, il riordino completo con impaginazione disabilitata non è supportato e i risultati vengono ordinati all'interno di una pagina. Le query Slice non sono supportate con la parola chiave. IN TOKENS non sono supportate con la parola chiave. IN Amazon Keyspaces elabora le query con la IN parola chiave creando sottoquery. Ogni sottoquery conta come connessione verso il limite di 3.000 query CQL per connessione TCP al secondo. Per ulteriori informazioni, consulta [the section called “Usa IN SELECT”](#).

FROZENcollezioni

La FROZEN parola chiave in Cassandra serializza più componenti di un tipo di dati di raccolta in un unico valore immutabile che viene trattato come un BLOB. INSERT e le UPDATE istruzioni sovrascrivono l'intera raccolta.

Amazon Keyspaces supporta per impostazione predefinita fino a 8 livelli di nidificazione per raccolte congelate. Per ulteriori informazioni, consulta [the section called “Quote di servizio Amazon Keyspaces”](#).

Amazon Keyspaces non supporta confronti di disuguaglianze che utilizzano l'intera raccolta congelata in un condizionale o in una dichiarazione. UPDATE SELECT Il comportamento delle raccolte e delle raccolte congelate è lo stesso in Amazon Keyspaces.

Quando utilizzi raccolte congelate con timestamp lato client, nel caso in cui il timestamp di un'operazione di scrittura sia lo stesso del timestamp di una colonna esistente che non è scaduta o non è stata salvata, Amazon Keyspaces non esegue confronti. Al contrario, consente al server di determinare l'autore più recente e l'autore più recente vince.

Per ulteriori informazioni sulle collezioni congelate, consulta [the section called “Tipi di raccolta”](#).

Transazioni leggere

Amazon Keyspaces (per Apache Cassandra) supporta completamente le funzionalità di confronto e impostazione su e DELETE comandi INSERTUPDATE, noti come transazioni leggere (LWTs) in Apache Cassandra. Essendo un'offerta serverless, Amazon Keyspaces (per Apache Cassandra) offre prestazioni costanti su qualsiasi scala, anche per transazioni leggere. Con Amazon Keyspaces, non vi è alcuna riduzione delle prestazioni per l'utilizzo di transazioni leggere.

Sistema di bilanciamento del carico

Le voci della `system.peers` tabella corrispondono ai sistemi di bilanciamento del carico di Amazon Keyspaces. Per ottenere i migliori risultati, ti consigliamo di utilizzare una politica di bilanciamento del carico round robin e di ottimizzare il numero di connessioni per IP in base alle esigenze dell'applicazione.

Paginazione

Amazon Keyspaces impagina i risultati in base al numero di righe lette per elaborare una richiesta, non al numero di righe restituite nel set di risultati. Di conseguenza, alcune pagine potrebbero contenere un numero di righe inferiore a quello specificato in PAGE SIZE per le query filtrate. Inoltre, Amazon Keyspaces impagina automaticamente i risultati dopo aver letto 1 MB di dati per fornire ai clienti prestazioni di lettura coerenti a una cifra in millisecondi. Per ulteriori informazioni, consulta [the section called “Impagina i risultati”](#).

Nelle tabelle con colonne statiche, sia Apache Cassandra che Amazon Keyspaces stabiliscono il valore della colonna statica della partizione all'inizio di ogni pagina in una query multipagina. Quando una tabella contiene righe di dati di grandi dimensioni, a causa del comportamento di impaginazione di Amazon Keyspaces, è più probabile che il risultato di un'operazione di lettura a intervalli restituisca più pagine per Amazon Keyspaces che per Apache Cassandra. Di conseguenza, esiste una maggiore probabilità in Amazon Keyspaces che gli aggiornamenti simultanei alla colonna statica possano far sì che il valore della colonna statica sia diverso nelle diverse pagine del set di risultati di lettura dell'intervallo.

Partizionatori

Il partizionatore predefinito in Amazon Keyspaces è compatibile con Cassandra.

`Murmur3Partitioner` Inoltre, puoi scegliere di utilizzare Amazon Keyspaces `DefaultPartitioner` o la versione compatibile con Cassandra. `RandomPartitioner`

Con Amazon Keyspaces, puoi modificare in sicurezza il partizionatore del tuo account senza dover ricaricare i dati di Amazon Keyspaces. Una volta completata la modifica della configurazione, che richiede circa 10 minuti, i client vedranno automaticamente la nuova impostazione del partizionatore alla successiva connessione. Per ulteriori informazioni, consulta [the section called “Lavorare con i partizionatori”](#).

Dichiarazioni preparate

Amazon Keyspaces supporta l'uso di istruzioni preparate per operazioni DML (Data Manipulation Language), come la lettura e la scrittura di dati. Amazon Keyspaces attualmente non supporta l'uso di istruzioni preparate per operazioni DDL (Data Definition Language), come la creazione di tabelle e keyspace. Le operazioni DDL devono essere eseguite al di fuori delle istruzioni preparate.

Eliminazione dell'intervallo

Amazon Keyspaces supporta l'eliminazione di righe nell'intervallo. Un intervallo è un insieme contiguo di righe all'interno di una partizione. Si specifica un intervallo in un'operazione DELETE utilizzando una clausola WHERE. È possibile specificare che l'intervallo sia un'intera partizione.

Inoltre, è possibile specificare un intervallo come sottoinsieme di righe contigue all'interno di una partizione utilizzando operatori relazionali (ad esempio, '>', '<') oppure includendo la chiave di partizione e omettendo una o più colonne di clustering. Con Amazon Keyspaces, puoi eliminare fino a 1.000 righe all'interno di un intervallo in un'unica operazione.

Le eliminazioni degli intervalli non sono isolate. Le eliminazioni di singole righe sono visibili ad altre operazioni mentre è in corso l'eliminazione di un intervallo.

Tabelle di sistema

Amazon Keyspaces popola le tabelle di sistema richieste dai driver Cassandra open source Apache 2.0. Le tabelle di sistema visibili a un client contengono informazioni esclusive per l'utente autenticato. Le tabelle di sistema sono completamente controllate da Amazon Keyspaces e sono di sola lettura. Per ulteriori informazioni, consulta [the section called “Spazi chiave di sistema”](#).

È richiesto l'accesso in sola lettura alle tabelle di sistema ed è possibile controllarlo con le policy di accesso IAM. Per ulteriori informazioni, consulta [the section called “Gestione dell’accesso tramite policy”](#). È necessario definire le politiche di controllo degli accessi basate su tag per le tabelle di sistema in modo diverso a seconda che si utilizzino le chiamate API AWS SDK o Cassandra Query Language (CQL) tramite i driver e gli strumenti di sviluppo di Cassandra. Per ulteriori informazioni sul controllo degli accessi basato su tag per le tabelle di sistema, consulta. [the section called “Accesso alle risorse Amazon Keyspaces basato su tag”](#)

Se accedi ad Amazon Keyspaces utilizzando gli endpoint Amazon [VPC, nella tabella vengono visualizzate le voci relative a `system.peers` ogni endpoint](#) Amazon VPC per il quale Amazon Keyspaces dispone delle autorizzazioni necessarie. Di conseguenza, il driver Cassandra potrebbe emettere un [messaggio di avviso relativo al nodo di controllo](#) stesso nella tabella. `system.peers` Puoi tranquillamente ignorare questo avviso.

Timestamp

In Amazon Keyspaces, i timestamp a livello di cella compatibili con i timestamp predefiniti in Apache Cassandra sono una funzionalità opzionale.

La `USING TIMESTAMP` clausola e la `WRITETIME` funzione sono disponibili solo quando i timestamp lato client sono attivati per una tabella. Per ulteriori informazioni sui timestamp lato client in Amazon Keyspaces, consulta. [the section called “Timestamp lato client”](#)

Tipi UDTs definiti dall'utente ()

L'operatore di disuguaglianza non è supportato UDTs in Amazon Keyspaces.

Per informazioni su come utilizzare Amazon Keyspaces, consulta. UDTs [the section called “Tipi definiti dall'utente \(\) UDTs”](#)

Per verificare quanti UDTs sono supportati per keyspace, i livelli di nidificazione supportati e altri valori e quote predefiniti correlati, consulta. UDTs [the section called “Quote e valori predefiniti per i tipi definiti dall'utente \(UDTs\) in Amazon Keyspaces”](#)

Cassandra APIs, operazioni, funzioni e tipi di dati supportati

Amazon Keyspaces (per Apache Cassandra) è compatibile con l'API Cassandra Query Language (CQL) 3.11 (retrocompatibile con la versione 2.x).

Amazon Keyspaces supporta tutte le operazioni sul piano dati Cassandra di uso comune, come la creazione di keyspace e tabelle, la lettura e la scrittura di dati.

Le seguenti sezioni elencano le funzionalità supportate.

Argomenti

- [Supporto per l'API Cassandra](#)
- [Supporto dell'API Cassandra Control Plane](#)
- [Supporto API Cassandra Data Plane](#)
- [Supporto della funzione Cassandra](#)
- [Supporto per i tipi di dati Cassandra](#)

Supporto per l'API Cassandra

Operazione API	Supportata
CREATE KEYSPACE	Sì
ALTER KEYSPACE	Sì
DROP KEYSPACE	Sì
CREATE TABLE	Sì
ALTER TABLE	Sì
DROP TABLE	Sì
CREATE INDEX	No
DROP INDEX	No
UNLOGGED BATCH	Sì
LOGGED BATCH	Sì
SELECT	Sì
INSERT	Sì

Operazione API	Supportata
DELETE	Sì
UPDATE	Sì
USE	Sì
CREATE TYPE	Sì
ALTER TYPE	No
DROP TYPE	Sì
CREATE TRIGGER	No
DROP TRIGGER	No
CREATE FUNCTION	No
DROP FUNCTION	No
CREATE AGGREGATE	No
DROP AGGREGATE	No
CREATE MATERIALIZED VIEW	No
ALTER MATERIALIZED VIEW	No
DROP MATERIALIZED VIEW	No
TRUNCATE	No

Supporto dell'API Cassandra Control Plane

Poiché Amazon Keyspaces è gestito, non sono necessarie le operazioni dell'API del piano di controllo Cassandra per gestire le impostazioni di cluster e nodi. Di conseguenza, le seguenti funzionalità di Cassandra non sono applicabili.

Funzionalità	Motivo
Interruttore Durable Writings	Tutte le scritture sono durevoli
Leggi le impostazioni di riparazione	Non applicabile
secondi di grazia GC	Non applicabile
Impostazioni del filtro Bloom	Non applicabile
Impostazioni di compattazione	Non applicabile
Compression settings (Impostazioni di compressione)	Non applicabile
Impostazioni di memorizzazione nella cache	Non applicabile
Impostazioni di sicurezza	Sostituito da IAM

Supporto API Cassandra Data Plane

Funzionalità	Supportato
Supporto JSON per le istruzioni SELECT e INSERT	Sì
Colonne statiche	Sì
Time to Live (TTL)	Sì

Supporto della funzione Cassandra

Per ulteriori informazioni sulle funzioni supportate, vedere [the section called “Funzioni integrate”](#).

Funzione	Supportata
Funzioni Aggregate	No

Funzione	Supportata
Blobconversione	Sì
Cast	Sì
Funzioni Datetime	Sì
funzioni di conversione del tempo	Sì
Funzioni TimeUuid	Sì
Token	Sì
User defined functions (UDF)	No
Uuid	Sì

Supporto per i tipi di dati Cassandra

La tabella seguente elenca i tipi di dati Apache Cassandra supportati in Amazon Keyspaces. Per ulteriori informazioni sui tipi di dati in Amazon Keyspaces, consulta [the section called “Tipi di dati”](#)

Tipo di dati	Supportata	
ascii	Sì	
bigint	Sì	
blob	Sì	
boolean	Sì	
counter	Sì	
date	Sì	
decimal	Sì	
double	Sì	

Tipo di dati	Supportata	
float	Sì	
frozen	Sì	
inet	Sì	
int	Sì	
list	Sì	
map	Sì	
set	Sì	
smallint	Sì	
text	Sì	
time	Sì	
timestamp	Sì	
timeuuid	Sì	
tinyint	Sì	
tuple	Sì	
user-defined types (UDTs)	Sì	
uuid	Sì	
varchar	Sì	
varint	Sì	

Livelli di coerenza di lettura e scrittura supportati da Apache Cassandra e costi associati

Gli argomenti di questa sezione descrivono quali livelli di coerenza di Apache Cassandra sono supportati per le operazioni di lettura e scrittura in Amazon Keyspaces (per Apache Cassandra).

Argomenti

- [Livelli di coerenza di scrittura](#)
- [Leggi i livelli di coerenza](#)
- [Livelli di coerenza non supportati](#)

Livelli di coerenza di scrittura

Amazon Keyspaces replica tutte le operazioni di scrittura tre volte su più zone di disponibilità per garantire durabilità e disponibilità elevata. Le scritture vengono archiviate in modo duraturo prima di essere riconosciute utilizzando il livello di coerenza. LOCAL_QUORUM Per ogni 1 KB di scrittura, viene fatturata 1 unità di capacità di scrittura (WCU) per le tabelle che utilizzano la modalità di capacità fornita o 1 unità di richiesta di scrittura (WRU) per le tabelle che utilizzano la modalità on demand.

È possibile utilizzare il `cqlsh` codice seguente per impostare la coerenza di tutte le query nella sessione corrente. LOCAL_QUORUM

```
CONSISTENCY LOCAL_QUORUM;
```

Per configurare il livello di coerenza a livello di codice, è possibile impostare la coerenza con i driver client Cassandra appropriati. Ad esempio, i driver Java della versione 4.x consentono di impostare il livello di coerenza nel `app config` file come illustrato di seguito.

```
basic.request.consistency = LOCAL_QUORUM
```

Se utilizzate un driver Java Cassandra versione 3.x, potete specificare il livello di coerenza per la sessione aggiungendolo `.withQueryOptions(new QueryOptions().setConsistencyLevel(ConsistencyLevel.LOCAL_QUORUM))` come mostrato nel seguente esempio di codice.

```
Session session = Cluster.builder()
```

```
.addContactPoint(endPoint)
.withPort(portNumber)
.withAuthProvider(new SigV4AuthProvider("us-east-1"))
.withSSL()
.withQueryOptions(new
QueryOptions().setConsistencyLevel(ConsistencyLevel.LOCAL_QUORUM)
.build())
.connect();
```

Per configurare il livello di coerenza per operazioni di scrittura specifiche, potete definire la coerenza quando chiamate `QueryBuilder.insertInto` con un `setConsistencyLevel` argomento quando utilizzate il driver Java.

Leggi i livelli di coerenza

Amazon Keyspaces supporta tre livelli di coerenza di lettura: `ONELOCAL_ONE`, e. `LOCAL_QUORUM`. Durante una `LOCAL_QUORUM` lettura, Amazon Keyspaces restituisce una risposta che riflette gli aggiornamenti più recenti di tutte le precedenti operazioni di scrittura riuscite. Utilizzando il livello di coerenza `ONE` o `LOCAL_ONE` può migliorare le prestazioni e la disponibilità delle richieste di lettura, ma la risposta potrebbe non riflettere i risultati di una scrittura completata di recente.

Per ogni utilizzo `ONE` o `LOCAL_ONE` coerenza di lettura di 4 KB, vengono fatturate 0,5 unità di capacità di lettura (RCUs) per le tabelle che utilizzano la modalità di capacità fornita o 0,5 unità di richiesta di lettura (RRUs) per le tabelle che utilizzano la modalità su richiesta. Per ogni 4 KB di lettura con `LOCAL_QUORUM` coerenza, viene fatturata 1 unità di capacità di lettura (RCU) per le tabelle che utilizzano la modalità di capacità fornita o 1 unità di richiesta di lettura (RRU) per le tabelle che utilizzano la modalità on demand.

Fatturazione basata sulla coerenza di lettura e sulla capacità di lettura (modalità di throughput) per tabella per ogni 4 KB di letture

Livello di coerenza	Assegnata	On demand
ONE	0,5 RCUs	0,5 RRUs
LOCAL_ONE	0,5 RCUs	0,5 RRUs
LOCAL_QUORUM	1 TAZZA	1 TRU

Per specificare una consistenza diversa per le operazioni di lettura, chiamate `QueryBuilder.select` con un `setConsistencyLevel` argomento quando utilizzate il driver Java.

Livelli di coerenza non supportati

I seguenti livelli di coerenza non sono supportati da Amazon Keyspaces e comporteranno delle eccezioni.

Livelli di coerenza non supportati

Apache Cassandra	Amazon Keyspaces
EACH_QUORUM	Non supportato
QUORUM	Non supportato
ALL	Non supportato
TWO	Non supportato
THREE	Non supportato
ANY	Non supportato
SERIAL	Non supportato
LOCAL_SERIAL	Non supportato

Migrazione ad Amazon Keyspaces (per Apache Cassandra)

La migrazione ad Amazon Keyspaces (per Apache Cassandra) offre una serie di vantaggi interessanti per aziende e organizzazioni. Ecco alcuni vantaggi chiave che rendono Amazon Keyspaces una scelta interessante per la migrazione.

- **Scalabilità:** Amazon Keyspaces è progettato per gestire carichi di lavoro enormi e scalare senza problemi per soddisfare volumi e traffico di dati in crescita. Con Cassandra tradizionale, la scalabilità non viene eseguita su richiesta e richiede la pianificazione per i picchi futuri. Con Amazon Keyspaces, puoi scalare facilmente le tabelle verso l'alto o verso il basso in base alla domanda, assicurando che le tue applicazioni siano in grado di gestire picchi improvvisi di traffico senza compromettere le prestazioni.
- **Prestazioni:** Amazon Keyspaces offre un accesso ai dati a bassa latenza, consentendo alle applicazioni di recuperare ed elaborare i dati con una velocità eccezionale. La sua architettura distribuita garantisce che le operazioni di lettura e scrittura siano distribuite su più nodi, offrendo tempi di risposta coerenti a una cifra di millisecondi anche a frequenze di richieste elevate.
- **Completamente gestito:** Amazon Keyspaces è un servizio completamente gestito fornito da AWS. Ciò significa che AWS gestisce gli aspetti operativi della gestione del database, tra cui il provisioning, la configurazione, l'applicazione di patch, i backup e la scalabilità. Ciò consente di concentrarsi maggiormente sullo sviluppo delle applicazioni e meno sulle attività di amministrazione del database.
- **Architettura serverless:** Amazon Keyspaces è serverless. Paghi solo per la capacità consumata e non è richiesto alcun provisioning anticipato della capacità. Non hai server da gestire o istanze tra cui scegliere. Questo pay-per-request modello offre efficienza in termini di costi e spese operative minime, in quanto si paga solo per le risorse consumate senza la necessità di fornire e monitorare la capacità.
- **Flessibilità NoSQL con schema:** Amazon Keyspaces segue un modello di dati NoSQL, offrendo flessibilità nella progettazione dello schema. Con Amazon Keyspaces, puoi archiviare dati strutturati, semistrutturati e non strutturati, rendendoli ideali per la gestione di tipi di dati diversi e in evoluzione. Inoltre, Amazon Keyspaces esegue la convalida dello schema in scrittura, consentendo un'evoluzione centralizzata del modello di dati. Questa flessibilità consente cicli di sviluppo più rapidi e un adattamento più semplice alle mutevoli esigenze aziendali.
- **Disponibilità e durabilità elevate:** Amazon Keyspaces replica i dati su più [zone di disponibilità](#) all'interno di un'unica area Regione AWS, garantendo elevata disponibilità e durabilità dei dati. Gestisce automaticamente la replica, il failover e il ripristino, riducendo al minimo il rischio di

perdita dei dati o interruzioni del servizio. Amazon Keyspaces offre uno SLA di disponibilità fino al 99,999%. [Per una resilienza ancora maggiore e letture locali a bassa latenza, Amazon Keyspaces offre la replica in più regioni.](#)

- Sicurezza e conformità: Amazon Keyspaces si integra AWS Identity and Access Management per un controllo granulare degli accessi. Fornisce la crittografia a riposo e in transito, contribuendo a migliorare la sicurezza dei dati. Amazon Keyspaces è stato valutato da revisori di terze parti per quanto riguarda la sicurezza e la conformità con programmi specifici, tra cui HIPAA, PCI DSS e SOC, che consentono di soddisfare i requisiti normativi. Per ulteriori informazioni, consulta [the section called “Convalida della conformità”](#).
- Integrazione con l' AWS ecosistema: come parte dell' AWS ecosistema, Amazon Keyspaces si integra perfettamente con altri Servizi AWS, ad esempio AWS CloudFormation e Amazon CloudWatch e AWS CloudTrail. Questa integrazione consente di creare architetture serverless, sfruttare l'infrastruttura come codice e creare applicazioni basate sui dati in tempo reale. Per ulteriori informazioni, consulta [Monitoraggio di Amazon Keyspaces](#).

Argomenti

- [Crea un piano di migrazione per la migrazione da Apache Cassandra ad Amazon Keyspaces](#)
- [Come scegliere lo strumento giusto per il caricamento in blocco o la migrazione di dati su Amazon Keyspaces](#)

Crea un piano di migrazione per la migrazione da Apache Cassandra ad Amazon Keyspaces

Per una migrazione di successo da Apache Cassandra ad Amazon Keyspaces, consigliamo una revisione dei concetti e delle best practice di migrazione applicabili, nonché un confronto tra le opzioni disponibili.

Questo argomento descrive come funziona il processo di migrazione introducendo diversi concetti chiave e gli strumenti e le tecniche a tua disposizione. È possibile valutare le diverse strategie di migrazione per selezionare quella che meglio soddisfa i propri requisiti.

Argomenti

- [Compatibilità funzionale](#)
- [Stima dei prezzi di Amazon Keyspaces](#)
- [Scegli una strategia di migrazione](#)

- [Migrazione online verso Amazon Keyspaces: strategie e best practice](#)
- [Processo di migrazione offline: da Apache Cassandra ad Amazon Keyspaces](#)
- [Utilizzo di una soluzione di migrazione ibrida: da Apache Cassandra ad Amazon Keyspaces](#)

Compatibilità funzionale

Valuta attentamente le differenze funzionali tra Apache Cassandra e Amazon Keyspaces prima della migrazione. Amazon Keyspaces supporta tutte le operazioni sul piano dati Cassandra di uso comune, come la creazione di keyspace e tabelle, la lettura e la scrittura di dati.

Tuttavia ci sono alcuni Cassandra APIs che Amazon Keyspaces non supporta. Per ulteriori informazioni sui supporti APIs, consulta [the section called “Cassandra APIs, operazioni, funzioni e tipi di dati supportati”](#) Per una panoramica di tutte le differenze funzionali tra Amazon Keyspaces e Apache Cassandra, consulta [the section called “Differenze funzionali con Apache Cassandra”](#)

Per confrontare Cassandra APIs e lo schema che stai utilizzando con le funzionalità supportate in Amazon Keyspaces, puoi eseguire uno script di compatibilità disponibile nel toolkit Amazon Keyspaces su [GitHub](#)

Come usare lo script di compatibilità

1. Scarica lo script Python di compatibilità da [GitHub](#) e spostalo in una posizione che abbia accesso al tuo cluster Apache Cassandra esistente.
2. Lo script di compatibilità utilizza parametri simili a CQLSH. Cerca `--host` e `--port` inserisci l'indirizzo IP e la porta che usi per connetterti ed eseguire query su uno dei nodi Cassandra del tuo cluster.

Se il tuo cluster Cassandra utilizza l'autenticazione, devi anche fornire `-username` e `-password`. Per eseguire lo script di compatibilità, puoi usare il seguente comando.

```
python toolkit-compat-tool.py --host hostname or IP -u "username" -p "password" --port native transport port
```

Stima dei prezzi di Amazon Keyspaces

Questa sezione fornisce una panoramica delle informazioni che devi raccogliere dalle tabelle Apache Cassandra per calcolare il costo stimato per Amazon Keyspaces. Ciascuna tabella richiede tipi di dati diversi, deve supportare diverse query CQL e mantiene un traffico di lettura/scrittura distintivo.

Pensare ai tuoi requisiti in base alle tabelle è in linea con le modalità di isolamento delle risorse a livello di tabella di Amazon Keyspaces [e](#) capacità di velocità effettiva di lettura/scrittura. Con Amazon Keyspaces, puoi definire la capacità di lettura/scrittura e le politiche di [scalabilità automatica per le tabelle in modo indipendente](#).

La comprensione dei requisiti delle tabelle ti aiuta a dare priorità alle tabelle per la migrazione in base a funzionalità, costi e sforzi di migrazione.

Raccogli le seguenti metriche della tabella Cassandra prima di una migrazione. Queste informazioni aiutano a stimare il costo del carico di lavoro su Amazon Keyspaces.

- Nome della tabella: il nome del keyspace completo e del nome della tabella.
- Descrizione: una descrizione della tabella, ad esempio come viene utilizzata o che tipo di dati vi sono memorizzati.
- Letture medie al secondo: il numero medio di letture a livello di coordinate sulla tabella in un ampio intervallo di tempo.
- Scritture medie al secondo: il numero medio di scritture a livello di coordinate sulla tabella in un ampio intervallo di tempo.
- Dimensione media delle righe in byte: la dimensione media delle righe in byte.
- Dimensione di archiviazione in GBs: la dimensione di archiviazione non elaborata per una tabella.
- Ripartizione della coerenza di lettura: la percentuale di letture che utilizzano una coerenza finale (LOCAL_ONEoONE) rispetto a una consistenza forte (). LOCAL_QUORUM

Questa tabella mostra un esempio delle informazioni sulle tabelle che è necessario raccogliere durante la pianificazione di una migrazione.

Nome tabella	Descrizione	Lectture medie al secondo	Scritture medie al secondo	Dimensione media delle righe in byte	Dimensioni di archiviazione in GBs	Leggi la ripartizione della coerenza
mykeyspace.mytable	Utilizzato per memorizzare la cronologia del carrello	10.000	5.000	2.200	2.000	100% LOCAL_ONE
mykeyspace.mytable2	Utilizzato per memorizzare le informazioni più recenti sul profilo	20.000	1.000	850	1.000	25% LOCAL_QUORUM 75% LOCAL_ONE

Come raccogliere le metriche delle tabelle

Questa sezione fornisce istruzioni dettagliate su come raccogliere le metriche di tabella necessarie dal cluster Cassandra esistente. Queste metriche includono la dimensione delle righe, la dimensione della tabella e le richieste di lettura/scrittura al secondo (RPS). Ti consentono di valutare i requisiti di capacità di throughput per una tabella Amazon Keyspaces e stimare i prezzi.

Come raccogliere i parametri della tabella nella tabella dei sorgenti di Cassandra

1. Determina la dimensione delle righe

La dimensione delle righe è importante per determinare la capacità di lettura e l'utilizzo della capacità di scrittura in Amazon Keyspaces. Il diagramma seguente mostra la distribuzione tipica dei dati su un intervallo di token Cassandra.



È possibile utilizzare uno script di campionamento delle dimensioni delle righe disponibile su [GitHub](#) per raccogliere le metriche delle dimensioni delle righe per ogni tabella del cluster Cassandra.

Lo script esporta i dati della tabella da Apache Cassandra utilizzando `cqlsh` e `awk` calcolando la deviazione minima, massima, media e standard della dimensione delle righe su un set di dati di tabella di esempio configurabile. Il campionatore della dimensione delle righe passa gli argomenti `acqlsh`, quindi è possibile utilizzare gli stessi parametri per connettersi e leggere dal cluster Cassandra.

La seguente dichiarazione ne è un esempio.

```
./row-size-sampler.sh 10.22.33.44 9142 \\  
-u "username" -p "password" --ssl
```

Per ulteriori informazioni su come viene calcolata la dimensione delle righe in Amazon Keyspaces, consulta [the section called “Stima della dimensione delle righe”](#)

2. Determina le dimensioni della tabella

Con Amazon Keyspaces, non è necessario effettuare il provisioning dello storage in anticipo. Amazon Keyspaces monitora continuamente le dimensioni fatturabili delle tabelle per determinare i costi di archiviazione. Lo storage viene fatturato per GB al mese. La dimensione della tabella Amazon Keyspaces si basa sulla dimensione raw (non compressa) di una singola replica.

Per monitorare la dimensione della tabella in Amazon Keyspaces, puoi utilizzare la metrica `BillableTableSizeInBytes`, che viene visualizzata per ogni tabella in. Console di gestione AWS

Per stimare la dimensione fatturabile della tabella Amazon Keyspaces, puoi utilizzare uno di questi due metodi:

- Usa la dimensione media delle righe e moltiplica per il numero o le righe.

Puoi stimare la dimensione della tabella Amazon Keyspaces moltiplicando la dimensione media delle righe per il numero di righe della tabella sorgente di Cassandra. Utilizza lo script di esempio relativo alla dimensione delle righe della sezione precedente per acquisire la dimensione media delle righe. Per acquisire il conteggio delle righe, puoi utilizzare strumenti come `dsbulk count` determinare il numero totale di righe nella tabella di origine.

- Usa `nodetool` per raccogliere i metadati della tabella.

`Nodetool` è uno strumento amministrativo fornito nella distribuzione Apache Cassandra che fornisce informazioni sullo stato del processo Cassandra e restituisce i metadati della tabella. Puoi utilizzarli `nodetool` per campionare i metadati sulla dimensione della tabella e quindi estrapolare la dimensione della tabella in Amazon Keyspaces.

Il comando da usare è. `nodetool tablestats` `Tablestats` restituisce le dimensioni e il rapporto di compressione della tabella. La dimensione della tabella viene memorizzata come riferimento `tablelivespace` per la tabella ed è possibile dividerla per. `compression ratio` Quindi moltiplica questo valore di dimensione per il numero di nodi. Infine dividi per il fattore di replica (in genere tre).

Questa è la formula completa per il calcolo che puoi usare per valutare le dimensioni della tabella.

```
((tablelivespace / compression ratio) * (total number of nodes)) / (replication factor)
```

Supponiamo che il cluster Cassandra abbia 12 nodi. L'esecuzione del `nodetool tablestats` comando restituisce un valore `tablelivespace` di 200 GB e uno `compression ratio` di 0,5. Il keyspace ha un fattore di replica pari a tre.

Ecco come appare il calcolo per questo esempio.

```
(200 GB / 0.5) * (12 nodes) / (replication factor of 3)
= 4,800 GB / 3
= 1,600 GB is the table size estimate for Amazon
Keyspaces
```

3. Registra il numero di letture e scritture

Per determinare i requisiti di capacità e scalabilità per le tabelle Amazon Keyspaces, acquisisci la frequenza di richiesta di lettura e scrittura delle tabelle Cassandra prima della migrazione.

Amazon Keyspaces è serverless e paghi solo per ciò che usi. In generale, il prezzo del throughput di lettura/scrittura in Amazon Keyspaces si basa sul numero e sulla dimensione delle richieste.

Esistono due modalità di capacità in Amazon Keyspaces:

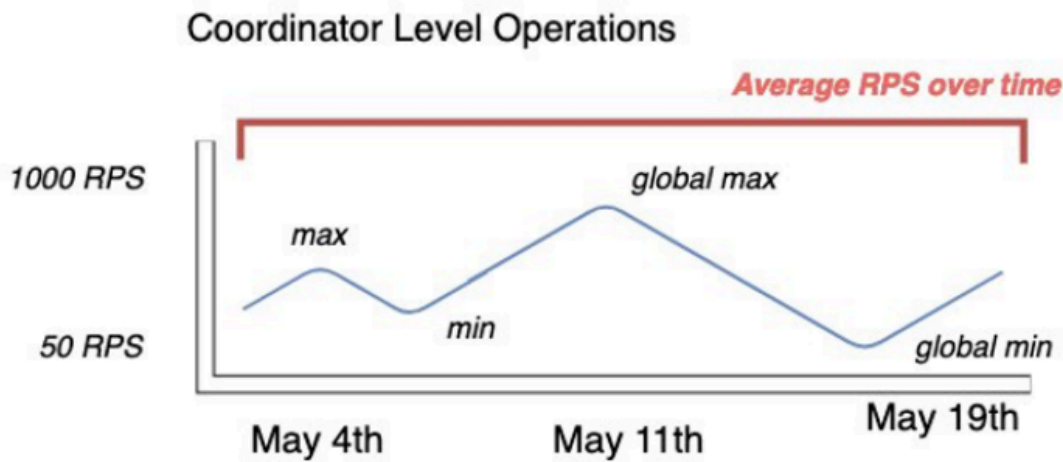
- [On-demand](#): si tratta di un'opzione di fatturazione flessibile in grado di soddisfare migliaia di richieste al secondo senza la necessità di pianificare la capacità. Offre pay-per-request prezzi per le richieste di lettura e scrittura in modo da pagare solo per ciò che si utilizza.
- [Provisioned](#): se si sceglie la modalità Provisioned Throughput Capacity, si specifica il numero di letture e scritture al secondo necessarie per l'applicazione. Questo ti aiuta a gestire l'utilizzo di Amazon Keyspaces in modo che rimanga pari o inferiore a una frequenza di richieste definita per mantenere la prevedibilità.

La modalità Provisioned offre [la scalabilità automatica](#) per regolare automaticamente la tariffa assegnata per aumentare o diminuire per migliorare l'efficienza operativa. Per ulteriori informazioni sulla gestione delle risorse senza server, vedere. [Gestione delle risorse serverless](#)

Poiché la capacità di throughput di lettura e scrittura viene fornita separatamente in Amazon Keyspaces, è necessario misurare la frequenza di richieste di lettura e scrittura nelle tabelle esistenti in modo indipendente.

Per raccogliere i parametri di utilizzo più accurati dal cluster Cassandra esistente, acquisisci la media delle richieste al secondo (RPS) per le operazioni di lettura e scrittura a livello di coordinatore per un periodo di tempo prolungato per una tabella aggregata su tutti i nodi di un singolo data center.

L'acquisizione dell'RPS medio in un periodo di almeno diverse settimane consente di rilevare picchi e valli nei modelli di traffico, come illustrato nel diagramma seguente.



Sono disponibili due opzioni per determinare la frequenza delle richieste di lettura e scrittura della tabella Cassandra.

- Usa il monitoraggio Cassandra esistente

Puoi utilizzare le metriche mostrate nella tabella seguente per osservare le richieste di lettura e scrittura. Tieni presente che i nomi delle metriche possono cambiare in base allo strumento di monitoraggio che stai utilizzando.

Dimensione	Metrica Cassandra JMX
Scrive	<code>org.apache.cassandra.metrics:type=ClientRequest,scope=Write,name=Latency#Count</code>
Legge	<code>org.apache.cassandra.metrics:type=ClientRequest,scope=Read,name=Latency#Count</code>

- Utilizzo dell'`nodetool`

Usa `nodetool tablestats` e `nodetool info` acquisisce le operazioni medie di lettura e scrittura dalla tabella. `tablestats` restituisce il conteggio totale di letture e scritture dal momento in cui il nodo è stato avviato. `nodetool info` fornisce il tempo di attività di un nodo in secondi.

Per ottenere la media delle letture e delle scritture al secondo, dividi il conteggio delle letture e delle scritture per il tempo di attività del nodo, espresso in secondi. Quindi, per le letture si divide per il livello di coerenza e per le scritture si divide per il fattore di replica. Questi calcoli sono espressi nelle seguenti formule.

Formula per la media delle letture al secondo:

```
((number of reads * number of nodes in cluster) / read consistency quorum
(2)) / uptime
```

Formula per la media delle scritture al secondo:

```
((number of writes * number of nodes in cluster) / replication factor of 3) /
uptime
```

Supponiamo di avere un cluster a 12 nodi attivo da 4 settimane. `nodetool info` restituisce 2.419.200 secondi di uptime e `nodetool tablestats` restituisce 1 miliardo di scritture e 2 miliardi di letture. Questo esempio comporterebbe il seguente calcolo.

```
((2 billion reads * 12 in cluster) / read consistency quorum (2)) / 2,419,200
seconds
= 12 billion reads / 2,419,200 seconds
= 4,960 read request per second

((1 billion writes * 12 in cluster) / replication
factor of 3) / 2,419,200 seconds
= 4 billion writes / 2,419,200 seconds
= 1,653 write request per second
```

4. Determinare l'utilizzo della capacità della tabella

Per stimare l'utilizzo medio della capacità, inizia con i tassi di richiesta medi e la dimensione media delle righe della tabella dei sorgenti di Cassandra.

Amazon Keyspaces utilizza unità di capacità di lettura (RCUs) e unità di capacità di scrittura (WCUs) per misurare la capacità di throughput assegnata per le letture e le scritture per le tabelle. Per questa stima utilizziamo queste unità per calcolare le esigenze di capacità di lettura e scrittura della nuova tabella Amazon Keyspaces dopo la migrazione.

Più avanti in questo argomento vedremo in che modo la scelta tra la modalità di capacità fornita e quella con capacità su richiesta influisce sulla fatturazione. Tuttavia, per la stima dell'utilizzo della capacità in questo esempio, supponiamo che la tabella sia in modalità provisioning.

- **Letture:** una RCU rappresenta una LOCAL_QUORUM o due richieste di LOCAL_ONE lettura per una riga di dimensioni fino a 4 KB. Se è necessario leggere una riga di dimensioni superiori a 4 KB, l'operazione di lettura utilizza elementi aggiuntivi. RCUs Il numero totale di elementi RCUs richiesti dipende dalla dimensione della riga e dal fatto che si desideri utilizzare LOCAL_QUORUM o LOCAL_ONE leggere la coerenza.

Ad esempio, per leggere una riga da 8 KB sono necessarie 2 unità che RCUs utilizzano la coerenza di LOCAL_QUORUM lettura e 1 RCU se si sceglie la coerenza di LOCAL_ONE lettura.

- **Scritture:** una WCU rappresenta una scrittura per una riga di dimensioni fino a 1 KB. Tutte le scritture utilizzano LOCAL_QUORUM la coerenza e non sono previsti costi aggiuntivi per l'utilizzo di transazioni leggere (LWTs).

Il numero totale di WCUs richieste dipende dalla dimensione della riga. Se è necessario scrivere una riga più grande di 1 KB, l'operazione di scrittura utilizza elementi aggiuntivi WCUs. Ad esempio, se la dimensione della riga è di 2 KB, ne occorrono 2 WCUs per eseguire una richiesta di scrittura.

La formula seguente può essere utilizzata per stimare la quantità RCUs e WCUs.

- La capacità di lettura in RCUs entrata può essere determinata moltiplicando le letture al secondo per il numero di righe lette per lettura moltiplicato per la dimensione media delle righe divisa per 4 KB e arrotondata al numero intero più vicino.
- La capacità di scrittura WCUs può essere determinata moltiplicando il numero di richieste per la dimensione media delle righe divisa per 1 KB e arrotondata al numero intero più vicino.

Ciò è espresso nelle seguenti formule.

```
Read requests per second * ROUNDUP((Average Row Size)/4096 per unit) = RCUs per second
```

```
Write requests per second * ROUNDUP(Average Row Size/1024 per unit) = WCUs per second
```

Ad esempio, se stai eseguendo 4.960 richieste di lettura con una dimensione di riga di 2,5 KB sulla tua tabella Cassandra, ne occorrono 4.960 in Amazon Keyspaces. RCUs Se attualmente stai eseguendo 1.653 richieste di scrittura al secondo con una dimensione di riga di 2,5 KB sulla tua tabella Cassandra, ne occorrono 4.959 al WCUs secondo in Amazon Keyspaces.

Questo esempio è espresso nelle seguenti formule.

```
4,960 read requests per second * ROUNDUP( 2.5KB /4KB bytes per unit)
= 4,960 read requests per second * 1 RCU
= 4,960 RCUs
```

```
1,653 write requests per second * ROUNDUP(2.5KB/1KB per unit)
= 1,653 requests per second * 3 WCUs
= 4,959 WCUs
```

L'utilizzo eventual consistency consente di risparmiare fino alla metà della capacità di throughput per ogni richiesta di lettura. Ogni lettura coerente alla fine può consumare fino a 8 KB. È possibile calcolare eventuali letture coerenti moltiplicando il calcolo precedente per 0,5, come illustrato nella formula seguente.

```
4,960 read requests per second * ROUNDUP( 2.5KB /4KB per unit) * .5
= 2,480 read request per second * 1 RCU
= 2,480 RCUs
```

5. Calcola la stima mensile dei prezzi per Amazon Keyspaces

Per stimare la fatturazione mensile della tabella in base alla velocità effettiva di lettura/scrittura, puoi calcolare i prezzi per la modalità on-demand e quella per la modalità provisioning utilizzando diverse formule e confrontare le opzioni disponibili per la tabella.

Modalità di provisioning: il consumo di capacità di lettura e scrittura viene fatturato in base a una tariffa oraria basata sulle unità di capacità al secondo. Innanzitutto, dividi tale percentuale per 0,7

per rappresentare l'obiettivo di utilizzo predefinito del 70% per la scalabilità automatica. Quindi moltiplica per 30 giorni di calendario, 24 ore al giorno e tariffazione regionale.

Questo calcolo è riepilogato nelle seguenti formule.

```
(read capacity per second / .7) * 24 hours * 30 days * regional rate
      (write capacity per second / .7) * 24 hours * 30 days * regional
      rate
```

Modalità su richiesta: la capacità di lettura e scrittura viene fatturata in base a una tariffa per richiesta. Innanzitutto, moltiplica la frequenza delle richieste per 30 giorni di calendario e 24 ore al giorno. Quindi dividi per un milione di unità di richiesta. Infine, moltiplicare per la tariffa regionale.

Questo calcolo è riassunto nelle seguenti formule.

```
((read capacity per second * 30 * 24 * 60 * 60) / 1 Million read request units) *
      regional rate
      ((write capacity per second * 30 * 24 * 60 * 60) / 1 Million write
      request units) * regional rate
```

Scegli una strategia di migrazione

Puoi scegliere tra le seguenti strategie di migrazione durante la migrazione da Apache Cassandra ad Amazon Keyspaces:

- **Online:** si tratta di una migrazione in tempo reale che utilizza la doppia scrittura per iniziare a scrivere nuovi dati contemporaneamente su Amazon Keyspaces e sul cluster Cassandra. Questo tipo di migrazione è consigliato per le applicazioni che non richiedono tempi di inattività durante la migrazione e la coerenza tra lettura e scrittura.

Per ulteriori informazioni su come pianificare e implementare una strategia di migrazione online, consulta [the section called “Migrazione online”](#).

- **Offline:** questa tecnica di migrazione prevede la copia di un set di dati da Cassandra ad Amazon Keyspaces durante una finestra di inattività. La migrazione offline può semplificare il processo di migrazione, poiché non richiede modifiche all'applicazione o la risoluzione dei conflitti tra dati storici e nuove scritture.

Per ulteriori informazioni su come pianificare una migrazione offline, consulta [the section called “Migrazione offline”](#).

- Ibrido: questa tecnica di migrazione consente di replicare le modifiche su Amazon Keyspaces quasi in tempo reale, ma senza coerenza tra lettura e scrittura.

Per ulteriori informazioni su come pianificare una migrazione ibrida, consulta [the section called “Migrazione ibrida”](#)

Dopo aver esaminato le tecniche di migrazione e le best practice illustrate in questo argomento, è possibile inserire le opzioni disponibili in un albero decisionale per progettare una strategia di migrazione basata sui requisiti e sulle risorse disponibili.

Migrazione online verso Amazon Keyspaces: strategie e best practice

Se devi mantenere la disponibilità delle applicazioni durante una migrazione da Apache Cassandra ad Amazon Keyspaces, puoi preparare una strategia di migrazione online personalizzata implementando i componenti chiave discussi in questo argomento. Seguendo queste best practice per le migrazioni online, puoi garantire che la disponibilità e la read-after-write coerenza delle applicazioni siano mantenute durante l'intero processo di migrazione, riducendo al minimo l'impatto sugli utenti.

Quando si progetta una strategia di migrazione online da Apache Cassandra ad Amazon Keyspaces, è necessario considerare i seguenti passaggi chiave.

1. Scrittura di nuovi dati

- Proxy ZDM Dual Write per la migrazione di Amazon Keyspaces: utilizza lo ZDM Dual Write Proxy disponibile su [Github per eseguire la migrazione senza downtime da Apache Cassandra](#) ad Amazon Keyspaces. Il proxy ZDM esegue due scritture senza la necessità di rifattorizzare le applicazioni esistenti ed esegue due letture per la convalida delle query.
- Scritture doppie delle applicazioni: è possibile implementare la doppia scrittura nell'applicazione utilizzando le librerie e i driver client Cassandra esistenti. Designate un database come leader e l'altro come seguace. Gli errori di scrittura nel database dei follower vengono registrati in una [coda di lettere morte \(DLQ\)](#) per l'analisi.
- Doppia scrittura a livello di messaggistica: in alternativa, puoi configurare la tua piattaforma di messaggistica esistente per inviare scritture sia a Cassandra che ad Amazon Keyspaces utilizzando un consumatore aggiuntivo. Ciò crea alla fine viste coerenti su entrambi i database.

2. Migrazione dei dati storici

- Copiare dati storici: puoi migrare i dati storici da Cassandra ad Amazon Keyspaces utilizzando AWS Glue i nostri script di estrazione, trasformazione e caricamento (ETL) personalizzati. Gestisci la risoluzione dei conflitti tra due scritture e carichi di massa utilizzando tecniche come transazioni leggere o timestamp.
- Utilizzo Time-To-Live (TTL): per periodi di conservazione dei dati più brevi, puoi utilizzare TTL sia in Cassandra che in Amazon Keyspaces per evitare di caricare dati storici non necessari. Man mano che i vecchi dati scadono in Cassandra e i nuovi dati vengono scritti tramite doppia scrittura, Amazon Keyspaces alla fine recupera il ritardo.

3. Convalida dei dati

- Doppie letture: implementa letture doppie da entrambi i database Cassandra (primario) e Amazon Keyspaces (secondario), confrontando i risultati in modo asincrono. Le differenze vengono registrate o inviate a un DLQ.
- Letture di esempio: utilizzate le funzioni λ per campionare e confrontare periodicamente i dati su entrambi i sistemi, registrando eventuali discrepanze su un DLQ.

4. Migrazione dell'applicazione

- Strategia blue-green: cambia la tua applicazione per trattare Amazon Keyspaces come principale e Cassandra come archivio dati secondario in un unico passaggio. Monitora le prestazioni ed esegui il rollback in caso di problemi.
- Implementazione Canary: distribuisce gradualmente la migrazione prima a un sottoinsieme di utenti, aumentando in modo incrementale il traffico verso Amazon Keyspaces come principale fino alla completa migrazione.

5. Smantellamento di Cassandra

Una volta completata la migrazione dell'applicazione ad Amazon Keyspaces e convalidata la coerenza dei dati, puoi pianificare la disattivazione del cluster Cassandra in base alle politiche di conservazione dei dati.

Pianificando una strategia di migrazione online con questi componenti, puoi passare senza problemi al servizio Amazon Keyspaces completamente gestito con tempi di inattività o interruzioni minimi. Le seguenti sezioni approfondiscono ogni componente in modo più dettagliato.

Argomenti

- [Scrittura di nuovi dati durante una migrazione online](#)
- [Caricamento di dati storici durante una migrazione online](#)

- [Convalida della coerenza dei dati durante una migrazione online](#)
- [Migrazione dell'applicazione durante una migrazione online](#)
- [Disattivazione di Cassandra dopo una migrazione online](#)

Scrittura di nuovi dati durante una migrazione online

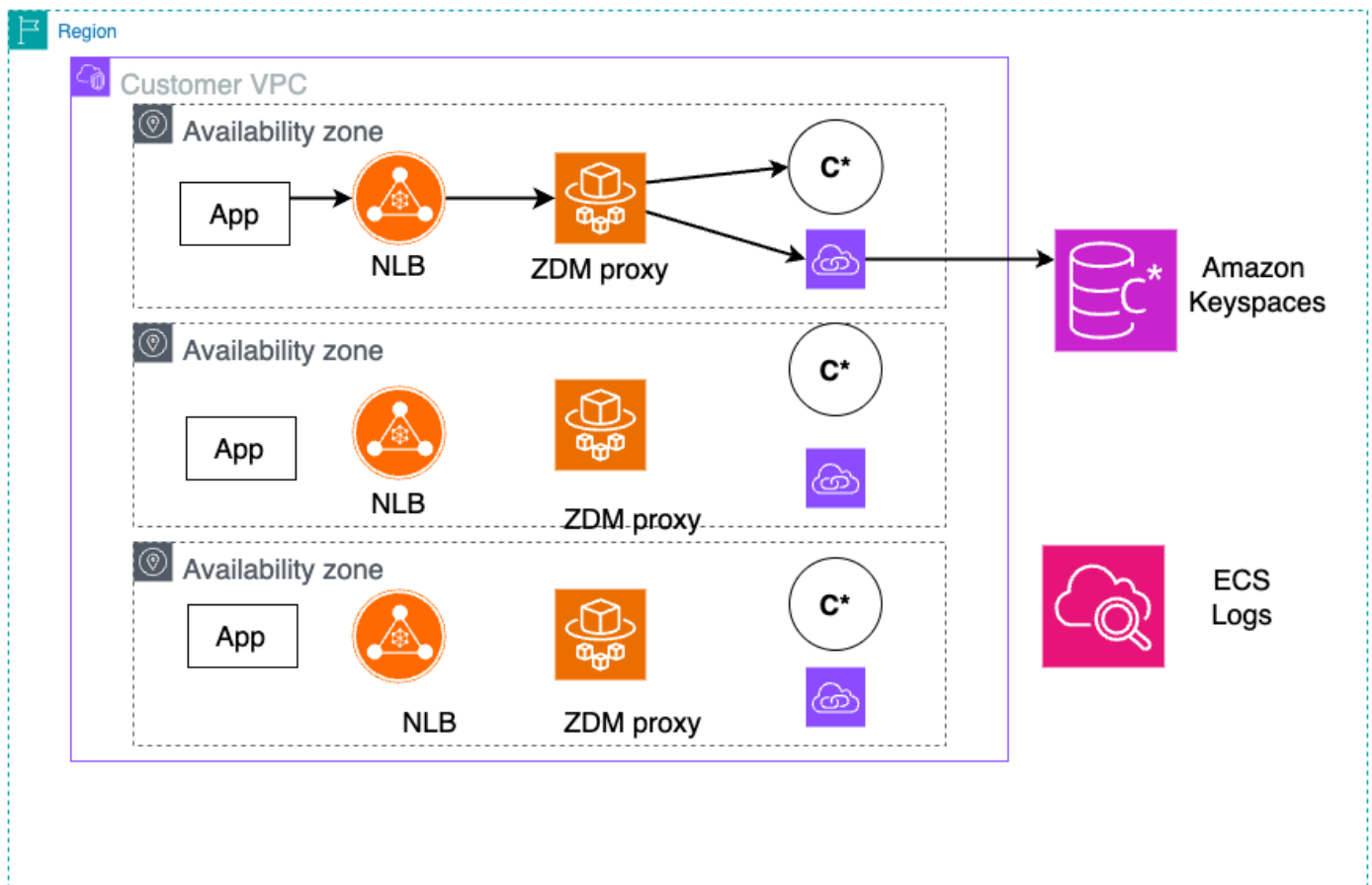
Il primo passo di un piano di migrazione online consiste nell'assicurare che tutti i nuovi dati scritti dall'applicazione siano archiviati in entrambi i database, nel cluster Cassandra esistente e in Amazon Keyspaces. L'obiettivo è fornire una visione coerente tra i due archivi di dati. È possibile farlo applicando tutte le nuove scritture a entrambi i database. Per implementare la doppia scrittura, considera una delle tre opzioni seguenti.

- Proxy ZDM Dual Write per la migrazione di Amazon Keyspaces: utilizzando il proxy ZDM per Amazon Keyspaces disponibile [su Github, puoi migrare i carichi di lavoro di Apache Cassandra su Amazon Keyspaces](#) senza tempi di inattività delle applicazioni. Questa soluzione avanzata implementa le migliori pratiche ed estende le funzionalità ufficiali di ZDM Proxy. AWS
 - Esegui migrazioni online tra Apache Cassandra e Amazon Keyspaces.
 - Scrivi dati su entrambe le tabelle di origine e di destinazione contemporaneamente senza rifattorizzare le applicazioni.
 - Convalida le query tramite operazioni di doppia lettura.

La soluzione offre i seguenti miglioramenti con cui lavorare e AWS Amazon Keyspaces.

- Distribuzione di container: utilizza un'immagine Docker preconfigurata da Amazon Elastic Container Registry (Amazon ECR) per distribuzioni accessibili tramite VPC.
- Infrastruttura come codice: esegui la distribuzione utilizzando modelli per la configurazione automatica su AWS CloudFormation AWS Fargate
- Compatibilità con Amazon Keyspaces: accedi alle tabelle di sistema con adattamenti personalizzati per Amazon Keyspaces.

La soluzione viene eseguita su Amazon ECS con Fargate, fornendo scalabilità serverless in base alle richieste del carico di lavoro. Un sistema di bilanciamento del carico di rete distribuisce il traffico delle applicazioni in entrata tra più attività Amazon ECS per un'elevata disponibilità.



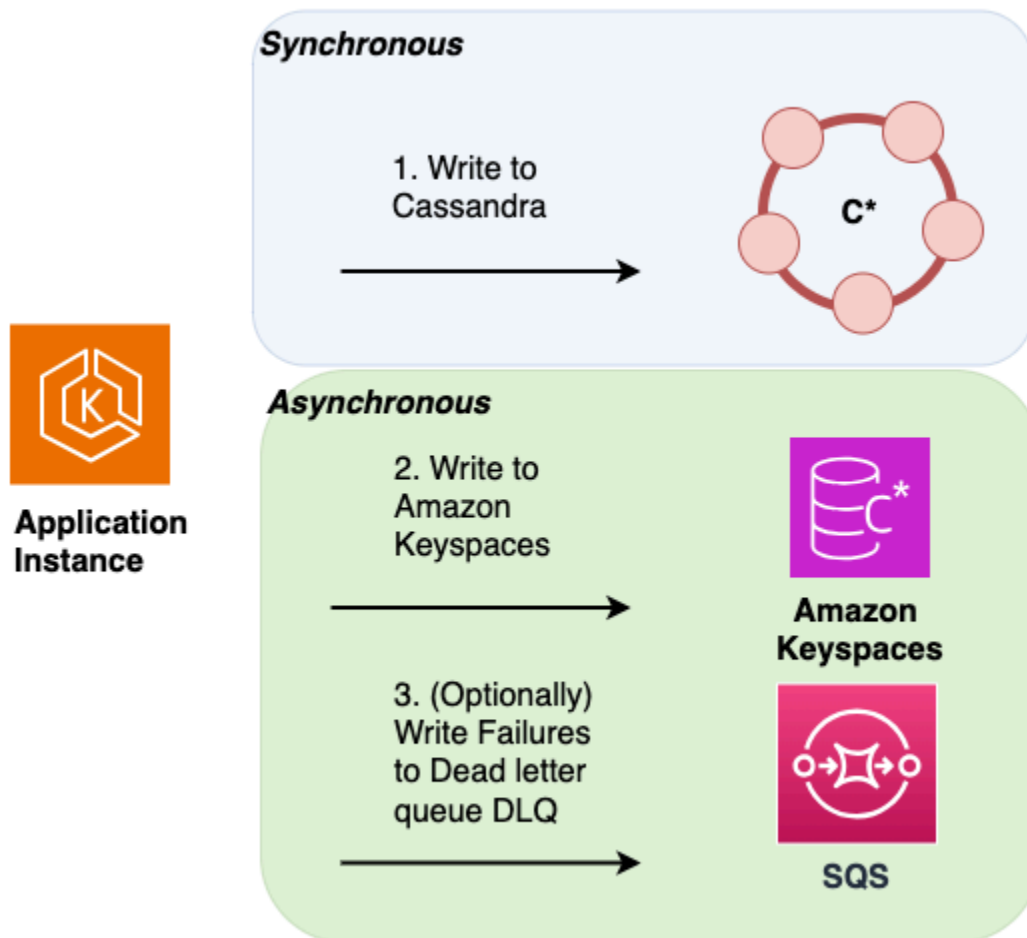
- Scritture doppie delle applicazioni: è possibile implementare la doppia scrittura con modifiche minime al codice dell'applicazione sfruttando le librerie e i driver client Cassandra esistenti. È possibile implementare la doppia scrittura nell'applicazione esistente o creare un nuovo livello nell'architettura per gestire le doppie scritture. Per ulteriori informazioni e per un case study di un cliente che mostra come sono state implementate le doppie scritture in un'applicazione esistente, consulta il [case study sulla migrazione di Cassandra](#).

Quando si implementano le doppie scritture, è possibile designare un database come leader e l'altro database come seguace. Ciò consente di continuare a scrivere sul database di origine o leader originale senza che errori di scrittura del database follower o di destinazione interrompano il percorso critico dell'applicazione.

Invece di riprovare le scritture non riuscite al follower, puoi utilizzare Amazon Simple Queue Service per registrare le scritture non riuscite in una [coda di lettere morte](#) (DLQ). Il DLQ consente di analizzare le scritture non riuscite al follower e determinare il motivo per cui l'elaborazione non è riuscita nel database di destinazione.

Per un'implementazione in doppia scrittura più sofisticata, puoi seguire le AWS migliori pratiche per progettare una sequenza di transazioni locali utilizzando il modello [saga](#). Un modello a catena garantisce che, se una transazione fallisce, la saga esegua transazioni di compensazione per ripristinare le modifiche al database apportate dalle transazioni precedenti.

Quando si utilizzano le doppie scritture per una migrazione online, è possibile configurare le doppie scritture seguendo lo schema saga in modo che ogni scrittura sia una transazione locale per garantire operazioni atomiche su database eterogenei. [Per ulteriori informazioni sulla progettazione di applicazioni distribuite utilizzando i modelli di progettazione consigliati per il Cloud AWS, consulta Modelli di progettazione, architetture e implementazioni del cloud.](#)



- Doppia scrittura a livello di messaggistica: invece di implementare la doppia scrittura a livello di applicazione, puoi utilizzare il livello di messaggistica esistente per eseguire due scritture su Cassandra e Amazon Keyspaces.

A tale scopo, puoi configurare un utente aggiuntivo sulla tua piattaforma di messaggistica per inviare scritture a entrambi gli archivi di dati. Questo approccio offre una semplice strategia low-code che utilizza il livello di messaggistica per creare due visualizzazioni su entrambi i database che alla fine sono coerenti.

Caricamento di dati storici durante una migrazione online

Dopo aver implementato la doppia scrittura per garantire che i nuovi dati vengano scritti in entrambi gli archivi di dati in tempo reale, il passaggio successivo del piano di migrazione consiste nel valutare la quantità di dati storici da copiare o caricare in blocco da Cassandra ad Amazon Keyspaces. Ciò garantisce che sia i nuovi dati che i dati storici siano disponibili nel nuovo database Amazon Keyspaces prima della migrazione dell'applicazione.

A seconda dei requisiti di conservazione dei dati, ad esempio della quantità di dati storici da conservare in base alle politiche dell'organizzazione, puoi prendere in considerazione una delle due opzioni seguenti.

- Caricamento in blocco di dati storici: la migrazione dei dati storici dalla distribuzione esistente di Cassandra ad Amazon Keyspaces può essere ottenuta attraverso varie tecniche, ad esempio utilizzando AWS Glue o script personalizzati per estrarre, trasformare e caricare (ETL) i dati. Per ulteriori informazioni sull'utilizzo per caricare dati storici, AWS Glue consulta [the section called “Migrazione offline”](#)

Quando si pianifica il caricamento in blocco di dati storici, è necessario considerare come risolvere i conflitti che possono verificarsi quando nuove scritture tentano di aggiornare gli stessi dati in fase di caricamento. Si prevede che il caricamento in blocco alla fine sia coerente, il che significa che alla fine i dati raggiungeranno tutti i nodi.

Se si verifica un aggiornamento degli stessi dati contemporaneamente a causa di una nuova scrittura, devi assicurarti che non vengano sovrascritti dal caricamento dei dati storici. Per garantire la conservazione degli aggiornamenti più recenti dei dati anche durante l'importazione in blocco, è necessario aggiungere la risoluzione dei conflitti negli script di caricamento in blocco o nella logica dell'applicazione per le scritture doppie.

Ad esempio, è possibile utilizzare [the section called “Transazioni leggere”](#) (LWT) per confrontare e impostare le operazioni. A tale scopo, è possibile aggiungere un campo aggiuntivo al modello di dati che rappresenti l'ora o lo stato della modifica.

Inoltre, Amazon Keyspaces supporta la funzione timestamp di CassandraWRITETIME. Puoi utilizzare i timestamp lato client di Amazon Keyspaces per conservare i timestamp del database di origine e implementare la risoluzione dei conflitti. last-writer-wins Per ulteriori informazioni, consulta [the section called “Timestamp lato client”](#).

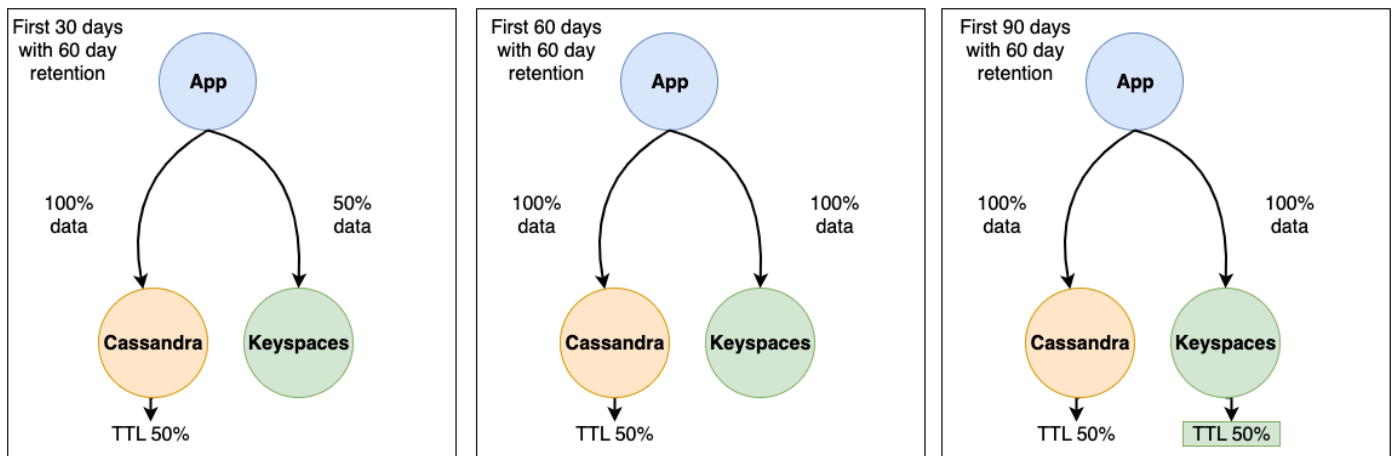
- Utilizzo di Time-to-Live (TTL): per periodi di conservazione dei dati inferiori a 30, 60 o 90 giorni, puoi utilizzare TTL in Cassandra e Amazon Keyspaces durante la migrazione per evitare di caricare dati storici non necessari su Amazon Keyspaces. TTL consente di impostare un periodo di tempo dopo il quale i dati vengono rimossi automaticamente dal database.

Durante la fase di migrazione, invece di copiare i dati storici su Amazon Keyspaces, puoi configurare le impostazioni TTL per far scadere automaticamente i dati storici nel vecchio sistema (Cassandra) applicando solo le nuove scritture ad Amazon Keyspaces utilizzando il metodo dual-write. Nel tempo e con i vecchi dati che scadono continuamente nel cluster Cassandra e i nuovi dati scritti utilizzando il metodo dual-write, Amazon Keyspaces recupera automaticamente il ritardo e contiene gli stessi dati di Cassandra.

Questo approccio può ridurre in modo significativo la quantità di dati da migrare, garantendo un processo di migrazione più efficiente e semplificato. È possibile prendere in considerazione questo approccio quando si ha a che fare con set di dati di grandi dimensioni con requisiti di conservazione dei dati diversi. Per ulteriori informazioni sul TTL, vedere. [the section called “Fai scadere i dati con Time to Live”](#)

Considera il seguente esempio di migrazione da Cassandra ad Amazon Keyspaces utilizzando la scadenza dei dati TTL. In questo esempio impostiamo il TTL per entrambi i database su 60 giorni e mostriamo come procede il processo di migrazione in un periodo di 90 giorni. Entrambi i database ricevono gli stessi dati appena scritti durante questo periodo utilizzando il metodo di doppia scrittura. Esamineremo tre diverse fasi della migrazione, ogni fase dura 30 giorni.

Il funzionamento del processo di migrazione per ciascuna fase è illustrato nelle immagini seguenti.



1. Dopo i primi 30 giorni, il cluster Cassandra e Amazon Keyspaces hanno ricevuto nuove scritture. Il cluster Cassandra contiene anche dati storici che non hanno ancora raggiunto i 60 giorni di conservazione, che rappresentano il 50% dei dati del cluster.

I dati più vecchi di 60 giorni vengono eliminati automaticamente nel cluster Cassandra tramite TTL. A questo punto Amazon Keyspaces contiene il 50% dei dati archiviati nel cluster Cassandra, che è composto dalle nuove scritture meno i dati storici.

2. Dopo 60 giorni, sia il cluster Cassandra che Amazon Keyspaces contengono gli stessi dati scritti negli ultimi 60 giorni.
3. Entro 90 giorni, sia Cassandra che Amazon Keyspaces contengono gli stessi dati e i dati in scadenza hanno la stessa frequenza.

Questo esempio illustra come evitare la fase di caricamento dei dati storici utilizzando TTL con una data di scadenza impostata su 60 giorni.

Convalida della coerenza dei dati durante una migrazione online

La fase successiva del processo di migrazione online è la convalida dei dati. Le doppie scritture aggiungono nuovi dati al tuo database Amazon Keyspaces e hai completato la migrazione dei dati storici utilizzando il caricamento in blocco o la scadenza dei dati con TTL.

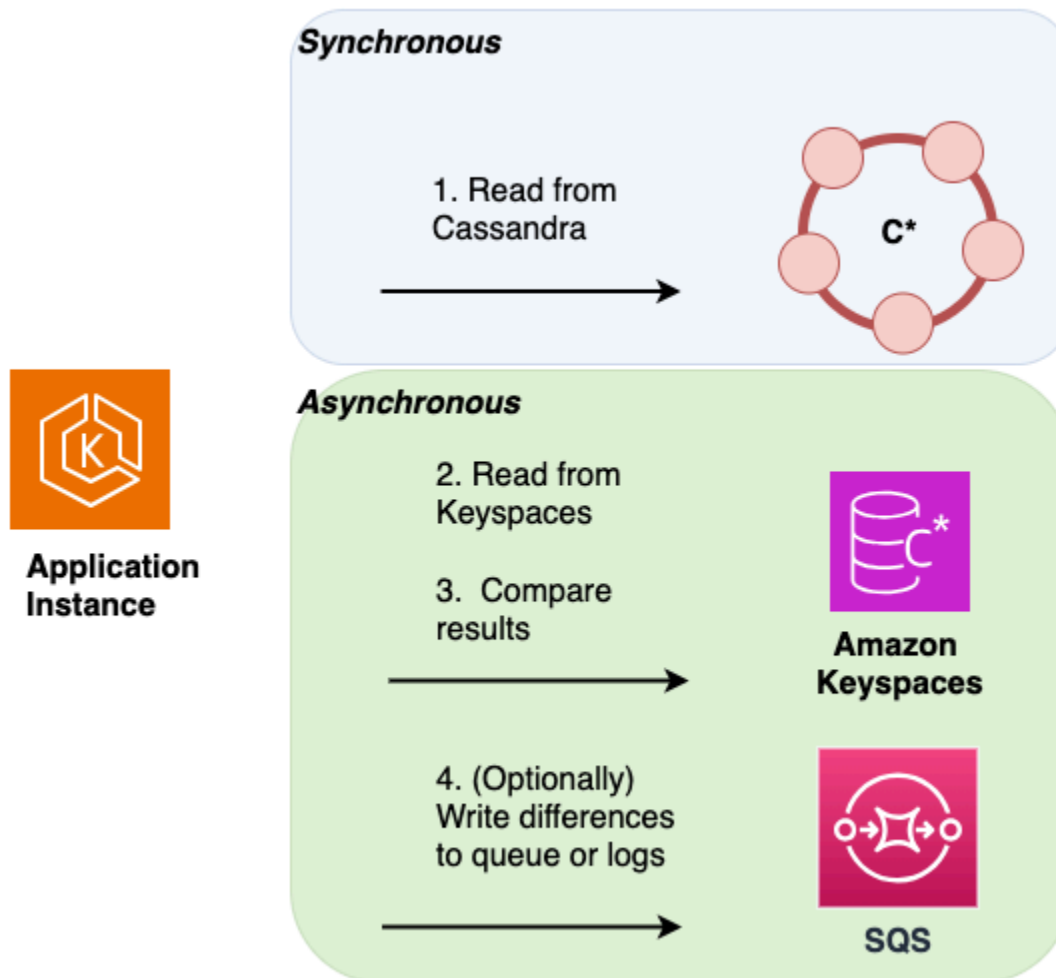
Ora puoi utilizzare la fase di convalida per confermare che entrambi gli archivi dati contengano effettivamente gli stessi dati e restituiscano gli stessi risultati di lettura. Puoi scegliere una delle due opzioni seguenti per verificare che entrambi i database contengano dati identici.

- Doppie letture: per verificare che sia il database di origine che quello di destinazione contengano lo stesso set di dati appena scritti e storici, puoi implementare letture doppie. A tale scopo, leggi sia

il database principale Cassandra che quello secondario di Amazon Keyspaces in modo analogo al metodo dual write e confronta i risultati in modo asincrono.

I risultati del database primario vengono restituiti al client e i risultati del database secondario vengono utilizzati per la convalida rispetto al set di risultati primario. Le differenze rilevate possono essere registrate o inviate a una [coda di lettere morte \(DLQ\)](#) per una successiva riconciliazione.

Nel diagramma seguente, l'applicazione esegue una lettura sincrona da Cassandra, che è l'archivio dati primario) e una lettura asincrona da Amazon Keyspaces, che è l'archivio dati secondario.

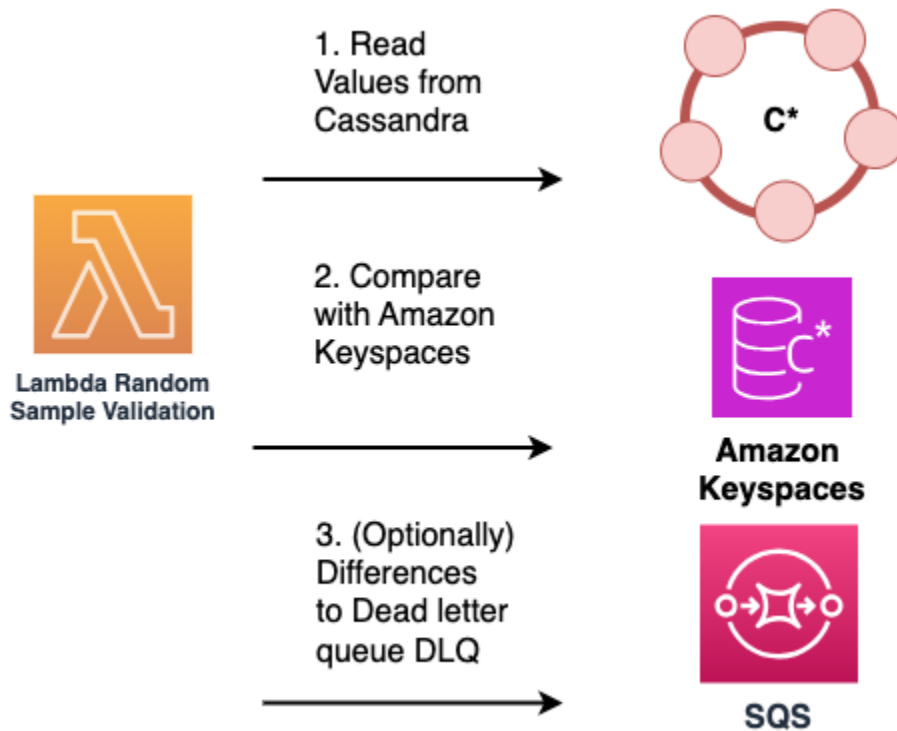


- Letture di esempio: una soluzione alternativa che non richiede modifiche al codice dell'applicazione consiste nell'utilizzare AWS Lambda le funzioni per campionare periodicamente e in modo casuale i dati sia dal cluster Cassandra di origine che dal database Amazon Keyspaces di destinazione.

Queste funzioni Lambda possono essere configurate per essere eseguite a intervalli regolari. La funzione Lambda recupera un sottoinsieme casuale di dati sia dal sistema di origine che da quello di destinazione, quindi esegue un confronto tra i dati campionati. Eventuali discrepanze o

discrepanze tra i due set di dati possono essere registrate e inviate a una coda di [lettere morte \(DLQ\) dedicata per una successiva riconciliazione](#).

Questo processo è illustrato nello schema seguente.



Migrazione dell'applicazione durante una migrazione online

Nella quarta fase di una migrazione online, stai migrando la tua applicazione e passando ad Amazon Keyspaces come archivio dati principale. Ciò significa che passi l'applicazione alla lettura e alla scrittura direttamente da e verso Amazon Keyspaces. Per garantire interruzioni minime agli utenti, questo deve essere un processo ben pianificato e coordinato.

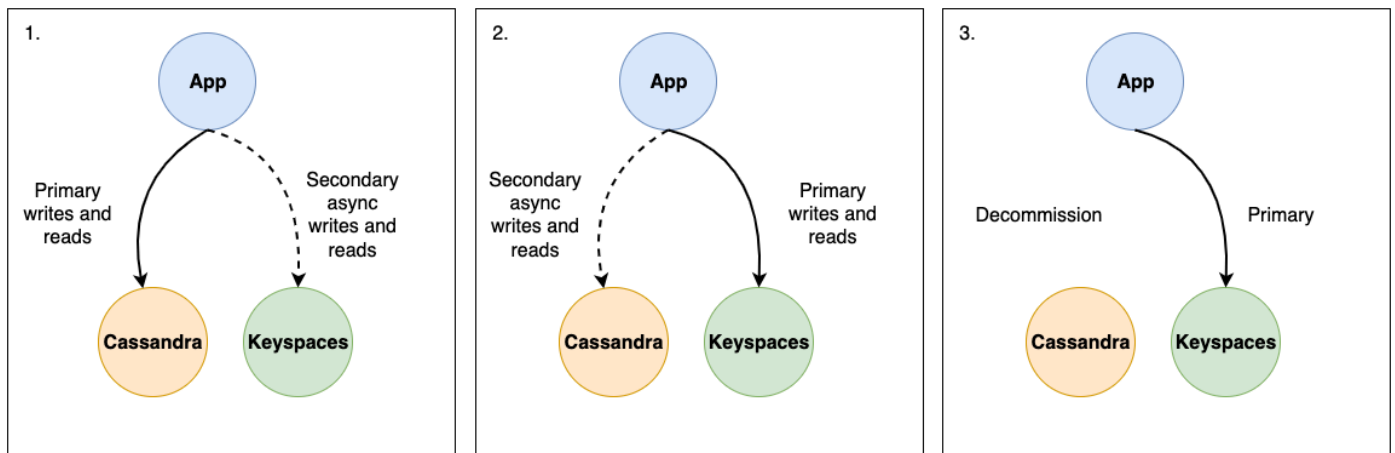
Sono disponibili due diverse soluzioni consigliate per la migrazione delle applicazioni, la strategia blue green cut over e la canary cut over strategy. Le sezioni seguenti descrivono queste strategie in modo più dettagliato.

- **Strategia blue green**: utilizzando questo approccio, puoi cambiare la tua applicazione per trattare Amazon Keyspaces come data store primario e Cassandra come data store secondario in un unico passaggio. Puoi farlo utilizzando un flag di AWS AppConfig funzionalità per controllare la selezione degli archivi dati primari e secondari nell'istanza dell'applicazione. Per ulteriori informazioni sui feature flag, consultate [Creazione di un profilo di configurazione dei feature flag in AWS AppConfig](#).

Dopo aver reso Amazon Keyspaces l'archivio dati principale, monitora il comportamento e le prestazioni dell'applicazione, assicurandoti che Amazon Keyspaces soddisfi i tuoi requisiti e che la migrazione abbia esito positivo.

Ad esempio, se hai implementato la doppia lettura per la tua applicazione, durante la fase di migrazione dell'applicazione trasferisci le letture primarie da Cassandra ad Amazon Keyspaces e le letture secondarie da Amazon Keyspaces a Cassandra. Dopo la transizione, continuerai a monitorare e confrontare i risultati come descritto nella sezione sulla [convalida dei dati per garantire la coerenza tra entrambi i database prima della disattivazione di Cassandra](#).

Se rilevi problemi, puoi tornare rapidamente allo stato precedente tornando a Cassandra come archivio dati principale. Puoi procedere alla fase di smantellamento della migrazione solo se Amazon Keyspaces soddisfa tutte le tue esigenze come data store primario.



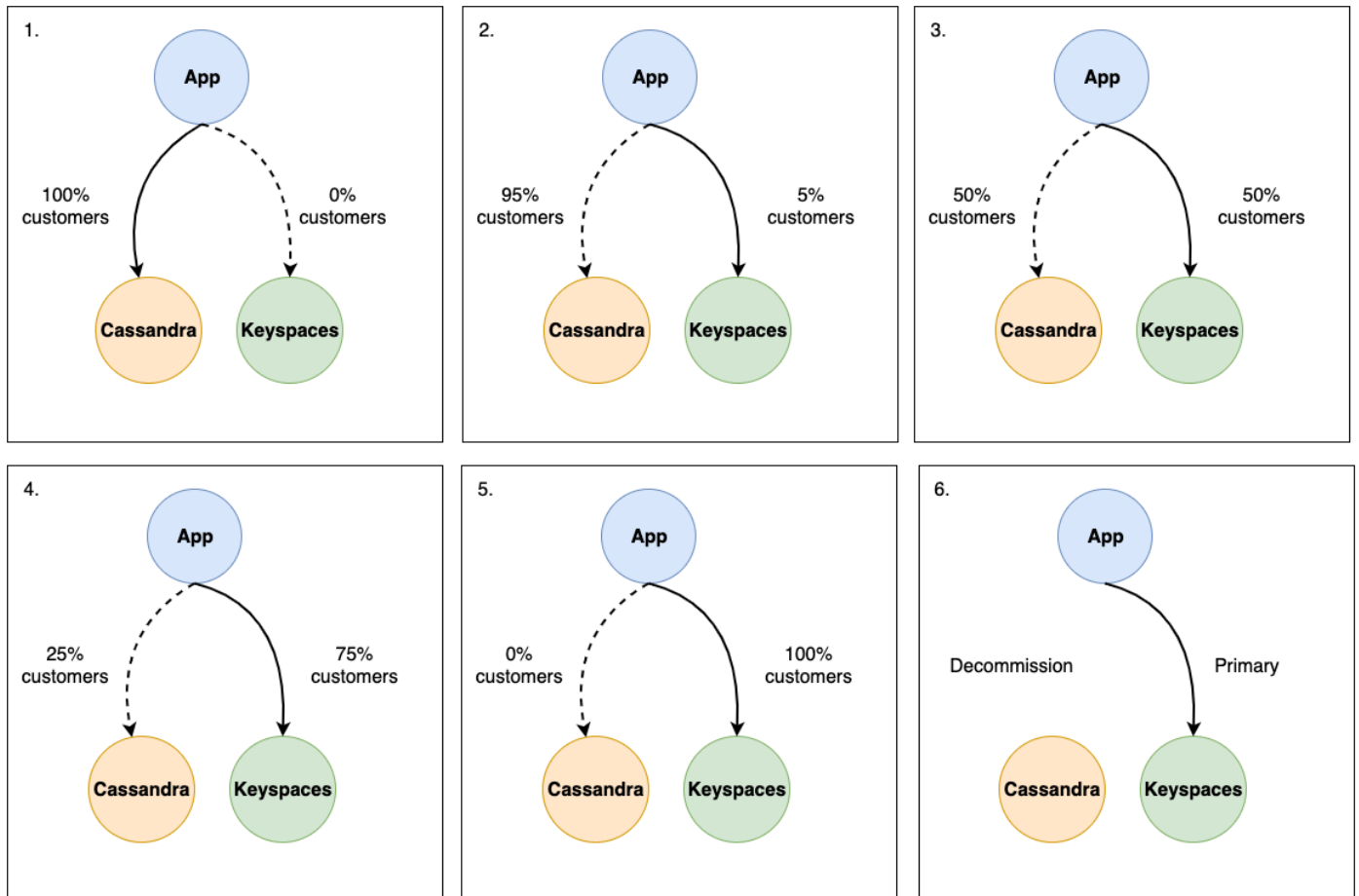
- **Strategia Canary:** con questo approccio, estendi gradualmente la migrazione a un sottoinsieme dei tuoi utenti o del tuo traffico. Inizialmente, una piccola percentuale del traffico dell'applicazione, ad esempio il 5% di tutto il traffico, viene indirizzato alla versione che utilizza Amazon Keyspaces come data store principale, mentre il resto del traffico continua a utilizzare Cassandra come data store principale.

In questo modo puoi testare a fondo la versione migrata con traffico reale, monitorarne le prestazioni e la stabilità e indagare su potenziali problemi. Se non rilevi alcun problema, puoi aumentare in modo incrementale la percentuale di traffico indirizzato ad Amazon Keyspaces fino a farlo diventare l'archivio dati principale per tutti gli utenti e il traffico.

Questa implementazione graduale riduce al minimo il rischio di interruzioni diffuse del servizio e consente un processo di migrazione più controllato. In caso di problemi critici durante l'implementazione di Canary, puoi tornare rapidamente alla versione precedente utilizzando

Cassandra come archivio dati principale per il segmento di traffico interessato. Puoi procedere alla fase di smantellamento della migrazione solo dopo aver verificato che Amazon Keyspaces elabori il 100% degli utenti e del traffico come previsto.

Il diagramma seguente illustra le singole fasi della strategia delle Canarie.



Disattivazione di Cassandra dopo una migrazione online

Una volta completata la migrazione dell'applicazione, con l'applicazione completamente in esecuzione su Amazon Keyspaces e aver convalidato la coerenza dei dati per un periodo di tempo, puoi pianificare la disattivazione del cluster Cassandra. Durante questa fase, puoi valutare se i dati rimanenti nel cluster Cassandra devono essere archiviati o possono essere eliminati. Ciò dipende dalle politiche dell'organizzazione per la gestione e la conservazione dei dati.

Seguendo questa strategia e considerando le best practice consigliate descritte in questo argomento durante la pianificazione della migrazione online da Cassandra ad Amazon Keyspaces, puoi garantire

una transizione senza interruzioni ad Amazon Keyspaces read-after-write mantenendo al contempo la coerenza e la disponibilità dell'applicazione.

La migrazione da Apache Cassandra ad Amazon Keyspaces può offrire numerosi vantaggi, tra cui riduzione del sovraccarico operativo, scalabilità automatica, maggiore sicurezza e un framework che ti aiuta a raggiungere i tuoi obiettivi di conformità. Pianificando una strategia di migrazione online con doppia scrittura, caricamento storico dei dati, convalida dei dati e implementazione graduale, puoi garantire una transizione fluida con interruzioni minime per l'applicazione e i suoi utenti.

L'implementazione della strategia di migrazione online illustrata in questo argomento consente di convalidare i risultati della migrazione, identificare e risolvere eventuali problemi e, infine, disattivare la distribuzione Cassandra esistente a favore del servizio Amazon Keyspaces completamente gestito.

Processo di migrazione offline: da Apache Cassandra ad Amazon Keyspaces

Le migrazioni offline sono adatte quando ci si può permettere tempi di inattività necessari per eseguire la migrazione. È comune tra le aziende disporre di finestre di manutenzione per l'applicazione di patch, rilasci di grandi dimensioni o tempi di inattività per aggiornamenti hardware o aggiornamenti importanti. La migrazione offline può utilizzare questa finestra per copiare i dati e trasferire il traffico delle applicazioni da Apache Cassandra ad Amazon Keyspaces.

La migrazione offline riduce le modifiche all'applicazione perché non richiede la comunicazione simultanea con Cassandra e Amazon Keyspaces. Inoltre, con il flusso di dati in pausa, è possibile copiare lo stato esatto senza mantenere le mutazioni.

In questo esempio, utilizziamo Amazon Simple Storage Service (Amazon S3) come area di gestione temporanea per i dati durante la migrazione offline per ridurre al minimo i tempi di inattività. Puoi importare automaticamente i dati archiviati in formato Parquet in Amazon S3 in una tabella Amazon Keyspaces utilizzando il connettore Spark Cassandra e AWS Glue. La sezione seguente mostrerà una panoramica di alto livello del processo. Puoi trovare esempi di codice per questo processo su [Github](#).

Il processo di migrazione offline da Apache Cassandra ad Amazon Keyspaces utilizza Amazon S3 e richiede i seguenti processi. AWS Glue AWS Glue

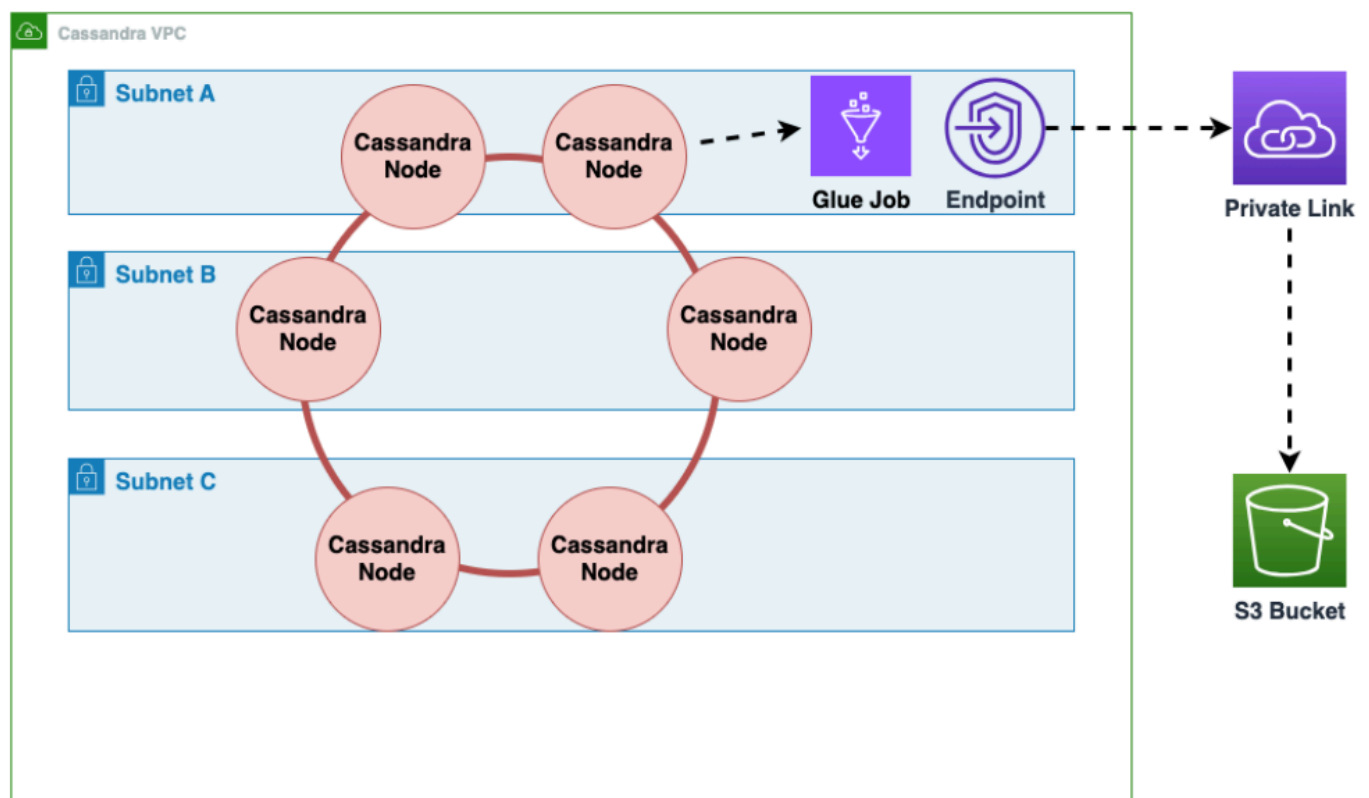
1. Un processo ETL che estrae e trasforma i dati CQL e li archivia in un bucket Amazon S3.
2. Un secondo processo che importa i dati dal bucket in Amazon Keyspaces.

3. Un terzo lavoro per importare dati incrementali.

Come eseguire una migrazione offline verso Amazon Keyspaces da Cassandra in esecuzione su Amazon EC2 in un Amazon Virtual Private Cloud

1. Innanzitutto devi AWS Glue esportare i dati della tabella da Cassandra in formato Parquet e salvarli in un bucket Amazon S3. È necessario eseguire un AWS Glue processo utilizzando un AWS Glue connettore a un VPC in cui risiede l' EC2 istanza Amazon che esegue Cassandra. Quindi, utilizzando l'endpoint privato Amazon S3, puoi salvare i dati nel bucket Amazon S3.

Il diagramma seguente illustra questi passaggi.



2. Mescola i dati nel bucket Amazon S3 per migliorare la randomizzazione dei dati. I dati importati in modo uniforme consentono una maggiore distribuzione del traffico nella tabella di destinazione.

Questo passaggio è necessario quando si esportano dati da Cassandra con partizioni di grandi dimensioni (partizioni con più di 1000 righe) per evitare schemi di tasti di scelta rapida durante l'inserimento dei dati in Amazon Keyspaces. I problemi relativi ai tasti di scelta rapida si

verificano `WriteThrottleEvents` in Amazon Keyspaces e comportano un aumento del tempo di caricamento.



3. Usa un altro AWS Glue processo per importare dati dal bucket Amazon S3 in Amazon Keyspaces. I dati mischiati nel bucket Amazon S3 vengono archiviati in formato Parquet.



Per ulteriori informazioni sul processo di migrazione offline, consulta il workshop [Amazon Keyspaces con AWS Glue](#)

Utilizzo di una soluzione di migrazione ibrida: da Apache Cassandra ad Amazon Keyspaces

La seguente soluzione di migrazione può essere considerata un ibrido tra migrazione online e offline. Con questo approccio ibrido, i dati vengono scritti nel database di destinazione quasi in tempo reale senza fornire coerenza tra lettura e scrittura. Ciò significa che i dati appena scritti non saranno immediatamente disponibili e sono prevedibili ritardi. Se hai bisogno di coerenza tra lettura e scrittura, vedi [the section called “Migrazione online”](#).

Per una migrazione quasi in tempo reale da Apache Cassandra ad Amazon Keyspaces, puoi scegliere tra due metodi disponibili.

- CQLReplicator— (Consigliata) CQLReplicator è un'utilità open source disponibile su [Github](#) che consente di migrare i dati da Apache Cassandra ad Amazon Keyspaces quasi in tempo reale.

Per determinare le scritture e gli aggiornamenti da propagare al database di destinazione, CQLReplicator analizza l'intervallo di token Apache Cassandra e utilizza un AWS Glue processo per rimuovere gli eventi duplicati e applicare scritture e aggiornamenti direttamente ad Amazon Keyspaces.

- **Change Data Capture (CDC):** se conosci Cassandra CDC, la funzionalità CDC integrata di Apache Cassandra che consente di acquisire le modifiche copiando il log di commit in una directory CDC separata è un'altra opzione per implementare una migrazione ibrida.

Puoi farlo replicando le modifiche ai dati in Amazon Keyspaces, rendendo CDC un'opzione alternativa per gli scenari di migrazione dei dati.

Se non hai bisogno di coerenza tra lettura e scrittura, puoi utilizzare la pipeline CQLReplicator o una pipeline CDC per migrare i dati da Apache Cassandra ad Amazon Keyspaces in base alle tue preferenze e alla tua familiarità con gli strumenti utilizzati in ciascuna soluzione. Servizi AWS L'utilizzo di questi metodi per migrare i dati quasi in tempo reale può essere considerato un approccio ibrido alla migrazione che offre un'alternativa alla migrazione online.

Questa strategia è considerata un approccio ibrido, poiché oltre alle opzioni descritte in questo argomento, è necessario implementare alcune fasi del progresso della migrazione online, ad esempio la copia storica dei dati e le strategie di migrazione delle applicazioni discusse nell'argomento sulla [migrazione online](#).

Le sezioni seguenti esaminano le opzioni di migrazione ibrida in modo più dettagliato.

Argomenti

- [Migra i dati utilizzando CQLReplicator](#)
- [Migra i dati utilizzando Change Data Capture \(CDC\)](#)

Migra i dati utilizzando CQLReplicator

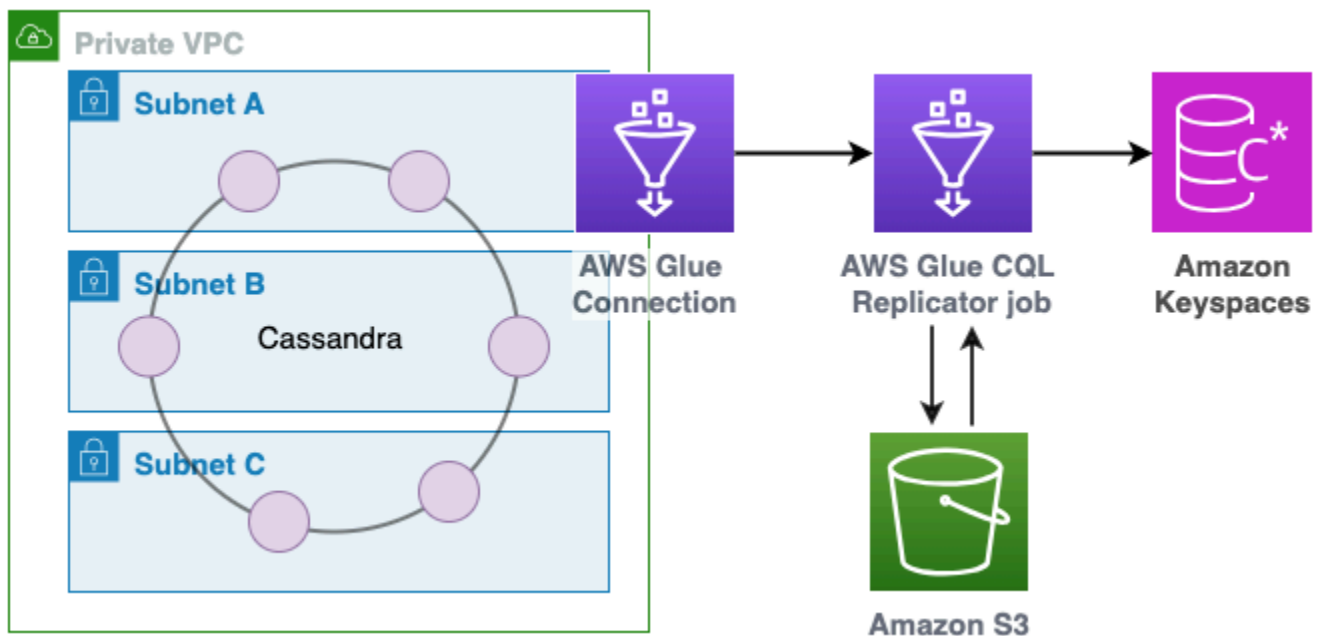
Con [CQLReplicator](#), puoi leggere i dati da Apache Cassandra quasi in tempo reale scansionando in modo intelligente il token ring di Cassandra utilizzando le query CQL. CQLReplicator non utilizza Cassandra CDC e implementa invece una strategia di caching per ridurre le penalità prestazionali delle scansioni complete.

Per ridurre il numero di scritture sulla destinazione, rimuove automaticamente gli eventi di replica duplicati CQLReplicator . Con CQLReplicator, puoi ottimizzare la replica delle modifiche dal database

di origine al database di destinazione, consentendo una migrazione quasi in tempo reale dei dati da Apache Cassandra ad Amazon Keyspaces.

Il diagramma seguente mostra l'architettura tipica di un job che utilizza CQLReplicator AWS Glue

1. Per consentire l'accesso ad Apache Cassandra in esecuzione in un VPC privato, configura una AWS Glue connessione con il tipo di connessione Rete.
2. Per rimuovere i duplicati e abilitare la memorizzazione nella cache delle chiavi con il CQLReplicator job, configura Amazon Simple Storage Service (Amazon S3).
3. Il database di origine verificato di CQLReplicator Job Streams viene modificato direttamente in Amazon Keyspaces.



Per ulteriori informazioni sull'utilizzo del processo di migrazione CQLReplicator, consulta il seguente post sul blog AWS Database [Migrate Cassandra to Amazon Keyspaces using CQLReplicator e la guida AWS prescrittiva Migrate Apache Cassandra workload](#) to Amazon Keyspaces using. AWS Glue

Migra i dati utilizzando Change Data Capture (CDC)

Se hai già familiarità con la configurazione di una pipeline di change data capture (CDC) con [Debezium](#), puoi utilizzare questa opzione per migrare i dati su Amazon Keyspaces come alternativa all'utilizzo. CQLReplicator Debezium è una piattaforma distribuita open source per CDC, progettata per monitorare un database e acquisire le modifiche a livello di riga in modo affidabile.

Il [connettore Debezium per Apache Cassandra carica le](#) modifiche su Amazon Managed Streaming for Apache Kafka (Amazon MSK) in modo che possano essere utilizzate ed elaborate dai consumatori downstream che a loro volta scrivono i dati su Amazon Keyspaces. Per ulteriori informazioni, consulta [Guida per la migrazione continua dei dati da Apache Cassandra ad Amazon Keyspaces](#).

Per risolvere eventuali problemi di coerenza dei dati, puoi implementare un processo con Amazon MSK in cui un consumatore confronta le chiavi o le partizioni di Cassandra con quelle di Amazon Keyspaces.

Per implementare correttamente questa soluzione, consigliamo di prendere in considerazione quanto segue.

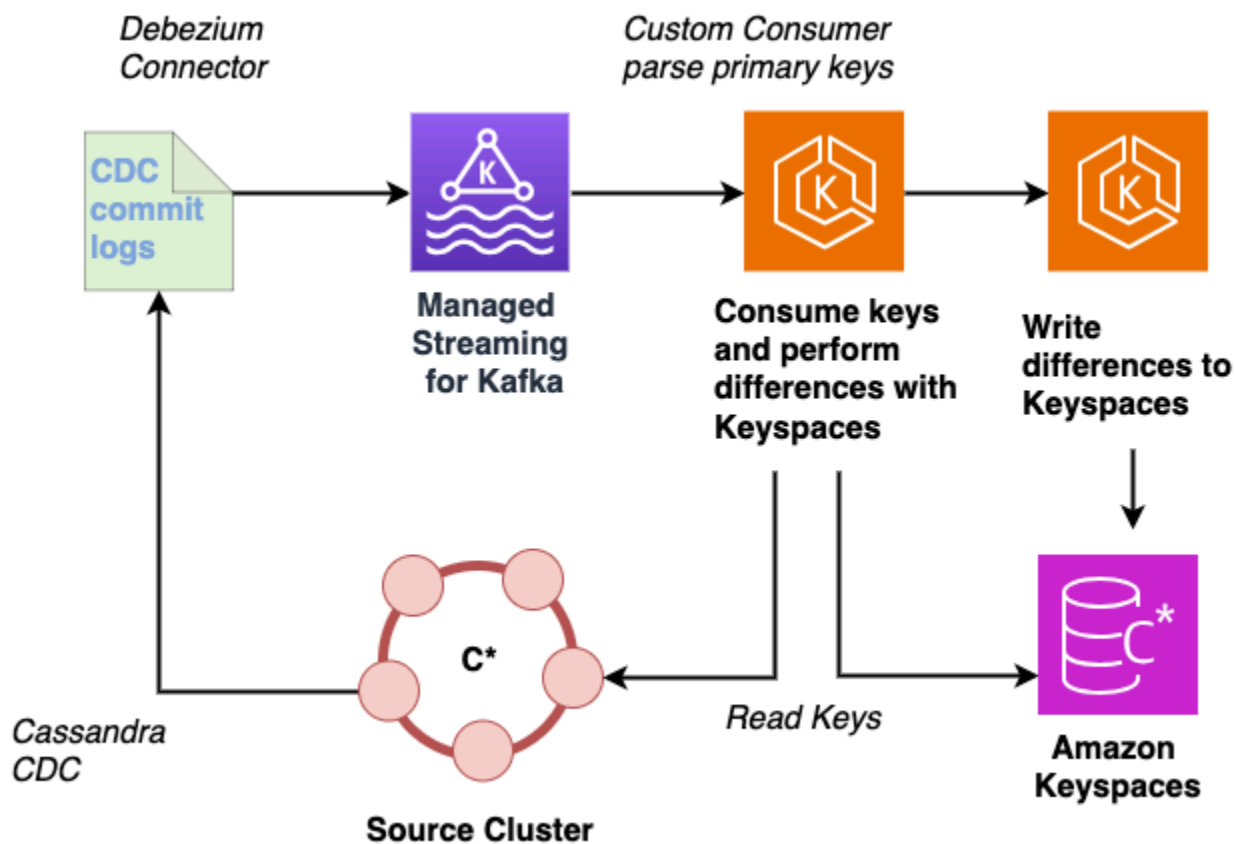
- Come analizzare il registro di commit del CDC, ad esempio come rimuovere gli eventi duplicati.
- Come mantenere la directory CDC, ad esempio come eliminare i vecchi log.
- Come gestire gli errori parziali in Apache Cassandra, ad esempio se una scrittura riesce solo in una replica su tre.
- Come gestire l'allocazione delle risorse, ad esempio aumentando le dimensioni dell'istanza per tenere conto dei requisiti aggiuntivi di CPU, memoria, DISCO e IO per il processo CDC che si verifica su un nodo.

Questo modello considera le modifiche apportate a Cassandra come un «indizio» del fatto che una chiave potrebbe essere cambiata rispetto al suo stato precedente. Per determinare se ci sono modifiche da propagare al database di destinazione, devi prima leggere dal cluster Cassandra di origine utilizzando un'LOCAL_QUORUMoperazione per ricevere i record più recenti e poi scriverli su Amazon Keyspaces.

In caso di eliminazioni o aggiornamenti di intervalli, potrebbe essere necessario eseguire un confronto con l'intera partizione per determinare quali eventi di scrittura o aggiornamento devono essere scritti nel database di destinazione.

Nei casi in cui le scritture non sono idempotenti, devi anche confrontare le tue scritture con quelle già presenti nel database di destinazione prima di scrivere su Amazon Keyspaces.

Il diagramma seguente mostra l'architettura tipica di una pipeline CDC che utilizza Debezium e Amazon MSK.



Come scegliere lo strumento giusto per il caricamento in blocco o la migrazione di dati su Amazon Keyspaces

In questa sezione puoi esaminare i diversi strumenti che puoi utilizzare per caricare in blocco o migrare dati su Amazon Keyspaces e scoprire come selezionare lo strumento corretto in base alle tue esigenze. Inoltre, questa sezione fornisce una panoramica e casi d'uso per step-by-step i tutorial disponibili che dimostrano come importare dati in Amazon Keyspaces.

Per esaminare le strategie disponibili per migrare i carichi di lavoro da Apache Cassandra ad Amazon Keyspaces, consulta. [the section called “Migrazione da Cassandra”](#)

- Strumenti di migrazione
 - Con il [calcolatore dei prezzi per Amazon Keyspaces \(per Apache Cassandra\)](#) disponibile su Github, puoi stimare i costi mensili per Amazon Keyspaces in base al tuo carico di lavoro Apache Cassandra esistente. Inserisci i parametri dello stato dello strumento Cassandra e della configurazione serverless prevista per Amazon Keyspaces per confrontare i costi diretti tra le due soluzioni. Tieni presente che questo calcolatore si concentra solo sui costi operativi di Amazon

Keyspaces rispetto alla distribuzione Cassandra esistente. Non include fattori del costo totale di proprietà (TCO) come la manutenzione dell'infrastruttura, le spese generali operative o i costi di supporto per Cassandra.

- Proxy ZDM Dual Write per Amazon Keyspaces Migration — ZDM Dual Write Proxy disponibile su [Github](#) supporta la migrazione senza downtime da Apache Cassandra ad Amazon Keyspaces.
- CQLReplicator— CQLReplicator è un'utilità open source disponibile su [Github](#) che consente di migrare i dati da Apache Cassandra ad Amazon Keyspaces quasi in tempo reale.

Per ulteriori informazioni, consulta [the section called “CQLReplicator”](#).

- Per ulteriori informazioni su come utilizzare Amazon Managed Streaming for Apache Kafka per [implementare un](#) processo di migrazione online con doppia scrittura, [consulta la Guida per la migrazione continua dei dati da](#) Apache Cassandra ad Amazon Keyspaces.
- Per migrazioni di grandi dimensioni, prendi in considerazione l'utilizzo di uno strumento di estrazione, trasformazione e caricamento (ETL). È possibile utilizzarlo AWS Glue per eseguire migrazioni di trasformazione dei dati in modo rapido ed efficace. Per ulteriori informazioni, consulta [the section called “Migrazione offline”](#).
- Per informazioni su come utilizzare il connettore Apache Cassandra Spark per scrivere dati su Amazon Keyspaces, consulta. [the section called “Integrazione con Apache Spark”](#)
- Inizia rapidamente a caricare dati in Amazon Keyspaces utilizzando il COPY FROM comando cqlsh. cqlsh è incluso in Apache Cassandra ed è ideale per caricare piccoli set di dati o dati di test. step-by-stepPer istruzioni, consulta. [the section called “Caricamento dei dati utilizzando cqlsh”](#)
- Puoi anche utilizzare DataStax Bulk Loader per Apache Cassandra per caricare dati in Amazon Keyspaces utilizzando il comando. dsbulk DSBulk[offre funzionalità di importazione più solide rispetto a cqlsh ed è disponibile nel repository. GitHub](#) Per step-by-step istruzioni, vedere. [the section called “Caricamento dei dati utilizzando DSBulk”](#)

Considerazioni generali per il caricamento di dati su Amazon Keyspaces

- Suddividi il caricamento dei dati in componenti più piccoli.

Considerate le seguenti unità di migrazione e il loro potenziale impatto in termini di dimensioni dei dati grezzi. Il caricamento di piccole quantità di dati in una o più fasi può contribuire a semplificare la migrazione.

- Per cluster: migra tutti i dati di Cassandra contemporaneamente. Questo approccio può essere [utile per i cluster più piccoli](#).

- Per spazio di chiavi o tabella: suddividi la migrazione in gruppi di spazi chiave o tabelle. Questo approccio può aiutarti a migrare i dati in fasi in base ai requisiti per ogni carico di lavoro.
- In base ai dati: valuta la possibilità di migrare i dati per un gruppo specifico di utenti o prodotti, per ridurre ulteriormente le dimensioni dei dati.
- Dai la priorità ai dati da caricare per primi in base alla semplicità.

Valuta se disponi di dati che potrebbero essere migrati per primi più facilmente, ad esempio dati che non cambiano in orari specifici, dati provenienti da processi in batch notturni, dati non utilizzati durante le ore offline o dati provenienti da app interne.

Argomenti

- [Tutorial: caricamento di dati in Amazon Keyspaces utilizzando cqlsh](#)
- [Tutorial: caricamento di dati in Amazon Keyspaces utilizzando DSBulk](#)

Tutorial: caricamento di dati in Amazon Keyspaces utilizzando cqlsh

Questo tutorial ti guida attraverso il processo di migrazione dei dati da Apache Cassandra ad Amazon Keyspaces utilizzando il comando `cqlsh COPY FROM`. Il comando `cqlsh COPY FROM` è utile per caricare rapidamente e facilmente piccoli set di dati su Amazon Keyspaces per scopi accademici o di test. Per ulteriori informazioni su come migrare i carichi di lavoro di produzione, consulta [the section called “Migrazione offline”](#). In questo tutorial, completerai i seguenti passaggi:

Prerequisiti: configura un AWS account con credenziali, crea un file di trust store JKS per il certificato e configura la connessione `cqlsh` ad Amazon Keyspaces.

1. Crea CSV di origine e tabella di destinazione: prepara un file CSV come dati di origine e crea lo spazio chiave e la tabella di destinazione in Amazon Keyspaces.
2. Preparazione dei dati: randomizza i dati nel file CSV e analizzali per determinare le dimensioni medie e massime delle righe.
3. Imposta la capacità di trasmissione: calcola le unità di capacità di scrittura richieste (WCUs) in base alla dimensione dei dati e al tempo di caricamento desiderato e configura la capacità assegnata alla tabella.
4. Configura i parametri `cqlsh`: determina i valori ottimali per `cqlsh COPY FROM` parametri come `INGESTRATE`, `NUMPROCESSES`, `MAXBATCHSIZE`, e distribuisce il carico di lavoro `CHUNKSIZE` in modo uniforme.

5. Esegui il **cqlsh COPY FROM** comando: esegui il `cqlsh COPY FROM` comando per caricare i dati dal file CSV nella tabella Amazon Keyspaces e monitorare l'avanzamento.

Risoluzione dei problemi: risolvi problemi comuni come richieste non valide, errori del parser, errori di capacità ed errori cqlsh durante il processo di caricamento dei dati.

Argomenti

- [Prerequisiti: passaggi da completare prima di poter caricare i dati utilizzando cqlsh COPY FROM](#)
- [Passaggio 1: crea il file CSV di origine e una tabella di destinazione per il caricamento dei dati](#)
- [Passaggio 2: prepara i dati di origine per un corretto caricamento dei dati](#)
- [Fase 3: Impostare la capacità di throughput per la tabella](#)
- [Fase 4: Configurare cqlsh COPY FROM le impostazioni](#)
- [Passaggio 5: Esegui il cqlsh COPY FROM comando per caricare i dati dal file CSV nella tabella di destinazione](#)
- [Risoluzione dei problemi](#)

Prerequisiti: passaggi da completare prima di poter caricare i dati utilizzando `cqlsh COPY FROM`

È necessario completare le seguenti attività prima di iniziare questo tutorial.

1. Se non l'hai già fatto, iscriviti a un account Account AWS seguendo i passaggi riportati in [the section called “Configurazione AWS Identity and Access Management”](#).
2. Crea credenziali specifiche per il servizio seguendo i passaggi riportati in [the section called “Crea credenziali specifiche per il servizio”](#)
3. Configura la connessione shell Cassandra Query Language (cqlsh) e conferma di poterti connettere ad Amazon Keyspaces seguendo i passaggi indicati. [the section called “Uso di cqlsh”](#)

Passaggio 1: crea il file CSV di origine e una tabella di destinazione per il caricamento dei dati

Per questo tutorial, utilizziamo un file con valori separati da virgole (CSV) con il nome `keyspaces_sample_table.csv` come file di origine per la migrazione dei dati. Il file di esempio fornito contiene alcune righe di dati per una tabella con il nome. `book_awards`

1. Crea il file sorgente. Puoi scegliere una delle seguenti opzioni:

- Scaricate il file CSV di esempio (`keyspaces_sample_table.csv`) contenuto nel seguente file di archivio [samplmigration.zip](#). Decomprimi l'archivio e prendi nota del percorso verso `keyspaces_sample_table.csv`
- Per compilare un file CSV con i propri dati memorizzati in un database Apache Cassandra, è possibile compilare il file CSV di origine utilizzando l'`cqlshCOPY TO`istruzione, come illustrato nell'esempio seguente.

```
cqlsh localhost 9042 -u "username" -p "password" --execute  
"COPY mykeyspace.mytable TO 'keyspaces_sample_table.csv' WITH HEADER=true";
```

Assicurati che il file CSV che crei soddisfi i seguenti requisiti:

- La prima riga contiene i nomi delle colonne.
- I nomi delle colonne nel file CSV di origine corrispondono ai nomi delle colonne nella tabella di destinazione.
- I dati sono delimitati da una virgola.
- Tutti i valori dei dati sono tipi di dati Amazon Keyspaces validi. Consultare [the section called “Tipi di dati”](#).

2. Crea lo spazio chiave e la tabella di destinazione in Amazon Keyspaces.

- a. Connettiti ad Amazon Keyspaces utilizzando `cqlsh` e sostituendo l'endpoint del servizio, il nome utente e la password nell'esempio seguente con i tuoi valori.

```
cqlsh cassandra.us-east-1.amazonaws.com 9142 -u "111122223333" -p  
"wJa1rXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY" --ssl
```

- b. Crea un nuovo keyspace con il nome mostrato nell'catalogoesempio seguente.

```
CREATE KEYSPACE catalog WITH REPLICATION = {'class': 'SingleRegionStrategy'};
```

- c. Quando il nuovo keyspace è disponibile, utilizzate il codice seguente per creare la tabella di destinazione. `book_awards`

```
CREATE TABLE "catalog.book_awards" (  
    year int,  
    award text,  
    rank int,
```



```
category text,  
book_title text,  
author text,  
publisher text,  
PRIMARY KEY ((year, award), category, rank)  
);
```

Se Apache Cassandra è la tua fonte di dati originale, un modo semplice per creare la tabella di destinazione di Amazon Keyspaces con intestazioni corrispondenti consiste nel generare l'`CREATE TABLE` istruzione dalla tabella di origine, come illustrato nell'istruzione seguente.

```
cqlsh localhost 9042 -u "username" -p "password" --execute "DESCRIBE  
TABLE mykeyspace.mytable;"
```

Quindi crea la tabella di destinazione in Amazon Keyspaces con i nomi delle colonne e i tipi di dati che corrispondono alla descrizione della tabella di origine di Cassandra.

Passaggio 2: prepara i dati di origine per un corretto caricamento dei dati

La preparazione dei dati di origine per un trasferimento efficiente è un processo in due fasi. Innanzitutto, i dati vengono randomizzati. Nella seconda fase, si analizzano i dati per determinare i valori dei `cqlsh` parametri appropriati e le impostazioni della tabella richieste per garantire che il caricamento dei dati abbia esito positivo.

Randomizza i dati

Il `cqlsh COPY FROM` comando legge e scrive i dati nello stesso ordine in cui appaiono nel file CSV. Se si utilizza il `cqlsh COPY TO` comando per creare il file sorgente, i dati vengono scritti in ordine di chiave nel file CSV. Internamente, Amazon Keyspaces partiziona i dati utilizzando chiavi di partizione. Sebbene Amazon Keyspaces disponga di una logica integrata per aiutare a bilanciare il carico delle richieste per la stessa chiave di partizione, il caricamento dei dati è più rapido ed efficiente se si rende casuale l'ordine. Questo perché puoi sfruttare il bilanciamento del carico integrato che si verifica quando Amazon Keyspaces scrive su partizioni diverse.

Per distribuire le scritture tra le partizioni in modo uniforme, devi randomizzare i dati nel file sorgente. [È possibile scrivere un'applicazione per eseguire questa operazione o utilizzare uno strumento open source, come Shuf.](#) Shuf è disponibile gratuitamente sulle distribuzioni Linux, su macOS (installando `coreutils` in [homebrew](#)) e su Windows (utilizzando Windows Subsystem for Linux (WSL)).

È necessario un passaggio aggiuntivo per evitare che la riga di intestazione con i nomi delle colonne venga mescolata in questo passaggio.

Per rendere casuale il file sorgente preservando l'intestazione, inserisci il codice seguente.

```
tail -n +2 keyspaces_sample_table.csv | shuf -o keyspace.table.csv && (head  
-1 keyspaces_sample_table.csv && cat keyspace.table.csv ) > keyspace.table.csv1 &&  
mv keyspace.table.csv1 keyspace.table.csv
```

Shuf riscrive i dati in un nuovo file CSV chiamato `keyspace.table.csv`. Ora puoi eliminare il `keyspaces_sample_table.csv` file, non ti serve più.

Analizza i dati

Determina la dimensione media e massima delle righe analizzando i dati.

Lo fai per i seguenti motivi:

- La dimensione media delle righe aiuta a stimare la quantità totale di dati da trasferire.
- È necessaria la dimensione media delle righe per fornire la capacità di scrittura necessaria per il caricamento dei dati.
- Puoi assicurarti che ogni riga abbia una dimensione inferiore a 1 MB, che è la dimensione massima delle righe in Amazon Keyspaces.

Note

Questa quota si riferisce alla dimensione della riga, non alla dimensione della partizione. A differenza delle partizioni Apache Cassandra, le partizioni Amazon Keyspaces possono avere dimensioni praticamente illimitate. Le chiavi di partizione e le colonne di clustering richiedono spazio di archiviazione aggiuntivo per i metadati, che è necessario aggiungere alla dimensione grezza delle righe. Per ulteriori informazioni, consulta [the section called “Stima della dimensione delle righe”](#).

Il codice seguente utilizza [AWK](#) per analizzare un file CSV e stampare la dimensione media e massima delle righe.

```
awk -F, 'BEGIN {samp=10000;max=-1;}{if(NR>1){len=length($0);t+=len;avg=t/
NR;max=(len>max ? len : max)}}NR==samp{exit}END{printf("{lines: %d, average: %d bytes,
max: %d bytes}\n",NR,avg,max);}' keyspace.table.csv
```

L'esecuzione di questo codice produce il seguente risultato.

```
using 10,000 samples:
{lines: 10000, avg: 123 bytes, max: 225 bytes}
```

Utilizzerai la dimensione media delle righe nel passaggio successivo di questo tutorial per fornire la capacità di scrittura per la tabella.

Fase 3: Impostare la capacità di throughput per la tabella

Questo tutorial mostra come ottimizzare cqlsh per caricare i dati entro un intervallo di tempo prestabilito. Poiché sapete quante letture e scritture eseguite in anticipo, utilizzate la modalità di capacità fornita. Al termine del trasferimento dei dati, è necessario impostare la modalità di capacità della tabella in modo che corrisponda ai modelli di traffico dell'applicazione. Per ulteriori informazioni sulla gestione della capacità, consulta [Gestione delle risorse serverless](#).

Con la modalità di capacità fornita, è possibile specificare in anticipo la quantità di capacità di lettura e scrittura che si desidera fornire alla tabella. La capacità di scrittura viene fatturata ogni ora e misurata in unità di capacità di scrittura (). WCUs Ogni WCU ha una capacità di scrittura sufficiente per supportare la scrittura di 1 KB di dati al secondo. Quando si caricano i dati, la velocità di scrittura deve essere inferiore al valore massimo WCUs (parametro:write_capacity_units) impostato nella tabella di destinazione.

Per impostazione predefinita, puoi eseguire il provisioning fino a 40.000 WCUs per tabella e 80.000 per WCUs tutte le tabelle del tuo account. Se hai bisogno di capacità aggiuntiva, puoi richiedere un aumento della quota nella console [Service Quotas](#). Per ulteriori informazioni sulle quote, consulta [Quote](#).

Calcola il numero medio di componenti WCUs necessari per un inserto

L'inserimento di 1 KB di dati al secondo richiede 1 WCU. Se il tuo file CSV ha 360.000 righe e desideri caricare tutti i dati in un'ora, devi scrivere 100 righe al secondo (360.000 righe/60 minuti/60 secondi = 100 righe al secondo). Se ogni riga contiene fino a 1 KB di dati, per inserire 100 righe al secondo, devi assegnarne 100 WCUs alla tabella. Se ogni riga contiene 1,5 KB di dati, ne occorrono

due WCUs per inserire una riga al secondo. Pertanto, per inserire 100 righe al secondo, è necessario predisporre 200 WCUs.

Per determinare quante WCUs righe sono necessarie per inserire una riga al secondo, dividi la dimensione media delle righe in byte per 1024 e arrotonda al numero intero più vicino.

Ad esempio, se la dimensione media delle righe è 3000 byte, ne occorrono tre WCUs per inserire una riga al secondo.

```
ROUNDUP(3000 / 1024) = ROUNDUP(2.93) = 3 WCUs
```

Calcola il tempo e la capacità di caricamento dei dati

Ora che conosci la dimensione e il numero medi di righe del tuo file CSV, puoi calcolare quante sono WCUs necessarie per caricare i dati in un determinato periodo di tempo e il tempo approssimativo necessario per caricare tutti i dati nel file CSV utilizzando diverse impostazioni WCU.

Ad esempio, se ogni riga del file è di 1 KB e il file CSV contiene 1.000.000 di righe, per caricare i dati in un'ora, è necessario fornire almeno 278 righe WCUs alla tabella per quell'ora.

```
1,000,000 rows * 1 KBs = 1,000,000 KBs  
1,000,000 KBs / 3600 seconds = 277.8 KBs / second = 278 WCUs
```

Configura le impostazioni della capacità assegnata

È possibile impostare le impostazioni della capacità di scrittura di una tabella al momento della creazione della tabella o utilizzando il comando ALTER TABLE CQL. Di seguito è riportata la sintassi per modificare le impostazioni della capacità assegnata a una tabella con l'istruzione CQL. ALTER TABLE

```
ALTER TABLE mykeyspace.mytable WITH custom_properties={'capacity_mode':  
{'throughput_mode': 'PROVISIONED', 'read_capacity_units': 100,  
'write_capacity_units': 278}} ;
```

Per il riferimento completo alla lingua, vedere. [the section called “ALTER TABLE”](#)

Fase 4: Configurare **cqlsh COPY FROM** le impostazioni

Questa sezione descrive come determinare i valori dei parametri per **cqlsh COPY FROM**. Il **cqlsh COPY FROM** comando legge il file CSV preparato in precedenza e inserisce i dati in Amazon

Keyspaces utilizzando CQL. Il comando divide le righe e distribuisce le operazioni tra un insieme di lavoratori. INSERT Ogni lavoratore stabilisce una connessione con Amazon Keyspaces e INSERT invia richieste attraverso questo canale.

Il `cqlsh COPY` comando non ha una logica interna per distribuire il lavoro in modo uniforme tra i suoi lavoratori. Tuttavia, puoi configurarlo manualmente per assicurarti che il lavoro sia distribuito in modo uniforme. Inizia esaminando questi parametri chiave di `cqlsh`:

- **DELIMITATORE**: se hai utilizzato un delimitatore diverso da una virgola, puoi impostare questo parametro, che per impostazione predefinita è la virgola.
- **INGESTRATE**: il numero target di righe che tenta di elaborare al secondo. `cqlsh COPY FROM` Se non è impostato, il valore predefinito è 100.000.
- **NUMPROCESSES**: il numero di processi di lavoro minorile che `cqlsh` crea per le attività. `COPY FROM` Il massimo per questa impostazione è 16, l'impostazione predefinita è `num_cores - 1`, `num_cores` dov'è il numero di core di elaborazione sull'host che esegue `cqlsh`.
- **MAXBATCHSIZE** — La dimensione del batch determina il numero massimo di righe inserite nella tabella di destinazione in un singolo batch. Se non è impostato, `cqlsh` utilizza batch di 20 righe inserite.
- **CHUNKSIZE** — La dimensione dell'unità di lavoro che passa al lavoratore minorile. Per impostazione predefinita, è impostata su 5.000.
- **MAXTENTEMENTS** — Il numero massimo di volte in cui riprovare un blocco di lavoro non riuscito. Una volta raggiunto il tentativo massimo, i record non riusciti vengono scritti in un nuovo file CSV che è possibile eseguire nuovamente in un secondo momento dopo aver esaminato l'errore.

INGESTRATE Impostato in base al numero di file WCUs che hai assegnato alla tabella di destinazione. Il INGESTRATE valore del `cqlsh COPY FROM` comando non è un limite, ma una media target. Ciò significa che può (e spesso succede) superare il numero impostato. Per consentire l'insorgenza di interruzioni e assicurarvi che sia disponibile una capacità sufficiente per gestire le richieste di caricamento dei dati, impostate INGESTRATE il 90% della capacità di scrittura della tabella.

```
INGESTRATE = WCUs * .90
```

Quindi, impostate il **NUMPROCESSES** parametro in modo che sia uguale a uno in meno rispetto al numero di core del sistema. Per scoprire qual è il numero di core del sistema, è possibile eseguire il codice seguente.

```
python -c "import multiprocessing; print(multiprocessing.cpu_count())"
```

Per questo tutorial, utilizziamo il seguente valore.

```
NUMPROCESSES = 4
```

Ogni processo crea un lavoratore e ogni lavoratore stabilisce una connessione ad Amazon Keyspaces. Amazon Keyspaces può supportare fino a 3.000 richieste CQL al secondo su ogni connessione. Ciò significa che devi assicurarti che ogni lavoratore elabori meno di 3.000 richieste al secondo.

Inoltre `INGESTRATE`, i lavoratori spesso superano il numero impostato e non sono limitati dai secondi. Pertanto, per tenere conto delle interruzioni, impostate i parametri `cqlsh` in modo che ciascun lavoratore elabori 2.500 richieste al secondo. Per calcolare la quantità di lavoro distribuita a un lavoratore, utilizzate le seguenti linee guida.

- Dividi `INGESTRATE` per `NUMPROCESSES`.
- Se $INGESTRATE/NUMPROCESSES > 2.500$, abbassa il valore `INGESTRATE` per rendere vera questa formula.

```
INGESTRATE / NUMPROCESSES <= 2,500
```

Prima di configurare le impostazioni per ottimizzare il caricamento dei nostri dati di esempio, esaminiamo le impostazioni `cqlsh` predefinite e vediamo come il loro utilizzo influisce sul processo di caricamento dei dati. Poiché `cqlsh COPY FROM` utilizza il `CHUNKSIZE` per creare blocchi di lavoro (`INSERT` trendiconti) da distribuire ai lavoratori, il lavoro non viene distribuito automaticamente in modo uniforme. Alcuni lavoratori potrebbero rimanere inattivi, a seconda dell'impostazione. `INGESTRATE`

Per distribuire il lavoro in modo uniforme tra i lavoratori e mantenere per ogni lavoratore la frequenza ottimale di 2.500 richieste al secondo, è necessario impostare `CHUNKSIZE` e modificare i `INGESTRATE` parametri di input. `MAXBATCHSIZE` Per ottimizzare l'utilizzo del traffico di rete durante il caricamento dei dati, scegliete un valore vicino al valore massimo di 30. `MAXBATCHSIZE` Passando `CHUNKSIZE` a 100 e `MAXBATCHSIZE` a 25, le 10.000 righe vengono distribuite uniformemente tra i quattro lavoratori ($10.000/2500 = 4$).

Il seguente esempio di codice lo illustra.

```
INGESTRATE = 10,000
```

```
NUMPROCESSES = 4
CHUNKSIZE = 100
MAXBATCHSIZE. = 25
Work Distribution:
Connection 1 / Worker 1 : 2,500 Requests per second
Connection 2 / Worker 2 : 2,500 Requests per second
Connection 3 / Worker 3 : 2,500 Requests per second
Connection 4 / Worker 4 : 2,500 Requests per second
```

Per riassumere, utilizzate le seguenti formule per impostare i parametri: `cqlsh COPY FROM`

- $INGESTRATE = write\_capacity\_units * .90$
- $NUMPROCESSES = num\_cores - 1$ (impostazione predefinita)
- $INGESTRATE/NUMPROCESSES = 2.500$ (deve essere una dichiarazione vera).
- $MAXBATCHSIZE = 30$ (il valore predefinito è 20. Amazon Keyspaces accetta batch fino a 30.)
- $CHUNKSIZE = (INGESTRATE/NUMPROCESSES)/MAXBATCHSIZE$

Ora che hai calcolato `CHUNKSIZE` sei pronto per `NUMPROCESSES` caricare `INGESTRATE` i tuoi dati.

Passaggio 5: Esegui il **`cqlsh COPY FROM`** comando per caricare i dati dal file CSV nella tabella di destinazione

Per eseguire il `cqlsh COPY FROM` comando, completare i passaggi seguenti.

1. Connect ad Amazon Keyspaces utilizzando `cqlsh`.
2. Scegli il tuo keyspace con il codice seguente.

```
USE catalog;
```

3. Imposta la coerenza di scrittura su `LOCAL_QUORUM`. Per garantire la durabilità dei dati, Amazon Keyspaces non consente altre impostazioni di coerenza di scrittura. Vedi il codice seguente.

```
CONSISTENCY LOCAL_QUORUM;
```

4. Preparate la `cqlsh COPY FROM` sintassi utilizzando il seguente esempio di codice.

```
COPY book_awards FROM './keyspace.table.csv' WITH HEADER=true
AND INGESTRATE=calculated ingestrate
AND NUMPROCESSES=calculated numprocess
```

```
AND MAXBATCHSIZE=20  
AND CHUNKSIZE=calculated chunksize;
```

5. Esegui l'istruzione preparata nel passaggio precedente. `cqlsh` riporta tutte le impostazioni che hai configurato.

a. Assicurati che le impostazioni corrispondano ai dati immessi. Guarda l'esempio seguente.

```
Reading options from the command line: {'chunksize': '120', 'header': 'true',  
    'ingestrate': '36000', 'numprocesses': '15', 'maxbatchsize': '20'}  
Using 15 child processes
```

b. Controlla il numero di righe trasferite e il tasso medio corrente, come mostrato nell'esempio seguente.

```
Processed: 57834 rows; Rate: 6561 rows/s; Avg. rate: 31751 rows/s
```

c. Quando `cqlsh` ha finito di caricare i dati, esamina il riepilogo delle statistiche di caricamento dei dati (il numero di file letti, di runtime e le righe ignorate) come mostrato nell'esempio seguente.

```
15556824 rows imported from 1 files in 8 minutes and 8.321 seconds (0 skipped).
```

In questo passaggio finale del tutorial, hai caricato i dati su Amazon Keyspaces.

Important

Ora che hai trasferito i dati, regola le impostazioni della modalità di capacità della tabella di destinazione in modo che corrispondano ai normali modelli di traffico dell'applicazione. Fino a quando non la modifichi, ti verranno addebitati i costi in base alla tariffa oraria per la capacità assegnata.

Risoluzione dei problemi

Una volta completato il caricamento dei dati, controlla se le righe sono state saltate. Per farlo, accedi alla directory dei sorgenti del file CSV di origine e cerca un file con il nome seguente.

```
import_yourcsvfilename.err.timestamp.csv
```


cqlsh scrive tutte le righe di dati saltate in un file con quel nome. Se il file esiste nella tua directory di origine e contiene dati, queste righe non sono state caricate su Amazon Keyspaces. Per riprovare queste righe, verifica innanzitutto la presenza di eventuali errori riscontrati durante il caricamento e modifica i dati di conseguenza. Per riprovare queste righe, puoi eseguire nuovamente il processo.

Errori comuni

I motivi più comuni per cui le righe non vengono caricate sono gli errori di capacità e gli errori di analisi.

Errori di richiesta non validi durante il caricamento di dati su Amazon Keyspaces

Nell'esempio seguente, la tabella di origine contiene una colonna counter, che genera chiamate batch registrate dal comando cqlsh. COPY Le chiamate batch registrate non sono supportate da Amazon Keyspaces.

```
Failed to import 10 rows: InvalidRequest - Error from server: code=2200 [Invalid query]
message="Only UNLOGGED Batches are supported at this time.", will retry later,
attempt 22 of 25
```

Per risolvere questo errore, usa DSBulk per migrare i dati. Per ulteriori informazioni, consulta [the section called “Caricamento dei dati utilizzando DSBulk”](#).

Errori del parser durante il caricamento di dati su Amazon Keyspaces

L'esempio seguente mostra una riga saltata a causa di un. ParseError

```
Failed to import 1 rows: ParseError - Invalid ... -
```

Per risolvere questo errore, devi assicurarti che i dati da importare corrispondano allo schema della tabella in Amazon Keyspaces. Controlla il file di importazione per verificare eventuali errori di analisi. Puoi provare a utilizzare una singola riga di dati utilizzando un'INSERTistruzione per isolare l'errore.

Errori di capacità durante il caricamento dei dati su Amazon Keyspaces

```
Failed to import 1 rows: WriteTimeout - Error from server: code=1100 [Coordinator node
timed out waiting for replica nodes' responses]
message="Operation timed out - received only 0 responses." info={'received_responses':
0, 'required_responses': 2, 'write_type': 'SIMPLE', 'consistency':
'LOCAL_QUORUM'}, will retry later, attempt 1 of 100
```

Amazon Keyspaces utilizza le `WriteTimeout` eccezioni `ReadTimeout` e per indicare quando una richiesta di scrittura non riesce a causa di una capacità di throughput insufficiente. Per aiutare a diagnosticare eccezioni di capacità insufficienti, Amazon Keyspaces pubblica parametri e `WriteThrottleEvents` parametri in Amazon. `ReadThrottledEvents` CloudWatch Per ulteriori informazioni, consulta [the section called “Monitoraggio con CloudWatch”](#).

errori `cqlsh` durante il caricamento di dati su Amazon Keyspaces

Per facilitare la risoluzione degli errori `cqlsh`, esegui nuovamente il comando che ha avuto esito negativo con il flag. `--debug`

Quando si utilizza una versione incompatibile di `cqlsh`, viene visualizzato il seguente errore.

```
AttributeError: 'NoneType' object has no attribute 'is_up'
Failed to import 3 rows: AttributeError - 'NoneType' object has no attribute 'is_up',
given up after 1 attempts
```

Verificare che sia installata la versione corretta di `cqlsh` eseguendo il comando seguente.

```
cqlsh --version
```

Dovreste vedere qualcosa di simile a quanto segue per l'output.

```
cqlsh 5.0.1
```

Se usi Windows, sostituisci tutte le istanze di `cqlsh` with `cqlsh.bat`. Ad esempio, per verificare la versione di `cqlsh` in Windows, esegui il comando seguente.

```
cqlsh.bat --version
```

La connessione ad Amazon Keyspaces fallisce dopo che il client `cqlsh` riceve tre errori consecutivi di qualsiasi tipo dal server. Il client `cqlsh` ha esito negativo e viene visualizzato il seguente messaggio.

```
Failed to import 1 rows: NoHostAvailable - , will retry later, attempt 3 of 100
```

Per risolvere questo errore, devi assicurarti che i dati da importare corrispondano allo schema della tabella in Amazon Keyspaces. Controlla il file di importazione per verificare eventuali errori di analisi. Puoi provare a utilizzare una singola riga di dati utilizzando un'istruzione `INSERT` per isolare l'errore.

Il client tenta automaticamente di ristabilire la connessione.

Tutorial: caricamento di dati in Amazon Keyspaces utilizzando DSBulk

Questo step-by-step tutorial ti guida nella migrazione dei dati da Apache Cassandra ad Amazon Keyspaces utilizzando DataStax Bulk Loader () disponibile su [DSBulk GitHub](#). L'utilizzo DSBulk è utile per caricare set di dati su Amazon Keyspaces per scopi accademici o di test. Per ulteriori informazioni su come migrare i carichi di lavoro di produzione, consulta [the section called "Migrazione offline"](#). In questo tutorial, completerai i seguenti passaggi.

Prerequisiti: configura un AWS account con credenziali, crea un file di trust store JKS per il certificato, configura, scarica cqlsh DSBulk, installa e configura un file. `application.conf`

1. Crea CSV di origine e tabella di destinazione: prepara un file CSV come dati di origine e crea lo spazio chiave e la tabella di destinazione in Amazon Keyspaces.
2. Preparazione dei dati: randomizza i dati nel file CSV e analizzali per determinare le dimensioni medie e massime delle righe.
3. Imposta la capacità di trasmissione: calcola le unità di capacità di scrittura richieste (WCUs) in base alla dimensione dei dati e al tempo di caricamento desiderato e configura la capacità assegnata alla tabella.
4. Configura DSBulk le impostazioni: crea un file di DSBulk configurazione con impostazioni come autenticazione, SSL/TLS, livello di coerenza e dimensione del pool di connessioni.
5. Esegui il comando DSBulk load: esegui il comando DSBulk load per caricare i dati dal file CSV nella tabella Amazon Keyspaces e monitorare l'avanzamento.

Argomenti

- [Prerequisiti: passaggi da completare prima di poter caricare i dati con DSBulk](#)
- [Passaggio 1: creare il file CSV di origine e una tabella di destinazione per il caricamento dei dati utilizzando DSBulk](#)
- [Passaggio 2: prepara i dati da caricare utilizzando DSBulk](#)
- [Fase 3: Impostare la capacità di throughput per la tabella di destinazione](#)
- [Passaggio 4: Configurare DSBulk le impostazioni per caricare i dati dal file CSV alla tabella di destinazione](#)
- [Passaggio 5: Esegui il DSBulk load comando per caricare i dati dal file CSV nella tabella di destinazione](#)

Prerequisiti: passaggi da completare prima di poter caricare i dati con DSBulk

È necessario completare le seguenti attività prima di iniziare questo tutorial.

1. Se non l'hai ancora fatto, crea un AWS account seguendo i passaggi riportati in [the section called “Configurazione AWS Identity and Access Management”](#).
2. Crea le credenziali seguendo i passaggi riportati in [the section called “Crea credenziali IAM per l'autenticazione AWS”](#).
3. Crea un file di trust store JKS.
 - a. Scaricate i seguenti certificati digitali e salvate i file localmente o nella vostra home directory.
 1. AmazonRootCA1
 2. AmazonRootCA2
 3. AmazonRootCA3
 4. AmazonRootCA4
 5. Starfield Class 2 Root (opzionale, per compatibilità con le versioni precedenti)

Per scaricare i certificati, puoi usare i seguenti comandi.

```
curl -O https://www.amazontrust.com/repository/AmazonRootCA1.pem
curl -O https://www.amazontrust.com/repository/AmazonRootCA2.pem
curl -O https://www.amazontrust.com/repository/AmazonRootCA3.pem
curl -O https://www.amazontrust.com/repository/AmazonRootCA4.pem
curl -O https://certs.secureserver.net/repository/sf-class2-root.crt
```

Note

Amazon Keyspaces utilizzava in precedenza certificati TLS ancorati alla CA Starfield Class 2. AWS sta migrando tutto Regioni AWS verso certificati emessi nell'ambito di Amazon Trust Services (Amazon Root CAs 1—4). Durante questa transizione, configura i client in modo che si fidino sia di Amazon Root CAs 1-4 che di Starfield root per garantire la compatibilità in tutte le regioni.

- b. Converti i certificati digitali in file TrustStore e aggiungili al keystore.

```
openssl x509 -outform der -in AmazonRootCA1.pem -out temp_file.der
```

```
keytool -import -alias amazon-root-ca-1 -keystore cassandra_truststore.jks -
file temp_file.der

openssl x509 -outform der -in AmazonRootCA2.pem -out temp_file.der
keytool -import -alias amazon-root-ca-2 -keystore cassandra_truststore.jks -
file temp_file.der

openssl x509 -outform der -in AmazonRootCA3.pem -out temp_file.der
keytool -import -alias amazon-root-ca-3 -keystore cassandra_truststore.jks -
file temp_file.der

openssl x509 -outform der -in AmazonRootCA4.pem -out temp_file.der
keytool -import -alias amazon-root-ca-4 -keystore cassandra_truststore.jks -
file temp_file.der

openssl x509 -outform der -in sf-class2-root.crt -out temp_file.der
keytool -import -alias cassandra -keystore cassandra_truststore.jks -file
temp_file.der
```

Nell'ultimo passaggio, è necessario creare una password per il keystore e considerare attendibile ogni certificato. Il comando interattivo ha questo aspetto.

```
Enter keystore password:
Re-enter new password:
Owner: CN=Amazon Root CA 1, O=Amazon, C=US
Issuer: CN=Amazon Root CA 1, O=Amazon, C=US
Serial number: 66c9fcf99bf8c0a39e2f0788a43e696365bca
Valid from: Tue May 26 00:00:00 UTC 2015 until: Sun Jan 17 00:00:00 UTC 2038
Certificate fingerprints:
    SHA1: 8D:A7:F9:65:EC:5E:FC:37:91:0F:1C:6E:59:FD:C1:CC:6A:6E:DE:16
    SHA256:
    8E:CD:E6:88:4F:3D:87:B1:12:5B:A3:1A:C3:FC:B1:3D:70:16:DE:7F:57:CC:90:4F:E1:CB:97:C6:AE
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 2048-bit RSA key
Version: 3

Extensions:

#1: ObjectId: 2.5.29.19 Criticality=true
BasicConstraints:[
    CA:true
    PathLen:2147483647
]
```

```
#2: ObjectId: 2.5.29.15 Criticality=true
```

```
KeyUsage [
  DigitalSignature
  Key_CertSign
  Crl_Sign
]
```

```
#3: ObjectId: 2.5.29.14 Criticality=false
```

```
SubjectKeyIdentifier [
KeyIdentifier [
0000: 84 18 CC 85 34 EC BC 0C   94 94 2E 08 59 9C C7 B2   ....4.....Y...
0010: 10 4E 0A 08                               .N..
]
]
```

```
Trust this certificate? [no]: yes
```

```
Certificate was added to keystore
```

```
Enter keystore password:
```

```
Owner: CN=Amazon Root CA 2, O=Amazon, C=US
```

```
Issuer: CN=Amazon Root CA 2, O=Amazon, C=US
```

```
Serial number: 66c9fd29635869f0a0fe58678f85b26bb8a37
```

```
Valid from: Tue May 26 00:00:00 UTC 2015 until: Sat May 26 00:00:00 UTC 2040
```

```
Certificate fingerprints:
```

```
SHA1: 5A:8C:EF:45:D7:A6:98:59:76:7A:8C:8B:44:96:B5:78:CF:47:4B:1A
```

```
SHA256:
```

```
1B:A5:B2:AA:8C:65:40:1A:82:96:01:18:F8:0B:EC:4F:62:30:4D:83:CE:C4:71:3A:19:C3:9C:01:1E
```

```
Signature algorithm name: SHA384withRSA
```

```
Subject Public Key Algorithm: 4096-bit RSA key
```

```
Version: 3
```

```
Extensions:
```

```
#1: ObjectId: 2.5.29.19 Criticality=true
```

```
BasicConstraints:[
  CA:true
  PathLen:2147483647
]
```

```
#2: ObjectId: 2.5.29.15 Criticality=true
```

```
KeyUsage [
  DigitalSignature
  Key_CertSign
  Crl_Sign
]
```

```
]
```

```
#3: ObjectId: 2.5.29.14 Criticality=false
```

```
SubjectKeyIdentifier [
```

```
KeyIdentifier [
```

```
0000: B0 0C F0 4C 30 F4 05 58    02 48 FD 33 E5 52 AF 4B    ...L0..X.H.3.R.K
```

```
0010: 84 E3 66 52                                ..fR
```

```
]
```

```
]
```

```
Trust this certificate? [no]: yes
```

```
Certificate was added to keystore
```

```
Enter keystore password:
```

```
Owner: CN=Amazon Root CA 3, O=Amazon, C=US
```

```
Issuer: CN=Amazon Root CA 3, O=Amazon, C=US
```

```
Serial number: 66c9fd5749736663f3b0b9ad9e89e7603f24a
```

```
Valid from: Tue May 26 00:00:00 UTC 2015 until: Sat May 26 00:00:00 UTC 2040
```

```
Certificate fingerprints:
```

```
SHA1: 0D:44:DD:8C:3C:8C:1A:1A:58:75:64:81:E9:0F:2E:2A:FF:B3:D2:6E
```

```
SHA256:
```

```
18:CE:6C:FE:7B:F1:4E:60:B2:E3:47:B8:DF:E8:68:CB:31:D0:2E:BB:3A:DA:27:15:69:F5:03:43:B4
```

```
Signature algorithm name: SHA256withECDSA
```

```
Subject Public Key Algorithm: 256-bit EC (secp256r1) key
```

```
Version: 3
```

```
Extensions:
```

```
#1: ObjectId: 2.5.29.19 Criticality=true
```

```
BasicConstraints:[
```

```
CA:true
```

```
PathLen:2147483647
```

```
]
```

```
#2: ObjectId: 2.5.29.15 Criticality=true
```

```
KeyUsage [
```

```
DigitalSignature
```

```
Key_CertSign
```

```
Crl_Sign
```

```
]
```

```
#3: ObjectId: 2.5.29.14 Criticality=false
```

```
SubjectKeyIdentifier [
```

```
KeyIdentifier [
```

```
0000: AB B6 DB D7 06 9E 37 AC    30 86 07 91 70 C7 9C C4    .....7.0...p...
```

```

0010: 19 B1 78 C0                                     ...x.
]
]

Trust this certificate? [no]: yes
Certificate was added to keystore
Enter keystore password:
Owner: CN=Amazon Root CA 4, O=Amazon, C=US
Issuer: CN=Amazon Root CA 4, O=Amazon, C=US
Serial number: 66c9fd7c1bb104c2943e5717b7b2cc81ac10e
Valid from: Tue May 26 00:00:00 UTC 2015 until: Sat May 26 00:00:00 UTC 2040
Certificate fingerprints:
    SHA1: F6:10:84:07:D6:F8:BB:67:98:0C:C2:E2:44:C2:EB:AE:1C:EF:63:BE
    SHA256:
    E3:5D:28:41:9E:D0:20:25:CF:A6:90:38:CD:62:39:62:45:8D:A5:C6:95:FB:DE:A3:C2:2B:0B:FB:25
Signature algorithm name: SHA384withECDSA
Subject Public Key Algorithm: 384-bit EC (secp384r1) key
Version: 3

Extensions:

#1: ObjectId: 2.5.29.19 Criticality=true
BasicConstraints:[
    CA:true
    PathLen:2147483647
]

#2: ObjectId: 2.5.29.15 Criticality=true
KeyUsage [
    DigitalSignature
    Key_CertSign
    Crl_Sign
]

#3: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: D3 EC C7 3A 65 6E CC E1    DA 76 9A 56 FB 9C F3 86    ...:en...v.V....
0010: 6D 57 E5 81                mW..
]
]

Trust this certificate? [no]: yes
Certificate was added to keystore

```



```

Enter keystore password:
Owner: OU=Starfield Class 2 Certification Authority, O="Starfield Technologies,
  Inc.", C=US
Issuer: OU=Starfield Class 2 Certification Authority, O="Starfield
  Technologies, Inc.", C=US
Serial number: 0
Valid from: Tue Jun 29 17:39:16 UTC 2004 until: Thu Jun 29 17:39:16 UTC 2034
Certificate fingerprints:
    SHA1: AD:7E:1C:28:B0:64:EF:8F:60:03:40:20:14:C3:D0:E3:37:0E:B5:8A
    SHA256:
    14:65:FA:20:53:97:B8:76:FA:A6:F0:A9:95:8E:55:90:E4:0F:CC:7F:AA:4F:B7:C2:C8:67:75:21:FB
Signature algorithm name: SHA1withRSA (weak)
Subject Public Key Algorithm: 2048-bit RSA key
Version: 3

Extensions:

#1: ObjectId: 2.5.29.35 Criticality=false
AuthorityKeyIdentifier [
  KeyIdentifier [
0000: BF 5F B7 D1 CE DD 1F 86   F4 5B 55 AC DC D7 10 C2   ._.....[U.....
0010: 0E A9 88 E7                               ....
  ]
[OU=Starfield Class 2 Certification Authority, O="Starfield Technologies,
  Inc.", C=US]
SerialNumber: [    00]
]

#2: ObjectId: 2.5.29.19 Criticality=false
BasicConstraints:[
  CA:true
  PathLen:2147483647
]

#3: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
  KeyIdentifier [
0000: BF 5F B7 D1 CE DD 1F 86   F4 5B 55 AC DC D7 10 C2   ._.....[U.....
0010: 0E A9 88 E7                               ....
  ]
]

Warning:

```

```
The input uses the SHA1withRSA signature algorithm which is considered a
security risk. This algorithm will be disabled in a future update.
```

```
Trust this certificate? [no]: yes
Certificate was added to keystore
```

4. Configura la connessione shell Cassandra Query Language (cqlsh) e conferma di poterti connettere ad Amazon Keyspaces seguendo i passaggi indicati. [the section called “Uso di cqlsh”](#)

5. Scarica e installa. DSBulk

- a. Per scaricare DSBulk, puoi usare il seguente codice.

```
curl -OL https://downloads.datastax.com/dsbulk/dsbulk-1.8.0.tar.gz
```

- b. Quindi decomprimi il file tar e aggiungilo DSBulk al tuo PATH come mostrato nell'esempio seguente.

```
tar -zxvf dsbulk-1.8.0.tar.gz
# add the DSBulk directory to the path
export PATH=$PATH:./dsbulk-1.8.0/bin
```

- c. Crea un `application.conf` file per memorizzare le impostazioni da utilizzare. DSBulk È possibile salvare il seguente esempio come `./dsbulk_keyspaces.conf`. Sostituiscilo `localhost` con il punto di contatto del cluster Cassandra locale se non ti trovi sul nodo locale, ad esempio il nome DNS o l'indirizzo IP. Prendi nota del nome e del percorso del file, poiché dovrai specificarlo più avanti nel `dsbulk load` comando.

```
datastax-java-driver {
  basic.contact-points = [ "localhost" ]
  advanced.auth-provider {
    class = software.aws.mcs.auth.SigV4AuthProvider
    aws-region = us-east-1
  }
}
```

- d. Per abilitare il supporto SigV4, scaricate il `jar` file ombreggiato da [GitHub](#) inseritelo nella DSBulk `lib` cartella come mostrato nell'esempio seguente.

```
curl -O -L https://github.com/aws/aws-sigv4-auth-cassandra-java-driver-plugin/releases/download/4.0.6-shaded-v2/aws-sigv4-auth-cassandra-java-driver-plugin-4.0.6-shaded.jar
```

Passaggio 1: creare il file CSV di origine e una tabella di destinazione per il caricamento dei dati utilizzando DSBulk

Per questo tutorial, utilizziamo un file con valori separati da virgole (CSV) con il nome `keyspaces_sample_table.csv` come file di origine per la migrazione dei dati. Il file di esempio fornito contiene alcune righe di dati per una tabella con il nome `book_awards`

1. Crea il file sorgente. Puoi scegliere una delle seguenti opzioni:

- Scaricate il file CSV di esempio (`keyspaces_sample_table.csv`) contenuto nel seguente file di archivio [samplemigration.zip](#). Decomprimi l'archivio e prendi nota del percorso verso `keyspaces_sample_table.csv`
- Per compilare un file CSV con i propri dati memorizzati in un database Apache Cassandra, è possibile compilare il file CSV di origine utilizzando `dsbulk unload` quanto illustrato nell'esempio seguente.

```
dsbulk unload -k mykeyspace -t mytable -f ./my_application.conf  
> keyspace_sample_table.csv
```

Assicurati che il file CSV che crei soddisfi i seguenti requisiti:

- La prima riga contiene i nomi delle colonne.
- I nomi delle colonne nel file CSV di origine corrispondono ai nomi delle colonne nella tabella di destinazione.
- I dati sono delimitati da una virgola.
- Tutti i valori dei dati sono tipi di dati Amazon Keyspaces validi. Per informazioni, consulta [the section called “Tipi di dati”](#).

2. Crea lo spazio chiave e la tabella di destinazione in Amazon Keyspaces.

- a. Connettiti ad Amazon Keyspaces utilizzando `cqlsh` e sostituendo l'endpoint del servizio, il nome utente e la password nell'esempio seguente con i tuoi valori.

```
cqlsh cassandra.us-east-1.amazonaws.com 9142 -u "111122223333" -p
"wJa1rXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY" --ssl
```

- b. Crea un nuovo keyspace con il nome mostrato nell'catalogo seguente.

```
CREATE KEYSPACE catalog WITH REPLICATION = {'class': 'SingleRegionStrategy'};
```

- c. Dopo che il nuovo keyspace ha lo stato di disponibile, utilizzate il codice seguente per creare la tabella di destinazione. `book_awards` Per ulteriori informazioni sulla creazione asincrona di risorse e su come verificare se una risorsa è disponibile, consulta [the section called “Controlla lo stato di creazione del keyspace”](#)

```
CREATE TABLE catalog.book_awards (
  year int,
  award text,
  rank int,
  category text,
  book_title text,
  author text,
  publisher text,
  PRIMARY KEY ((year, award), category, rank)
);
```

Se Apache Cassandra è la tua fonte di dati originale, un modo semplice per creare la tabella di destinazione di Amazon Keyspaces con intestazioni corrispondenti consiste nel generare l'`CREATE TABLE` istruzione dalla tabella di origine, come mostrato nell'istruzione seguente.

```
cqlsh localhost 9042 -u "username" -p "password" --execute "DESCRIBE
TABLE mykeyspace.mytable;"
```

Quindi crea la tabella di destinazione in Amazon Keyspaces con i nomi delle colonne e i tipi di dati che corrispondono alla descrizione della tabella di origine di Cassandra.

Passaggio 2: prepara i dati da caricare utilizzando DSBulk

La preparazione dei dati di origine per un trasferimento efficiente è un processo in due fasi. Innanzitutto, i dati vengono randomizzati. Nella seconda fase, si analizzano i dati per determinare i valori dei `dsbulk` parametri appropriati e le impostazioni della tabella richieste.

Randomizza i dati

Il `dsbulk` comando legge e scrive i dati nello stesso ordine in cui appaiono nel file CSV. Se si utilizza il `dsbulk` comando per creare il file sorgente, i dati vengono scritti in ordine di chiave nel file CSV. Internamente, Amazon Keyspaces partiziona i dati utilizzando chiavi di partizione. Sebbene Amazon Keyspaces disponga di una logica integrata per aiutare a bilanciare il carico delle richieste per la stessa chiave di partizione, il caricamento dei dati è più rapido ed efficiente se si rende casuale l'ordine. Questo perché puoi sfruttare il bilanciamento del carico integrato che si verifica quando Amazon Keyspaces scrive su partizioni diverse.

Per distribuire le scritture tra le partizioni in modo uniforme, devi randomizzare i dati nel file sorgente. [È possibile scrivere un'applicazione per eseguire questa operazione o utilizzare uno strumento open source, come Shuf.](#) Shuf è disponibile gratuitamente sulle distribuzioni Linux, su macOS (installando `coreutils` in [homebrew](#)) e su Windows (utilizzando Windows Subsystem for Linux (WSL)). È necessario un passaggio aggiuntivo per evitare che la riga di intestazione con i nomi delle colonne venga mescolata in questo passaggio.

Per rendere casuale il file sorgente preservando l'intestazione, inserisci il codice seguente.

```
tail -n +2 keyspaces_sample_table.csv | shuf -o keyspace.table.csv && (head  
-1 keyspaces_sample_table.csv && cat keyspace.table.csv ) > keyspace.table.csv1 &&  
mv keyspace.table.csv1 keyspace.table.csv
```

Shuf riscrive i dati in un nuovo file CSV chiamato `keyspace.table.csv`. Ora puoi eliminare il `keyspaces_sample_table.csv` file, non è più necessario.

Analizza i dati

Determina la dimensione media e massima delle righe analizzando i dati.

Lo fai per i seguenti motivi:

- La dimensione media delle righe aiuta a stimare la quantità totale di dati da trasferire.
- È necessaria la dimensione media delle righe per fornire la capacità di scrittura necessaria per il caricamento dei dati.
- Puoi assicurarti che ogni riga abbia una dimensione inferiore a 1 MB, che è la dimensione massima delle righe in Amazon Keyspaces.

Note

Questa quota si riferisce alla dimensione della riga, non alla dimensione della partizione. A differenza delle partizioni Apache Cassandra, le partizioni Amazon Keyspaces possono avere dimensioni praticamente illimitate. Le chiavi di partizione e le colonne di clustering richiedono spazio di archiviazione aggiuntivo per i metadati, che è necessario aggiungere alla dimensione grezza delle righe. Per ulteriori informazioni, consulta [the section called “Stima della dimensione delle righe”](#).

Il codice seguente utilizza [AWK](#) per analizzare un file CSV e stampare la dimensione media e massima delle righe.

```
awk -F, 'BEGIN {samp=10000;max=-1;}{if(NR>1){len=length($0);t+=len;avg=t/NR;max=(len>max ? len : max)}}NR==samp{exit}END{printf("{lines: %d, average: %d bytes, max: %d bytes}\n",NR,avg,max);}' keyspace.table.csv
```

L'esecuzione di questo codice produce il seguente risultato.

```
using 10,000 samples:
{lines: 10000, avg: 123 bytes, max: 225 bytes}
```

Assicurati che la dimensione massima delle righe non superi 1 MB. In caso affermativo, devi suddividere la riga o comprimere i dati per portare la dimensione della riga al di sotto di 1 MB. Nel passaggio successivo di questo tutorial, utilizzerai la dimensione media delle righe per fornire la capacità di scrittura per la tabella.

Fase 3: Impostare la capacità di throughput per la tabella di destinazione

Questo tutorial mostra come eseguire la regolazione per DSBulk caricare i dati entro un intervallo di tempo prestabilito. Poiché sapete quante letture e scritture eseguite in anticipo, utilizzate la modalità di capacità fornita. Al termine del trasferimento dei dati, è necessario impostare la modalità di capacità della tabella in modo che corrisponda ai modelli di traffico dell'applicazione. Per ulteriori informazioni sulla gestione della capacità, consulta [Gestione delle risorse serverless](#).

Con la modalità di capacità fornita, è possibile specificare in anticipo la quantità di capacità di lettura e scrittura che si desidera fornire alla tabella. La capacità di scrittura viene fatturata ogni ora e misurata in unità di capacità di scrittura (). WCUs Ogni WCU ha una capacità di scrittura sufficiente

per supportare la scrittura di 1 KB di dati al secondo. Quando si caricano i dati, la velocità di scrittura deve essere inferiore al valore massimo WCUs (parametro: `write_capacity_units`) impostato nella tabella di destinazione.

Per impostazione predefinita, puoi eseguire il provisioning fino a 40.000 WCUs per tabella e 80.000 per WCUs tutte le tabelle del tuo account. Se hai bisogno di capacità aggiuntiva, puoi richiedere un aumento della quota nella console [Service Quotas](#). Per ulteriori informazioni sulle quote, consulta [Quote](#).

Calcola il numero medio di componenti WCUs necessari per un inserto

L'inserimento di 1 KB di dati al secondo richiede 1 WCU. Se il tuo file CSV ha 360.000 righe e desideri caricare tutti i dati in un'ora, devi scrivere 100 righe al secondo (360.000 righe/60 minuti/60 secondi = 100 righe al secondo). Se ogni riga contiene fino a 1 KB di dati, per inserire 100 righe al secondo, devi assegnarne 100 WCUs alla tabella. Se ogni riga contiene 1,5 KB di dati, ne occorrono due WCUs per inserire una riga al secondo. Pertanto, per inserire 100 righe al secondo, è necessario predisporre 200 WCUs.

Per determinare quante WCUs righe sono necessarie per inserire una riga al secondo, dividi la dimensione media delle righe in byte per 1024 e arrotonda al numero intero più vicino.

Ad esempio, se la dimensione media delle righe è 3000 byte, ne occorrono tre WCUs per inserire una riga al secondo.

```
ROUNDUP(3000 / 1024) = ROUNDUP(2.93) = 3 WCUs
```

Calcola il tempo e la capacità di caricamento dei dati

Ora che conosci la dimensione e il numero medio di righe del tuo file CSV, puoi calcolare quante sono WCUs necessarie per caricare i dati in un determinato periodo di tempo e il tempo approssimativo necessario per caricare tutti i dati nel file CSV utilizzando diverse impostazioni WCU.

Ad esempio, se ogni riga del file è di 1 KB e il file CSV contiene 1.000.000 di righe, per caricare i dati in un'ora, è necessario fornire almeno 278 righe WCUs alla tabella per quell'ora.

```
1,000,000 rows * 1 KBs = 1,000,000 KBs  
1,000,000 KBs / 3600 seconds = 277.8 KBs / second = 278 WCUs
```

Configura le impostazioni della capacità assegnata

È possibile impostare le impostazioni della capacità di scrittura di una tabella al momento della creazione della tabella o utilizzando il `ALTER TABLE` comando. Di seguito è riportata la sintassi per modificare le impostazioni della capacità assegnata a una tabella con il comando. `ALTER TABLE`

```
ALTER TABLE catalog.book_awards WITH custom_properties={'capacity_mode':
{'throughput_mode': 'PROVISIONED', 'read_capacity_units': 100, 'write_capacity_units':
278}} ;
```

Per il riferimento completo alla lingua, vedere e. [the section called “CREATE TABLE”](#) [the section called “ALTER TABLE”](#)

Passaggio 4: Configurare **DSBulk** le impostazioni per caricare i dati dal file CSV alla tabella di destinazione

Questa sezione descrive i passaggi necessari DSBulk per configurare il caricamento dei dati su Amazon Keyspaces. La configurazione DSBulk viene effettuata utilizzando un file di configurazione. Il file di configurazione viene specificato direttamente dalla riga di comando.

1. Crea un file di DSBulk configurazione per la migrazione ad Amazon Keyspaces, in questo esempio utilizziamo il nome del file. `dsbulk_keyspaces.conf` Specificate le seguenti impostazioni nel file DSBulk di configurazione.
 - a. *PlainTextAuthProvider*— Crea il provider di autenticazione con la `PlainTextAuthProvider` classe. `ServiceUsername` e `ServicePassword` deve corrispondere al nome utente e alla password ottenuti al momento della generazione delle credenziali specifiche del servizio seguendo la procedura riportata in. [the section called “Crea credenziali di accesso programmatico”](#)
 - b. *local-datacenter*— Imposta il valore `local-datacenter` per il quale Regione AWS ti stai connettendo. Ad esempio, se l'applicazione si connette a `cassandra.us-east-1.amazonaws.com`, imposta il data center locale su `us-east-1`. Per tutte le opzioni disponibili Regioni AWS, vedi [the section called “Endpoint di servizio”](#). Per evitare repliche, imposta `suslow-replica-avoidance`. `false`
 - c. *SSLEngineFactory*— Per configurare SSL/TLS, inizializza `SSLEngineFactory` aggiungendo una sezione nel file di configurazione con una sola riga che specifica la classe con. `class = DefaultSslEngineFactory` Fornisci il percorso `cassandra_truststore.jks` e la password che hai creato in precedenza.

- d. *consistency*— Imposta il livello di coerenza su `LOCAL QUORUM`. Altri livelli di coerenza di scrittura non sono supportati, per ulteriori informazioni, vedere [the section called “Livelli di coerenza di Cassandra supportati”](#).
- e. Il numero di connessioni per pool è configurabile nel driver Java. Per questo esempio, imposta su `advanced.connection.pool.local.size 3`.

Di seguito è riportato il file di configurazione di esempio completo.

```
datastax-java-driver {
  basic.contact-points = [ "cassandra.us-east-1.amazonaws.com:9142" ]
  advanced.auth-provider {
    class = PlainTextAuthProvider
    username = "ServiceUserName"
    password = "ServicePassword"
  }

  basic.load-balancing-policy {
    local-datacenter = "us-east-1"
    slow-replica-avoidance = false
  }

  basic.request {
    consistency = LOCAL_QUORUM
    default-idempotence = true
  }
  advanced.ssl-engine-factory {
    class = DefaultSslEngineFactory
    truststore-path = "./cassandra_truststore.jks"
    truststore-password = "my_password"
    hostname-validation = false
  }
  advanced.connection.pool.local.size = 3
}
```

2. Esamine i parametri del `DSBulk load` comando.

- a. *executor.maxPerSecond*— Il numero massimo di righe che il comando `load` tenta di elaborare contemporaneamente al secondo. Se non è impostata, questa impostazione viene disabilitata con `-1`.

Imposta in `executor.maxPerSecond` base al numero di WCUs elementi che hai assegnato alla tabella di destinazione. Il valore `executor.maxPerSecond` del `load` comando non è un limite, ma una media obiettivo. Ciò significa che può (e spesso succede) superare il numero impostato. Per consentire l'insorgenza di interruzioni e assicurarvi che sia disponibile una capacità sufficiente per gestire le richieste di caricamento dei dati, impostate `executor.maxPerSecond` il 90% della capacità di scrittura della tabella.

```
executor.maxPerSecond = WCUs * .90
```

In questo tutorial, abbiamo impostato su `executor.maxPerSecond` 5.

 Note

Se stai usando DSBulk 1.6.0 o versioni successive, puoi usare `dsbulk.engine.maxConcurrentQueries` invece.

- b. Configura questi parametri aggiuntivi per il DSBulk `load` comando.
- *batch-mode*— Questo parametro indica al sistema di raggruppare le operazioni per chiave di partizione. Si consiglia di disabilitare la modalità batch, poiché può causare scenari e cause `WriteThrottleEvents` con tasti di scelta rapida.
 - *driver.advanced.retry-policy-max-retries*— Ciò determina quante volte riprovare un'interrogazione non riuscita. Se non è impostata, l'impostazione predefinita è 10. È possibile modificare questo valore in base alle esigenze.
 - *driver.basic.request.timeout*— Il tempo in minuti in cui il sistema attende la restituzione di una query. Se non è impostata, l'impostazione predefinita è «5 minuti». È possibile modificare questo valore in base alle esigenze.

Passaggio 5: Esegui il DSBulk **load** comando per caricare i dati dal file CSV nella tabella di destinazione

Nella fase finale di questo tutorial, carichi i dati in Amazon Keyspaces.

Per eseguire il DSBulk `load` comando, completa i seguenti passaggi.

1. Esegui il codice seguente per caricare i dati dal tuo file csv nella tabella Amazon Keyspaces. Assicurati di aggiornare il percorso del file di configurazione dell'applicazione che hai creato in precedenza.

```
dsbulk load -f ./dsbulk_keyspaces.conf --connector.csv.url keyspace.table.csv
--header true --batch.mode DISABLED --executor.maxPerSecond 5 --
driver.basic.request.timeout "5 minutes" --driver.advanced.retry-policy.max-
retries 10 -k catalog -t book_awards
```

2. L'output include la posizione di un file di registro che riporta i dettagli delle operazioni riuscite e non riuscite. Il file è memorizzato nella seguente directory.

```
Operation directory: /home/user_name/logs/UNLOAD_20210308-202317-801911
```

3. Le voci del file di registro includeranno le metriche, come nell'esempio seguente. Assicurati che il numero di righe sia coerente con il numero di righe del tuo file csv.

total	failed	rows/s	p50ms	p99ms	p999ms
200	0	200	21.63	21.89	21.89

Important

Ora che hai trasferito i dati, regola le impostazioni della modalità di capacità della tabella di destinazione in modo che corrispondano ai normali modelli di traffico dell'applicazione. Fino a quando non la modifichi, ti verranno addebitati i costi in base alla tariffa oraria per la capacità assegnata. Per ulteriori informazioni, consulta [the section called “Configura le modalità di capacità di lettura/scrittura”](#).

Accesso ad Amazon Keyspaces (per Apache Cassandra)

Puoi accedere ad Amazon Keyspaces utilizzando la console AWS CloudShell, a livello di codice eseguendo un `cqlsh` client, l' AWS SDK o utilizzando un driver Cassandra con licenza Apache 2.0. Amazon Keyspaces supporta driver e client compatibili con Apache Cassandra 3.11.2. Prima di accedere ad Amazon Keyspaces, devi completare la configurazione AWS Identity and Access Management e quindi concedere le autorizzazioni di accesso all'identità IAM ad Amazon Keyspaces.

Configurazione AWS Identity and Access Management

Registrati per un Account AWS

Se non ne hai uno Account AWS, completa i seguenti passaggi per crearne uno.

Per iscriverti a un Account AWS

1. Apri la <https://portal.aws.amazon.com/billing/registrazione>.
2. Segui le istruzioni online.

Nel corso della procedura di registrazione riceverai una telefonata o un messaggio di testo e ti verrà chiesto di inserire un codice di verifica attraverso la tastiera del telefono.

Quando ti iscrivi a un Account AWS, Utente root dell'account AWS viene creato un. L'utente root dispone dell'accesso a tutte le risorse e tutti i Servizi AWS nell'account. Come best practice di sicurezza, assegna l'accesso amministrativo a un utente e utilizza solo l'utente root per eseguire le [attività che richiedono l'accesso di un utente root](#).

AWS ti invia un'email di conferma dopo il completamento della procedura di registrazione. In qualsiasi momento, puoi visualizzare l'attività corrente del tuo account e gestirlo accedendo a <https://aws.amazon.com/> e scegliendo Il mio account.

Crea un utente con accesso amministrativo

Dopo esserti registrato Account AWS, proteggi Utente root dell'account AWS AWS IAM Identity Center, abilita e crea un utente amministrativo in modo da non utilizzare l'utente root per le attività quotidiane.

Proteggi i tuoi Utente root dell'account AWS

1. Accedi [Console di gestione AWS](#) come proprietario dell'account scegliendo Utente root e inserendo il tuo indirizzo Account AWS email. Nella pagina successiva, inserisci la password.

Per informazioni sull'accesso utilizzando un utente root, consulta la pagina [Signing in as the root user](#) della Guida per l'utente di Accedi ad AWS .

2. Abilita l'autenticazione a più fattori (MFA) per l'utente root.

Per istruzioni, consulta [Abilitare un dispositivo MFA virtuale per l'utente Account AWS root \(console\)](#) nella Guida per l'utente IAM.

Crea un utente con accesso amministrativo

1. Abilita Centro identità IAM.

Per istruzioni, consulta [Abilitazione di AWS IAM Identity Center](#) nella Guida per l'utente di AWS IAM Identity Center .

2. In IAM Identity Center, assegna l'accesso amministrativo a un utente.

Per un tutorial sull'utilizzo di IAM Identity Center directory come fonte di identità, consulta [Configurare l'accesso utente con l'impostazione predefinita IAM Identity Center directory](#) nella Guida per l'AWS IAM Identity Center utente.

Accesso come utente amministratore

- Per accedere con l'utente IAM Identity Center, utilizza l'URL di accesso che è stato inviato al tuo indirizzo e-mail quando hai creato l'utente IAM Identity Center.

Per informazioni sull'accesso utilizzando un utente IAM Identity Center, consulta [AWS Accedere al portale di accesso](#) nella Guida per l'Accedi ad AWS utente.

Assegna l'accesso a ulteriori utenti

1. In IAM Identity Center, crea un set di autorizzazioni conforme alla best practice dell'applicazione di autorizzazioni con il privilegio minimo.

Segui le istruzioni riportate nella pagina [Creazione di un set di autorizzazioni](#) nella Guida per l'utente di AWS IAM Identity Center .

2. Assegna al gruppo prima gli utenti e poi l'accesso con autenticazione unica (Single Sign-On).

Per istruzioni, consulta [Aggiungere gruppi](#) nella Guida per l'utente di AWS IAM Identity Center .

Configurazione di Amazon Keyspaces

[L'accesso alle risorse di Amazon Keyspaces è gestito tramite IAM.](#) Utilizzando IAM, puoi associare policy a utenti, ruoli e identità federate IAM che concedono autorizzazioni di lettura e scrittura a risorse specifiche in Amazon Keyspaces.

Per iniziare a concedere le autorizzazioni a un'identità IAM, puoi utilizzare una delle policy AWS gestite per Amazon Keyspaces:

- [AmazonKeyspacesFullAccess](#)— questa politica concede le autorizzazioni per accedere a tutte le risorse in Amazon Keyspaces con accesso completo a tutte le funzionalità.
- [AmazonKeyspacesReadOnlyAccess_v2](#): questa politica concede autorizzazioni di sola lettura ad Amazon Keyspaces.

Per una spiegazione dettagliata delle azioni definite nelle politiche gestite, consulta. [the section called “AWS politiche gestite”](#)

Per limitare l'ambito delle azioni che un'identità IAM può eseguire o limitare le risorse a cui l'identità può accedere, puoi creare una policy personalizzata che utilizzi la policy `AmazonKeyspacesFullAccess` gestita come modello e rimuovere tutte le autorizzazioni non necessarie. Puoi anche limitare l'accesso a spazi chiave o tabelle specifici. Per ulteriori informazioni su come limitare le azioni o limitare l'accesso a risorse specifiche in Amazon Keyspaces, consulta. [the section called “Come funziona Amazon Keyspaces con IAM”](#)

Per accedere ad Amazon Keyspaces dopo aver creato Account AWS e creato una policy che concede un accesso tramite identità IAM ad Amazon Keyspaces, continua con una delle seguenti sezioni:

- [Utilizzo della console](#)
- [Utilizzando AWS CloudShell](#)

Accesso ad Amazon Keyspaces tramite la console

Puoi accedere alla console per Amazon Keyspaces all'indirizzo. <https://console.aws.amazon.com/keyspaces/home> Per ulteriori informazioni sull' Console di gestione AWS accesso, consulta [Controlling IAM users access to Console di gestione AWS nella](#) IAM User Guide.

Puoi utilizzare la console per eseguire le seguenti operazioni in Amazon Keyspaces:

- Crea, elimina e gestisci spazi chiave e tabelle.
- Monitora le metriche importanti della tabella nella scheda Monitor di una tabella:
 - Dimensioni fatturabili della tabella (byte)
 - Metriche della capacità
- Esegui query utilizzando l'editor CQL, ad esempio inserisci, aggiorna ed elimina dati.
- Modifica la configurazione del partizionatore dell'account.
- Visualizza le metriche relative alle prestazioni e agli errori per l'account sulla dashboard.

Per informazioni su come creare uno spazio di chiavi e una tabella Amazon Keyspaces e configurarlo con dati applicativi di esempio, consulta. [Guida introduttiva ad Amazon Keyspaces \(per Apache Cassandra\)](#)

Utilizzo AWS CloudShell per accedere ad Amazon Keyspaces

AWS CloudShell è una shell preautenticata basata su browser che è possibile avviare direttamente da. Console di gestione AWS È possibile eseguire AWS CLI comandi AWS sui servizi utilizzando la shell preferita (Bash o Z shell). PowerShell Per utilizzare Amazon Keyspaces utilizzando `cqlsh`, devi installare il. `cqlsh-expansion` Per le istruzioni `cqlsh-expansion` di installazione, consulta [the section called "Utilizzo di cqlsh-expansion"](#).

Si [avvia AWS CloudShell da Console di gestione AWS](#), e AWS le credenziali utilizzate per accedere alla console sono automaticamente disponibili in una nuova sessione di shell. Questa preautenticazione degli AWS CloudShell utenti consente di saltare la configurazione delle credenziali quando si interagisce con servizi AWS come Amazon Keyspaces utilizzando `cqlsh` o la AWS CLI versione 2 (preinstallata nell'ambiente di calcolo della shell).

Ottenere le autorizzazioni IAM per AWS CloudShell

Utilizzando le risorse di gestione degli accessi fornite da AWS Identity and Access Management, gli amministratori possono concedere le autorizzazioni agli utenti IAM in modo che possano accedere AWS CloudShell e utilizzare le funzionalità dell'ambiente.

Il modo più rapido per un amministratore di concedere l'accesso agli utenti è tramite una AWS policy gestita. Una [policy gestita da AWS](#) è una policy autonoma che viene creata e amministrata da AWS. La seguente policy AWS gestita per CloudShell può essere allegata alle identità IAM:

- **AWSCloudShellFullAccess**: concede l'autorizzazione all'uso AWS CloudShell con accesso completo a tutte le funzionalità.

Se desideri limitare l'ambito di azioni che un utente IAM può eseguire AWS CloudShell, puoi creare una policy personalizzata che utilizzi la policy **AWSCloudShellFullAccess** gestita come modello. Per ulteriori informazioni sulla limitazione delle azioni disponibili per gli utenti in CloudShell, consulta [Gestire AWS CloudShell l'accesso e l'utilizzo con le politiche IAM](#) nella Guida per l'AWS CloudShell utente.

Note

La tua identità IAM richiede anche una policy che conceda l'autorizzazione a effettuare chiamate verso Amazon Keyspaces.

Puoi utilizzare una policy AWS gestita per consentire alla tua identità IAM di accedere ad Amazon Keyspaces oppure iniziare con la policy gestita come modello e rimuovere le autorizzazioni che non ti servono. Puoi anche limitare l'accesso a spazi chiave e tabelle specifici per creare una policy personalizzata. La seguente policy gestita per Amazon Keyspaces può essere allegata alle identità IAM:

- [AmazonKeyspacesFullAccess](#)— Questa politica concede l'autorizzazione a utilizzare Amazon Keyspaces con accesso completo a tutte le funzionalità.

Per una spiegazione dettagliata delle azioni definite nella politica gestita, consulta. [the section called “AWS politiche gestite”](#)

Per ulteriori informazioni su come limitare le azioni o limitare l'accesso a risorse specifiche in Amazon Keyspaces, consulta [the section called “Come funziona Amazon Keyspaces con IAM”](#)

Interazione con Amazon Keyspaces tramite AWS CloudShell

Dopo l'avvio AWS CloudShell da Console di gestione AWS, puoi iniziare immediatamente a interagire con Amazon Keyspaces utilizzando `cqlsh` o l'interfaccia a riga di comando. Se non l'hai ancora installato `cqlsh-expansion`, consulta [the section called “Utilizzo di `cqlsh-expansion`”](#) la procedura dettagliata.

Note

Quando usi `cqlsh-expansion` in AWS CloudShell, non è necessario configurare le credenziali prima di effettuare chiamate, perché sei già autenticato all'interno della shell.

Connettiti ad Amazon Keyspaces e crea un nuovo keyspace. Quindi leggi da una tabella di sistema per confermare che il keyspace è stato creato utilizzando AWS CloudShell

1. Da Console di gestione AWS, puoi avviarlo CloudShell scegliendo le seguenti opzioni disponibili nella barra di navigazione:
 - Scegli l' CloudShell icona.
 - Inizia a digitare «cloudshell» nella casella di ricerca, quindi scegli l'opzione. CloudShell
2. Puoi stabilire una connessione ad Amazon Keyspaces utilizzando il seguente comando. Assicurati di sostituirlo `cassandra.us-east-1.amazonaws.com` con l'endpoint corretto per la tua regione.

```
cqlsh-expansion cassandra.us-east-1.amazonaws.com 9142 --ssl
```

Se la connessione è riuscita, dovresti vedere un output simile a quello dell'esempio seguente.

```
Connected to Amazon Keyspaces at cassandra.us-east-1.amazonaws.com:9142
[cqlsh 6.1.0 | Cassandra 3.11.2 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
cqlsh current consistency level is ONE.
cqlsh>
```

3. Crea un nuovo keyspace con il nome `mykeyspace`. È possibile utilizzare il seguente comando per farlo.

```
CREATE KEYSPACE mykeyspace WITH REPLICATION = {'class': 'SingleRegionStrategy'};
```

4. Per confermare che il keyspace è stato creato, potete leggere da una tabella di sistema usando il seguente comando.

```
SELECT * FROM system_schema_mcs.keyspaces WHERE keyspace_name = 'mykeyspace';
```

Se la chiamata ha esito positivo, la riga di comando visualizza una risposta del servizio simile al seguente output:

```
keyspace_name | durable_writes | replication
-----+-----
+-----+-----+-----
mykeyspace    |                True | {'class':
'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '3'}

(1 rows)
```

Crea credenziali per l'accesso programmatico ad Amazon Keyspaces

Per fornire agli utenti e alle applicazioni le credenziali per l'accesso programmatico alle risorse di Amazon Keyspaces, puoi effettuare una delle seguenti operazioni:

- Crea credenziali specifiche per il servizio simili al nome utente e alla password tradizionali utilizzati da Cassandra per l'autenticazione e la gestione degli accessi. AWS Le credenziali specifiche del servizio sono associate a un utente specifico AWS Identity and Access Management (IAM) e possono essere utilizzate solo per il servizio per cui sono state create. Per ulteriori informazioni, consulta [Using IAM with Amazon Keyspaces \(per Apache Cassandra\)](#) nella IAM User Guide.

Warning

Gli utenti IAM dispongono di credenziali a lungo termine, il che rappresenta un rischio per la sicurezza. Per ridurre questo rischio, si consiglia di fornire a questi utenti solo

le autorizzazioni necessarie per eseguire l'attività e di rimuoverli quando non sono più necessari.

- Per una maggiore sicurezza, consigliamo di creare identità IAM da utilizzare in tutti i AWS servizi e di utilizzare credenziali temporanee. Il plug-in di autenticazione Amazon Keyspaces SigV4 per i driver client Cassandra consente di autenticare le chiamate ad Amazon Keyspaces utilizzando chiavi di accesso IAM anziché nome utente e password. [Per ulteriori informazioni su come il plug-in Amazon Keyspaces SigV4 consente agli utenti, ai ruoli e alle identità federate IAM di autenticarsi nelle richieste API di Amazon Keyspaces, consulta il processo Signature Version 4 \(SigV4\).AWS](#)

Puoi scaricare i plugin SigV4 dalle seguenti posizioni.

- Java: <https://github.com/aws/aws-sigv4-auth-cassandra-java-driver-plugin>
- Node.js: <https://github.com/aws/aws-sigv4-auth-cassandra-nodejs-driver-plugin>.
- Python: <https://github.com/aws/aws-sigv4-auth-cassandra-python-driver-plugin>.
- Vai: <https://github.com/aws/aws-sigv4-auth-cassandra-gocql-driver-plugin>.

Per esempi di codice che mostrano come stabilire connessioni utilizzando il plug-in di autenticazione SigV4, vedi. [the section called “Utilizzo di un driver client Cassandra”](#)

Argomenti

- [Crea credenziali specifiche del servizio per l'accesso programmatico ad Amazon Keyspaces](#)
- [Crea e configura AWS credenziali per Amazon Keyspaces](#)

Crea credenziali specifiche del servizio per l'accesso programmatico ad Amazon Keyspaces

Le credenziali specifiche del servizio sono simili al nome utente e alla password tradizionali utilizzati da Cassandra per l'autenticazione e la gestione degli accessi. Le credenziali specifiche del servizio consentono agli utenti IAM di accedere a un servizio specifico. AWS Queste credenziali a lungo termine non possono essere utilizzate per accedere ad altri servizi. AWS Sono associate a un utente IAM specifico e non possono essere utilizzate da altri utenti IAM.

Important

Le credenziali specifiche del servizio sono credenziali a lungo termine associate a uno specifico utente IAM e possono essere utilizzate solo per il servizio per cui sono state create.

Per concedere ai ruoli IAM o alle identità federate le autorizzazioni per accedere a tutte le tue AWS risorse utilizzando credenziali temporanee, devi utilizzare [AWS l'autenticazione con il plug-in di autenticazione SigV4](#) per Amazon Keyspaces.

Utilizza una delle seguenti procedure per generare credenziali specifiche del servizio.

Console

Crea credenziali specifiche del servizio utilizzando la console

1. Accedi a Console di gestione AWS e apri la console all' AWS Identity and Access Management indirizzo. <https://console.aws.amazon.com/iam/home>
2. Nel pannello di navigazione, scegli Utenti, quindi scegli l'utente che hai creato in precedenza che dispone delle autorizzazioni Amazon Keyspaces (policy allegata).
3. Seleziona Security Credentials (Credenziali di sicurezza). In Credenziali per Amazon Keyspaces, scegli Genera credenziali per generare le credenziali specifiche del servizio.

Le credenziali specifiche del servizio sono ora disponibili. Questa è l'unica volta che puoi scaricare o visualizzare la password. Non puoi recuperarla successivamente. Tuttavia, è possibile reimpostare la password in qualsiasi momento. Salva l'utente e la password in una posizione sicura, perché ne avrai bisogno in un secondo momento.

CLI

Crea credenziali specifiche per il servizio utilizzando il AWS CLI

Prima di generare credenziali specifiche per il servizio, è necessario scaricare, installare e configurare (): AWS Command Line Interface AWS CLI

1. [Scaricatele all' AWS CLI indirizzo http://aws.amazon.com/cli](http://aws.amazon.com/cli).



Note

AWS CLI Funziona su Windows, macOS o Linux.

2. Segui le istruzioni per l'[installazione della AWS CLI](#) e la [configurazione della AWS CLI nella Guida](#) per l'utente.AWS Command Line Interface

3. Utilizzando AWS CLI, esegui il comando seguente per generare credenziali specifiche del servizio per l'utente `alice`, in modo che possa accedere ad Amazon Keyspaces.

```
aws iam create-service-specific-credential \  
  --user-name alice \  
  --service-name cassandra.amazonaws.com
```

L'output sarà simile al seguente.

```
{  
  "ServiceSpecificCredential": {  
    "CreateDate": "2019-10-09T16:12:04Z",  
    "ServiceName": "cassandra.amazonaws.com",  
    "ServiceUserName": "alice-at-111122223333",  
    "ServicePassword": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",  
    "ServiceSpecificCredentialId": "ACCAYFI33SINPGJEYESF",  
    "UserName": "alice",  
    "Status": "Active"  
  }  
}
```

Nell'output, annota i valori per `e. ServiceUserName ServicePassword`. Salva questi valori in un luogo sicuro, perché ti serviranno in seguito.

Important

Questa è l'unica volta che `ServicePassword` saranno disponibili per te.

Crea e configura AWS credenziali per Amazon Keyspaces

Per accedere ad Amazon Keyspaces in modo programmatico con AWS CLI, l'AWS SDK o con i driver client Cassandra e il plug-in SigV4, è necessario un utente IAM con chiavi di accesso. Quando usi Amazon Keyspaces a livello di codice, fornisci le tue chiavi di AWS accesso per AWS verificare la tua identità nelle chiamate programmatiche. Le tue chiavi di accesso sono costituite da un ID di chiave di accesso (ad `AKIAIOSFODNN7` esempio, `EXAMPLE`) e da una chiave di accesso segreta (ad esempio, `wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`). Questo argomento illustra i passaggi necessari di questo processo.

Le migliori pratiche di sicurezza consigliano di creare utenti IAM con autorizzazioni limitate e di associare invece i ruoli IAM alle autorizzazioni necessarie per eseguire attività specifiche. Gli utenti IAM possono quindi assumere temporaneamente ruoli IAM per eseguire le attività richieste. Ad esempio, gli utenti IAM del tuo account che utilizzano la console Amazon Keyspaces possono passare a un ruolo per utilizzare temporaneamente le autorizzazioni del ruolo nella console. Gli utenti abbandonano le loro autorizzazioni originali e assumono le autorizzazioni assegnate al ruolo. Quando gli utenti escono dal ruolo, le autorizzazioni originali vengono ripristinate. Le credenziali utilizzate dagli utenti per assumere il ruolo sono temporanee. Al contrario, gli utenti IAM dispongono di credenziali a lungo termine, il che rappresenta un rischio per la sicurezza se, invece di assumere ruoli, dispongono di autorizzazioni assegnate loro direttamente. Per ridurre questo rischio, si consiglia di fornire a questi utenti solo le autorizzazioni necessarie per eseguire l'attività e di rimuoverli quando non sono più necessari. Per ulteriori informazioni sui ruoli, consulta [Scenari comuni per i ruoli: utenti, applicazioni e servizi](#) nella Guida per l'utente IAM.

Argomenti

- [Credenziali richieste dal AWS CLI, dall' AWS SDK o dal plug-in Amazon Keyspaces SigV4 per i driver client Cassandra](#)
- [Crea credenziali temporanee per connetterti ad Amazon Keyspaces utilizzando un ruolo IAM e il plug-in SigV4](#)
- [Crea un utente IAM per l'accesso programmatico ad Amazon Keyspaces nel tuo account AWS](#)
- [Crea nuove chiavi di accesso per un utente IAM](#)
- [Memorizza le chiavi di accesso per l'accesso programmatico](#)

Credenziali richieste dal AWS CLI, dall' AWS SDK o dal plug-in Amazon Keyspaces SigV4 per i driver client Cassandra

Le seguenti credenziali sono necessarie per autenticare l'utente o il ruolo IAM:

AWS_ACCESS_KEY_ID

Specifica una chiave di AWS accesso associata a un utente o a un ruolo IAM.

La chiave di accesso `aws_access_key_id` è necessaria per connettersi ad Amazon Keyspaces in modo programmatico.

AWS_SECRET_ACCESS_KEY

Specifica la chiave segreta associata alla chiave di accesso. Si tratta essenzialmente della "password" per la chiave di accesso.

`aws_secret_access_key` È necessario per connettersi ad Amazon Keyspaces a livello di codice.

AWS_SESSION_TOKEN— Facoltativo

Specifica il valore del token di sessione richiesto se si utilizzano credenziali di sicurezza temporanee recuperate direttamente dalle operazioni AWS Security Token Service . Per ulteriori informazioni, consulta [the section called “Crea credenziali temporanee per connetterti ad Amazon Keyspaces”](#).

Se ti connetti con un utente IAM, non `aws_session_token` è necessario.

Crea credenziali temporanee per connetterti ad Amazon Keyspaces utilizzando un ruolo IAM e il plug-in SigV4

Il modo consigliato per accedere ad Amazon Keyspaces a livello di codice consiste nell'utilizzare [credenziali temporanee](#) per l'autenticazione con il plug-in SigV4. In molti scenari, non è necessaria una chiave di accesso a lungo termine a validità illimitata (come accade invece per gli utenti IAM). Puoi invece creare un ruolo IAM e generare credenziali di sicurezza temporanee. Tali credenziali sono composte dall'ID della chiave di accesso e dalla chiave di accesso segreta, ma includono anche un token di sicurezza che ne indica la scadenza. Per ulteriori informazioni su come utilizzare i ruoli IAM anziché le chiavi di accesso a lungo termine, consulta [Switching to an IAM role \(AWS API\)](#).

Per iniziare con le credenziali temporanee, devi prima creare un ruolo IAM.

Crea un ruolo IAM che garantisca l'accesso in sola lettura ad Amazon Keyspaces

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 Ruoli, quindi Crea ruolo.
3. Nella pagina Crea ruolo, in Seleziona il tipo di entità affidabile, scegli AWS servizio. In Scegli un caso d'uso, scegli Amazon EC2, quindi scegli Avanti.
4. Nella pagina Aggiungi autorizzazioni, in Politiche di autorizzazione, scegli Amazon Keyspaces Read Only Access dall'elenco delle policy, quindi scegli Avanti.

5. Nella pagina Nome, rivedi e crea, inserisci un nome per il ruolo e consulta le sezioni Seleziona entità affidabili e Aggiungi autorizzazioni. In questa pagina puoi anche aggiungere tag opzionali per il ruolo. Quando hai finito, seleziona Crea ruolo. Ricorda questo nome perché ti servirà quando lancerai la tua EC2 istanza Amazon.

Per utilizzare credenziali di sicurezza temporanee nel codice, richiami a livello di codice un' AWS Security Token Service API simile a un'API AssumeRole ed estrai le credenziali e il token di sessione risultanti dal tuo ruolo IAM creato nel passaggio precedente. Questi valori vengono quindi utilizzati come credenziali per le chiamate successive a AWS. L'esempio seguente mostra lo pseudocodice su come utilizzare le credenziali di sicurezza temporanee:

```
assumeRoleResult = AssumeRole(role-arn);
tempCredentials = new SessionAWSCredentials(
    assumeRoleResult.AccessKeyId,
    assumeRoleResult.SecretAccessKey,
    assumeRoleResult.SessionToken);
cassandraRequest = CreateAmazoncassandraClient(tempCredentials);
```

Per un esempio che implementa credenziali temporanee utilizzando il driver Python per accedere ad Amazon Keyspaces, consulta. [???](#)

Per dettagli su come richiamare AssumeRole, GetFederationToken e altre operazioni API, consulta la [Documentazione di riferimento delle API AWS Security Token Service](#). Per informazioni su come ottenere le credenziali di sicurezza provvisorie e il token di sessione dal risultato, consulta la documentazione dell'SDK in uso. Puoi trovare la documentazione per tutti nella AWS SDKs [pagina di AWS documentazione](#) principale, nella sezione e Toolkits. SDKs

Crea un utente IAM per l'accesso programmatico ad Amazon Keyspaces nel tuo account AWS

Per ottenere le credenziali per l'accesso programmatico ad Amazon Keyspaces con AWS CLI, l' AWS SDK o il plug-in SigV4, devi prima creare un utente o un ruolo IAM. Il processo di creazione di un utente IAM e di configurazione di tale utente IAM per l'accesso programmatico ad Amazon Keyspaces è illustrato nei seguenti passaggi:

1. Crea l'utente negli Console di gestione AWS strumenti per Windows PowerShell o utilizzando un' AWS operazione API. AWS CLI Se si crea l'utente in Console di gestione AWS, le credenziali vengono create automaticamente.

2. Se si crea l'utente a livello di codice, è necessario creare una chiave di accesso (ID chiave di accesso e chiave di accesso segreta) per quell'utente in un passaggio aggiuntivo.
3. Concedi all'utente le autorizzazioni per accedere ad Amazon Keyspaces.

Per informazioni sulle autorizzazioni necessarie per creare un utente IAM, consulta [Autorizzazioni necessarie per accedere alle](#) risorse IAM.

Console

Crea un utente IAM con accesso programmatico (console)

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
2. Nel pannello di navigazione seleziona Utenti, quindi seleziona Aggiungi utenti.
3. Digita un nome utente per il nuovo utente. Questo è il nome di accesso per AWS.

Note

I nomi utente possono essere una combinazione di un massimo di 64 lettere, cifre e i seguenti caratteri: più (+), uguale (=), virgola (,), punto (.), chiocciola (@), trattino basso (_) e trattino (-). I nomi devono essere univoci nell'account. Non fanno distinzione tra maiuscole e minuscole. Ad esempio, non è possibile creare due utenti chiamati TESTUSER e testuser.

4. Seleziona Chiave di accesso - Accesso programmatico per creare una chiave di accesso per il nuovo utente. Puoi visualizzare o scaricare la chiave di accesso quando arrivi alla pagina finale.

Scegli Successivo: autorizzazioni.

5. Nella pagina Imposta autorizzazioni, scegli Allega direttamente le politiche esistenti per assegnare le autorizzazioni al nuovo utente.

Questa opzione mostra l'elenco delle politiche AWS gestite e gestite dai clienti disponibili nel tuo account. Puoi keyspaces accedere al campo di ricerca per visualizzare solo le politiche relative ad Amazon Keyspaces.

Per Amazon Keyspaces, le policy gestite disponibili sono `AmazonKeyspacesFullAccess` e `AmazonKeyspacesReadOnlyAccess`. Per ulteriori informazioni su ciascuna politica, consulta [the section called “AWS politiche gestite”](#).

A scopo di test e per seguire i tutorial di connessione, seleziona la `AmazonKeyspacesReadOnlyAccess` policy per il nuovo utente IAM. Nota: come best practice, ti consigliamo di seguire il principio del privilegio minimo e di creare politiche personalizzate che limitino l'accesso a risorse specifiche e consentano solo le azioni richieste. Per ulteriori informazioni sulle policy IAM e per visualizzare esempi di policy per Amazon Keyspaces, consulta [the section called “Politiche basate sull'identità di Amazon Keyspaces”](#). Dopo aver creato politiche di autorizzazione personalizzate, associa le politiche ai ruoli e consenti agli utenti di assumere temporaneamente i ruoli appropriati.

Scegli Successivo: Tag.

6. Nella pagina Aggiungi tag (opzionale) puoi aggiungere tag per l'utente oppure scegliere Avanti: revisione.
7. Nella pagina Revisione puoi vedere tutte le scelte che hai fatto fino a questo punto. Quando sei pronto per procedere, scegli Crea utente.
8. Per visualizzare le chiavi di accesso dell'utente (chiave di accesso IDs e chiavi di accesso segrete), scegli Mostra accanto alla password e alla chiave di accesso. Per salvare le chiavi di accesso, scegliere Download .csv (Scarica .csv) e quindi salvare il file in una posizione sicura.

 Important

Questa è la tua unica opportunità per visualizzare o scaricare le chiavi di accesso segrete e hai bisogno di queste informazioni prima che possano utilizzare il plugin SigV4. Salva i nuovi ID chiave di accesso e Secret Access Key dell'utente in un luogo sicuro. Successivamente a questa fase non sarà più possibile accedere alle chiavi segrete.

CLI

Crea un utente IAM con accesso programmatico ()AWS CLI

1. Crea un utente con il AWS CLI codice seguente.

- [aws iam create-user](#)
2. Concedi all'utente l'accesso programmatico. Ciò richiede chiavi di accesso, che possono essere generate nei seguenti modi.
 - AWS CLI: [aws iam create-access-key](#)
 - Strumenti per Windows PowerShell: [New-IAMAccessKey](#)
 - API IAM: [CreateAccessKey](#)

 Important

Questa è la tua unica opportunità per visualizzare o scaricare le chiavi di accesso segrete e hai bisogno di queste informazioni prima che possano utilizzare il plugin SigV4. Salva i nuovi ID chiave di accesso e Secret Access Key dell'utente in un luogo sicuro. Successivamente a questa fase non sarà più possibile accedere alle chiavi segrete.

3. Allega la `AmazonKeyspacesReadOnlyAccess` policy all'utente che definisce le autorizzazioni dell'utente. Nota: come procedura consigliata, si consiglia di gestire le autorizzazioni degli utenti aggiungendo l'utente a un gruppo e allegando una politica al gruppo anziché allegarla direttamente a un utente.
 - AWS CLI: [aws iam attach-user-policy](#)

Crea nuove chiavi di accesso per un utente IAM

Se hai già un utente IAM, puoi creare nuove chiavi di accesso in qualsiasi momento. Per ulteriori informazioni sulla gestione delle chiavi, ad esempio su come aggiornare le chiavi di accesso, consulta [Gestione delle chiavi di accesso per gli utenti IAM](#).

Per creare chiavi di accesso per un utente IAM (console)

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
2. Nel pannello di navigazione, seleziona Utenti.
3. Scegli il nome dell'utente di cui desideri creare le chiavi di accesso.
4. Nella pagina di riepilogo dell'utente, scegli la scheda Credenziali di sicurezza.

5. Nella sezione Chiavi di accesso, in Procedure consigliate e alternative alle chiavi di accesso, scegli il caso d'uso Altro. Fai clic su Avanti, inserisci le informazioni opzionali necessarie e scegli Crea chiave di accesso.

Per visualizzare la nuova chiave di accesso, seleziona Mostra. Le credenziali saranno simili a quanto segue:

- ID della chiave di accesso: AKIAIOSFODNN7 ESEMPIO
- Chiave di accesso segreta wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY

 Note

Non sarà possibile accedere nuovamente alla chiave di accesso segreta dopo la chiusura di questa finestra di dialogo.

Prendi in considerazione le seguenti best practice per la key pair che hai creato.

- Non archiviate mai la chiave di accesso in testo semplice, in un archivio di codice o nel codice.
 - Disattiva o elimina le chiavi di accesso quando non sono più necessarie.
 - Abilita le autorizzazioni con privilegi minimi.
 - Ruota regolarmente le chiavi di accesso.
6. Per fare il download della coppia di chiavi, scegliere Download .csv file. Conserva le chiavi in un posto sicuro.
 7. Dopo avere scaricato il file .csv, scegli Chiudi.

Quando si crea una chiave di accesso, la coppia di chiavi è attiva per impostazione predefinita ed è possibile utilizzarla immediatamente.

Memorizza le chiavi di accesso per l'accesso programmatico

Come best practice, ti consigliamo di non incorporare le chiavi di accesso direttamente nel codice. Gli strumenti AWS SDKs e la AWS riga di comando consentono di inserire le chiavi di accesso in posizioni note in modo da non doverle conservare nel codice. Colloca le chiavi di accesso in una delle posizioni seguenti:

- Variabili di ambiente: in un sistema multitenant, scegliete le variabili di ambiente utente, non le variabili di ambiente di sistema.
- File delle credenziali CLI: il **config** file credentials and viene aggiornato quando si esegue il comando. `aws configure` Il credentials file si trova `~/.aws/credentials` in Linux, macOS o Unix o in `C:\Users\USERNAME\.aws\credentials` Windows. Questo file può contenere i dettagli delle credenziali per il profilo default ed eventuali profili denominati.
- File di configurazione CLI: il **config** file credentials and viene aggiornato quando si esegue il comando. `aws configure` Il config file si trova `~/.aws/config` in Linux, macOS o Unix o in `C:\Users\USERNAME\.aws\config` Windows. Questo file contiene le impostazioni di configurazione per il profilo predefinito e gli eventuali profili denominati.

La memorizzazione delle chiavi di accesso come variabili di ambiente è un prerequisito per. [the section called “Plugin di autenticazione per Java 4.x”](#) Si noti che ciò include l'impostazione predefinita Regione AWS. Il client cerca le credenziali utilizzando la catena di provider di credenziali predefinita e le chiavi di accesso memorizzate come variabili di ambiente hanno la precedenza su tutte le altre posizioni, ad esempio i file di configurazione. Per ulteriori informazioni, consulta Impostazioni e priorità di [configurazione](#).

L'esempio seguente mostra come configurare le variabili di ambiente per l'utente predefinito.

Linux, macOS, or Unix

```
$ export AWS_ACCESS_KEY_ID=AKIAIOSFODNN7EXAMPLE
$ export AWS_SECRET_ACCESS_KEY=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
$ export AWS_SESSION_TOKEN=AQoDYXdzEJr...<remainder of security token>
$ export AWS_DEFAULT_REGION=us-east-1
```

L'impostazione della variabile di ambiente modifica il valore utilizzato fino al termine della sessione della shell o finché non imposti la variabile su un valore diverso. Puoi rendere le variabili persistenti per le sessioni future impostandole nello script di avvio della shell.

Windows Command Prompt

```
C:\> setx AWS_ACCESS_KEY_ID AKIAIOSFODNN7EXAMPLE
C:\> setx AWS_SECRET_ACCESS_KEY wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
C:\> setx AWS_SESSION_TOKEN AQoDYXdzEJr...<remainder of security token>
C:\> setx AWS_DEFAULT_REGION us-east-1
```

Se si utilizza [set](#) per impostare una variabile di ambiente, il valore utilizzato viene modificato fino al termine della sessione del prompt dei comandi corrente o finché non imposti la variabile su un valore diverso. Se si utilizza [setx](#) per impostare una variabile di ambiente, il valore utilizzato viene modificato nella sessione del prompt dei comandi corrente e in tutte le sessioni del prompt dei comandi create dopo l'esecuzione del comando. Ciò non ha alcun impatto su altre shell di comando già in esecuzione quando esegui il comando.

PowerShell

```
PS C:\> $Env:AWS_ACCESS_KEY_ID="AKIAIOSFODNN7EXAMPLE"  
PS C:\> $Env:AWS_SECRET_ACCESS_KEY="wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY"  
PS C:\> $Env:AWS_SESSION_TOKEN="AQoDYXdzEJr...<remainder of security token>"  
PS C:\> $Env:AWS_DEFAULT_REGION="us-east-1"
```

Se impostate una variabile di ambiente al PowerShell prompt, come mostrato negli esempi precedenti, il valore viene salvato solo per la durata della sessione corrente. Per rendere persistente l'impostazione della variabile di ambiente in tutte PowerShell le sessioni del prompt dei comandi, memorizzatela utilizzando l'applicazione System nel Pannello di controllo. In alternativa, puoi impostare la variabile per tutte le PowerShell sessioni future aggiungendola al tuo PowerShell profilo. Consulta la [PowerShell documentazione](#) per ulteriori informazioni sulla memorizzazione delle variabili di ambiente o sulla loro persistenza tra le sessioni.

Endpoint di servizio per Amazon Keyspaces

Argomenti

- [Porte e protocolli](#)
- [Endpoint globali](#)
- [AWS GovCloud \(US\) Region Endpoint FIPS](#)
- [Endpoint delle regioni della Cina](#)
- [Endpoint Streams](#)
- [Connessione a endpoint dual-stack](#)

Porte e protocolli

Puoi accedere ad Amazon Keyspaces a livello di codice eseguendo un `cqlsh` client, con un driver Cassandra con licenza Apache 2.0 o utilizzando e l'SDK. AWS CLI AWS

La tabella seguente mostra le porte e i protocolli per i diversi meccanismi di accesso.

Accesso programmatico	Porta	Protocollo
CQLSH	1942	TLS
Cassandra Driver	1942	TLS
AWS CLI	443	HTTPS
AWS SDK	443	HTTPS

Per le connessioni TLS, Amazon Keyspaces utilizza certificati emessi nell'ambito di Amazon Trust Services (Amazon CAs Root 1—4) per l'autenticazione sul server. Per ulteriori informazioni, consulta [the section called “Come configurare manualmente le cqlsh connessioni per TLS”](#) la sezione [Prima di iniziare](#) del capitolo dedicata all'autista. [the section called “Utilizzo di un driver client Cassandra”](#)

Endpoint globali

Amazon Keyspaces supporta sia IPv4 gli endpoint pubblici che quelli IPv6 pubblici. Puoi scegliere tra IPv4 endpoint ed endpoint dual-stack. Gli endpoint utilizzano la seguente convenzione di denominazione, che è possibile sostituire *us-east-1* con un'altra disponibile nella tabella. Regione AWS

- IPv4 endpoint — `cassandra.us-east-1.amazonaws.com`
- Endpoint dual-stack: `cassandra.us-east-1.api.aws`

Per ulteriori informazioni sugli endpoint dual-stack e su come configurare le connessioni, consulta [the section called “Connessione a endpoint dual-stack”](#)

Amazon Keyspaces è disponibile nelle seguenti regioni.

Nome della regione	Regione	Endpoint	Protocollo
US East (Ohio)	us-east-2	<code>cassandra.us-east-2.amazonaws.com</code>	HTTPS e TLS

Nome della regione	Regione	Endpoint	Protocollo	
		cassandra.us-east-2.api.aws	HTTPS e TLS	
US East (N. Virginia)	us-east-1	cassandra.us-east-1.amazonaws.com	HTTPS e TLS	
		cassandra-fips.us-east-1.api.aws	HTTPS e TLS	
		cassandra-fips.us-east-1.amazonaws.com	HTTPS e TLS	
		cassandra.us-east-1.api.aws	HTTPS e TLS	
Stati Uniti occidentali (California settentrionale)	us-west-1	cassandra.us-west-1.amazonaws.com	HTTPS e TLS	
		cassandra.us-west-1.api.aws	HTTPS e TLS	
US West (Oregon)	us-west-2	cassandra.us-west-2.amazonaws.com	HTTPS e TLS	
		cassandra-fips.us-west-2.api.aws	HTTPS e TLS	
		cassandra-fips.us-west-2.amazonaws.com	HTTPS e TLS	
		cassandra.us-west-2.api.aws	HTTPS e TLS	

Nome della regione	Regione	Endpoint	Protocollo	
Africa (Cape Town)	af-south-1	cassandra.af-south-1.amazonaws.com cassandra.af-south-1.api.aws	HTTPS e TLS HTTPS e TLS	
Asia Pacifico (Hong Kong)	ap-east-1	cassandra.ap-east-1.amazonaws.com cassandra.ap-east-1.api.aws	HTTPS e TLS HTTPS e TLS	
Asia Pacifico (Mumbai)	ap-south-1	cassandra.ap-south-1.amazonaws.com cassandra.ap-south-1.api.aws	HTTPS e TLS HTTPS e TLS	
Asia Pacifico (Seoul)	ap-northeast-2	cassandra.ap-northeast-2.amazonaws.com cassandra.ap-northeast-2.api.aws	HTTPS e TLS HTTPS e TLS	
Asia Pacifico (Singapore)	ap-southeast-1	cassandra.ap-southeast-1.amazonaws.com cassandra.ap-southeast-1.api.aws	HTTPS e TLS HTTPS e TLS	
Asia Pacifico (Sydney)	ap-southeast-2	cassandra.ap-southeast-2.amazonaws.com cassandra.ap-southeast-2.api.aws	HTTPS e TLS HTTPS e TLS	

Nome della regione	Regione	Endpoint	Protocollo	
Asia Pacifico (Tokyo)	ap-northeast-1	cassandra.ap-northeast-1.amazonaws.com cassandra.ap-northeast-1.api.aws	HTTPS e TLS HTTPS e TLS	
Canada (Centrale)	ca-central-1	cassandra.ca-central-1.amazonaws.com cassandra.ca-central-1.api.aws	HTTPS e TLS HTTPS e TLS	
Europa (Francoforte)	eu-central-1	cassandra.eu-central-1.amazonaws.com cassandra.eu-central-1.api.aws	HTTPS e TLS HTTPS e TLS	
Europa (Irlanda)	eu-west-1	cassandra.eu-west-1.amazonaws.com cassandra.eu-west-1.api.aws	HTTPS e TLS HTTPS e TLS	
Europa (Londra)	eu-west-2	cassandra.eu-west-2.amazonaws.com cassandra.eu-west-2.api.aws	HTTPS e TLS HTTPS e TLS	
Europa (Parigi)	eu-west-3	cassandra.eu-west-3.amazonaws.com cassandra.eu-west-3.api.aws	HTTPS e TLS HTTPS e TLS	

Nome della regione	Regione	Endpoint	Protocollo	
Europa (Stoccolma)	eu-north-1	cassandra.eu-north-1.amazonaws.com cassandra.eu-north-1.api.aws	HTTPS e TLS HTTPS e TLS	
Medio Oriente (Bahrein)	me-south-1	cassandra.me-south-1.amazonaws.com cassandra.me-south-1.api.aws	HTTPS e TLS HTTPS e TLS	
Medio Oriente (Emirati Arabi Uniti)	me-central-1	cassandra.me-central-1.amazonaws.com cassandra.me-central-1.api.aws	HTTPS e TLS HTTPS e TLS	
Sud America (São Paulo)	sa-east-1	cassandra.sa-east-1.amazonaws.com cassandra.sa-east-1.api.aws	HTTPS e TLS HTTPS e TLS	
AWS GovCloud (Stati Uniti orientali)	us-gov-east-1	cassandra.us-gov-east-1.amazonaws.com cassandra.us-gov-east-1.api.aws	HTTPS e TLS HTTPS e TLS	

Nome della regione	Regione	Endpoint	Protocollo	
AWS GovCloud (Stati Uniti occidentali)	us-gov-west-1	cassandra.us-gov-west-1.amazonaws.com cassandra.us-gov-west-1.api.aws	HTTPS e TLS HTTPS e TLS	

AWS GovCloud (US) Region Endpoint FIPS

Endpoint FIPS disponibili in. AWS GovCloud (US) Region Amazon Keyspaces supporta sia gli endpoint FIPS che quelli IPv4 IPv6 FIPS. Puoi scegliere tra IPv4 endpoint ed endpoint dual-stack. Per ulteriori informazioni, consulta [Amazon Keyspaces nella Guida per l'AWS GovCloud \(US\) utente](#).

Nome Regione	Regione	Endpoint FIPS IPv4	Endpoint FIPS dual-stack	Protocollo
AWS GovCloud (Stati Uniti orientali)	us-gov-east-1	cassandra.us-gov-east-1.amazonaws.com	cassandra.us-gov-east-1.api.aws	HTTPS e TLS
AWS GovCloud (Stati Uniti occidentali)	us-gov-west-1	cassandra.us-gov-west-1.amazonaws.com	cassandra.us-gov-west-1.api.aws	HTTPS e TLS

Endpoint delle regioni della Cina

Amazon Keyspaces supporta gli IPv4 endpoint nelle regioni della Cina AWS .

Per accedere a questi endpoint, devi registrarti per un set separato di credenziali di account uniche per le regioni della Cina. Per ulteriori informazioni, consulta [Registrazione, account e credenziali in Cina](#).

Nome Regione	Regione	Endpoint	Protocollo
China (Beijing)	cn-north-1	cassandra.cn-north-1.amazonaws.com.cn	HTTPS e TLS
Cina (Ningxia)	cn-northwest-1	cassandra.cn-northwest-1.amazonaws.com.cn	HTTPS e TLS

Endpoint Streams

Amazon Keyspaces CDC streams è disponibile nelle seguenti versioni. Regioni AWS Questa tabella mostra l'endpoint di servizio dual-stack disponibile per ogni regione. Per ulteriori informazioni su, vedere. Amazon Keyspaces CDC streams [the section called “Accedi agli endpoint di flusso CDC”](#)

Nome della regione	Regione	Endpoint	Protocollo
US East (Ohio)	us-east-2	cassandra-streams.us-east-2.api.aws	HTTPS
US East (N. Virginia)	us-east-1	cassandra-streams.us-east-1.api.aws	HTTPS
Stati Uniti occidentali	us-west-1	cassandra-streams.us-west-1.api.aws	HTTPS

Nome della regione	Regione	Endpoint	Protocollo	
(California settentrionale)				
US West (Oregon)	us-west-2	cassandra-streams.us-west-2.api.aws	HTTPS	
Africa (Cape Town)	af-south-1	cassandra-streams.af-south-1.api.aws	HTTPS	
Asia Pacifico (Hong Kong)	ap-east-1	cassandra-streams.ap-east-1.api.aws	HTTPS	
Asia Pacifico (Mumbai)	ap-south-1	cassandra-streams.ap-south-1.api.aws	HTTPS	
Asia Pacifico (Seoul)	ap-northeast-2	cassandra-streams.ap-northeast-2.api.aws	HTTPS	
Asia Pacifico (Singapore)	ap-southeast-1	cassandra-streams.ap-southeast-1.api.aws	HTTPS	
Asia Pacifico (Sydney)	ap-southeast-2	cassandra-streams.ap-southeast-2.api.aws	HTTPS	

Nome della regione	Regione	Endpoint	Protocollo	
Asia Pacifico (Tokyo)	ap-northeast-1	cassandra-streams.ap-northeast-1.api.aws	HTTPS	
Canada (Centrale)	ca-central-1	cassandra-streams.ca-central-1.api.aws	HTTPS	
Europa (Francoforte)	eu-central-1	cassandra-streams.eu-central-1.api.aws	HTTPS	
Europa (Irlanda)	eu-west-1	cassandra-streams.eu-west-1.api.aws	HTTPS	
Europa (Londra)	eu-west-2	cassandra-streams.eu-west-2.api.aws	HTTPS	
Europa (Parigi)	eu-west-3	cassandra-streams.eu-west-3.api.aws	HTTPS	
Europa (Stoccolma)	eu-north-1	cassandra-streams.eu-north-1.api.aws	HTTPS	
Medio Oriente (Bahrein)	me-south-1	cassandra-streams.me-south-1.api.aws	HTTPS	
Medio Oriente (Emirati Arabi Uniti)	me-central-1	cassandra-streams.me-central-1.api.aws	HTTPS	

Nome della regione	Regione	Endpoint	Protocollo	
Sud America (São Paulo)	sa-east-1	cassandra-streams.sa-east-1.api.aws	HTTPS	
AWS GovCloud (Stati Uniti orientali)	us-gov-east-1	cassandra-streams.us-gov-east-1.api.aws	HTTPS	
AWS GovCloud (Stati Uniti occidentali)	us-gov-west-1	cassandra-streams.us-gov-west-1.api.aws	HTTPS	

Connessione a endpoint dual-stack

Gli endpoint globali di Amazon Keyspaces sono endpoint dual-stack che accettano e richiedono. IPv4 IPv6

Quando ti connetti ad Amazon Keyspaces utilizzando IPv6, il servizio adatta automaticamente le risposte della tabella di sistema in base al protocollo di connessione. Ciò garantisce che le applicazioni ricevano informazioni coerenti sugli indirizzi di rete corrispondenti al tipo di connessione. Ciò fornisce al client informazioni accurate sulla topologia di rete, mantenendo al contempo la compatibilità con le versioni precedenti per le applicazioni CQL esistenti.

Amazon Keyspaces rileva automaticamente il protocollo di rete (IPv4 o IPv6) utilizzato dalla connessione client e regola di conseguenza le risposte della tabella di sistema. Questo rilevamento avviene in modo trasparente durante l'handshake iniziale della connessione e non richiede alcuna configurazione aggiuntiva da parte dell'applicazione client.

Amazon Keyspaces restituisce gli indirizzi IP in base al protocollo di connessione. Ad esempio, una richiesta proveniente da una IPv4 rete restituisce la seguente risposta.

```
SELECT * FROM system.peers;  
-- Returns IPv4 addresses in peer column  
-- Example: 172.31.1.1, 172.31.1.2, etc.
```

Una connessione da una IPv6 rete a un endpoint dual-stack, ad esempio `cassandra.us-east-1.api.aws`, restituisce la seguente risposta.

```
SELECT * FROM system.peers;  
-- Returns IPv6 addresses in peer column  
-- Example: 2001:db8::1, 2001:db8::2, etc.
```

Per ulteriori informazioni sul IPv6 supporto in Amazon Keyspaces, consulta [the section called “IPv6 supporto in Amazon Keyspaces”](#)

IPv6 supporto in Amazon Keyspaces

IPv6 il supporto in Amazon Keyspaces consente alle applicazioni di stabilire connessioni utilizzando Internet Protocol versione 6, il protocollo Internet di nuova generazione che fornisce uno spazio di indirizzi notevolmente ampliato rispetto a. IPv4 L'implementazione utilizza endpoint dual-stack che supportano entrambi IPv4 e IPv6 contemporaneamente, garantendo la compatibilità con le versioni precedenti e abilitando una connettività pronta per il futuro. Per un elenco di endpoint, consulta [the section called “Endpoint globali”](#).

Amazon Keyspaces implementa il IPv6 supporto tramite un'architettura dual-stack che mantiene la completa compatibilità con le versioni precedenti abilitando al contempo la connettività. IPv6

Risoluzione DNS in Amazon Keyspaces

Quando le applicazioni si connettono a endpoint dual-stack, il processo di risoluzione DNS restituisce entrambi i tipi di indirizzi:

Un record () IPv4

IPv4 Indirizzi tradizionali per la compatibilità con le versioni precedenti

Record AAAA () IPv6

IPv6 Nuovi indirizzi per una connettività moderna

Il sistema operativo e lo stack di rete del client selezionano automaticamente il protocollo più appropriato in base alla configurazione locale, alla disponibilità della rete e alle preferenze di sistema.

Il protocollo Cassandra Query Language (CQL) supporta perfettamente la IPv6 connettività senza richiedere modifiche al codice dell'applicazione.

Selezione automatica del protocollo

- Le applicazioni specificano l'endpoint dual-stack
- Lo stack di rete sceglie o si basa sulla disponibilità IPv4 IPv6
- Non sono necessarie modifiche al codice per le applicazioni CQL esistenti

Compatibilità dei driver

- Tutti i principali driver CQL supportano IPv6 in modo trasparente
- DataStax i driver gestiscono IPv6 gli indirizzi in modo nativo
- I driver open source funzionano senza modifiche

Coerenza della connessione

- Le tabelle di sistema riflettono il protocollo di connessione utilizzato
- IPv6 le connessioni mostrano IPv6 gli indirizzi in `system.peers`
- IPv4 le connessioni continuano a mostrare IPv4 gli indirizzi

Utilizzo **cqlsh** per connettersi ad Amazon Keyspaces

Per connetterti ad Amazon Keyspaces utilizzandocqlsh, puoi utilizzare il. cqlsh-expansion. Si tratta di un toolkit che contiene strumenti comuni come Apache Cassandra cqlsh e helper preconfigurati per Amazon Keyspaces, pur mantenendo la piena compatibilità con Apache Cassandra. cqlsh-expansionIntegra il plug-in di autenticazione SigV4 e consente di connettersi utilizzando le chiavi di accesso IAM anziché il nome utente e la password. Devi solo installare cqlsh gli script per stabilire una connessione e non la distribuzione completa di Apache Cassandra, perché Amazon Keyspaces è serverless. Questo pacchetto di installazione leggero include cqlsh-expansion cqlsh gli script classici che puoi installare su qualsiasi piattaforma che supporti Python.

Note

Murmur3Partitioner è il partizionatore consigliato per Amazon Keyspaces e. `cqlsh-expansion` `cqlsh-expansion` Non supporta `Amazon KeyspacesDefaultPartitioner`. Per ulteriori informazioni, consulta [the section called “Lavorare con i partizionatori”](#).

Per informazioni generali su `cqlsh`, vedi [cqlsh: la shell CQL](#).

Argomenti

- [Utilizzo di `cqlsh-expansion` per connettersi ad Amazon Keyspaces](#)
- [Come configurare manualmente le `cqlsh` connessioni per TLS](#)

Utilizzo di `cqlsh-expansion` per connettersi ad Amazon Keyspaces**Installazione e configurazione di `cqlsh-expansion`**

1. Per installare il pacchetto `cqlsh-expansion` Python, puoi eseguire un `pip` comando. Questo installa `cqlsh-expansion` gli script sulla tua macchina usando un'installazione `pip` insieme a un file contenente un elenco di dipendenze. Le `--user` flag istruzioni `pip` per usare la directory di installazione degli utenti Python per la tua piattaforma. Su un sistema basato su Unix, quella dovrebbe essere la `~/local/` directory.

Hai bisogno di Python 3 per installare `cqlsh-expansion`, per scoprire la tua versione di Python, usa `python --version` Per installarlo, puoi eseguire il seguente comando.

```
python3 -m pip install --user cqlsh-expansion
```

L'output dovrebbe essere simile a questo.

```
Collecting cqlsh-expansion
  Downloading cqlsh_expansion-0.9.6-py3-none-any.whl (153 kB)
##### 153.7/153.7 KB 3.3 MB/s eta 0:00:00
Collecting cassandra-driver
  Downloading cassandra_driver-3.28.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (19.1 MB)
##### 19.1/19.1 MB 44.5 MB/s eta 0:00:00
```

```

Requirement already satisfied: six>=1.12.0 in /usr/lib/python3/dist-packages (from
  cqlsh-expansion) (1.16.0)
Collecting boto3
  Downloading boto3-1.29.2-py3-none-any.whl (135 kB)
##### 135.8/135.8 KB 17.2 MB/s eta 0:00:00
Collecting cassandra-sigv4>=4.0.2
  Downloading cassandra_sigv4-4.0.2-py2.py3-none-any.whl (9.8 kB)
Collecting botocore<1.33.0,>=1.32.2
  Downloading botocore-1.32.2-py3-none-any.whl (11.4 MB)
##### 11.4/11.4 MB 60.9 MB/s eta 0:00:00
Collecting s3transfer<0.8.0,>=0.7.0
  Downloading s3transfer-0.7.0-py3-none-any.whl (79 kB)
##### 79.8/79.8 KB 13.1 MB/s eta 0:00:00
Collecting jmespath<2.0.0,>=0.7.1
  Downloading jmespath-1.0.1-py3-none-any.whl (20 kB)
Collecting geomet<0.3,>=0.1
  Downloading geomet-0.2.1.post1-py3-none-any.whl (18 kB)
Collecting python-dateutil<3.0.0,>=2.1
  Downloading python_dateutil-2.8.2-py2.py3-none-any.whl (247 kB)
##### 247.7/247.7 KB 33.1 MB/s eta 0:00:00
Requirement already satisfied: urllib3<2.1,>=1.25.4 in /usr/lib/python3/dist-
  packages (from botocore<1.33.0,>=1.32.2->boto3->cqlsh-expansion) (1.26.5)
Requirement already satisfied: click in /usr/lib/python3/dist-packages (from
  geomet<0.3,>=0.1->cassandra-driver->cqlsh-expansion) (8.0.3)
Installing collected packages: python-dateutil, jmespath, geomet, cassandra-driver,
  botocore, s3transfer, boto3, cassandra-sigv4, cqlsh-expansion
  WARNING: The script geomet is installed in '/home/ubuntu/.local/bin' which is not
  on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this
  warning, use --no-warn-script-location.
  WARNING: The scripts cqlsh, cqlsh-expansion and cqlsh-expansion.init are
  installed in '/home/ubuntu/.local/bin' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this
  warning, use --no-warn-script-location.
Successfully installed boto3-1.29.2 botocore-1.32.2 cassandra-driver-3.28.0
  cassandra-sigv4-4.0.2 cqlsh-expansion-0.9.6 geomet-0.2.1.post1 jmespath-1.0.1
  python-dateutil-2.8.2 s3transfer-0.7.0

```

Se la directory di installazione non si trova in PATH, è necessario aggiungerla seguendo le istruzioni del sistema operativo. Di seguito è riportato un esempio per Ubuntu Linux.

```
export PATH="$PATH:/home/ubuntu/.local/bin"
```

Per confermare che il pacchetto è installato, puoi eseguire il seguente comando.

```
cqlsh-expansion --version
```

L'output dovrebbe essere simile a questo.

```
cqlsh 6.1.0
```

2. Per configurare `cqlsh-expansion`, puoi eseguire uno script di post-installazione per completare automaticamente i seguenti passaggi:

1. Crea la `.cassandra` directory nella home directory dell'utente se non esiste già.
2. Copia un file di `cqlshrc` configurazione preconfigurato nella `.cassandra` directory.
3. Copia il file di certificato combinato nella `.cassandra` directory. Amazon Keyspaces utilizza questo certificato per configurare la connessione sicura con Transport Layer Security (TLS). La crittografia in transito fornisce un ulteriore livello di protezione dei dati crittografando i dati mentre viaggiano da e verso Amazon Keyspaces. Per ulteriori informazioni sui certificati, consulta [the section called "Come configurare manualmente le cqlsh connessioni per TLS"](#)

Per esaminare prima lo script, puoi accedervi dal repository Github all'indirizzo [post_install.py](#)

Per utilizzare lo script, puoi eseguire il comando seguente.

```
cqlsh-expansion.init
```

Note

La directory e il file creati dallo script di post-installazione non vengono rimossi quando si disinstalla `cqlsh-expansion` utilizzando `pip uninstall` e devono essere eliminati manualmente.

Connessione ad Amazon Keyspaces utilizzando il **cqlsh-expansion**

1. Configura la tua Regione AWS e aggiungila come variabile di ambiente utente.

Per aggiungere la tua regione predefinita come variabile di ambiente su un sistema basato su Unix, puoi eseguire il seguente comando. Per questo esempio, utilizziamo `us-east-1`.

```
export AWS_DEFAULT_REGION=us-east-1
```

Per ulteriori informazioni su come impostare le variabili di ambiente, anche per altre piattaforme, vedi [Come impostare le variabili di ambiente](#).

2. Trova il tuo endpoint di servizio.

Scegli l'endpoint di servizio appropriato per la tua regione. Per esaminare gli endpoint disponibili per Amazon Keyspaces, consulta [the section called "Endpoint di servizio"](#). Per questo esempio, utilizziamo l'endpoint `cassandra.us-east-1.amazonaws.com`.

3. Configura il metodo di autenticazione.

La connessione con le chiavi di accesso IAM (utenti, ruoli e identità federate IAM) è il metodo consigliato per una maggiore sicurezza.

Prima di poterti connettere con le chiavi di accesso IAM, devi completare i seguenti passaggi:

- Crea un utente IAM o segui le best practice e crea un ruolo IAM che gli utenti IAM possono assumere. Per ulteriori informazioni su come creare chiavi di accesso IAM, consulta [the section called "Crea credenziali IAM per l'autenticazione AWS"](#).
- Crea una policy IAM che conceda al ruolo (o all'utente IAM) almeno l'accesso in sola lettura ad Amazon Keyspaces. Per ulteriori informazioni sulle autorizzazioni richieste all'utente o al ruolo IAM per connettersi ad Amazon Keyspaces, consulta [the section called "Accesso alle tabelle Amazon Keyspaces"](#).
- Aggiungi le chiavi di accesso dell'utente IAM alle variabili di ambiente dell'utente come mostrato nell'esempio seguente.

```
export AWS_ACCESS_KEY_ID=AKIAIOSFODNN7EXAMPLE
export AWS_SECRET_ACCESS_KEY=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
```

Per ulteriori informazioni su come impostare le variabili di ambiente, anche per altre piattaforme, vedi [Come impostare le variabili di ambiente](#).

 Note

Se ti connetti da un' EC2 istanza Amazon, devi anche configurare una regola in uscita nel gruppo di sicurezza che consenta il traffico dall'istanza ad Amazon Keyspaces. Per ulteriori informazioni su come visualizzare e modificare le regole EC2 in uscita, consulta [Aggiungere regole a un gruppo di sicurezza nella Amazon EC2 User Guide](#).

4. Connect ad Amazon Keyspaces utilizzando l'autenticazione `cqlsh-expansion` e SigV4.

Per connetterti ad Amazon Keyspaces con `cqlsh-expansion`, puoi usare il seguente comando. Assicurati di sostituire l'endpoint di servizio con l'endpoint corretto per la tua regione.

```
cqlsh-expansion cassandra.us-east-1.amazonaws.com 9142 --ssl
```

Se la connessione è riuscita, dovresti vedere un output simile a quello dell'esempio seguente.

```
Connected to Amazon Keyspaces at cassandra.us-east-1.amazonaws.com:9142
[cqlsh 6.1.0 | Cassandra 3.11.2 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
cqlsh current consistency level is ONE.
cqlsh>
```

Se si verifica un errore di connessione, consulta la sezione [the section called “Errori di connessione Cqlsh”](#) per informazioni sulla risoluzione dei problemi.

- Connect ad Amazon Keyspaces con credenziali specifiche del servizio.

Per connetterti con la combinazione tradizionale di nome utente e password utilizzata da Cassandra per l'autenticazione, devi prima creare credenziali specifiche del servizio per Amazon Keyspaces come descritto in. [the section called “Crea credenziali specifiche per il servizio”](#) È inoltre necessario concedere all'utente le autorizzazioni per accedere ad Amazon Keyspaces, per ulteriori informazioni, consulta. [the section called “Accesso alle tabelle Amazon Keyspaces”](#)

Dopo aver creato le credenziali e le autorizzazioni specifiche del servizio per l'utente, devi aggiornare il `cqlshrc` file, che in genere si trova nel percorso della directory utente. `~/.cassandra/` Nel `cqlshrc` file, vai alla `[authentication]` sezione Cassandra e

commenta il modulo SigV4 e la classe sotto `[auth_provider]` usando il carattere «;», come mostrato nell'esempio seguente.

```
[auth_provider]

; module = cassandra_sigv4.auth

; classname = SigV4AuthProvider
```

Dopo aver aggiornato il `cqlshrc` file, puoi connetterti ad Amazon Keyspaces con credenziali specifiche del servizio utilizzando il seguente comando.

```
cqlsh-expansion cassandra.us-east-1.amazonaws.com 9142 -u myUserName -
p myPassword --ssl
```

Pulizia

- Per rimuovere il `cqlsh-expansion` pacchetto puoi usare il comando. `pip uninstall`

```
pip3 uninstall cqlsh-expansion
```

Il `pip3 uninstall` comando non rimuove la directory e i file correlati creati dallo script di post-installazione. Per rimuovere la cartella e i file creati dallo script di post-installazione, è possibile eliminare la `.cassandra` directory.

Come configurare manualmente le **cqlsh** connessioni per TLS

Amazon Keyspaces accetta solo connessioni sicure tramite Transport Layer Security (TLS). Puoi utilizzare l'`cqlsh-expansion` utilità che scarica automaticamente i certificati e installa un file di configurazione `cqlshrc` preconfigurato. Per ulteriori informazioni, consulta questa [the section called “Utilizzo di cqlsh-expansion”](#) pagina.

Se desideri scaricare i certificati e configurare la connessione manualmente, puoi farlo utilizzando i seguenti passaggi.

1. Scaricate i seguenti certificati digitali e salvate i file localmente o nella vostra home directory.

- a. AmazonRootCA1

- b. AmazonRootCA2
- c. AmazonRootCA3
- d. AmazonRootCA4
- e. Starfield Class 2 Root (opzionale, per compatibilità con le versioni precedenti)

Per scaricare i certificati, puoi usare i seguenti comandi.

```
curl -O https://www.amazontrust.com/repository/AmazonRootCA1.pem
curl -O https://www.amazontrust.com/repository/AmazonRootCA2.pem
curl -O https://www.amazontrust.com/repository/AmazonRootCA3.pem
curl -O https://www.amazontrust.com/repository/AmazonRootCA4.pem
curl -O https://certs.secureserver.net/repository/sf-class2-root.crt
```

Note

Amazon Keyspaces utilizzava in precedenza certificati TLS ancorati alla CA Starfield Class 2. AWS sta migrando tutte le Regioni AWS verso certificati emessi nell'ambito di Amazon Trust Services (Amazon Root CAs 1—4). Durante questa transizione, configura i client in modo che si fidino sia di Amazon Root CAs 1-4 che di Starfield root per garantire la compatibilità in tutte le regioni.

2. Combina tutti i certificati scaricati in un unico pem file con il nome indicato *keyspaces-bundle.pem* nei nostri esempi. Per aggiungerlo, eseguire il comando seguente: Prendi nota del percorso del file, ne avrai bisogno in seguito.

```
cat AmazonRootCA1.pem \
    AmazonRootCA2.pem \
    AmazonRootCA3.pem \
    AmazonRootCA4.pem \
    sf-class2-root.crt \
    > keyspaces-bundle.pem
```

3. Apri il file di `cqlshrc` configurazione nella home directory di Cassandra, ad esempio, `${HOME}/.cassandra/cqlshrc` e aggiungi le seguenti righe.

```
[connection]
port = 9142
factory = cqlshlib.ssl.ssl_transport_factory
```

```
[ssl]
validate = true
certfile = path_to_file/keysaces-bundle.pem
```

Utilizzo di AWS CLI per connettersi ad Amazon Keyspaces

Puoi usare il AWS Command Line Interface (AWS CLI) per controllare più AWS servizi dalla riga di comando e automatizzarli tramite script. Con Amazon Keyspaces puoi utilizzare le AWS CLI operazioni DDL (Data Definition Language), come la creazione di una tabella. Inoltre, puoi utilizzare servizi e strumenti di infrastruttura come codice (IaC) come AWS CloudFormation Terraform.

Prima di poterlo utilizzare AWS CLI con Amazon Keyspaces, devi ottenere un ID della chiave di accesso e una chiave di accesso segreta. Per ulteriori informazioni, consulta [the section called “Crea credenziali IAM per l'autenticazione AWS”](#).

Per un elenco completo di tutti i comandi disponibili per Amazon Keyspaces in AWS CLI, consulta il [AWS CLI Command Reference](#).

Argomenti

- [Download e configurazione della AWS CLI](#)
- [Utilizzo di AWS CLI con Amazon Keyspaces](#)

Download e configurazione della AWS CLI

AWS CLI È disponibile all'indirizzo <https://aws.amazon.com/cli>. Viene eseguita su Windows, macOS o Linux. Dopo averlo scaricato AWS CLI, segui questi passaggi per installarlo e configurarlo:

1. Vai alla [guida per l'AWS Command Line Interface utente](#)
2. Segui le istruzioni per [l'installazione AWS CLI e la configurazione di AWS CLI](#)

Utilizzo di AWS CLI con Amazon Keyspaces

Il formato della riga di comando è costituito da un nome di operazione Amazon Keyspaces seguito dai parametri per tale operazione. AWS CLI Supporta una sintassi abbreviata per i valori dei parametri, oltre a JSON. I seguenti esempi di Amazon Keyspaces utilizzano una sintassi AWS CLI abbreviata. Per ulteriori informazioni, consulta [Uso della sintassi abbreviata con la CLI](#). AWS

Il comando seguente crea uno spazio di chiavi con il nome catalog.

```
aws keyspaces create-keyspace --keyspace-name 'catalog'
```

Il comando restituisce la risorsa Amazon Resource Name (ARN) nell'output.

```
{
  "resourceArn": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/catalog/"
}
```

Per confermare l'esistenza del catalogo keyspace, puoi usare il seguente comando.

```
aws keyspaces get-keyspace --keyspace-name 'catalog'
```

L'output del comando restituisce i seguenti valori.

```
{
  "keyspaceName": "catalog",
  "resourceArn": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/catalog/"
}
```

Il comando seguente crea una tabella con il nome book_awards. La chiave di partizione della tabella è costituita dalle colonne year award e la chiave di clustering è costituita dalle colonne category ed rank entrambe le colonne di clustering utilizzano l'ordinamento crescente. Per facilitare la leggibilità, in questa sezione i comandi lunghi sono suddivisi su righe separate.

```
aws keyspaces create-table --keyspace-name 'catalog' --table-name 'book_awards'
                        --schema-definition 'allColumns=[{name=year,type=int},
{name=award,type=text},{name=rank,type=int},
                        {name=category,type=text}, {name=author,type=text},
{name=book_title,type=text},{name=publisher,type=text}],
                        partitionKeys=[{name=year},
{name=award}],clusteringKeys=[{name=category,orderBy=ASC},{name=rank,orderBy=ASC}]'
```

Questo comando produce l'output seguente.

```
{
  "resourceArn": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/catalog/table/
book_awards"
}
```

Per confermare i metadati e le proprietà della tabella, è possibile utilizzare il seguente comando.

```
aws keyspaces get-table --keyspace-name 'catalog' --table-name 'book_awards'
```

Questo comando restituisce il seguente output.

```
{
  "keyspaceName": "catalog",
  "tableName": "book_awards",
  "resourceArn": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/catalog/table/
book_awards",
  "creationTimestamp": 1645564368.628,
  "status": "ACTIVE",
  "schemaDefinition": {
    "allColumns": [
      {
        "name": "year",
        "type": "int"
      },
      {
        "name": "award",
        "type": "text"
      },
      {
        "name": "category",
        "type": "text"
      },
      {
        "name": "rank",
        "type": "int"
      },
      {
        "name": "author",
        "type": "text"
      },
      {
        "name": "book_title",
        "type": "text"
      },
      {
        "name": "publisher",
        "type": "text"
      }
    ]
  }
}
```

```

    ],
    "partitionKeys": [
      {
        "name": "year"
      },
      {
        "name": "award"
      }
    ],
    "clusteringKeys": [
      {
        "name": "category",
        "orderBy": "ASC"
      },
      {
        "name": "rank",
        "orderBy": "ASC"
      }
    ],
    "staticColumns": []
  },
  "capacitySpecification": {
    "throughputMode": "PAY_PER_REQUEST",
    "lastUpdateToPayPerRequestTimestamp": 1645564368.628
  },
  "encryptionSpecification": {
    "type": "AWS_OWNED_KMS_KEY"
  },
  "pointInTimeRecovery": {
    "status": "DISABLED"
  },
  "ttl": {
    "status": "ENABLED"
  },
  "defaultTimeToLive": 0,
  "comment": {
    "message": ""
  }
}

```

Quando si creano tabelle con schemi complessi, può essere utile caricare la definizione dello schema della tabella da un file JSON. Di seguito è riportato un esempio di ciò. Scaricate il file JSON di esempio di definizione dello schema da [schema_definition.zip](#) ed

estraeteloschema_definition.json, prendendo nota del percorso del file. In questo esempio, il file JSON di definizione dello schema si trova nella directory corrente. Per le diverse opzioni relative al percorso dei file, consultate [Come caricare i parametri da un file](#).

```
aws keyspaces create-table --keyspace-name 'catalog'
                        --table-name 'book_awards' --schema-definition 'file://
schema_definition.json'
```

Gli esempi seguenti mostrano come creare una tabella semplice con il nome myTable con opzioni aggiuntive. Nota che i comandi sono suddivisi in righe separate per migliorare la leggibilità. Questo comando mostra come creare una tabella e:

- imposta la modalità di capacità della tabella
- abilita Point-in-time il ripristino per la tabella
- imposta il valore Time to Live (TTL) predefinito per la tabella su un anno
- aggiungi due tag per la tabella

```
aws keyspaces create-table --keyspace-name 'catalog' --table-name 'myTable'
                        --schema-definition 'allColumns=[{name=id,type=int},{name=name,type=text},
{name=date,type=timestamp}],partitionKeys=[{name=id}]'
                        --capacity-specification
                        'throughputMode=PROVISIONED,readCapacityUnits=5,writeCapacityUnits=5'
                        --point-in-time-recovery 'status=ENABLED'
                        --default-time-to-live '31536000'
                        --tags 'key=env,value=test' 'key=dpt,value=sec'
```

Questo esempio mostra come creare una nuova tabella che utilizza una chiave gestita dal cliente per la crittografia e ha il TTL abilitato per consentire di impostare le date di scadenza per colonne e righe. Per eseguire questo esempio, devi sostituire l'ARN della risorsa per la AWS KMS chiave gestita dal cliente con la tua chiave e assicurarti che Amazon Keyspaces possa accedervi.

```
aws keyspaces create-table --keyspace-name 'catalog' --table-name 'myTable'
                        --schema-definition 'allColumns=[{name=id,type=int},{name=name,type=text},
{name=date,type=timestamp}],partitionKeys=[{name=id}]'
                        --encryption-specification
                        'type=CUSTOMER_MANAGED_KMS_KEY,kmsKeyId=arn:aws:kms:us-
east-1:111122223333:key/11111111-2222-3333-4444-555555555555'
                        --ttl 'status=ENABLED'
```

Utilizzo dell'API per la connessione ad Amazon Keyspaces

Puoi utilizzare l' AWS SDK e AWS Command Line Interface (AWS CLI) per lavorare in modo interattivo con Amazon Keyspaces. Puoi utilizzare l'API per operazioni DDL (Data Language Definition), come la creazione di un keyspace o di una tabella. Inoltre, puoi utilizzare servizi e strumenti di infrastruttura come codice (IaC) come AWS CloudFormation Terraform.

Prima di poterlo utilizzare AWS CLI con Amazon Keyspaces, devi ottenere un ID della chiave di accesso e una chiave di accesso segreta. Per ulteriori informazioni, consulta [the section called “Crea credenziali IAM per l'autenticazione AWS”](#).

Per un elenco completo di tutte le operazioni disponibili per Amazon Keyspaces nell'API, consulta Amazon [Keyspaces](#) API Reference.

Utilizzo di un driver client Cassandra per accedere ad Amazon Keyspaces a livello di codice

Puoi utilizzare molti driver Cassandra open source di terze parti per connetterti ad Amazon Keyspaces. Amazon Keyspaces è compatibile con i driver Cassandra che supportano la versione 3.11.2 di Apache Cassandra. Questi sono i driver e le versioni più recenti che abbiamo testato e che consigliamo di utilizzare con Amazon Keyspaces:

- Java v3.3
- Java v4.17
- Python Cassandra-driver 3.29.1
- Node.js cassandra driver -v 4.7.2
- GO using GOCQL v1.6
- .NET CassandraCSharpDriver -v 3.20.1

Per ulteriori informazioni sui driver Cassandra, consulta Driver del client [Apache Cassandra](#).

Note

Per aiutarti a iniziare, puoi visualizzare e scaricare esempi di end-to-end codice che stabiliscono connessioni ad Amazon Keyspaces con i driver più diffusi. Guarda gli [esempi di Amazon Keyspaces](#) su. GitHub

I tutorial in questo capitolo includono una semplice query CQL per confermare che la connessione ad Amazon Keyspaces è stata stabilita correttamente. Per informazioni su come utilizzare keyspace e tabelle dopo la connessione a un endpoint Amazon Keyspaces, consulta. [Riferimento al linguaggio CQL](#) Per un step-by-step tutorial che mostra come connettersi ad Amazon Keyspaces da un endpoint Amazon VPC, consulta. [the section called “Connessione con endpoint VPC”](#)

Argomenti

- [Utilizzo di un driver client Java Cassandra per accedere ad Amazon Keyspaces a livello di programmazione](#)
- [Utilizzo di un driver client Cassandra Python per accedere ad Amazon Keyspaces a livello di codice](#)
- [Utilizzo di un driver client Cassandra Node.js per accedere ad Amazon Keyspaces a livello di codice](#)
- [Utilizzo di un driver client Cassandra.NET Core per accedere ad Amazon Keyspaces a livello di codice](#)
- [Utilizzo di un driver client Cassandra Go per accedere ad Amazon Keyspaces a livello di codice](#)
- [Utilizzo di un driver client Cassandra Perl per accedere ad Amazon Keyspaces a livello di codice](#)

Utilizzo di un driver client Java Cassandra per accedere ad Amazon Keyspaces a livello di programmazione

Questa sezione mostra come connettersi ad Amazon Keyspaces utilizzando un driver client Java.

Note

Java 17 e DataStax Java Driver 4.17 sono attualmente supportati solo in versione beta. Per ulteriori informazioni, consulta https://docs.datastax.com/en/developer/java-driver/4.17/upgrade_guide/.

Per fornire agli utenti e alle applicazioni le credenziali per l'accesso programmatico alle risorse di Amazon Keyspaces, puoi effettuare una delle seguenti operazioni:

- Crea credenziali specifiche del servizio associate a un utente specifico (IAM). AWS Identity and Access Management
- Per una maggiore sicurezza, consigliamo di creare chiavi di accesso IAM per le identità IAM utilizzate in tutti i servizi. AWS Il plug-in di autenticazione Amazon Keyspaces SigV4 per i driver

client Cassandra consente di autenticare le chiamate ad Amazon Keyspaces utilizzando chiavi di accesso IAM anziché nome utente e password. Per ulteriori informazioni, consulta [the section called “Crea credenziali IAM per l'autenticazione AWS”](#).

Note

Per un esempio di come usare Amazon Keyspaces con Spring Boot, vedi. <https://github.com/aws-samples/amazon-keyspaces-examples/tree/main/java/datastax-v4/spring>

Argomenti

- [Prima di iniziare](#)
- [Step-by-step tutorial per connettersi ad Amazon Keyspaces utilizzando il driver DataStax Java per Apache Cassandra utilizzando credenziali specifiche del servizio](#)
- [Step-by-step tutorial per connettersi ad Amazon Keyspaces utilizzando il driver DataStax Java 4.x per Apache Cassandra e il plug-in di autenticazione SigV4](#)
- [Connect ad Amazon Keyspaces utilizzando il driver DataStax Java 3.x per Apache Cassandra e il plug-in di autenticazione SigV4](#)

Prima di iniziare

Per connetterti ad Amazon Keyspaces, devi completare le seguenti attività prima di iniziare.

1. Amazon Keyspaces richiede l'uso di Transport Layer Security (TLS) per proteggere le connessioni con i client.
 - a. Scarica i seguenti certificati digitali e salva i file localmente o nella tua home directory.
 1. AmazonRootCA1
 2. AmazonRootCA2
 3. AmazonRootCA3
 4. AmazonRootCA4
 5. Starfield Class 2 Root (opzionale, per compatibilità con le versioni precedenti)

Per scaricare i certificati, puoi usare i seguenti comandi.

```
curl -O https://www.amazontrust.com/repository/AmazonRootCA1.pem
curl -O https://www.amazontrust.com/repository/AmazonRootCA2.pem
curl -O https://www.amazontrust.com/repository/AmazonRootCA3.pem
curl -O https://www.amazontrust.com/repository/AmazonRootCA4.pem
curl -O https://certs.secureserver.net/repository/sf-class2-root.crt
```

 Note

Amazon Keyspaces utilizzava in precedenza certificati TLS ancorati alla CA Starfield Class 2. AWS sta migrando tutto Regioni AWS verso certificati emessi nell'ambito di Amazon Trust Services (Amazon Root CAs 1—4). Durante questa transizione, configura i client in modo che si fidino sia di Amazon Root CAs 1-4 che di Starfield root per garantire la compatibilità in tutte le regioni.

- b. Converti i certificati digitali in file TrustStore e aggiungili al keystore.

```
openssl x509 -outform der -in AmazonRootCA1.pem -out temp_file.der
keytool -import -alias amazon-root-ca-1 -keystore cassandra_truststore.jks -
file temp_file.der

openssl x509 -outform der -in AmazonRootCA2.pem -out temp_file.der
keytool -import -alias amazon-root-ca-2 -keystore cassandra_truststore.jks -
file temp_file.der

openssl x509 -outform der -in AmazonRootCA3.pem -out temp_file.der
keytool -import -alias amazon-root-ca-3 -keystore cassandra_truststore.jks -
file temp_file.der

openssl x509 -outform der -in AmazonRootCA4.pem -out temp_file.der
keytool -import -alias amazon-root-ca-4 -keystore cassandra_truststore.jks -
file temp_file.der

openssl x509 -outform der -in sf-class2-root.crt -out temp_file.der
keytool -import -alias cassandra -keystore cassandra_truststore.jks -file
temp_file.der
```

Nell'ultimo passaggio, è necessario creare una password per il keystore e considerare attendibile ogni certificato. Il comando interattivo ha questo aspetto.

```

Enter keystore password:
Re-enter new password:
Owner: CN=Amazon Root CA 1, O=Amazon, C=US
Issuer: CN=Amazon Root CA 1, O=Amazon, C=US
Serial number: 66c9fcf99bf8c0a39e2f0788a43e696365bca
Valid from: Tue May 26 00:00:00 UTC 2015 until: Sun Jan 17 00:00:00 UTC 2038
Certificate fingerprints:
    SHA1: 8D:A7:F9:65:EC:5E:FC:37:91:0F:1C:6E:59:FD:C1:CC:6A:6E:DE:16
    SHA256:
      8E:CD:E6:88:4F:3D:87:B1:12:5B:A3:1A:C3:FC:B1:3D:70:16:DE:7F:57:CC:90:4F:E1:CB:97:C6:AE
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 2048-bit RSA key
Version: 3

Extensions:

#1: ObjectId: 2.5.29.19 Criticality=true
BasicConstraints:[
    CA:true
    PathLen:2147483647
]

#2: ObjectId: 2.5.29.15 Criticality=true
KeyUsage [
    DigitalSignature
    Key_CertSign
    Crl_Sign
]

#3: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 84 18 CC 85 34 EC BC 0C   94 94 2E 08 59 9C C7 B2   ....4.....Y...
0010: 10 4E 0A 08                               .N..
]
]

Trust this certificate? [no]: yes
Certificate was added to keystore
Enter keystore password:
Owner: CN=Amazon Root CA 2, O=Amazon, C=US
Issuer: CN=Amazon Root CA 2, O=Amazon, C=US
Serial number: 66c9fd29635869f0a0fe58678f85b26bb8a37

```

```

Valid from: Tue May 26 00:00:00 UTC 2015 until: Sat May 26 00:00:00 UTC 2040
Certificate fingerprints:
    SHA1: 5A:8C:EF:45:D7:A6:98:59:76:7A:8C:8B:44:96:B5:78:CF:47:4B:1A
    SHA256:
    1B:A5:B2:AA:8C:65:40:1A:82:96:01:18:F8:0B:EC:4F:62:30:4D:83:CE:C4:71:3A:19:C3:9C:01:1E
Signature algorithm name: SHA384withRSA
Subject Public Key Algorithm: 4096-bit RSA key
Version: 3

Extensions:

#1: ObjectId: 2.5.29.19 Criticality=true
BasicConstraints:[
    CA:true
    PathLen:2147483647
]

#2: ObjectId: 2.5.29.15 Criticality=true
KeyUsage [
    DigitalSignature
    Key_CertSign
    Crl_Sign
]

#3: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: B0 0C F0 4C 30 F4 05 58    02 48 FD 33 E5 52 AF 4B    ...L0..X.H.3.R.K
0010: 84 E3 66 52                                ..fR
]
]

Trust this certificate? [no]: yes
Certificate was added to keystore
Enter keystore password:
Owner: CN=Amazon Root CA 3, O=Amazon, C=US
Issuer: CN=Amazon Root CA 3, O=Amazon, C=US
Serial number: 66c9fd5749736663f3b0b9ad9e89e7603f24a
Valid from: Tue May 26 00:00:00 UTC 2015 until: Sat May 26 00:00:00 UTC 2040
Certificate fingerprints:
    SHA1: 0D:44:DD:8C:3C:8C:1A:1A:58:75:64:81:E9:0F:2E:2A:FF:B3:D2:6E
    SHA256:
    18:CE:6C:FE:7B:F1:4E:60:B2:E3:47:B8:DF:E8:68:CB:31:D0:2E:BB:3A:DA:27:15:69:F5:03:43:B4
Signature algorithm name: SHA256withECDSA

```

Subject Public Key Algorithm: 256-bit EC (secp256r1) key

Version: 3

Extensions:

#1: ObjectId: 2.5.29.19 Criticality=true

BasicConstraints:[

CA:true

PathLen:2147483647

]

#2: ObjectId: 2.5.29.15 Criticality=true

KeyUsage [

DigitalSignature

Key_CertSign

Crl_Sign

]

#3: ObjectId: 2.5.29.14 Criticality=false

SubjectKeyIdentifier [

KeyIdentifier [

0000: AB B6 DB D7 06 9E 37 AC 30 86 07 91 70 C7 9C C4 7.0...p...

0010: 19 B1 78 C0 ...x.

]

]

Trust this certificate? [no]: yes

Certificate was added to keystore

Enter keystore password:

Owner: CN=Amazon Root CA 4, O=Amazon, C=US

Issuer: CN=Amazon Root CA 4, O=Amazon, C=US

Serial number: 66c9fd7c1bb104c2943e5717b7b2cc81ac10e

Valid from: Tue May 26 00:00:00 UTC 2015 until: Sat May 26 00:00:00 UTC 2040

Certificate fingerprints:

SHA1: F6:10:84:07:D6:F8:BB:67:98:0C:C2:E2:44:C2:EB:AE:1C:EF:63:BE

SHA256:

E3:5D:28:41:9E:D0:20:25:CF:A6:90:38:CD:62:39:62:45:8D:A5:C6:95:FB:DE:A3:C2:2B:0B:FB:25

Signature algorithm name: SHA384withECDSA

Subject Public Key Algorithm: 384-bit EC (secp384r1) key

Version: 3

Extensions:

#1: ObjectId: 2.5.29.19 Criticality=true

```

BasicConstraints:[
  CA:true
  PathLen:2147483647
]

#2: ObjectId: 2.5.29.15 Criticality=true
KeyUsage [
  DigitalSignature
  Key_CertSign
  Crl_Sign
]

#3: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: D3 EC C7 3A 65 6E CC E1   DA 76 9A 56 FB 9C F3 86   ...:en...v.V....
0010: 6D 57 E5 81                               mW..
]
]

Trust this certificate? [no]:  yes
Certificate was added to keystore
Enter keystore password:
Owner: OU=Starfield Class 2 Certification Authority, O="Starfield Technologies,
  Inc.", C=US
Issuer: OU=Starfield Class 2 Certification Authority, O="Starfield
  Technologies, Inc.", C=US
Serial number: 0
Valid from: Tue Jun 29 17:39:16 UTC 2004 until: Thu Jun 29 17:39:16 UTC 2034
Certificate fingerprints:
    SHA1: AD:7E:1C:28:B0:64:EF:8F:60:03:40:20:14:C3:D0:E3:37:0E:B5:8A
    SHA256:
      14:65:FA:20:53:97:B8:76:FA:A6:F0:A9:95:8E:55:90:E4:0F:CC:7F:AA:4F:B7:C2:C8:67:75:21:FB
Signature algorithm name: SHA1withRSA (weak)
Subject Public Key Algorithm: 2048-bit RSA key
Version: 3

Extensions:

#1: ObjectId: 2.5.29.35 Criticality=false
AuthorityKeyIdentifier [
KeyIdentifier [
0000: BF 5F B7 D1 CE DD 1F 86   F4 5B 55 AC DC D7 10 C2   ._.....[U.....
0010: 0E A9 88 E7                               ....

```

```

]
[OU=Starfield Class 2 Certification Authority, O="Starfield Technologies,
  Inc.", C=US]
SerialNumber: [    00]
]

#2: ObjectId: 2.5.29.19 Criticality=false
BasicConstraints:[
  CA:true
  PathLen:2147483647
]

#3: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: BF 5F B7 D1 CE DD 1F 86    F4 5B 55 AC DC D7 10 C2  ._.....[U.....
0010: 0E A9 88 E7                      ....
]
]

Warning:
The input uses the SHA1withRSA signature algorithm which is considered a
security risk. This algorithm will be disabled in a future update.

Trust this certificate? [no]: yes
Certificate was added to keystore

```

2. Allega il file TrustStore negli argomenti JVM:

```

-Djavax.net.ssl.trustStore=path_to_file/cassandra_truststore.jks
-Djavax.net.ssl.trustStorePassword=my_password

```

Step-by-step tutorial per connettersi ad Amazon Keyspaces utilizzando il driver DataStax Java per Apache Cassandra utilizzando credenziali specifiche del servizio

Il seguente step-by-step tutorial illustra come connettersi ad Amazon Keyspaces utilizzando un driver Java per Cassandra utilizzando credenziali specifiche del servizio. In particolare, utilizzerai la versione 4.0 del driver DataStax Java per Apache Cassandra.

Argomenti

- [Fase 1: prerequisiti](#)
- [Fase 2: Configurare il driver](#)
- [Fase 3: Esegui l'applicazione di esempio](#)

Fase 1: prerequisiti

Per seguire questo tutorial, è necessario generare credenziali specifiche del servizio e aggiungere il driver DataStax Java per Apache Cassandra al progetto Java.

- Genera credenziali specifiche del servizio per il tuo utente IAM di Amazon Keyspaces completando i passaggi indicati in. [the section called “Crea credenziali specifiche per il servizio”](#) Se preferisci utilizzare le chiavi di accesso IAM per l'autenticazione, consulta. [the section called “Plugin di autenticazione per Java 4.x”](#)
- Aggiungi il driver DataStax Java per Apache Cassandra al tuo progetto Java. Assicurati di utilizzare una versione del driver che supporti Apache Cassandra 3.11.2. Per ulteriori informazioni, consulta la documentazione del [driver DataStax Java per Apache Cassandra](#).

Fase 2: Configurare il driver

È possibile specificare le impostazioni per il driver DataStax Java Cassandra creando un file di configurazione per l'applicazione. Questo file di configurazione sostituisce le impostazioni predefinite e indica al driver di connettersi all'endpoint del servizio Amazon Keyspaces utilizzando la porta 9142. Per un elenco degli endpoint di servizio disponibili, consulta. [the section called “Endpoint di servizio”](#)

Create un file di configurazione e salvatelo nella cartella delle risorse dell'applicazione, ad esempio. `src/main/resources/application.conf` Apri `application.conf` e aggiungi le seguenti impostazioni di configurazione.

1. Provider di autenticazione: crea il provider di autenticazione con la `PlainTextAuthProvider` classe. `ServiceUsername` e `ServicePassword` deve corrispondere al nome utente e alla password ottenuti al momento della generazione delle credenziali specifiche del servizio seguendo la procedura riportata di seguito. [Crea credenziali specifiche del servizio per l'accesso programmatico ad Amazon Keyspaces](#)

Note

È possibile utilizzare credenziali a breve termine utilizzando il plug-in di autenticazione per il driver DataStax Java per Apache Cassandra anziché codificare le credenziali nel

file di configurazione del driver. Per ulteriori informazioni, segui le istruzioni per. [the section called “Plugin di autenticazione per Java 4.x”](#)

2. Data center locale: imposta il valore per la regione `local-datacenter` a cui ti stai connettendo. Ad esempio, se l'applicazione si connette a `cassandra.us-east-2.amazonaws.com`, imposta il data center locale su `us-east-2`. Per tutte le opzioni disponibili Regioni AWS, vedi [???](#). Impostato `slow-replica-avoidance = false` per bilanciare il carico su un minor numero di nodi.
3. SSL/TLS: inizializza `SSL Engine Factory` aggiungendo una sezione nel file di configurazione con una singola riga che specifica la classe con `class = DefaultSslEngineFactory`. Fornisci il percorso del file `TrustStore` e la password che hai creato in precedenza. Amazon Keyspaces non supporta `hostname-validation` i peer, quindi imposta questa opzione su `false`.

```
datastax-java-driver {  
  
    basic.contact-points = [ "cassandra.us-east-2.amazonaws.com:9142"  
    advanced.auth-provider{  
        class = PlainTextAuthProvider  
        username = "ServiceUserName"  
        password = "ServicePassword"  
    }  
    basic.load-balancing-policy {  
        local-datacenter = "us-east-2"  
        slow-replica-avoidance = false  
    }  
  
    advanced.ssl-engine-factory {  
        class = DefaultSslEngineFactory  
        truststore-path = "./src/main/resources/cassandra_truststore.jks"  
        truststore-password = "my_password"  
        hostname-validation = false  
    }  
}
```

 Note

Invece di aggiungere il percorso al TrustStore nel file di configurazione, puoi anche aggiungere il percorso TrustStore direttamente nel codice dell'applicazione oppure puoi aggiungere il percorso al TrustStore agli argomenti JVM.

Fase 3: Esegui l'applicazione di esempio

Questo esempio di codice mostra una semplice applicazione a riga di comando che crea un pool di connessioni ad Amazon Keyspaces utilizzando il file di configurazione creato in precedenza. Conferma che la connessione è stata stabilita eseguendo una semplice query.

```
package <your package>;
// add the following imports to your project
import com.datastax.oss.driver.api.core.CqlSession;
import com.datastax.oss.driver.api.core.config.DriverConfigLoader;
import com.datastax.oss.driver.api.core.cql.ResultSet;
import com.datastax.oss.driver.api.core.cql.Row;

public class App
{
    public static void main( String[] args )
    {
        //Use DriverConfigLoader to load your configuration file
        DriverConfigLoader loader =
        DriverConfigLoader.fromClasspath("application.conf");
        try (CqlSession session = CqlSession.builder()
            .withConfigLoader(loader)
            .build()) {

            ResultSet rs = session.execute("select * from system_schema.keyspaces");
            Row row = rs.one();
            System.out.println(row.getString("keyspace_name"));

        }
    }
}
```

 Note

Usa un `try` blocco per stabilire la connessione per assicurarti che sia sempre chiusa. Se non usi un `try` blocco, ricordati di chiudere la connessione per evitare perdite di risorse.

Step-by-step tutorial per connettersi ad Amazon Keyspaces utilizzando il driver DataStax Java 4.x per Apache Cassandra e il plug-in di autenticazione SigV4

La sezione seguente descrive come utilizzare il plug-in di autenticazione SigV4 per il driver DataStax Java 4.x open source per Apache Cassandra per accedere ad Amazon Keyspaces (per Apache Cassandra). [Il plugin è disponibile nel repository. GitHub](#)

Il plug-in di autenticazione SigV4 consente di utilizzare le credenziali IAM per utenti o ruoli durante la connessione ad Amazon Keyspaces. Invece di richiedere un nome utente e una password, questo plugin firma le richieste API utilizzando le chiavi di accesso. Per ulteriori informazioni, consulta [the section called “Crea credenziali IAM per l'autenticazione AWS”](#).

Fase 1: prerequisiti

Per seguire questo tutorial, devi completare le seguenti attività.

- Se non l'hai già fatto, crea le credenziali per il tuo utente o ruolo IAM seguendo la procedura riportata qui. [the section called “Crea credenziali IAM per l'autenticazione AWS”](#) Questo tutorial presuppone che le chiavi di accesso siano memorizzate come variabili di ambiente. Per ulteriori informazioni, consulta [the section called “Gestione delle chiavi di accesso”](#).
- Aggiungi il driver DataStax Java per Apache Cassandra al tuo progetto Java. Assicurati di utilizzare una versione del driver che supporti Apache Cassandra 3.11.2. Per ulteriori informazioni, consulta la documentazione del [driver DataStax Java per Apache Cassandra](#).
- Aggiungi il plugin di autenticazione alla tua applicazione. Il plugin di autenticazione supporta la versione 4.x del driver DataStax Java per Apache Cassandra. Se utilizzi Apache Maven o un sistema di compilazione in grado di utilizzare le dipendenze Maven, aggiungi le seguenti dipendenze al tuo file. `pom.xml`

 Important

[Sostituisci la versione del plugin con l'ultima versione mostrata nel repository. GitHub](#)

```
<dependency>
  <groupId>software.aws.mcs</groupId>
  <artifactId>aws-sigv4-auth-cassandra-java-driver-plugin</artifactId>
  <version>4.0.9</version>
</dependency>
```

Fase 2: Configurare il driver

È possibile specificare le impostazioni per il driver DataStax Java Cassandra creando un file di configurazione per l'applicazione. Questo file di configurazione sostituisce le impostazioni predefinite e indica al driver di connettersi all'endpoint del servizio Amazon Keyspaces utilizzando la porta 9142. Per un elenco degli endpoint di servizio disponibili, consulta [the section called “Endpoint di servizio”](#)

Create un file di configurazione e salvatelo nella cartella delle risorse dell'applicazione, ad esempio. `src/main/resources/application.conf` Apri `application.conf` e aggiungi le seguenti impostazioni di configurazione.

1. **Provider di autenticazione:** imposta una nuova istanza `software.aws.mcs.auth.SigV4AuthProvider`. `advanced.auth-provider.class SigV4 AuthProvider` è il gestore di autenticazione fornito dal plugin per eseguire l'autenticazione SigV4.
2. **Data center locale:** imposta il valore per la regione `local-datacenter` a cui ti stai connettendo. Ad esempio, se l'applicazione si connette a `cassandra.us-east-2.amazonaws.com`, imposta il data center locale su `us-east-2`. Per tutte le opzioni disponibili Regioni AWS, vedi [AWS Regions](#). Impostato `slow-replica-avoidance = false` per bilanciare il carico rispetto a tutti i nodi disponibili.
3. **Idempotenza:** imposta l'impostazione predefinita `idempotence` per l'applicazione in modo da configurare il driver in modo `true` che riprovi sempre le richieste non riuscite. `read/write/prepare/execute` Si tratta di una procedura ottimale per le applicazioni distribuite che consente di gestire gli errori temporanei riprovando le richieste non riuscite.
4. **SSL/TLS:** inizializza la `SSL Engine` fabbrica aggiungendo una sezione nel file di configurazione con una sola riga che specifica la classe con. `class = DefaultSslEngineFactory` Fornisci il percorso del file `TrustStore` e la password che hai creato in precedenza. Amazon Keyspaces non supporta `hostname-validation` i peer, quindi imposta questa opzione su `false`.

5. Connessioni: crea almeno 3 connessioni locali per endpoint impostando. `local.size = 3` Si tratta di una best practice che aiuta l'applicazione a gestire i sovraccarichi e i picchi di traffico. Per ulteriori informazioni su come calcolare il numero di connessioni locali per endpoint necessarie all'applicazione in base ai modelli di traffico previsti, consulta [the section called “Come configurare le connessioni”](#)
6. Politica di riprova: implementa la `AmazonKeyspacesExponentialRetryPolicy` politica di riprova di Amazon Keyspaces anziché quella fornita con `DefaultRetryPolicy` il driver Cassandra. Ciò ti consente di configurare il numero di tentativi di riprova per quello che soddisfa le `AmazonKeyspacesExponentialRetryPolicy` tue esigenze. Per impostazione predefinita, il numero di tentativi di nuovo tentativo `AmazonKeyspacesExponentialRetryPolicy` è impostato su 3. Per ulteriori informazioni, consulta [the section called “Come configurare le politiche di nuovo tentativo”](#).
7. Dichiarazioni preparate: imposta su `false` `prepare-on-all-nodes` per ottimizzare l'utilizzo della rete.

```
datastax-java-driver {  
  basic {  
    contact-points = [ "cassandra.us-east-2.amazonaws.com:9142"  
    request {  
      timeout = 2 seconds  
      consistency = LOCAL_QUORUM  
      page-size = 1024  
      default-idempotence = true  
    }  
    load-balancing-policy {  
      local-datacenter = "us-east-2"  
      class = DefaultLoadBalancingPolicy  
      slow-replica-avoidance = false  
    }  
  }  
  advanced {  
    auth-provider {  
      class = software.aws.mcs.auth.SigV4AuthProvider  
      aws-region = us-east-2  
    }  
    ssl-engine-factory {  
      class = DefaultSslEngineFactory  
      truststore-path = "./src/main/resources/cassandra_truststore.jks"  
      truststore-password = "my_password"
```

```

        hostname-validation = false
    }
    connection {
        connect-timeout = 5 seconds
        max-requests-per-connection = 512
        pool {
            local.size = 3
        }
    }
    retry-policy {
        class = com.aws.ssa.keyspaces.retry.AmazonKeyspacesExponentialRetryPolicy
    }
    max-attempts = 3
    min-wait = 10 mills
    max-wait = 100 mills
    prepared-statements {
        prepare-on-all-nodes = false
    }
}
}

```

Note

Invece di aggiungere il percorso al TrustStore nel file di configurazione, puoi anche aggiungere il percorso TrustStore direttamente nel codice dell'applicazione oppure puoi aggiungere il percorso al TrustStore agli argomenti JVM.

Passaggio 3: Esegui l'applicazione

Questo esempio di codice mostra una semplice applicazione a riga di comando che crea un pool di connessioni ad Amazon Keyspaces utilizzando il file di configurazione creato in precedenza. Conferma che la connessione è stata stabilita eseguendo una semplice query.

```

package <your package>;
// add the following imports to your project
import com.datastax.oss.driver.api.core.CqlSession;
import com.datastax.oss.driver.api.core.config.DriverConfigLoader;
import com.datastax.oss.driver.api.core.cql.ResultSet;
import com.datastax.oss.driver.api.core.cql.Row;

public class App

```

```
{

    public static void main( String[] args )
    {
        //Use DriverConfigLoader to load your configuration file
        DriverConfigLoader loader =
        DriverConfigLoader.fromClasspath("application.conf");
        try (CqlSession session = CqlSession.builder()
            .withConfigLoader(loader)
            .build()) {

            ResultSet rs = session.execute("select * from system_schema.keyspaces");
            Row row = rs.one();
            System.out.println(row.getString("keyspace_name"));

        }
    }
}
```

Note

Usa un try blocco per stabilire la connessione per assicurarti che sia sempre chiusa. Se non usi un try blocco, ricordati di chiudere la connessione per evitare perdite di risorse.

Connect ad Amazon Keyspaces utilizzando il driver DataStax Java 3.x per Apache Cassandra e il plug-in di autenticazione SigV4

La sezione seguente descrive come utilizzare il plug-in di autenticazione SigV4 per il driver DataStax Java open source 3.x per Apache Cassandra per accedere ad Amazon Keyspaces. [Il plugin è disponibile nel repository. GitHub](#)

Il plug-in di autenticazione SigV4 consente di utilizzare le credenziali IAM per utenti e ruoli durante la connessione ad Amazon Keyspaces. Invece di richiedere un nome utente e una password, questo plugin firma le richieste API utilizzando le chiavi di accesso. Per ulteriori informazioni, consulta [the section called “Crea credenziali IAM per l'autenticazione AWS”](#).

Fase 1: prerequisiti

Per eseguire questo esempio di codice, è necessario innanzitutto completare le seguenti attività.

- Crea le credenziali per il tuo utente o ruolo IAM seguendo la procedura descritta in [the section called “Crea credenziali IAM per l'autenticazione AWS”](#). Questo tutorial presuppone che le chiavi di accesso siano memorizzate come variabili di ambiente. Per ulteriori informazioni, consulta [the section called “Gestione delle chiavi di accesso”](#).
- Segui i passaggi indicati [the section called “Prima di iniziare”](#) per scaricare i certificati digitali, convertirli in file TrustStore e allegare il keystore negli argomenti JVM all'applicazione.
- Aggiungi il driver DataStax Java per Apache Cassandra al tuo progetto Java. Assicurati di utilizzare una versione del driver che supporti Apache Cassandra 3.11.2. Per ulteriori informazioni, consulta la documentazione del [driver DataStax Java per Apache Cassandra](#).
- Aggiungi il plugin di autenticazione alla tua applicazione. Il plugin di autenticazione supporta la versione 3.x del driver DataStax Java per Apache Cassandra. Se utilizzi Apache Maven o un sistema di compilazione in grado di utilizzare le dipendenze Maven, aggiungi le seguenti dipendenze al tuo file. pom.xml [Sostituisci la versione del plugin con l'ultima versione mostrata nel repository. GitHub](#)

```
<dependency>
  <groupId>software.aws.mcs</groupId>
  <artifactId>aws-sigv4-auth-cassandra-java-driver-plugin_3</artifactId>
  <version>3.0.3</version>
</dependency>
```

Fase 2: Esegui l'applicazione

Questo esempio di codice mostra una semplice applicazione a riga di comando che crea un pool di connessioni ad Amazon Keyspaces. Conferma che la connessione è stata stabilita eseguendo una semplice query.

```
package <your package>;
// add the following imports to your project

import software.aws.mcs.auth.SigV4AuthProvider;
import com.datastax.driver.core.Cluster;
import com.datastax.driver.core.ResultSet;
import com.datastax.driver.core.Row;
import com.datastax.driver.core.Session;

public class App
{
```



```
public static void main( String[] args )
{
    String endPoint = "cassandra.us-east-2.amazonaws.com";
    int portNumber = 9142;
    Session session = Cluster.builder()
                            .addContactPoint(endPoint)
                            .withPort(portNumber)
                            .withAuthProvider(new SigV4AuthProvider("us-east-2"))

                            .withSSL()
                            .build()
                            .connect();

    ResultSet rs = session.execute("select * from system_schema.keyspaces");
    Row row = rs.one();
    System.out.println(row.getString("keyspace_name"));
}
}
```

Note d'uso:

Per un elenco degli endpoint disponibili, vedere [the section called “Endpoint di servizio”](#).

Consulta il seguente repository per utili politiche, esempi e best practice dei driver Java per l'utilizzo del driver Java con Amazon <https://github.com/aws-samples/amazon-keyspaces-java-driver-helpers> Keyspaces:.

Utilizzo di un driver client Cassandra Python per accedere ad Amazon Keyspaces a livello di codice

In questa sezione, ti mostriamo come connetterti ad Amazon Keyspaces usando un driver client Python. Per fornire agli utenti e alle applicazioni le credenziali per l'accesso programmatico alle risorse di Amazon Keyspaces, puoi effettuare una delle seguenti operazioni:

- Crea credenziali specifiche del servizio associate a un utente specifico (IAM). AWS Identity and Access Management
- Per una maggiore sicurezza, consigliamo di creare chiavi di accesso IAM per utenti o ruoli IAM utilizzati in tutti i servizi. AWS Il plug-in di autenticazione Amazon Keyspaces SigV4 per i driver client Cassandra consente di autenticare le chiamate ad Amazon Keyspaces utilizzando chiavi

di accesso IAM anziché nome utente e password. Per ulteriori informazioni, consulta [the section called “Crea credenziali IAM per l'autenticazione AWS”](#).

Argomenti

- [Prima di iniziare](#)
- [Connect ad Amazon Keyspaces utilizzando il driver Python per Apache Cassandra e le credenziali specifiche del servizio](#)
- [Connect ad Amazon Keyspaces utilizzando il driver DataStax Python per Apache Cassandra e il plug-in di autenticazione SigV4](#)

Prima di iniziare

È necessario completare la seguente attività prima di iniziare.

Amazon Keyspaces richiede l'uso di Transport Layer Security (TLS) per proteggere le connessioni con i client. Per connetterti ad Amazon Keyspaces tramite TLS, devi scaricare i certificati digitali Amazon e configurare il driver Python per utilizzare TLS.

Scarica i seguenti certificati digitali e salva i file localmente o nella tua home directory.

1. AmazonRootCA1
2. AmazonRootCA2
3. AmazonRootCA3
4. AmazonRootCA4
5. Starfield Class 2 Root (opzionale, per compatibilità con le versioni precedenti)

Per scaricare i certificati, puoi usare i seguenti comandi.

```
curl -O https://www.amazontrust.com/repository/AmazonRootCA1.pem
curl -O https://www.amazontrust.com/repository/AmazonRootCA2.pem
curl -O https://www.amazontrust.com/repository/AmazonRootCA3.pem
curl -O https://www.amazontrust.com/repository/AmazonRootCA4.pem
curl -O https://certs.seureserver.net/repository/sf-class2-root.crt
```

 Note

Amazon Keyspaces utilizzava in precedenza certificati TLS ancorati alla CA Starfield Class 2. AWS sta migrando tutte le Regioni AWS verso certificati emessi nell'ambito di Amazon Trust Services (Amazon Root CAs 1—4). Durante questa transizione, configura i client in modo che si fidino sia di Amazon Root CAs 1-4 che di Starfield root per garantire la compatibilità in tutte le regioni.

Combina tutti i certificati scaricati in un unico pem file con il nome indicato *keyspaces-bundle.pem* nei nostri esempi. Per aggiungerlo, eseguire il comando seguente: Prendi nota del percorso del file, ne avrai bisogno in seguito.

```
cat AmazonRootCA1.pem \  
    AmazonRootCA2.pem \  
    AmazonRootCA3.pem \  
    AmazonRootCA4.pem \  
    sf-class2-root.crt \  
    > keyspaces-bundle.pem
```

Connect ad Amazon Keyspaces utilizzando il driver Python per Apache Cassandra e le credenziali specifiche del servizio

Il seguente esempio di codice mostra come connettersi ad Amazon Keyspaces con un driver client Python e credenziali specifiche del servizio.

```
from cassandra.cluster import Cluster  
from ssl import SSLContext, PROTOCOL_TLSv1_2 , CERT_REQUIRED  
from cassandra.auth import PlainTextAuthProvider  
  
ssl_context = SSLContext(PROTOCOL_TLSv1_2 )  
  
ssl_context.load_verify_locations('path_to_file/keyspaces-bundle.pem')  
  
ssl_context.verify_mode = CERT_REQUIRED  
auth_provider = PlainTextAuthProvider(username='ServiceUserName',  
    password='ServicePassword')  
cluster = Cluster(['cassandra.us-east-2.amazonaws.com'], ssl_context=ssl_context,  
    auth_provider=auth_provider, port=9142)  
session = cluster.connect()
```

```
r = session.execute('select * from system_schema.keyspaces')
print(r.current_rows)
```

Note sull'utilizzo:

1. Sostituisci "*path_to_file/keyspaces-bundle.pem*" con il percorso del file di certificato combinato salvato nel primo passaggio.
2. Assicurati che *ServiceUserName* e *ServicePassword* corrispondano al nome utente e alla password che hai ottenuto quando hai generato le credenziali specifiche del servizio seguendo i passaggi per. [Crea credenziali specifiche del servizio per l'accesso programmatico ad Amazon Keyspaces](#)
3. Per un elenco degli endpoint disponibili, consulta. [the section called “Endpoint di servizio”](#)

Connect ad Amazon Keyspaces utilizzando il driver DataStax Python per Apache Cassandra e il plug-in di autenticazione SigV4

La sezione seguente mostra come utilizzare il plug-in di autenticazione SigV4 per il driver DataStax Python open source per Apache Cassandra per accedere ad Amazon Keyspaces (per Apache Cassandra).

Se non l'hai già fatto, inizia a creare le credenziali per il tuo ruolo IAM seguendo i passaggi riportati di seguito. [the section called “Crea credenziali IAM per l'autenticazione AWS”](#) Questo tutorial utilizza credenziali temporanee, che richiedono un ruolo IAM. Per ulteriori informazioni sulle credenziali temporanee, consulta. [the section called “Crea credenziali temporanee per connetterti ad Amazon Keyspaces”](#)

[Quindi, aggiungi il plug-in di autenticazione Python SigV4 al tuo ambiente dal repository. GitHub](#)

```
pip install cassandra-sigv4
```

Il seguente esempio di codice mostra come connettersi ad Amazon Keyspaces utilizzando il driver DataStax Python open source per Cassandra e il plug-in di autenticazione SigV4. Il plugin dipende dall' AWS SDK per Python (Boto3). Viene utilizzato per `boto3.session` ottenere credenziali temporanee.

```
from cassandra.cluster import Cluster
from ssl import SSLContext, PROTOCOL_TLSv1_2 , CERT_REQUIRED
```

```

from cassandra.auth import PlainTextAuthProvider
import boto3
from cassandra_sigv4.auth import SigV4AuthProvider

ssl_context = SSLContext(PROTOCOL_TLSv1_2)
ssl_context.load_verify_locations('path_to_file/keyspaces-bundle.pem')
ssl_context.verify_mode = CERT_REQUIRED

# use this if you want to use Boto to set the session parameters.
boto_session = boto3.Session(aws_access_key_id="AKIAIOSFODNN7EXAMPLE",
                             aws_secret_access_key="wJalrXUtnFEMI/K7MDENG/
bPxRfiCYEXAMPLEKEY",
                             aws_session_token="AQoDYXdzEJr...<remainder of token>",
                             region_name="us-east-2")
auth_provider = SigV4AuthProvider(boto_session)

# Use this instead of the above line if you want to use the Default Credentials and not
# bother with a session.
# auth_provider = SigV4AuthProvider()

cluster = Cluster(['cassandra.us-east-2.amazonaws.com'], ssl_context=ssl_context,
                  auth_provider=auth_provider,
                  port=9142)
session = cluster.connect()
r = session.execute('select * from system_schema.keyspaces')
print(r.current_rows)

```

Note d'uso:

1. Sostituisci "*path_to_file/keyspaces-bundle.pem*" con il percorso del certificato salvato nel primo passaggio.
2. Assicurati che il *aws_access_key_id*, *aws_secret_access_key*, e il *aws_session_token* corrispondano a Access Key, Secret Access Key, e Session Token che hai ottenuto utilizzando `boto3.session`. Per ulteriori informazioni, consulta [Credentials](#) nella AWS SDK per Python (Boto3).
3. Per un elenco degli endpoint disponibili, consulta [the section called “Endpoint di servizio”](#).

Utilizzo di un driver client Cassandra Node.js per accedere ad Amazon Keyspaces a livello di codice

Questa sezione mostra come connettersi ad Amazon Keyspaces utilizzando un driver client Node.js. Per fornire agli utenti e alle applicazioni le credenziali per l'accesso programmatico alle risorse di Amazon Keyspaces, puoi effettuare una delle seguenti operazioni:

- Crea credenziali specifiche del servizio associate a un utente specifico (IAM). AWS Identity and Access Management
- Per una maggiore sicurezza, consigliamo di creare chiavi di accesso IAM per utenti o ruoli IAM utilizzati in tutti i servizi. AWS Il plug-in di autenticazione Amazon Keyspaces SigV4 per i driver client Cassandra consente di autenticare le chiamate ad Amazon Keyspaces utilizzando chiavi di accesso IAM anziché nome utente e password. Per ulteriori informazioni, consulta [the section called “Crea credenziali IAM per l'autenticazione AWS”](#).

Argomenti

- [Prima di iniziare](#)
- [Connect ad Amazon Keyspaces utilizzando il DataStax driver Node.js per Apache Cassandra e le credenziali specifiche del servizio](#)
- [Connect ad Amazon Keyspaces utilizzando il driver DataStax Node.js per Apache Cassandra e il plug-in di autenticazione SigV4](#)

Prima di iniziare

È necessario completare la seguente attività prima di iniziare.

Amazon Keyspaces richiede l'uso di Transport Layer Security (TLS) per proteggere le connessioni con i client. Per connetterti ad Amazon Keyspaces tramite TLS, devi scaricare un certificato digitale Amazon e configurare il driver Python per utilizzare TLS.

Scarica i seguenti certificati digitali e salva i file localmente o nella tua home directory.

1. AmazonRootCA1
2. AmazonRootCA2
3. AmazonRootCA3

4. AmazonRootCA4

5. Starfield Class 2 Root (opzionale, per compatibilità con le versioni precedenti)

Per scaricare i certificati, puoi usare i seguenti comandi.

```
curl -O https://www.amazontrust.com/repository/AmazonRootCA1.pem
curl -O https://www.amazontrust.com/repository/AmazonRootCA2.pem
curl -O https://www.amazontrust.com/repository/AmazonRootCA3.pem
curl -O https://www.amazontrust.com/repository/AmazonRootCA4.pem
curl -O https://certs.secureserver.net/repository/sf-class2-root.crt
```

Note

Amazon Keyspaces utilizzava in precedenza certificati TLS ancorati alla CA Starfield Class 2. AWS sta migrando tutte le Regioni AWS verso certificati emessi nell'ambito di Amazon Trust Services (Amazon Root CAs 1—4). Durante questa transizione, configura i client in modo che si fidino sia di Amazon Root CAs 1-4 che di Starfield root per garantire la compatibilità in tutte le regioni.

Combina tutti i certificati scaricati in un unico pem file con il nome indicato *keyspaces-bundle.pem* nei nostri esempi. Per aggiungerlo, eseguire il comando seguente: Prendi nota del percorso del file, ne avrai bisogno in seguito.

```
cat AmazonRootCA1.pem \
    AmazonRootCA2.pem \
    AmazonRootCA3.pem \
    AmazonRootCA4.pem \
    sf-class2-root.crt \
    > keyspaces-bundle.pem
```

Connect ad Amazon Keyspaces utilizzando il DataStax driver Node.js per Apache Cassandra e le credenziali specifiche del servizio

Configura il driver per utilizzare il file di certificato combinato *keyspaces-bundle.pem* per TLS e autenticarsi utilizzando credenziali specifiche del servizio. Ad esempio:

```
const cassandra = require('cassandra-driver');
```

```
const fs = require('fs');
const auth = new cassandra.auth.PlainTextAuthProvider('ServiceUserName',
  'ServicePassword');
const sslOptions1 = {
  ca: [
    fs.readFileSync('path_to_file/keyspaces-bundle.pem', 'utf-8')],

  host: 'cassandra.us-west-2.amazonaws.com',
  rejectUnauthorized: true
};
const client = new cassandra.Client({
  contactPoints: ['cassandra.us-west-2.amazonaws.com'],
  localDataCenter: 'us-west-2',
  authProvider: auth,
  sslOptions: sslOptions1,
  protocolOptions: { port: 9142 }
});
const query = 'SELECT * FROM system_schema.keyspaces';

client.execute(query)
  .then( result => console.log('Row from Keyspaces %s',
    result.rows[0]))
  .catch( e=> console.log(`${e}`));
```

Note sull'utilizzo:

1. Sostituisci "*path_to_file/keyspaces-bundle.pem*" con il percorso del file di certificato combinato salvato nel primo passaggio.
2. Assicurati che *ServiceUserName* e *ServicePassword* corrispondano al nome utente e alla password che hai ottenuto quando hai generato le credenziali specifiche del servizio seguendo i passaggi per. [Crea credenziali specifiche del servizio per l'accesso programmatico ad Amazon Keyspaces](#)
3. Per un elenco degli endpoint disponibili, consulta. [the section called "Endpoint di servizio"](#)

Connect ad Amazon Keyspaces utilizzando il driver DataStax Node.js per Apache Cassandra e il plug-in di autenticazione SigV4

La sezione seguente mostra come utilizzare il plug-in di autenticazione SigV4 per il driver open source DataStax Node.js per Apache Cassandra per accedere ad Amazon Keyspaces (per Apache Cassandra).

Se non l'hai già fatto, crea le credenziali per il tuo utente o ruolo IAM seguendo la procedura descritta qui. [the section called “Crea credenziali IAM per l'autenticazione AWS”](#)

[Aggiungi il plug-in di autenticazione SigV4 Node.js all'applicazione dal repository. GitHub](#) Il plugin supporta la versione 4.x del driver DataStax Node.js per Cassandra e dipende dall'SDK per Node.js. AWS Viene utilizzato per AWSCredentialsProvider ottenere le credenziali.

```
$ npm install aws-sigv4-auth-cassandra-plugin --save
```

Questo esempio di codice mostra come impostare un'istanza specifica della regione SigV4AuthProvider come provider di autenticazione.

```
const cassandra = require('cassandra-driver');
const fs = require('fs');
const sigV4 = require('aws-sigv4-auth-cassandra-plugin');

const auth = new sigV4.SigV4AuthProvider({
  region: 'us-west-2',
  accessKeyId: 'AKIAIOSFODNN7EXAMPLE',
  secretAccessKey: 'wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY'});

const sslOptions1 = {
  ca: [
    fs.readFileSync('path_to_file/keyspaces-bundle.pem', 'utf-8')],
  host: 'cassandra.us-west-2.amazonaws.com',
  rejectUnauthorized: true
};

const client = new cassandra.Client({
  contactPoints: ['cassandra.us-west-2.amazonaws.com'],
  localDataCenter: 'us-west-2',
  authProvider: auth,
  sslOptions: sslOptions1,
  protocolOptions: { port: 9142 }
});

const query = 'SELECT * FROM system_schema.keyspaces';

client.execute(query).then(
  result => console.log('Row from Keyspaces %s', result.rows[0]))
```

```
.catch( e=> console.log(`${e}`));
```

Note sull'utilizzo:

1. Sostituisci "*path_to_file/keyspaces-bundle.pem*" con il percorso del certificato salvato nel primo passaggio.
2. Assicurati che *accessKeyId* e *secretAccessKey* corrispondano alla chiave di accesso e alla chiave di accesso segreta che hai ottenuto utilizzando `AWSCredentialsProvider`. Per ulteriori informazioni, consulta [Impostazione delle credenziali in Node.js](#) nell'AWS SDK per JavaScript in Node.js.
3. Per archiviare le chiavi di accesso al di fuori del codice, consulta le best practice all'indirizzo. [the section called "Gestione delle chiavi di accesso"](#)
4. Per un elenco degli endpoint disponibili, consulta [the section called "Endpoint di servizio"](#).

Utilizzo di un driver client Cassandra.NET Core per accedere ad Amazon Keyspaces a livello di codice

Questa sezione mostra come connettersi ad Amazon Keyspaces utilizzando un driver client.NET Core. I passaggi di configurazione variano a seconda dell'ambiente e del sistema operativo, potrebbe essere necessario modificarli di conseguenza. Amazon Keyspaces richiede l'uso di Transport Layer Security (TLS) per proteggere le connessioni con i client. Per connetterti ad Amazon Keyspaces tramite TLS, configura il driver per utilizzare il trust store di sistema, che include Amazon Root 1-4. CAs

1. Installa il CSharp driver Cassandra tramite nuget, utilizzando la console nuget.

```
PM> Install-Package CassandraCSharpDriver
```

2. L'esempio seguente utilizza un progetto console.NET Core C# per connettersi ad Amazon Keyspaces ed eseguire una query.

```
using Cassandra;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net.Security;
using System.Runtime.ConstrainedExecution;
using System.Security.Cryptography.X509Certificates;
```

```
using System.Text;
using System.Threading.Tasks;

namespace CSharpKeyspacesExample
{
    class Program
    {
        public Program(){}

        static void Main(string[] args)
        {
            var userName = "ServiceUserName";
            var pwd = "ServicePassword";
            certCollection.Add(amazoncert);

            var awsEndpoint = "cassandra.us-east-2.amazonaws.com" ;

            var cluster = Cluster.Builder()
                .AddContactPoints(awsEndpoint)
                .WithPort(9142)
                .WithAuthProvider(new PlainTextAuthProvider(userName, pwd))
                .WithSSL(new
SSLOptions().SetCertificateCollection(certCollection))
                .Build();

            var session = cluster.Connect();
            var rs = session.Execute("SELECT * FROM system_schema.tables;");
            foreach (var row in rs)
            {
                var name = row.GetValue<String>("keyspace_name");
                Console.WriteLine(name);
            }
        }
    }
}
```

Note sull'utilizzo:

1. Assicurati di utilizzare l'archivio di fiducia di sistema predefinito, che include Amazon Root CAs 1-4.
2. Assicurati che *ServiceUserName* e *ServicePassword* corrispondano al nome utente e alla password che hai ottenuto quando hai generato le credenziali specifiche del servizio seguendo

i passaggi per. [Crea credenziali specifiche del servizio per l'accesso programmatico ad Amazon Keyspaces](#)

3. Per un elenco degli endpoint disponibili, consulta. [the section called “Endpoint di servizio”](#)

Utilizzo di un driver client Cassandra Go per accedere ad Amazon Keyspaces a livello di codice

Questa sezione mostra come connettersi ad Amazon Keyspaces utilizzando un driver client Go Cassandra. Per fornire agli utenti e alle applicazioni le credenziali per l'accesso programmatico alle risorse di Amazon Keyspaces, puoi effettuare una delle seguenti operazioni:

- Crea credenziali specifiche del servizio associate a un utente specifico (IAM). AWS Identity and Access Management
- Per una maggiore sicurezza, consigliamo di creare chiavi di accesso IAM per i principali IAM che vengono utilizzate in tutti i servizi. AWS Il plug-in di autenticazione Amazon Keyspaces SigV4 per i driver client Cassandra consente di autenticare le chiamate ad Amazon Keyspaces utilizzando chiavi di accesso IAM anziché nome utente e password. Per ulteriori informazioni, consulta [the section called “Crea credenziali IAM per l'autenticazione AWS ”](#).

Argomenti

- [Prima di iniziare](#)
- [Connect ad Amazon Keyspaces utilizzando il driver Gocql per Apache Cassandra e le credenziali specifiche del servizio](#)
- [Connect ad Amazon Keyspaces utilizzando il driver Go per Apache Cassandra e il plug-in di autenticazione SigV4](#)

Prima di iniziare

È necessario completare la seguente attività prima di iniziare.

Amazon Keyspaces richiede l'uso di Transport Layer Security (TLS) per proteggere le connessioni con i client. Per connetterti ad Amazon Keyspaces tramite TLS, devi scaricare un certificato digitale Amazon e configurare il driver Go per utilizzare TLS.

Scarica i seguenti certificati digitali e salva i file localmente o nella tua home directory.

1. AmazonRootCA1
2. AmazonRootCA2
3. AmazonRootCA3
4. AmazonRootCA4
5. Starfield Class 2 Root (opzionale, per compatibilità con le versioni precedenti)

Per scaricare i certificati, puoi usare i seguenti comandi.

```
curl -O https://www.amazontrust.com/repository/AmazonRootCA1.pem
curl -O https://www.amazontrust.com/repository/AmazonRootCA2.pem
curl -O https://www.amazontrust.com/repository/AmazonRootCA3.pem
curl -O https://www.amazontrust.com/repository/AmazonRootCA4.pem
curl -O https://certs.secureserver.net/repository/sf-class2-root.crt
```

Note

Amazon Keyspaces utilizzava in precedenza certificati TLS ancorati alla CA Starfield Class 2. AWS sta migrando tutte le Regioni AWS verso certificati emessi nell'ambito di Amazon Trust Services (Amazon Root CAs 1—4). Durante questa transizione, configura i client in modo che si fidino sia di Amazon Root CAs 1-4 che di Starfield root per garantire la compatibilità in tutte le regioni.

Combina tutti i certificati scaricati in un unico pem file con il nome indicato *keyspaces-bundle.pem* nei nostri esempi. Per aggiungerlo, eseguire il comando seguente: Prendi nota del percorso del file, ne avrai bisogno in seguito.

```
cat AmazonRootCA1.pem \
    AmazonRootCA2.pem \
    AmazonRootCA3.pem \
    AmazonRootCA4.pem \
    sf-class2-root.crt \
    > keyspaces-bundle.pem
```

Connect ad Amazon Keyspaces utilizzando il driver Gocql per Apache Cassandra e le credenziali specifiche del servizio

1. Crea una directory per l'applicazione.

```
mkdir ./gocqlexample
```

2. Accedere alla nuova directory.

```
cd gocqlexample
```

3. Crea un file per la tua applicazione.

```
touch cqlapp.go
```

4. Scarica il driver Go.

```
go get github.com/gocql/gocql
```

5. Aggiungi il seguente codice di esempio al file cqlapp.go.

```
package main

import (
    "fmt"
    "github.com/gocql/gocql"
    "log"
)

func main() {

    // add the Amazon Keyspaces service endpoint
    cluster := gocql.NewCluster("cassandra.us-east-2.amazonaws.com")
    cluster.Port=9142
    // add your service specific credentials
    cluster.Authenticator = gocql.PasswordAuthenticator{
        Username: "ServiceUserName",
        Password: "ServicePassword"}

    // provide the path to the keyspaces-bundle.pem
    cluster.SslOpts = &gocql.SslOptions{
        CaPath: "path_to_file/keyspaces-bundle.pem",
        EnableHostVerification: false,
    }

    // Override default Consistency to LocalQuorum
    cluster.Consistency = gocql.LocalQuorum
}
```

```
cluster.DisableInitialHostLookup = false

session, err := cluster.CreateSession()
if err != nil {
    fmt.Println("err>", err)
}
defer session.Close()

// run a sample query from the system keyspace
var text string
iter := session.Query("SELECT keyspace_name FROM system_schema.tables;").Iter()
for iter.Scan(&text) {
    fmt.Println("keyspace_name:", text)
}
if err := iter.Close(); err != nil {
    log.Fatal(err)
}
session.Close()
}
```

Note sull'utilizzo:

- a. Sostituisci "*path_to_file/keyspaces-bundle.pem*" con il percorso del file di certificato combinato salvato nel primo passaggio.
 - b. Assicurati che *ServiceUserName* e *ServicePassword* corrispondano al nome utente e alla password che hai ottenuto quando hai generato le credenziali specifiche del servizio seguendo i passaggi per. [Crea credenziali specifiche del servizio per l'accesso programmatico ad Amazon Keyspaces](#)
 - c. Per un elenco degli endpoint disponibili, consulta. [the section called "Endpoint di servizio"](#)
6. Costruisci il programma.

```
go build cqlapp.go
```

7. Esegui il programma.

```
./cqlapp
```

Connect ad Amazon Keyspaces utilizzando il driver Go per Apache Cassandra e il plug-in di autenticazione SigV4

Il seguente esempio di codice mostra come utilizzare il plug-in di autenticazione SigV4 per il driver open source Go per accedere ad Amazon Keyspaces (per Apache Cassandra).

Se non l'hai già fatto, crea le credenziali per il tuo principale IAM seguendo la procedura riportata qui. [the section called “Crea credenziali IAM per l'autenticazione AWS”](#) Se un'applicazione è in esecuzione su Lambda o un' EC2 istanza Amazon, utilizza automaticamente le credenziali dell'istanza. Per eseguire questo tutorial localmente, puoi memorizzare le credenziali come variabili di ambiente locali.

[Aggiungi il plug-in di autenticazione Go SigV4 all'applicazione dal repository. GitHub](#) Il plugin supporta la versione 1.2.x del driver open source Go per Cassandra e dipende dall'SDK for AWS Go.

```
$ go mod init
$ go get github.com/aws/aws-sigv4-auth-cassandra-gocql-driver-plugin
```

In questo esempio di codice, l'endpoint Amazon Keyspaces è rappresentato dalla classe. `Cluster` Utilizza la proprietà `AwsAuthenticator` for the authenticator del cluster per ottenere le credenziali.

```
package main

import (
    "fmt"
    "github.com/aws/aws-sigv4-auth-cassandra-gocql-driver-plugin/sigv4"
    "github.com/gocql/gocql"
    "log"
)

func main() {
    // configuring the cluster options
    cluster := gocql.NewCluster("cassandra.us-west-2.amazonaws.com")
    cluster.Port=9142

    // the authenticator uses the default credential chain to find AWS credentials
    cluster.Authenticator = sigv4.NewAwsAuthenticator()

    cluster.SslOpts = &gocql.SslOptions{

        CaPath: "path_to_file/keyspaces-bundle.pem",
```



```
        EnableHostVerification: false,
    }
    cluster.Consistency = gocql.LocalQuorum
    cluster.DisableInitialHostLookup = false

    session, err := cluster.CreateSession()
    if err != nil {
        fmt.Println("err>", err)
        return
    }
    defer session.Close()

    // doing the query
    var text string
    iter := session.Query("SELECT keyspace_name FROM system_schema.tables;").Iter()
    for iter.Scan(&text) {
        fmt.Println("keyspace_name:", text)
    }
    if err := iter.Close(); err != nil {
        log.Fatal(err)
    }
}
```

Note sull'utilizzo:

1. Sostituisci "*path_to_file/keyspaces-bundle.pem*" con il percorso del file di certificato salvato nel primo passaggio.
2. Affinché questo esempio venga eseguito localmente, è necessario definire le seguenti variabili come variabili di ambiente:
 - AWS_ACCESS_KEY_ID
 - AWS_SECRET_ACCESS_KEY
 - AWS_DEFAULT_REGION
3. Per memorizzare le chiavi di accesso al di fuori del codice, consulta le best practice all'indirizzo [the section called “Gestione delle chiavi di accesso”](#).
4. Per un elenco degli endpoint disponibili, consulta [the section called “Endpoint di servizio”](#).

Utilizzo di un driver client Cassandra Perl per accedere ad Amazon Keyspaces a livello di codice

Questa sezione mostra come connettersi ad Amazon Keyspaces utilizzando un driver client Perl. Per questo esempio di codice, abbiamo usato Perl 5. Amazon Keyspaces richiede l'uso di Transport Layer Security (TLS) per proteggere le connessioni con i client.

Important

I certificati Amazon Keyspaces stanno passando alla gerarchia di Amazon Trust Services (ATS). Assicurati che il tuo ambiente si fidi di Amazon Root CAs 1-4 per evitare errori di connessione durante questa rotazione. Il driver Perl non convalida il certificato Amazon SSL del server, il che significa che non puoi confermare che ti stai connettendo ad Amazon Keyspaces. Il secondo passaggio, configurare il driver per utilizzare TLS durante la connessione ad Amazon Keyspaces, è ancora necessario e garantisce che i dati trasferiti tra il client e il server siano crittografati.

1. Scarica il driver Cassandra DBI da <https://metacpan.org/pod/DBD::Cassandra> e installalo nel tuo ambiente Perl. I passaggi esatti dipendono dall'ambiente. Di seguito è riportato un esempio comune.

```
cpanm DBD::Cassandra
```

2. Crea un file per la tua applicazione.

```
touch cqlapp.pl
```

3. Aggiungi il seguente codice di esempio al file cqlapp.pl.

```
use DBI;
my $user = "ServiceUserName";
my $password = "ServicePassword";
my $db = DBI->connect("dbi:Cassandra:host=cassandra.us-east-2.amazonaws.com;port=9142;tls=1;",
    $user, $password);

my $rows = $db->selectall_arrayref("select * from system_schema.keyspaces");
print "Found the following Keyspaces...\n";
for my $row (@$rows) {
```

```
print join(" ",@ $row['keyspace_name']),"\n";  
}  
  
$db->disconnect;
```

Important

Assicurati che *ServiceUserName* e *ServicePassword* corrispondano al nome utente e alla password che hai ottenuto quando hai generato le credenziali specifiche del servizio seguendo i passaggi da. [Crea credenziali specifiche del servizio per l'accesso programmatico ad Amazon Keyspaces](#)

Note

Per un elenco degli endpoint disponibili, consulta. [the section called “Endpoint di servizio”](#)

4. Esegui l'applicazione.

```
perl cqlapp.pl
```

Configura l'accesso tra account ad Amazon Keyspaces con endpoint VPC

Puoi creare e utilizzare risorse separate Account AWS per isolare le risorse e utilizzarle in ambienti diversi, ad esempio sviluppo e produzione. Questo argomento illustra l'accesso tra più account per Amazon Keyspaces utilizzando gli endpoint VPC dell'interfaccia in un. Amazon Virtual Private Cloud Per ulteriori informazioni sulla configurazione dell'accesso tra account IAM, consulta [Scenario di esempio con account di sviluppo e produzione separati nella Guida per l'utente IAM](#).

Per ulteriori informazioni su Amazon Keyspaces e sugli endpoint VPC privati, consulta. [the section called “Utilizzo di endpoint VPC dell'interfaccia ”](#)

Argomenti

- [Configura l'accesso tra account ad Amazon Keyspaces utilizzando endpoint VPC in un VPC condiviso](#)

- [Configurazione dell'accesso tra account ad Amazon Keyspaces senza un VPC condiviso](#)

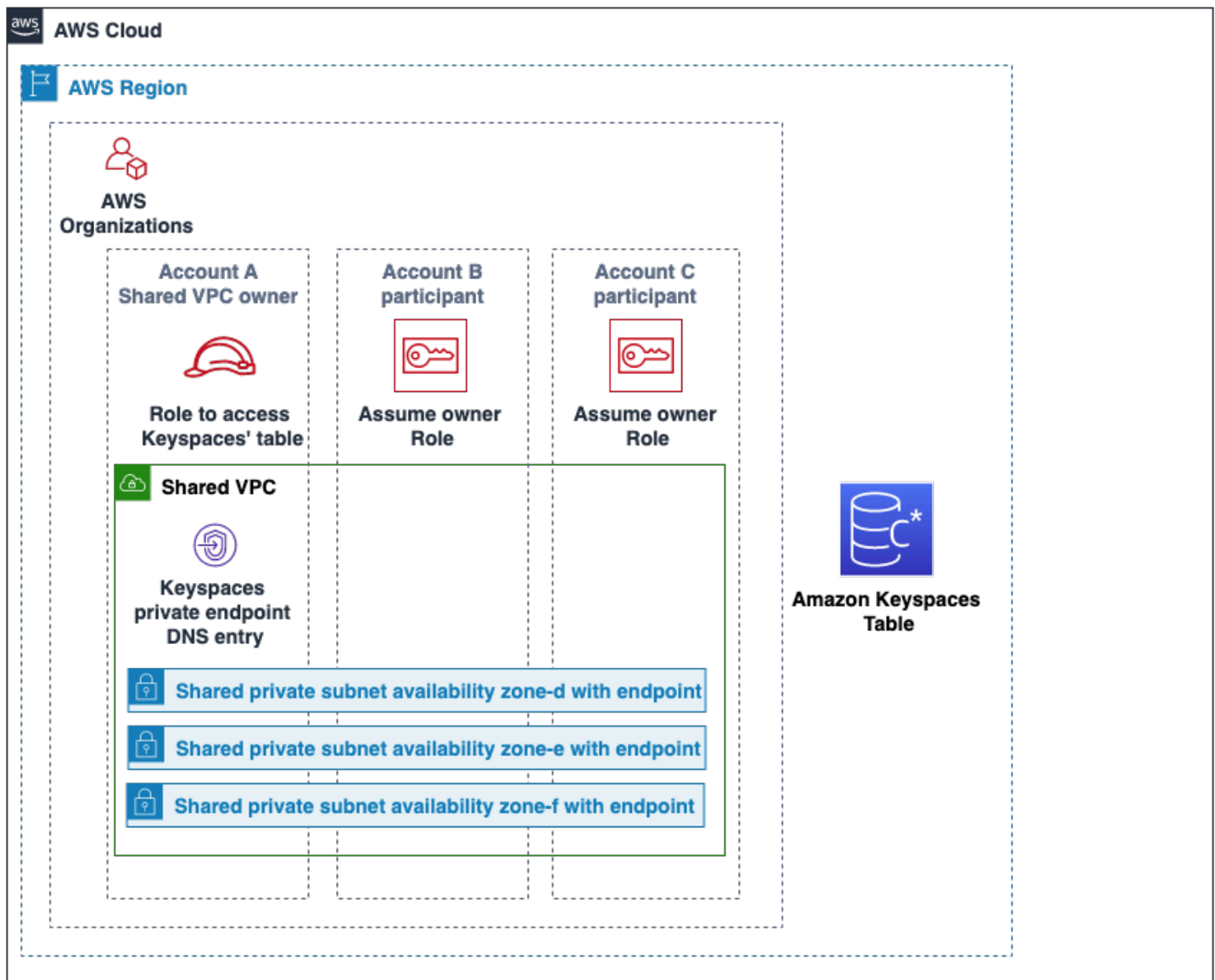
Configura l'accesso tra account ad Amazon Keyspaces utilizzando endpoint VPC in un VPC condiviso

È possibile creare risorse diverse Account AWS o separate dalle applicazioni. Ad esempio, puoi creare un account per le tabelle Amazon Keyspaces, un account diverso per le applicazioni in un ambiente di sviluppo e un altro account per le applicazioni in un ambiente di produzione. Questo argomento illustra i passaggi di configurazione necessari per configurare l'accesso tra account diversi per Amazon Keyspaces utilizzando gli endpoint VPC dell'interfaccia in un VPC condiviso.

Per i passaggi dettagliati su come configurare un endpoint VPC per Amazon Keyspaces, consulta [the section called “Fase 3: creare un endpoint VPC per Amazon Keyspaces”](#)

In questo esempio utilizziamo i seguenti tre account in un VPC condiviso:

- Account A:111111111111— Questo account contiene l'infrastruttura, inclusi gli endpoint VPC, le sottoreti VPC e le tabelle Amazon Keyspaces.
- Account B:222222222222— Questo account contiene un'applicazione in un ambiente di sviluppo che deve connettersi alla tabella Amazon Keyspaces in Account A:111111111111
- Account C:333333333333— Questo account contiene un'applicazione in un ambiente di produzione che deve connettersi alla tabella Amazon Keyspaces in Account A:111111111111



Account A:111111111111 è l'account che contiene le risorse (una tabella Amazon Keyspaces) a Account C:333333333333 cui dobbiamo accedere, così come Account A:111111111111 l'account trusting. Account B:222222222222 Account B:222222222222 e Account C:333333333333 sono gli account con i principali che devono accedere alle risorse (una tabella Amazon Keyspaces) Account A:111111111111 Account C:333333333333 e sono Account B:222222222222 quindi gli account affidabili. L'account trusting concede le autorizzazioni agli account fidati condividendo un ruolo IAM. La procedura seguente descrive i passaggi di configurazione richiesti in Account A:111111111111

Configurazione per **Account A:111111111111**

1. AWS Resource Access Manager Da utilizzare per creare una condivisione di risorse per la sottorete e condividere la sottorete privata con Account B:222222222222 e Account C:333333333333

Account B:222222222222e ora Account C:333333333333 possono vedere e creare risorse nella sottorete che è stata condivisa con loro.

2. Crea un endpoint VPC privato Amazon Keyspaces basato su AWS PrivateLink Questo crea più endpoint tra sottoreti condivise e voci DNS per l'endpoint del servizio Amazon Keyspaces.
3. Crea uno spazio di chiavi e una tabella Amazon Keyspaces.
4. Crea un ruolo IAM Account A:111111111111 che abbia accesso completo alla tabella Amazon Keyspaces, accesso in lettura alle tabelle di sistema Amazon Keyspaces e sia in grado di descrivere le risorse Amazon EC2 VPC come mostrato nel seguente esempio di policy.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CrossAccountAccess",
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeVpcEndpoints",
        "cassandra:*"
      ],
      "Resource": "*"
    }
  ]
}
```

5. Configura una policy di fiducia per il ruolo IAM in Account A:111111111111 modo che Account B:222222222222 e Account C:333333333333 possano assumere il ruolo di account affidabili. Questo viene mostrato nell'esempio seguente.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
```

```

    "Principal": {
      "AWS": [
        "arn:aws:iam::222222222222:role/Cross-Account-Role-B",
        "arn:aws:iam::333333333333:role/Cross-Account-Role-C"
      ]
    },
    "Action": "sts:AssumeRole",
    "Condition": {}
  }
]
}

```

Per ulteriori informazioni sulle politiche IAM tra account, consulta le politiche [tra account nella Guida](#) per l'utente IAM.

Configurazione in e **Account B:222222222222** **Account C:333333333333**

1. In Account B:222222222222 and Account C:333333333333, crea nuovi ruoli e allega la seguente politica che consente al principale di assumere il ruolo condiviso creato in Account A:111111111111.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}

```

Consentire al principale di assumere il ruolo condiviso viene implementato utilizzando l'AssumeRoleAPI di AWS Security Token Service (AWS STS). Per ulteriori informazioni, consulta [Fornire l'accesso a un utente IAM in un altro Account AWS di tua proprietà](#) nella Guida per l'utente IAM.

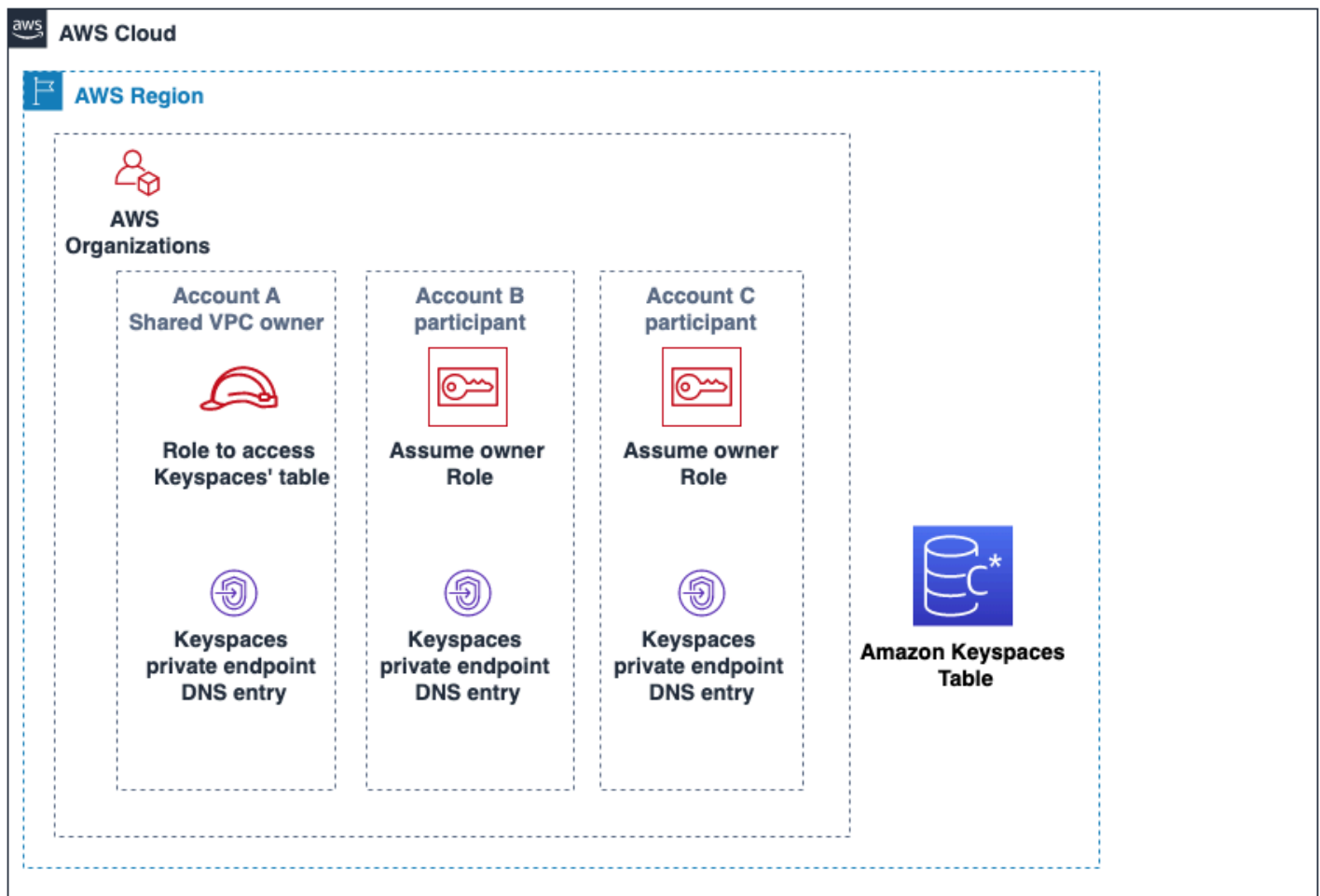
2. Inoltre Account B:222222222222 Account C:333333333333, puoi creare applicazioni che utilizzano il plug-in di SIGV4 autenticazione, che consente a un'applicazione di assumere

il ruolo condiviso per connettersi alla tabella Amazon Keyspaces situata tramite l'endpoint VPC Account A:111111111111 nel VPC condiviso. Per ulteriori informazioni sul plug-in di autenticazione, consulta [SIGV4 the section called “Crea credenziali di accesso programmatico”](#). Per ulteriori informazioni su come configurare un'applicazione in modo che assuma un ruolo in un altro AWS account, consulta [Autenticazione e accesso](#) nella Guida di riferimento agli strumenti AWS SDKs e agli strumenti.

Configurazione dell'accesso tra account ad Amazon Keyspaces senza un VPC condiviso

Se la tabella Amazon Keyspaces e l'endpoint VPC privato sono di proprietà di account diversi ma non condividono un VPC, le applicazioni possono comunque connettersi tra account diversi utilizzando endpoint VPC. Perché gli account non condividono gli endpoint VPC e Account C:333333333333 richiedono i Account A:111111111111 propri endpoint VPC. Account B:222222222222 Per il driver client Cassandra, Amazon Keyspaces appare come un singolo nodo anziché un cluster multinodo. Al momento della connessione, il driver client raggiunge il server DNS che restituisce uno degli endpoint disponibili nel VPC dell'account.

Puoi anche accedere alle tabelle Amazon Keyspaces da diversi account senza un endpoint VPC condiviso utilizzando gli endpoint pubblici o implementando un endpoint VPC privato in ogni account. Quando non si utilizza un VPC condiviso, ogni account richiede il proprio endpoint VPC. In questo esempio Account A:111111111111 Account B:222222222222, Account C:333333333333 richiedono i propri endpoint VPC per accedere alla tabella in Account A:111111111111. Quando si utilizzano gli endpoint VPC in questa configurazione, Amazon Keyspaces appare come un cluster a nodo singolo per il driver client Cassandra anziché un cluster multinodo. Al momento della connessione, il driver client raggiunge il server DNS che restituisce uno degli endpoint disponibili nel VPC dell'account. Ma il driver del client non è in grado di accedere alla `system.peers` tabella per scoprire endpoint aggiuntivi. Poiché ci sono meno host disponibili, il driver effettua meno connessioni. Per regolare questa impostazione, aumenta l'impostazione del pool di connessioni del driver di un fattore tre.



Account A:111111111111 è l'account che contiene le risorse (una tabella Amazon Keyspaces) a Account C:333333333333 cui dobbiamo accedere, così come Account A:111111111111 l'account trusting. Account B:222222222222 Account B:222222222222 e Account C:333333333333 sono gli account con i principali che devono accedere alle risorse (una tabella Amazon Keyspaces) Account A:111111111111 Account C:333333333333 e sono Account B:222222222222 quindi gli account affidabili. L'account trusting concede le autorizzazioni agli account fidati condividendo un ruolo IAM. La procedura seguente descrive i passaggi di configurazione richiesti in Account A:111111111111

Configurazione per **Account A:111111111111**

1. Crea uno spazio di chiavi Amazon Keyspaces e una tabella in Account A:111111111111
2. Crea un ruolo IAM con accesso completo alla tabella Amazon Keyspaces e accesso in lettura alle tabelle di sistema Amazon Keyspaces. Account A:111111111111

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cassandra:Select",
        "cassandra:Modify"
      ],
      "Resource": [
        "arn:aws:cassandra:us-east-1:111111111111:/keyspace/mykeyspace/table/mytable",
        "arn:aws:cassandra:us-east-1:111111111111:/keyspace/system*"
      ]
    }
  ]
}
```

3. Configura una policy di fiducia per il ruolo IAM in Account A:111111111111 modo che i responsabili abbiano Account B:222222222222 e Account C:333333333333 possano assumere il ruolo di account affidabili. Questo viene mostrato nell'esempio seguente.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::222222222222:role/Cross-Account-Role-B",
          "arn:aws:iam::333333333333:role/Cross-Account-Role-C"
        ]
      },
      "Action": "sts:AssumeRole",
      "Condition": {}
    }
  ]
}
```

Per ulteriori informazioni sulle policy IAM tra account, consulta le policy relative a più [account](#) nella IAM User Guide.

4. Configura l'endpoint VPC Account A:111111111111 e assegna all'endpoint le autorizzazioni che consentono ai ruoli da Account B:222222222222 e Account C:333333333333 di assumere il ruolo nell'utilizzo Account A dell'endpoint VPC. Queste autorizzazioni sono valide per l'endpoint VPC a cui sono collegate. Per ulteriori informazioni sulle policy degli endpoint VPC, consulta [the section called “Controllo dell'accesso agli endpoint VPC di interfaccia per Amazon Keyspaces”](#)

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowAccessfromSpecificIAMroles",
      "Effect": "Allow",
      "Action": "cassandra:*",
      "Resource": "*",
      "Principal": "*",
      "Condition": {
        "ArnEquals": {
          "aws:PrincipalArn": [
            "arn:aws:iam::222222222222:role/Cross-Account-Role-B",
            "arn:aws:iam::333333333333:role/Cross-Account-Role-C"
          ]
        }
      }
    }
  ]
}
```

Configurazione in e **Account B:222222222222** **Account C:333333333333**

1. In Account B:222222222222 and Account C:333333333333, crea nuovi ruoli e allega la seguente politica che consente al principale di assumere il ruolo condiviso creato in Account A:111111111111.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "sts:AssumeRole",
    "Resource": "arn:aws:iam::111111111111:role/keyspaces_access"
```

```
}  
}
```

Consentire al principale di assumere il ruolo condiviso viene implementato utilizzando l'AssumeRoleAPI di AWS Security Token Service (AWS STS). Per ulteriori informazioni, consulta [Fornire l'accesso a un utente IAM in un altro Account AWS di tua proprietà](#) nella Guida per l'utente IAM.

2. In Account B:222222222222 and Account C:333333333333, puoi creare applicazioni che utilizzano il plug-in di SIGV4 autenticazione, che consente a un'applicazione di assumere il ruolo condiviso per connettersi alla tabella Amazon Keyspaces situata in Account A:111111111111 Per ulteriori informazioni sul plug-in di SIGV4 autenticazione, consulta. [the section called “Crea credenziali di accesso programmatico”](#) Per ulteriori informazioni su come configurare un'applicazione in modo che assuma un ruolo in un altro AWS account, consulta [Autenticazione e accesso](#) nella Guida di riferimento agli strumenti AWS SDKs e agli strumenti.

Guida introduttiva ad Amazon Keyspaces (per Apache Cassandra)

Se non conosci Apache Cassandra e Amazon Keyspaces, questo tutorial ti guida nell'installazione dei programmi e degli strumenti necessari per utilizzare Amazon Keyspaces con successo. Imparerai come creare uno spazio chiave e una tabella utilizzando Cassandra Query Language (CQL), the o the (). Console di gestione AWS AWS Command Line Interface AWS CLI Utilizza quindi Cassandra Query Language (CQL) per eseguire operazioni di creazione, lettura, aggiornamento ed eliminazione (CRUD) sui dati nella tabella Amazon Keyspaces.

Questo tutorial illustra i seguenti passaggi.

- **Prerequisiti:** prima di iniziare il tutorial, segui le istruzioni di AWS configurazione per registrarti AWS e creare un utente IAM con accesso ad Amazon Keyspaces. Quindi configuri e. cqhsh-expansion AWS CloudShell In alternativa, puoi usare il AWS CLI per creare risorse in Amazon Keyspaces.
- **Passaggio 1:** Crea uno spazio di chiavi e una tabella: in questa sezione, creerai uno spazio di chiavi denominato «catalog» e al suo interno una tabella denominata «book_awards». Specificherai le colonne, i tipi di dati, la chiave di partizione e la colonna di clustering della tabella utilizzando CQL o. Console di gestione AWS AWS CLI
- **Passaggio 2:** Esegui le operazioni CRUD: qui utilizzerai l'cq1sh-expansionin CloudShell per inserire, leggere, aggiornare ed eliminare i dati nella tabella «book_awards». Imparerai come utilizzare varie istruzioni CQL come SELECT, INSERT, UPDATE e DELETE e come esercitarti a filtrare e modificare i dati.
- **Passaggio 3:** Pulisci le risorse: per evitare di incorrere in addebiti per le risorse non utilizzate, questa sezione ti guida nell'eliminazione della tabella «book_awards» e dello spazio chiave «catalog» utilizzando la console, CQL o il. AWS CLI

Per i tutorial sulla connessione programmatica ad Amazon Keyspaces utilizzando diversi driver client Apache Cassandra, consulta. [the section called “Utilizzo di un driver client Cassandra”](#) Per esempi di codice che utilizzano metodi diversi AWS SDKs, consulta [Esempi di codice per l'utilizzo di Amazon Keyspaces](#). AWS SDKs

Argomenti

- [Prerequisiti e considerazioni del tutorial](#)

- [Crea uno spazio chiave in Amazon Keyspaces](#)
- [Verifica lo stato di creazione del keyspace in Amazon Keyspaces](#)
- [Crea una tabella in Amazon Keyspaces](#)
- [Verifica lo stato di creazione della tabella in Amazon Keyspaces](#)
- [Crea, leggi, aggiorna ed elimina dati \(CRUD\) utilizzando CQL in Amazon Keyspaces](#)
- [Eliminare una tabella in Amazon Keyspaces](#)
- [Eliminare un keyspace in Amazon Keyspaces](#)

Prerequisiti e considerazioni del tutorial

Prima di iniziare a usare Amazon Keyspaces, segui le istruzioni di AWS configurazione riportate in [Accesso ad Amazon Keyspaces \(per Apache Cassandra\)](#). Questi passaggi includono la registrazione AWS e la creazione di un utente AWS Identity and Access Management (IAM) con accesso ad Amazon Keyspaces.

Per completare tutti i passaggi del tutorial, devi `cqlsh` installarlo. Puoi seguire le istruzioni di configurazione all'indirizzo [Utilizzo cqlsh per connettersi ad Amazon Keyspaces](#).

Per accedere ad Amazon Keyspaces utilizzando `cqlsh` o AWS CLI, consigliamo di utilizzare AWS CloudShell. CloudShell è una shell preautenticata basata su browser che puoi avviare direttamente da Console di gestione AWS. Puoi eseguire comandi AWS Command Line Interface (AWS CLI) su Amazon Keyspaces usando la tua shell preferita (Bash PowerShell o Z shell). Per utilizzarlo `cqlsh`, devi installare `cqlsh-expansion`. Per le istruzioni di `cqlsh-expansion` installazione, vedere [the section called “Utilizzo di cqlsh-expansion”](#). Per ulteriori informazioni su CloudShell see [the section called “Usando AWS CloudShell”](#).

Per utilizzare l'opzione AWS CLI per creare, visualizzare ed eliminare risorse in Amazon Keyspaces, segui le istruzioni di configurazione disponibili all'indirizzo. [the section called “Download e configurazione della AWS CLI”](#)

Dopo aver completato i passaggi preliminari, procedi a [Crea uno spazio chiave in Amazon Keyspaces](#)

Crea uno spazio chiave in Amazon Keyspaces

In questa sezione, crei uno spazio di chiavi utilizzando la `cqlsh` console o AWS CLI

 Note

Prima di iniziare, assicuratevi di aver configurato tutti i [prerequisiti del tutorial](#).

Un keyspace raggruppa le tabelle correlate che sono rilevanti per una o più applicazioni. Un keyspace contiene una o più tabelle e definisce la strategia di replica per tutte le tabelle che contiene. Per ulteriori informazioni sugli spazi chiave, consultate i seguenti argomenti:

- Istruzioni DDL (Data Definition Language) nel riferimento al linguaggio CQL: [Keyspaces](#)
- [Quote per Amazon Keyspaces \(per Apache Cassandra\)](#)

In questo tutorial creiamo uno spazio chiave a regione singola e la strategia di replica dello spazio chiave è. `SingleRegionStrategy` Utilizzando `SingleRegionStrategy`, Amazon Keyspaces replica i dati su tre [zone di disponibilità](#) in una. Regione AWS Per informazioni su come creare spazi chiave multiregionali, consulta. [the section called “Crea uno spazio chiave multiregionale”](#)

Utilizzo della console

Per creare uno spazio chiave utilizzando la console

1. [Accedi a e apri Console di gestione AWS la console Amazon Keyspaces a casahttps://console.aws.amazon.com/keyspaces/.](https://console.aws.amazon.com/keyspaces/)
2. Nel riquadro di navigazione, scegli Keyspaces.
3. Scegli Crea spazio chiave.
4. Nella casella Nome keyspace, immetti **catalog** come nome per il tuo keyspace.

Vincoli relativi al nome:

- Il nome non può essere vuoto.
 - Caratteri consentiti: caratteri alfanumerici e trattino basso (`.`). `_`
 - La lunghezza massima è di 48 caratteri.
5. Sotto Regioni AWS, conferma che la replica a regione singola è la strategia di replica per il keyspace.
 6. Per creare lo spazio chiave, scegli Crea spazio chiave.
 7. Verifica che lo spazio chiave sia `catalog` stato creato effettuando le seguenti operazioni:

- a. Nel riquadro di navigazione, scegli Keyspaces.
- b. Individua il tuo spazio chiave `catalog` nell'elenco degli spazi chiave.

Usare CQL

La procedura seguente crea uno spazio chiave utilizzando CQL.

Per creare uno spazio chiave utilizzando CQL

1. Apri AWS CloudShell e connettiti ad Amazon Keyspaces utilizzando il seguente comando. Assicurati di eseguire l'aggiornamento `us-east-1` con la tua regione.

```
cqlsh-expansion cassandra.us-east-1.amazonaws.com 9142 --ssl
```

L'output di quel comando dovrebbe essere simile a questo.

```
Connected to Amazon Keyspaces at cassandra.us-east-1.amazonaws.com:9142
[cqlsh 6.1.0 | Cassandra 3.11.2 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
cqlsh current consistency level is ONE.
```

2. Crea il tuo keyspace usando il seguente comando CQL.

```
CREATE KEYSPACE catalog WITH REPLICATION = {'class': 'SingleRegionStrategy'};
```

`SingleRegionStrategy` utilizza un fattore di replica pari a tre e replica i dati in tre zone di AWS disponibilità della propria regione.

Note

Amazon Keyspaces imposta per impostazione predefinita tutti gli input in lettere minuscole, a meno che non vengano racchiusi tra virgolette.

3. Verifica che il tuo keyspace sia stato creato.

```
SELECT * from system_schema.keyspaces;
```

L'output di questo comando dovrebbe essere simile a questo.


```
cqlsh> SELECT * from system_schema.keyspaces;
```

keyspace_name	durable_writes	replication
system_schema	True	{'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '3'}
system_schema_mcs	True	{'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '3'}
system	True	{'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '3'}
system_multiregion_info	True	{'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '3'}
catalog	True	{'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '3'}

(5 rows)

Usando il AWS CLI

La procedura seguente crea uno spazio chiave utilizzando AWS CLI

Per creare uno spazio chiave utilizzando il AWS CLI

1. Per confermare che l'ambiente è configurato, è possibile eseguire il seguente comando in CloudShell.

```
aws keyspaces help
```

2. Create il vostro keyspace utilizzando la seguente AWS CLI istruzione.

```
aws keyspaces create-keyspace --keyspace-name 'catalog'
```

3. Verifica che il tuo keyspace sia stato creato con la seguente dichiarazione AWS CLI

```
aws keyspaces get-keyspace --keyspace-name 'catalog'
```

L'output di questo comando dovrebbe essere simile a quello di questo esempio.

```
{
```

```

    "keyspaceName": "catalog",
    "resourceArn": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/catalog/",
    "replicationStrategy": "SINGLE_REGION"
}

```

Verifica lo stato di creazione del keyspace in Amazon Keyspaces

Amazon Keyspaces esegue operazioni DDL (Data Definition Language), come la creazione e l'eliminazione di keyspace, in modo asincrono.

Puoi monitorare lo stato di creazione di nuovi keyspace in Console di gestione AWS, che indica quando uno spazio chiave è in sospeso o attivo. È inoltre possibile monitorare lo stato di creazione di un nuovo keyspace a livello di codice utilizzando lo spazio chiave. `system_schema_mcs` Uno spazio chiave diventa visibile nella `system_schema_mcs` `keyspaces` tabella quando è pronto per l'uso.

Il modello di progettazione consigliato per verificare quando un nuovo keyspace è pronto per l'uso consiste nel polling della `system_schema_mcs` `keyspaces` tabella Amazon Keyspaces (`system_schema_mcs.*`). Per un elenco di istruzioni DDL per gli spazi chiave, consulta la sezione nella guida di riferimento al linguaggio CQL. [the section called “Keyspaces”](#)

La seguente query mostra se un keyspace è stato creato con successo.

```
SELECT * FROM system_schema_mcs.keyspaces WHERE keyspace_name = 'mykeyspace';
```

Per uno spazio chiave che è stato creato correttamente, l'output della query è simile al seguente.

```

keyspace_name | durable_writes | replication
-----+-----+-----
mykeyspace   | true           | {...} 1 item

```

Crea una tabella in Amazon Keyspaces

In questa sezione, crei una tabella utilizzando la console o `cqlsh` il. AWS CLI

Una tabella è il luogo in cui i dati vengono organizzati e archiviati. La chiave primaria della tabella determina il modo in cui i dati vengono partizionati nella tabella. La chiave primaria è composta da

una chiave di partizione richiesta e da una o più colonne di clustering opzionali. I valori combinati che compongono la chiave primaria devono essere univoci per tutti i dati della tabella. Per ulteriori informazioni sulle tabelle, consulta i seguenti argomenti:

- Progettazione delle chiavi di partizione: [the section called “Progettazione delle chiavi di partizione”](#)
- Lavorare con le tabelle: [the section called “Controlla lo stato di creazione della tabella”](#)
- Istruzioni DDL nel riferimento al linguaggio CQL: [Tabelle](#)
- Gestione delle risorse delle tabelle: [Gestione delle risorse serverless](#)
- Monitoraggio dell'utilizzo delle risorse della tabella: [the section called “Monitoraggio con CloudWatch”](#)
- [Quote per Amazon Keyspaces \(per Apache Cassandra\)](#)

Quando si crea una tabella, si specifica quanto segue:

- Nome della tabella.
- Il nome e il tipo di dati di ogni colonna della tabella.
- La chiave primaria per la tabella.
 - Chiave di partizione: obbligatoria
 - Colonne di raggruppamento: facoltativo

Utilizzare la procedura seguente per creare una tabella con le colonne, i tipi di dati, le chiavi di partizione e le colonne di clustering specificati.

Utilizzo della console

La procedura seguente crea la tabella `book_awards` con queste colonne e questi tipi di dati.

<code>year</code>	<code>int</code>
<code>award</code>	<code>text</code>
<code>rank</code>	<code>int</code>
<code>category</code>	<code>text</code>
<code>book_title</code>	<code>text</code>
<code>author</code>	<code>text</code>
<code>publisher</code>	<code>text</code>

Per creare una tabella utilizzando la console

1. [Accedi a e apri Console di gestione AWS la console Amazon Keyspaces a casa](https://console.aws.amazon.com/keyspaces/)<https://console.aws.amazon.com/keyspaces/>.
2. Nel riquadro di navigazione, scegli Keyspaces.
3. Scegli catalog come spazio chiave in cui vuoi creare questa tabella.
4. Scegliere Create table (Crea tabella).
5. Nella casella Nome tabella, inserisci **book_awards** un nome per la tua tabella.

Vincoli relativi al nome:

- Il nome non può essere vuoto.
 - Caratteri consentiti: caratteri alfanumerici e trattino basso (.). _
 - La lunghezza massima è di 48 caratteri.
6. Nella sezione Colonne, ripeti i passaggi seguenti per ogni colonna che desideri aggiungere a questa tabella.

Aggiungi le colonne e i tipi di dati seguenti.

year	int
award	text
rank	int
category	text
book_title	text
author	text
publisher	text

- a. Nome: inserisci un nome per la colonna.

Vincoli relativi al nome:

- Il nome non può essere vuoto.
 - Caratteri consentiti: caratteri alfanumerici e trattino basso (.). _
 - La lunghezza massima è di 48 caratteri.
- b. Tipo: nell'elenco dei tipi di dati, scegli il tipo di dati per questa colonna.
 - c. Per aggiungere un'altra colonna, scegli Aggiungi colonna.

7. Scegli `award` e `year` immetti le chiavi di partizione in Partition Key. È necessaria una chiave di partizione per ogni tabella. Una chiave di partizione può essere composta da una o più colonne.
8. Aggiungi `category` e `rank` come colonne di clustering. Le colonne di clustering sono opzionali e determinano l'ordinamento all'interno di ciascuna partizione.
 - a. Per aggiungere una colonna di clustering, scegli Aggiungi colonna di clustering.
 - b. Nell'elenco Colonna, scegli la categoria. Nell'elenco Ordine, scegliete ASC per ordinare in ordine crescente i valori di questa colonna. (Scegliete DESC per l'ordine decrescente).
 - c. Quindi seleziona Aggiungi colonna di raggruppamento e scegli classifica.
9. Nella sezione Impostazioni della tabella, scegli Impostazioni predefinite.
10. Scegliere Create table (Crea tabella).
11. Verifica che la tabella sia stata creata.
 - a. Nel pannello di navigazione, seleziona Tabelle.
 - b. Verifica che la tua tabella sia nell'elenco delle tabelle.
 - c. Scegli il nome del tuo tavolo.
 - d. Verifica che tutte le colonne e i tipi di dati siano corretti.

 Note

Le colonne potrebbero non essere elencate nello stesso ordine in cui le hai aggiunte alla tabella.

Usare CQL

Questa procedura crea una tabella con le seguenti colonne e tipi di dati utilizzando CQL. `award` e `year` sono chiavi di partizione con `category` e `rank` come colonne di clustering, insieme costituiscono la chiave primaria della tabella.

```
year          int
award         text
rank          int
category      text
book_title    text
author        text
publisher     text
```

Per creare una tabella utilizzando CQL

1. Apri AWS CloudShell e connettiti ad Amazon Keyspaces utilizzando il seguente comando. Assicurati di eseguire l'aggiornamento *us-east-1* con la tua regione.

```
cqlsh-expansion cassandra.us-east-1.amazonaws.com 9142 --ssl
```

L'output di quel comando dovrebbe essere simile a questo.

```
Connected to Amazon Keyspaces at cassandra.us-east-1.amazonaws.com:9142
[cqlsh 6.1.0 | Cassandra 3.11.2 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
cqlsh current consistency level is ONE.
```

2. Al prompt del keyspace (`cqlsh:keyspace_name>`), create la tabella inserendo il seguente codice nella finestra di comando.

```
CREATE TABLE catalog.book_awards (
  year int,
  award text,
  rank int,
  category text,
  book_title text,
  author text,
  publisher text,
  PRIMARY KEY ((year, award), category, rank)
);
```

Note

ASC è l'ordine di clustering predefinito. È inoltre possibile specificare DESC l'ordine decrescente.

Nota che `year` e `award` sono colonne chiave di partizione. Quindi, `category` e `rank` sono le colonne di raggruppamento ordinate in ordine crescente (). ASC. Insieme, queste colonne costituiscono la chiave principale della tabella.

3. Verifica che la tabella sia stata creata.

```
SELECT * from system_schema.tables WHERE keyspace_name='catalog.book_awards' ;
```

L'output dovrebbe essere simile a questo.

```
keyspace_name | table_name | bloom_filter_fp_chance | caching | cdc | comment
| compaction | compression | crc_check_chance | dclocal_read_repair_chance
| default_time_to_live | extensions | flags | gc_grace_seconds | id |
max_index_interval | memtable_flush_period_in_ms | min_index_interval |
read_repair_chance | speculative_retry
-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+

(0 rows)
```

4. Verifica la struttura della tabella.

```
SELECT * FROM system_schema.columns WHERE keyspace_name = 'catalog' AND table_name
= 'book_awards';
```

L'output di questa istruzione dovrebbe essere simile a quello di questo esempio.

```
keyspace_name | table_name | column_name | clustering_order | column_name_bytes
| kind      | position | type
-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+
catalog | book_awards | year | none |
0x79656172 | partition_key | 0 | int
catalog | book_awards | award | none |
0x6177617264 | partition_key | 1 | text
catalog | book_awards | category | asc |
0x63617465676f7279 | clustering | 0 | text
catalog | book_awards | rank | asc |
0x72616e6b | clustering | 1 | int
catalog | book_awards | author | none |
0x617574686672 | regular | -1 | text
catalog | book_awards | book_title | none |
0x626f666b5f7469746c65 | regular | -1 | text
```

```

      catalog | book_awards | publisher |      none |
0x7075626c6973686572 |      regular |      -1 | text
(7 rows)

```

Verifica che tutte le colonne e i tipi di dati siano quelli previsti. L'ordine delle colonne potrebbe essere diverso da quello dell'CREATEistruzione.

Usando il AWS CLI

Questa procedura crea una tabella con le colonne e i tipi di dati seguenti utilizzando AWS CLI. Le award colonne year and costituiscono la chiave di partizione con category e rank come colonne di clustering.

```

year          int
award         text
rank          int
category      text
book_title    text
author        text
publisher     text

```

Per creare una tabella utilizzando il AWS CLI

Il comando seguente crea una tabella con il nome book_awards. La chiave di partizione della tabella è costituita dalle colonne year award e la chiave di clustering è costituita dalle colonne category ed rank entrambe le colonne di clustering utilizzano l'ordinamento crescente. (Per facilitare la leggibilità, il schema-definition comando table create in questa sezione è suddiviso in righe separate.)

1. È possibile creare la tabella utilizzando la seguente istruzione.

```

aws keyspaces create-table --keyspace-name 'catalog' \
                          --table-name 'book_awards' \
                          --schema-definition 'allColumns=[{name=year,type=int},
{name=award,type=text},{name=rank,type=int},
                          {name=category,type=text}, {name=author,type=text},
{name=book_title,type=text},{name=publisher,type=text}],
                          partitionKeys=[{name=year},
{name=award}],clusteringKeys=[{name=category,orderBy=ASC},{name=rank,orderBy=ASC}]'

```


Questo comando produce l'output seguente.

```
{
  "resourceArn": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/catalog/
table/book_awards"
}
```

2. Per confermare i metadati e le proprietà della tabella, è possibile utilizzare il seguente comando.

```
aws keyspaces get-table --keyspace-name 'catalog' --table-name 'book_awards'
```

Questo comando restituisce il seguente output.

```
{
  "keyspaceName": "catalog",
  "tableName": "book_awards",
  "resourceArn": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/catalog/
table/book_awards",
  "creationTimestamp": "2024-07-11T15:12:55.571000+00:00",
  "status": "ACTIVE",
  "schemaDefinition": {
    "allColumns": [
      {
        "name": "year",
        "type": "int"
      },
      {
        "name": "award",
        "type": "text"
      },
      {
        "name": "category",
        "type": "text"
      },
      {
        "name": "rank",
        "type": "int"
      },
      {
        "name": "author",
        "type": "text"
      }
    ]
  }
}
```

```
        {
            "name": "book_title",
            "type": "text"
        },
        {
            "name": "publisher",
            "type": "text"
        }
    ],
    "partitionKeys": [
        {
            "name": "year"
        },
        {
            "name": "award"
        }
    ],
    "clusteringKeys": [
        {
            "name": "category",
            "orderBy": "ASC"
        },
        {
            "name": "rank",
            "orderBy": "ASC"
        }
    ],
    "staticColumns": [],
},
"capacitySpecification": {
    "throughputMode": "PAY_PER_REQUEST",
    "lastUpdateToPayPerRequestTimestamp": "2024-07-11T15:12:55.571000+00:00"
},
"encryptionSpecification": {
    "type": "AWS_OWNED_KMS_KEY"
},
"pointInTimeRecovery": {
    "status": "DISABLED"
},
"defaultTimeToLive": 0,
"comment": {
    "message": ""
},
"replicaSpecifications": []
```

}

Per eseguire operazioni CRUD (creazione, lettura, aggiornamento ed eliminazione) sui dati della tabella, procedi a [the section called “Operazioni CRUD”](#)

Verifica lo stato di creazione della tabella in Amazon Keyspaces

Amazon Keyspaces esegue operazioni DDL (Data Definition Language), come la creazione e l'eliminazione di tabelle, in modo asincrono. Puoi monitorare lo stato di creazione di nuove tabelle in Console di gestione AWS, che indica quando una tabella è in sospeso o attiva. È inoltre possibile monitorare lo stato di creazione di una nuova tabella a livello di codice utilizzando la tabella dello schema di sistema.

Una tabella mostra come attiva nello schema di sistema quando è pronta per l'uso. Il modello di progettazione consigliato per verificare quando una nuova tabella è pronta per l'uso consiste nel polling delle tabelle `system_schema_mcs.*` dello schema del sistema Amazon Keyspaces (`system_schema_mcs`). Per un elenco di istruzioni DDL per le tabelle, consulta la [the section called “Tabelle”](#) sezione nella guida di riferimento al linguaggio CQL.

La seguente query mostra lo stato di una tabella.

```
SELECT keyspace_name, table_name, status FROM system_schema_mcs.tables WHERE
keyspace_name = 'mykeyspace' AND table_name = 'mytable';
```

Per una tabella ancora in fase di creazione e in sospeso, l'output della query è simile al seguente.

```
keyspace_name | table_name | status
-----+-----+-----
mykeyspace   | mytable   | CREATING
```

Per una tabella che è stata creata correttamente ed è attiva, l'output della query è simile al seguente.

```
keyspace_name | table_name | status
-----+-----+-----
mykeyspace   | mytable   | ACTIVE
```

Crea, leggi, aggiorna ed elimina dati (CRUD) utilizzando CQL in Amazon Keyspaces

In questa fase del tutorial, imparerai come inserire, leggere, aggiornare ed eliminare dati in una tabella Amazon Keyspaces utilizzando istruzioni CQL Data Manipulation Language (DML). In Amazon Keyspaces, puoi creare solo istruzioni DML in linguaggio CQL. In questo tutorial, ti eserciterai a eseguire istruzioni DML usando `with in thecqlsh-expansion`. [AWS CloudShell](#) Console di gestione AWS

- **Inserimento di dati:** questa sezione tratta l'inserimento di record singoli o multipli in una tabella utilizzando l'istruzione. `INSERT` Imparerai come caricare dati da un file CSV e verificare gli inserimenti corretti utilizzando le query. `SELECT`
- **Lettura dei dati:** qui esaminerai diverse varianti dell'`SELECT`istruzione per recuperare i dati da una tabella. Gli argomenti includono la selezione di tutti i dati, la selezione di colonne specifiche, il filtraggio delle righe in base alle condizioni utilizzando la `WHERE` clausola e la comprensione di condizioni semplici e composte.
- **Aggiornamento dei dati:** in questa sezione imparerai come modificare i dati esistenti in una tabella utilizzando l'`UPDATE`istruzione. Ti eserciterai ad aggiornare colonne singole e multiple comprendendo al contempo le restrizioni relative all'aggiornamento delle colonne chiave primarie.
- **Eliminazione dei dati:** l'ultima sezione riguarda l'eliminazione dei dati da una tabella utilizzando l'`DELETE`istruzione. Imparerai come eliminare celle specifiche, intere righe e le implicazioni dell'eliminazione dei dati rispetto all'eliminazione dell'intera tabella o dello spazio chiave.

Nel corso del tutorial, troverai esempi, suggerimenti e opportunità per esercitarti a scrivere le tue query CQL per vari scenari.

Argomenti

- [Inserimento e caricamento di dati in una tabella Amazon Keyspaces](#)
- [Leggi i dati da una tabella utilizzando l'`SELECT`istruzione CQL in Amazon Keyspaces](#)
- [Aggiorna i dati in una tabella Amazon Keyspaces utilizzando CQL](#)
- [Eliminare i dati da una tabella utilizzando l'istruzione CQL `DELETE`](#)

Inserimento e caricamento di dati in una tabella Amazon Keyspaces

Per creare dati nella `book_awards` tabella, usa l'`INSERT`istruzione per aggiungere una singola riga.

1. Apri AWS CloudShell e connettiti ad Amazon Keyspaces utilizzando il seguente comando. Assicurati di eseguire l'aggiornamento `us-east-1` con la tua regione.

```
cqlsh-expansion cassandra.us-east-1.amazonaws.com 9142 --ssl
```

L'output di quel comando dovrebbe essere simile a questo.

```
Connected to Amazon Keyspaces at cassandra.us-east-1.amazonaws.com:9142
[cqlsh 6.1.0 | Cassandra 3.11.2 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
cqlsh current consistency level is ONE.
```

2. Prima di poter scrivere dati sulla tabella Amazon Keyspaces utilizzando cqlsh, devi impostare la consistenza di scrittura per la sessione cqlsh corrente su. LOCAL_QUORUM Per ulteriori informazioni sui livelli di coerenza supportati, consulta. [the section called “Livelli di coerenza di scrittura”](#) Si noti che questo passaggio non è necessario se si utilizza l'editor CQL in. Console di gestione AWS

```
CONSISTENCY LOCAL_QUORUM;
```

3. Per inserire un singolo record, eseguite il seguente comando nell'editor CQL.

```
INSERT INTO catalog.book_awards (award, year, category, rank, author, book_title,
publisher)
VALUES ('Wolf', 2023, 'Fiction',3,'Shirley Rodriguez','Mountain', 'AnyPublisher') ;
```

4. Verifica che i dati siano stati aggiunti correttamente alla tabella eseguendo il comando seguente.

```
SELECT * FROM catalog.book_awards;
```

L'output della dichiarazione dovrebbe essere simile al seguente.

```
year | award | category | rank | author           | book_title | publisher
-----+-----+-----+-----+-----+-----+-----
2023 |  Wolf |  Fiction |    3 | Shirley Rodriguez |  Mountain | AnyPublisher

(1 rows)
```

Per inserire più record da un file usando cqlsh

1. [Scaricate il file CSV di esempio \(keyspaces_sample_table.csv\) contenuto nel file di archivio samplemigration.zip](#). Decomprimi l'archivio e prendi nota del percorso verso. `keyspaces_sample_table.csv`

award	year	category	rank	author	book_title	publisher
Kwesi Manu Prize	2020	Fiction	1	Akua Mansa	Where did you go?	SomePublisher
Kwesi Manu Prize	2020	Fiction	2	John Stiles	Yesterday	Example Books
Kwesi Manu Prize	2020	Fiction	3	Nikki Wolf	Moving to the Chateau	AnyPublisher
Wolf	2020	Non-Fiction	1	Wang Xiulan	History of Ideas	Example Books
Wolf	2020	Non-Fiction	2	Ana Carolina Silva	Science Today	SomePublisher
Wolf	2020	Non-Fiction	3	Shirley Rodriguez	The Future of Sea Ice	AnyPublisher
Richard Roe	2020	Fiction	1	Alejandro Rosalez	Long Summer	SomePublisher
Richard Roe	2020	Fiction	2	Arnav Desai	The Key	Example Books
Richard Roe	2020	Fiction	3	Mateo Jackson	Inside the Whale	AnyPublisher

2. Apri AWS CloudShell Console di gestione AWS e connettiti ad Amazon Keyspaces utilizzando il seguente comando. Assicurati di eseguire l'aggiornamento `us-east-1` con la tua regione.

```
cqlsh-expansion cassandra.us-east-1.amazonaws.com 9142 --ssl
```

3. Al cqlsh prompt (`cqlsh>`), specificate uno spazio chiave.

```
USE catalog ;
```

4. Imposta la coerenza di scrittura su. `LOCAL_QUORUM` Per ulteriori informazioni sui livelli di coerenza supportati, consulta [the section called “Livelli di coerenza di scrittura”](#).

```
CONSISTENCY LOCAL_QUORUM;
```

5. Nella sezione AWS CloudShell Azioni in alto a destra dello schermo, quindi scegli Carica file per caricare il file csv scaricato in precedenza. Prendi nota del percorso del file.
6. Al prompt del keyspace (`cqlsh:catalog>`), eseguite la seguente istruzione.

```
COPY book_awards (award, year, category, rank, author, book_title, publisher) FROM
'/home/cloudshell-user/keyspaces_sample_table.csv' WITH header=TRUE ;
```

L'output dell'istruzione dovrebbe essere simile a questo.

```
cqlsh:catalog> COPY book_awards (award, year, category, rank, author,
book_title, publisher) FROM '/home/cloudshell-user/
keyspaces_sample_table.csv' WITH delimiter=',' AND header=TRUE ;
```

```

cqlsh current consistency level is LOCAL_QUORUM.
Reading options from /home/cloudshell-user/.cassandra/cqlshrc:[copy]:
{'numprocesses': '16', 'maxattempts': '1000'}
Reading options from /home/cloudshell-user/.cassandra/cqlshrc:[copy-from]:
{'ingestrate': '1500', 'maxparseerrors': '1000', 'maxinserterrors': '-1',
 'maxbatchsize': '10', 'minbatchsize': '1', 'chunksize': '30'}
Reading options from the command line: {'delimiter': ',', 'header': 'TRUE'}
Using 16 child processes

Starting copy of catalog.book_awards with columns [award, year, category, rank,
author, book_title, publisher].
OSError: handle is closed      0 rows/s; Avg. rate:      0 rows/s
Processed: 9 rows; Rate:      0 rows/s; Avg. rate:      0 rows/s
9 rows imported from 1 files in 0 day, 0 hour, 0 minute, and 26.706 seconds (0
skipped).

```

7. Verifica che i dati siano stati aggiunti correttamente alla tabella eseguendo la seguente query.

```
SELECT * FROM book_awards ;
```

Vedrai il seguente output.

year	award publisher	category	rank	author	book_title
2020	Wolf Example Books	Non-Fiction	1	Wang Xiulan	History of Ideas
2020	Wolf SomePublisher	Non-Fiction	2	Ana Carolina Silva	Science Today
2020	Wolf AnyPublisher	Non-Fiction	3	Shirley Rodriguez	The Future of Sea Ice
2020	Kwesi Manu Prize SomePublisher	Fiction	1	Akua Mansa	Where did you go?
2020	Kwesi Manu Prize Example Books	Fiction	2	John Stiles	Yesterday
2020	Kwesi Manu Prize AnyPublisher	Fiction	3	Nikki Wolf	Moving to the Chateau
2020	Richard Roe SomePublisher	Fiction	1	Alejandro Rosalez	Long Summer
2020	Richard Roe Example Books	Fiction	2	Arnav Desai	The Key

2020	Richard Roe	Fiction	3	Mateo Jackson	Inside
the Whale	AnyPublisher				

(9 rows)

Per ulteriori informazioni sull'utilizzo `cqlsh COPY` per caricare dati da file csv in una tabella Amazon Keyspaces, consulta. [the section called “Caricamento dei dati utilizzando cqlsh”](#)

Leggi i dati da una tabella utilizzando l'**SELECT**istruzione CQL in Amazon Keyspaces

Nella [Inserimento e caricamento di dati in una tabella Amazon Keyspaces](#) sezione, hai utilizzato l'**SELECT**istruzione per verificare di aver aggiunto correttamente i dati alla tabella. In questa sezione, si perfeziona l'uso **SELECT** per visualizzare colonne specifiche e solo le righe che soddisfano criteri specifici.

La forma generale della **SELECT** dichiarazione è la seguente.

```
SELECT column_list FROM table_name [WHERE condition [ALLOW FILTERING]] ;
```

Argomenti

- [Seleziona tutti i dati nella tua tabella](#)
- [Seleziona un sottoinsieme di colonne](#)
- [Seleziona un sottoinsieme di righe](#)

Seleziona tutti i dati nella tua tabella

La forma più semplice dell'**SELECT**istruzione restituisce tutti i dati della tabella.

Important

In un ambiente di produzione, in genere non è consigliabile eseguire questo comando, poiché restituisce tutti i dati della tabella.

Per selezionare tutti i dati della tabella

1. Apri AWS CloudShell e connettiti ad Amazon Keyspaces utilizzando il seguente comando. Assicurati di eseguire l'aggiornamento *us-east-1* con la tua regione.

```
cqlsh-expansion cassandra.us-east-1.amazonaws.com 9142 --ssl
```

2. Esegui la seguente query.

```
SELECT * FROM catalog.book_awards ;
```

Usa il carattere jolly (*) per `column_list` selezionare tutte le colonne. L'output dell'istruzione è simile all'esempio seguente.

year	award publisher	category	rank	author	book_title
2020	Wolf	Non-Fiction	1	Wang Xiulan	History of Ideas
2020	Wolf	Non-Fiction	2	Ana Carolina Silva	Science Today
2020	Wolf	Non-Fiction	3	Shirley Rodriguez	The Future of Sea Ice
2020	Kwesi Manu Prize	Fiction	1	Akua Mansa	Where did you go?
2020	Kwesi Manu Prize	Fiction	2	John Stiles	Yesterday
2020	Kwesi Manu Prize	Fiction	3	Nikki Wolf	Moving to the Chateau
2020	Richard Roe	Fiction	1	Alejandro Rosalez	Long Summer
2020	Richard Roe	Fiction	2	Arnav Desai	The Key
2020	Richard Roe	Fiction	3	Mateo Jackson	Inside the Whale

Seleziona un sottoinsieme di colonne

Per eseguire una query per un sottoinsieme di colonne

1. Apri AWS CloudShell e connettiti ad Amazon Keyspaces utilizzando il seguente comando. Assicurati di eseguire l'aggiornamento *us-east-1* con la tua regione.

```
cqlsh-expansion cassandra.us-east-1.amazonaws.com 9142 --ssl
```

2. Per recuperare solo le year colonneaward, ecategory, esegui la seguente query.

```
SELECT award, category, year FROM catalog.book_awards ;
```

L'output contiene solo le colonne specificate nell'ordine elencato nell'SELECTistruzione.

award	category	year
Wolf	Non-Fiction	2020
Wolf	Non-Fiction	2020
Wolf	Non-Fiction	2020
Kwesi Manu Prize	Fiction	2020
Kwesi Manu Prize	Fiction	2020
Kwesi Manu Prize	Fiction	2020
Richard Roe	Fiction	2020
Richard Roe	Fiction	2020
Richard Roe	Fiction	2020

Seleziona un sottoinsieme di righe

Quando si esegue una query su un set di dati di grandi dimensioni, è possibile che si desiderino solo record che soddisfino determinati criteri. A tale scopo, puoi aggiungere una WHERE clausola alla fine della nostra dichiarazione. SELECT

Per eseguire una ricerca su un sottoinsieme di righe

1. Apri AWS CloudShell e connettiti ad Amazon Keyspaces utilizzando il seguente comando. Assicurati di eseguire l'aggiornamento *us-east-1* con la tua regione.

```
cqlsh-expansion cassandra.us-east-1.amazonaws.com 9142 --ssl
```

2. Per recuperare solo i record relativi ai premi di un determinato anno, esegui la seguente query.

```
SELECT * FROM catalog.book_awards WHERE year=2020 AND award='Wolf' ;
```

L'SELECTistruzione precedente restituisce l'output seguente.

```

year | award | category      | rank | author                | book_title          |
publisher
-----+-----+-----+-----+-----+-----+-----
+-----+
2020 | Wolf | Non-Fiction | 1 | Wang Xiulan | History of Ideas |
AnyPublisher
2020 | Wolf | Non-Fiction | 2 | Ana Carolina Silva | Science Today |
SomePublisher
2020 | Wolf | Non-Fiction | 3 | Shirley Rodriguez | The Future of Sea Ice |
AnyPublisher

```

Comprensione della clausola **WHERE**

La **WHERE** clausola viene utilizzata per filtrare i dati e restituire solo i dati che soddisfano i criteri specificati. I criteri specificati possono essere una condizione semplice o una condizione composta.

Come utilizzare le condizioni in una **WHERE** clausola

- Una condizione semplice: una singola colonna.

```
WHERE column_name=value
```

È possibile utilizzare una condizione semplice in una **WHERE** clausola se viene soddisfatta una delle seguenti condizioni:

- La colonna è l'unica colonna chiave di partizione della tabella.
- Si aggiunge **ALLOW FILTERING** dopo la condizione nella **WHERE** clausola.

Tieni presente che l'utilizzo **ALLOW FILTERING** può comportare prestazioni incoerenti, specialmente con tabelle di grandi dimensioni e multipartizionate.

- Una condizione composta: più condizioni semplici collegate da **AND**

```
WHERE column_name1=value1 AND column_name2=value2 AND column_name3=value3...
```

È possibile utilizzare condizioni composte in una WHERE clausola se viene soddisfatta una delle seguenti condizioni:

- Le colonne che è possibile utilizzare nella WHERE clausola devono includere tutte o un sottoinsieme delle colonne nella chiave di partizione della tabella. Se si desidera utilizzare solo un sottoinsieme delle colonne della WHERE clausola, è necessario includere un insieme contiguo di colonne chiave di partizione da sinistra a destra, a partire dalla colonna principale della chiave di partizione. Ad esempio, se le colonne della chiave di partizione sono yearmonth, e award allora è possibile utilizzare le seguenti colonne nella clausola: WHERE
 - year
 - yearE month
 - yearE month E award
- Si aggiunge ALLOW FILTERING dopo la condizione composta nella WHERE clausola, come nell'esempio seguente.

```
SELECT * FROM my_table WHERE col1=5 AND col2='Bob' ALLOW FILTERING ;
```

Tieni presente che l'utilizzo ALLOW FILTERING può comportare prestazioni incoerenti, specialmente con tabelle di grandi dimensioni e multipartizionate.

Provalo

Crea le tue query CQL per trovare quanto segue dalla tua book_awards tabella:

- Trova i vincitori dei premi Wolf 2020 e mostra i titoli e gli autori dei libri, ordinati per classifica.
- Mostra i vincitori del primo premio di tutti i premi del 2020 e mostra i titoli dei libri e i nomi dei premi.

Aggiorna i dati in una tabella Amazon Keyspaces utilizzando CQL

Per aggiornare i dati nella tua book_awards tabella, usa l'UPDATEistruzione.

La forma generale della UPDATE dichiarazione è la seguente.

```
UPDATE table_name SET column_name=new_value WHERE primary_key=value ;
```

Tip

- È possibile aggiornare più colonne utilizzando un elenco di valori `column_names` e separati da virgole, come nell'esempio seguente.

```
UPDATE my_table SET col1='new_value_1', col2='new_value2' WHERE col3='1' ;
```

- Se la chiave primaria è composta da più colonne, tutte le colonne della chiave primaria e i relativi valori devono essere inclusi nella clausola `WHERE`
- Non è possibile aggiornare alcuna colonna nella chiave primaria perché ciò modificherebbe la chiave primaria del record.

Per aggiornare una singola cella

Usando la tua `book_awards` tabella, cambia il nome di un editore che ha vinto i Wolf Awards di saggistica nel 2020.

```
UPDATE book_awards SET publisher='new Books' WHERE year = 2020 AND award='Wolf' AND category='Non-Fiction' AND rank=1;
```

Verifica che l'editore lo sia adesso `new Books`.

```
SELECT * FROM book_awards WHERE year = 2020 AND award='Wolf' AND category='Non-Fiction' AND rank=1;
```

L'istruzione dovrebbe restituire il seguente risultato.

year	award	category	rank	author	book_title	publisher
2020	Wolf	Non-Fiction	1	Wang Xiulan	History of Ideas	new Books

Provalo

Avanzato: il vincitore della fiction «Kwezi Manu Prize» del 2020 ha cambiato nome. Aggiorna questo record per cambiare il nome in. 'Akua Mansa-House '

Eliminare i dati da una tabella utilizzando l'istruzione CQL **DELETE**

Per eliminare i dati nella `book_awards` tabella, usa l'`DELETE`istruzione.

È possibile eliminare i dati da una riga o da una partizione. Fai attenzione quando elimini i dati, poiché le eliminazioni sono irreversibili.

L'eliminazione di una o tutte le righe da una tabella non elimina la tabella. In questo modo puoi ripopolarla con i dati. L'eliminazione di una tabella comporta l'eliminazione della tabella e di tutti i dati in essa contenuti. Per utilizzare nuovamente la tabella, è necessario ricrearla e aggiungervi dati. L'eliminazione di uno spazio chiave elimina lo spazio chiave e tutte le tabelle al suo interno. Per utilizzare lo spazio chiave e le tabelle, è necessario ricrearli e quindi popolarli con i dati. Puoi utilizzare il ripristino di Amazon Keyspaces Point-in-time (PITR) per ripristinare le tabelle eliminate; per ulteriori informazioni, consulta [the section called “Backup e ripristino con point-in-time ripristino”](#). Per informazioni su come ripristinare una tabella eliminata con PITR abilitato, consulta [the section called “Ripristina una tabella eliminata”](#).

Eliminare le celle

L'eliminazione di una colonna da una riga rimuove i dati dalla cella specificata. Quando si visualizza la colonna utilizzando un'`SELECT`istruzione, i dati vengono visualizzati come `null`, sebbene in quella posizione non sia memorizzato un valore nullo.

La sintassi generale per eliminare una o più colonne specifiche è la seguente.

```
DELETE column_name1[, column_name2...] FROM table_name WHERE condition ;
```

Nella tua `book_awards` tabella, puoi vedere che il titolo del libro che ha vinto il primo premio del premio «Richard Roe» 2020 è «Long Summer». Immagina che questo titolo sia stato richiamato e che tu debba eliminare i dati da questa cella.

Per eliminare una cella specifica

1. Apri AWS CloudShell e connettiti ad Amazon Keyspaces utilizzando il seguente comando. Assicurati di eseguire l'aggiornamento `us-east-1` con la tua regione.

```
cqlsh-expansion cassandra.us-east-1.amazonaws.com 9142 --ssl
```

2. Esegui la seguente `DELETE` interrogazione.

```
DELETE book_title FROM catalog.book_awards WHERE year=2020 AND award='Richard Roe'
AND category='Fiction' AND rank=1;
```

3. Verifica che la richiesta di eliminazione sia stata effettuata come previsto.

```
SELECT * FROM catalog.book_awards WHERE year=2020 AND award='Richard Roe' AND
category='Fiction' AND rank=1;
```

L'output di questa dichiarazione è simile al seguente.

year	award	category	rank	author	book_title	publisher
2020	Richard Roe	Fiction	1	Alejandro Rosalez	null	SomePublisher

Eliminazione di righe

Potrebbe esserci un momento in cui è necessario eliminare un'intera riga, ad esempio per soddisfare una richiesta di cancellazione dei dati. La sintassi generale per l'eliminazione di una riga è la seguente.

```
DELETE FROM table_name WHERE condition ;
```

Per eliminare una riga

1. Apri AWS CloudShell e connettiti ad Amazon Keyspaces utilizzando il seguente comando. Assicurati di eseguire l'aggiornamento *us-east-1* con la tua regione.

```
cqlsh-expansion cassandra.us-east-1.amazonaws.com 9142 --ssl
```

2. Esegui la seguente DELETE interrogazione.

```
DELETE FROM catalog.book_awards WHERE year=2020 AND award='Richard Roe' AND
category='Fiction' AND rank=1;
```

3. Verificate che l'eliminazione sia stata effettuata come previsto.

```
SELECT * FROM catalog.book_awards WHERE year=2020 AND award='Richard Roe' AND
category='Fiction' AND rank=1;
```

L'output di questa istruzione è simile al seguente dopo l'eliminazione della riga.

```
year | award | category | rank | author | book_title | publisher
-----+-----+-----+-----+-----+-----+-----
(0 rows)
```

Puoi eliminare automaticamente i dati scaduti dalla tua tabella utilizzando Amazon Keyspaces Time to Live, per ulteriori informazioni, consulta [the section called “Fai scadere i dati con Time to Live”](#)

Eliminare una tabella in Amazon Keyspaces

Per evitare che ti vengano addebitati costi per tabelle e dati che non ti servono, elimina tutte le tabelle che non utilizzi. Quando elimini una tabella, la tabella e i relativi dati vengono eliminati e non ti vengono più addebitati costi per tali tabelle. Tuttavia, lo spazio delle chiavi rimane. Quando si elimina uno spazio chiave, lo spazio chiave e tutte le relative tabelle vengono eliminati e si interrompe l'addebito per tali spazi.

È possibile eliminare una tabella utilizzando la console, CQL o. AWS CLI. Quando si elimina una tabella, la tabella e tutti i relativi dati vengono eliminati.

Utilizzo della console

La procedura seguente elimina una tabella e tutti i relativi dati utilizzando il Console di gestione AWS.

Per eliminare una tabella utilizzando la console

1. [Accedi a e apri Console di gestione AWS la console Amazon Keyspaces a casahttps://console.aws.amazon.com/keyspaces/.](https://console.aws.amazon.com/keyspaces/)
2. Nel pannello di navigazione, seleziona Tabelle.
3. Scegli la casella a sinistra del nome di ogni tabella che desideri eliminare.
4. Scegliere Delete (Elimina).
5. Nella schermata Elimina tabella, inserisci **Delete** la casella. Quindi, scegli Elimina tabella.

6. Per verificare che la tabella sia stata eliminata, scegli Tabelle nel riquadro di navigazione e conferma che la `book_awards` tabella non è più elencata.

Usare CQL

La procedura seguente elimina una tabella e tutti i relativi dati utilizzando CQL.

Per eliminare una tabella utilizzando CQL

1. Apri AWS CloudShell e connettiti ad Amazon Keyspaces utilizzando il seguente comando. Assicurati di eseguire l'aggiornamento `us-east-1` con la tua regione.

```
cqlsh-expansion cassandra.us-east-1.amazonaws.com 9142 --ssl
```

2. Elimina la tabella inserendo la seguente dichiarazione.

```
DROP TABLE IF EXISTS catalog.book_awards ;
```

3. Verifica che la tabella sia stata eliminata.

```
SELECT * FROM system_schema.tables WHERE keyspace_name = 'catalog' ;
```

L'output dovrebbe assomigliare a questo. Nota che questa operazione potrebbe richiedere del tempo, quindi esegui nuovamente l'istruzione dopo un minuto se non vedi questo risultato.

```
keyspace_name | table_name | bloom_filter_fp_chance | caching | cdc | comment
| compaction | compression | crc_check_chance | dclocal_read_repair_chance
| default_time_to_live | extensions | flags | gc_grace_seconds | id |
max_index_interval | memtable_flush_period_in_ms | min_index_interval |
read_repair_chance | speculative_retry
-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+-----+

(0 rows)
```

Usando il AWS CLI

La procedura seguente elimina una tabella e tutti i relativi dati utilizzando AWS CLI

Per eliminare una tabella utilizzando il AWS CLI

1. Aprire CloudShell
2. Elimina la tabella con la seguente dichiarazione.

```
aws keyspaces delete-table --keyspace-name 'catalog' --table-name 'book_awards'
```

3. Per verificare che la tabella sia stata eliminata, puoi elencare tutte le tabelle in un keyspace.

```
aws keyspaces list-tables --keyspace-name 'catalog'
```

Vedrai il seguente output. Tieni presente che questa operazione asincrona può richiedere del tempo. Esegui nuovamente il comando dopo qualche istante per confermare che la tabella è stata eliminata.

```
{  
  "tables": []  
}
```

Eliminare un keyspace in Amazon Keyspaces

Per evitare che ti vengano addebitati costi per gli spazi chiave, elimina tutti i keyspace che non stai utilizzando. Quando elimini uno spazio chiave, lo spazio chiave e tutte le relative tabelle vengono eliminati e non ti vengono più addebitati costi per tali spazi.

È possibile eliminare uno spazio chiave utilizzando la console, CQL o AWS CLI

Utilizzo della console

La procedura seguente elimina uno spazio chiave e tutte le relative tabelle e dati utilizzando la console.

Per eliminare uno spazio chiave utilizzando la console

1. [Accedi a e apri Console di gestione AWS la console Amazon Keyspaces a casa](https://console.aws.amazon.com/keyspaces/)<https://console.aws.amazon.com/keyspaces/>.
2. Nel riquadro di navigazione, scegli Keyspaces.
3. Scegli la casella a sinistra del nome di ogni keyspace che desideri eliminare.
4. Scegliere Delete (Elimina).
5. Nella schermata Elimina spazio chiave, inserisci **Delete** nella casella. Quindi, scegli Elimina spazio chiave.
6. Per verificare che il keyspace sia stato eliminato, scegli Keyspaces nel riquadro di navigazione e conferma che non è più elencato. Poiché ne hai eliminato lo spazio chiave, anche la book_awards tabella in Tabelle non dovrebbe essere elencata.

Usare CQL

La procedura seguente elimina uno spazio di chiavi e tutte le relative tabelle e dati utilizzando CQL.

Per eliminare uno spazio chiave utilizzando CQL

1. Apri AWS CloudShell e connettiti ad Amazon Keyspaces utilizzando il seguente comando. Assicurati di eseguire l'aggiornamento *us-east-1* con la tua regione.

```
cqlsh-expansion cassandra.us-east-1.amazonaws.com 9142 --ssl
```

2. Elimina il tuo keyspace inserendo la seguente dichiarazione.

```
DROP KEYSPACE IF EXISTS catalog ;
```

3. Verifica che il tuo keyspace sia stato eliminato.

```
SELECT * from system_schema.keyspaces ;
```

Il tuo keyspace non dovrebbe essere elencato. Nota che, poiché si tratta di un'operazione asincrona, può verificarsi un ritardo prima che il keyspace venga eliminato. Dopo l'eliminazione del keyspace, l'output dell'istruzione dovrebbe avere il seguente aspetto.

```
keyspace_name | durable_writes | replication
```

```

-----+-----
+-----
      system_schema |          True | {'class':
'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '3'}
      system_schema_mcs |          True | {'class':
'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '3'}
      system |          True | {'class':
'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '3'}
system_multiregion_info |          True | {'class':
'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '3'}

(4 rows)

```

Usando il AWS CLI

La procedura seguente elimina un keyspace e tutte le relative tabelle e dati utilizzando AWS CLI

Per eliminare uno spazio chiave utilizzando AWS CLI

1. Aprire AWS CloudShell
2. Elimina il tuo keyspace inserendo la seguente dichiarazione.

```
aws keyspaces delete-keyspace --keyspace-name 'catalog'
```

3. Verifica che il tuo keyspace sia stato eliminato.

```
aws keyspaces list-keyspaces
```

L'output di questa istruzione dovrebbe essere simile a questo ed elencare solo gli spazi chiave di sistema. Nota che, poiché si tratta di un'operazione asincrona, può verificarsi un ritardo prima che il keyspace venga eliminato.

```

{
  "keyspaces": [
    {
      "keyspaceName": "system_schema",
      "resourceArn": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/
system_schema/",
      "replicationStrategy": "SINGLE_REGION"
    },
  ],
}

```

```
{
  "keyspaceName": "system_schema_mcs",
  "resourceArn": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/
system_schema_mcs/",
  "replicationStrategy": "SINGLE_REGION"
},
{
  "keyspaceName": "system",
  "resourceArn": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/
system/",
  "replicationStrategy": "SINGLE_REGION"
},
{
  "keyspaceName": "system_multiregion_info",
  "resourceArn": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/
system_multiregion_info/",
  "replicationStrategy": "SINGLE_REGION"
}
]
```

Tutorial e soluzioni per Amazon Keyspaces

Questo argomento include vari tutorial che illustrano i passaggi dettagliati necessari per connettersi ad Amazon Keyspaces a livello di codice o per utilizzare Amazon Keyspaces con altri servizi per creare soluzioni. AWS I tutorial disponibili coprono diversi scenari di accesso multiservizio più diffusi e dimostrano come integrare Amazon Keyspaces con Amazon Virtual Private Cloud, Amazon Elastic Kubernetes Service, Amazon Simple Storage Service e/o con tecnologie open source come Apache Spark. AWS Glue

Per step-by-step i tutorial che mostrano come connettersi ad Amazon Keyspaces utilizzando i driver Apache Cassandra, consulta [the section called “Utilizzo di un driver client Cassandra”](#)

Argomenti

- [Tutorial: Connettiti ad Amazon Keyspaces utilizzando un endpoint VPC di interfaccia](#)
- [Tutorial: Integrazione con Apache Spark per importare o esportare dati](#)
- [Tutorial: Connect da un'applicazione containerizzata ospitata su Amazon Elastic Kubernetes Service](#)
- [Tutorial: esporta una tabella Amazon Keyspaces in Amazon S3 utilizzando AWS Glue](#)

Tutorial: Connettiti ad Amazon Keyspaces utilizzando un endpoint VPC di interfaccia

Questo tutorial illustra la configurazione e l'utilizzo di un endpoint VPC di interfaccia per Amazon Keyspaces.

Gli endpoint VPC di interfaccia consentono la comunicazione privata tra il tuo cloud privato virtuale (VPC) in esecuzione su Amazon VPC e Amazon Keyspaces. Gli endpoint VPC di interfaccia sono alimentati da AWS PrivateLink, un AWS servizio che consente la comunicazione privata tra e VPCs servizi. AWS Per ulteriori informazioni, consulta [the section called “Utilizzo di endpoint VPC dell'interfaccia ”](#).

Argomenti

- [Prerequisiti e considerazioni del tutorial](#)
- [Passaggio 1: avviare un' EC2istanza Amazon](#)
- [Passaggio 2: configura la tua EC2 istanza Amazon](#)

- [Fase 3: creare un endpoint VPC per Amazon Keyspaces](#)
- [Passaggio 4: Configurare le autorizzazioni per la connessione agli endpoint VPC](#)
- [Fase 5: configura il monitoraggio con CloudWatch](#)
- [Passaggio 6: \(Facoltativo\) Procedure consigliate per configurare la dimensione del pool di connessioni per l'applicazione](#)
- [Fase 7: \(Facoltativo\) Pulizia](#)

Prerequisiti e considerazioni del tutorial

Prima di iniziare questo tutorial, segui le istruzioni di configurazione riportate in AWS . [Accesso ad Amazon Keyspaces \(per Apache Cassandra\)](#) Questi passaggi includono la registrazione AWS e la creazione di un principale AWS Identity and Access Management (IAM) con accesso ad Amazon Keyspaces. Prendi nota del nome dell'utente IAM e delle chiavi di accesso perché ti serviranno più avanti in questo tutorial.

Crea uno spazio chiave con il nome myKeyspace e almeno una tabella per testare la connessione utilizzando l'endpoint VPC più avanti in questo tutorial. Puoi trovare istruzioni dettagliate in. [Nozioni di base](#)

Dopo aver completato i passaggi relativi ai prerequisiti, procedi a [Passaggio 1: avviare un' EC2istanza Amazon](#).

Passaggio 1: avviare un' EC2istanza Amazon

In questa fase, avvii un' EC2 istanza Amazon nel tuo Amazon VPC predefinito. Puoi quindi creare e utilizzare un endpoint VPC per Amazon Keyspaces.

Per avviare un' EC2 istanza Amazon

1. Apri la EC2 console Amazon all'indirizzo <https://console.aws.amazon.com/ec2/>.
2. Seleziona Avvia istanza e completa le seguenti operazioni:

Dalla dashboard della EC2 console, nella casella Launch instance, scegli Launch instance, quindi scegli Launch instance tra le opzioni visualizzate.

In Nome e tag, per Nome, inserisci un nome descrittivo per l'istanza.

In Immagini dell'applicazione e del sistema operativo (Amazon Machine Image):

- Scegli Quick Start, quindi scegli Ubuntu. Questo è il sistema operativo (OS) per l'istanza.
- In Amazon Machine Image (AMI), puoi utilizzare l'immagine predefinita contrassegnata come idonea al piano gratuito. Un'Amazon Machine Image (AMI) è una configurazione di base che serve come modello per l'istanza.

In Tipo di istanza:

- Dall'elenco Tipo di istanza, scegliete il tipo di istanza t2.micro, selezionato per impostazione predefinita.

In Coppia di chiavi (login), per Nome della coppia di chiavi, scegli una delle seguenti opzioni per questo tutorial:

- Se non disponi di una coppia di EC2 chiavi Amazon, scegli Crea una nuova coppia di chiavi e segui le istruzioni. Ti verrà chiesto di scaricare un file di chiave privata (file.pem). Questo file ti servirà in seguito quando accederai alla tua EC2 istanza Amazon, quindi prendi nota del percorso del file.
- Se disponi già di una coppia di EC2 chiavi Amazon, vai a Seleziona una coppia di chiavi e scegli la tua coppia di chiavi dall'elenco. Devi già avere il file della chiave privata (file.pem) disponibile per accedere alla tua istanza Amazon EC2 .

In Impostazioni di rete:

- Scegli Modifica.
- Scegli Seleziona un gruppo di sicurezza esistente.
- Nell'elenco dei gruppi di sicurezza, scegli default (predefinito). Questo è il gruppo di sicurezza di default per il tuo ambiente VPC.

Continua con il riepilogo.

- Esamina un riepilogo della configurazione dell'istanza nel pannello Riepilogo. Al termine, scegli Avvia istanza.
3. Nella schermata di completamento della nuova EC2 istanza Amazon, scegli il riquadro Connect to instance. La schermata successiva mostra le informazioni necessarie e i passaggi necessari per connettersi alla nuova istanza. Prendi nota delle seguenti informazioni:

- Il comando di esempio per proteggere il file chiave
- La stringa di connessione
- Il nome IPv4 DNS pubblico

Dopo aver preso nota delle informazioni in questa pagina, puoi continuare con il passaggio successivo di questo tutorial ([Passaggio 2: configura la tua EC2 istanza Amazon](#)).

Note

Bastano alcuni minuti prima che la tua EC2 istanza Amazon diventi disponibile. Prima di continuare, verificare che Stato istanza sia `running` e che tutti i controlli di verifica stato siano stati superati.

Passaggio 2: configura la tua EC2 istanza Amazon

Quando la tua EC2 istanza Amazon è disponibile, puoi accedervi e prepararla per il primo utilizzo.

Note

I passaggi seguenti presuppongono che ti stia connettendo alla tua EC2 istanza Amazon da un computer che esegue Linux. Per altri modi di connessione, consulta [Connetti alla tua istanza Linux](#) nella Amazon EC2 User Guide.

Per configurare la tua EC2 istanza Amazon

1. Devi autorizzare il traffico SSH in entrata verso la tua istanza Amazon. EC2 Per fare ciò, crea un nuovo gruppo EC2 di sicurezza, quindi assegna il gruppo di sicurezza alla tua istanza. EC2
 - a. Fare clic su Security Groups (Gruppi di sicurezza) nel pannello di navigazione.
 - b. Scegli Crea gruppo di sicurezza. Nella finestra Crea gruppo di sicurezza effettua le operazioni seguenti:
 - Nome del gruppo di sicurezza: inserisci un nome per il tuo gruppo di sicurezza. Ad esempio: `my-ssh-access`

- Descrizione: inserisci una breve descrizione per il gruppo di sicurezza.
- VPC: scegli il tuo VPC predefinito.
- Nella sezione Regole in entrata, scegli Aggiungi regola e procedi come segue:
 - Tipo: scegli SSH.
 - Fonte: scegli il mio IP.
 - Scegli Aggiungi regola.

Nella parte inferiore della pagina, conferma le impostazioni di configurazione e scegli Crea gruppo di sicurezza.

- Nel riquadro di navigazione, scegliere Instances (Istanze).
 - Scegli l' EC2 istanza Amazon in cui hai effettuato il lancio [Passaggio 1: avviare un' EC2 istanza Amazon](#).
 - Scegli Azioni, scegli Sicurezza, quindi scegli Cambia gruppi di sicurezza.
 - In Modifica gruppi di sicurezza, vai a Gruppi di sicurezza associati e inserisci il gruppo di sicurezza che hai creato in precedenza in questa procedura (ad esempio, my-ssh-access). Dovrebbe essere selezionato anche il gruppo di sicurezza default esistente. Conferma le impostazioni di configurazione e scegli Salva.
- Usa il seguente comando per proteggere il tuo file di chiave privata dall'accesso. Se salti questo passaggio, la connessione fallisce.

```
chmod 400 path_to_file/my-keypair.pem
```

- Usa il ssh comando per accedere alla tua EC2 istanza Amazon, come nell'esempio seguente.

```
ssh -i path_to_file/my-keypair.pem ubuntu@public-dns-name
```

Devi specificare il tuo file di chiave privata (file.pem) e il nome DNS pubblico dell'istanza. Consulta [Passaggio 1: avviare un' EC2 istanza Amazon](#).

L'ID accesso è ubuntu. Non è richiesta una password.

Per ulteriori informazioni su come consentire le connessioni alla tua EC2 istanza Amazon e per AWS CLI istruzioni, consulta [Autorizza il traffico in entrata per le tue istanze Linux](#) nella Amazon EC2 User Guide.

- a. Installare unzip.

```
sudo apt install unzip
```

- b. Scarica il zip file con AWS CLI.

```
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o  
"awscliv2.zip"
```

- c. Decomprimere il file.

```
unzip awscliv2.zip
```

- d. Installa il AWS CLI.

```
sudo ./aws/install
```

- e. Conferma la versione dell' AWS CLI installazione.

```
aws --version
```

L'output dovrebbe essere simile al seguente:

```
aws-cli/2.9.19 Python/3.9.11 Linux/5.15.0-1028-aws exe/x86_64.ubuntu.22 prompt/  
off
```

5. Configura AWS le tue credenziali, come illustrato nell'esempio seguente. Quando richiesto, inserisci AWS l'ID della chiave di accesso, la chiave segreta e il nome predefinito della regione.

aws configure

```
AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE  
AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY  
Default region name [None]: us-east-1  
Default output format [None]:
```

6. Aggiungi regole al tuo gruppo VPCs di sicurezza che consentano l'accesso HTTP, HTTPS e SSH in entrata dagli indirizzi IPv6
7. Per confermare che il tuo endpoint VPC è stato configurato correttamente, devi utilizzare una connessione `cqlsh` ad Amazon Keyspaces. Se utilizzi il tuo ambiente locale o l'editor CQL

di Amazon Keyspaces in Console di gestione AWS, la connessione passa automaticamente attraverso l'endpoint pubblico anziché l'endpoint VPC. Da utilizzare `cqlsh` per testare la connessione dell'endpoint VPC in questo tutorial, completa le istruzioni di configurazione in.

[Utilizzo `cqlsh` per connettersi ad Amazon Keyspaces](#)

Ora sei pronto per creare un endpoint VPC dual-stack per Amazon Keyspaces.

Fase 3: creare un endpoint VPC per Amazon Keyspaces

In questa fase, crei un endpoint VPC dual-stack per Amazon Keyspaces utilizzando AWS CLI. Per creare l'endpoint VPC utilizzando la console VPC, puoi seguire le istruzioni per la creazione di [un endpoint VPC](#) nella Guida AWS PrivateLink. Quando filtrate per il nome del servizio, immettete.

Cassandra

Per creare un endpoint VPC utilizzando il AWS CLI

1. Prima di iniziare, verifica di poter comunicare con Amazon Keyspaces utilizzando il suo endpoint pubblico.

```
aws keyspaces list-tables --keyspace-name 'myKeyspace'
```

L'output mostra un elenco di tabelle Amazon Keyspaces contenute nel keyspace specificato. Se non disponi di tabelle, l'elenco è vuoto.

```
{
  "tables": [
    {
      "keyspaceName": "myKeyspace",
      "tableName": "myTable1",
      "resourceArn": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/catalog/table/myTable1"
    },
    {
      "keyspaceName": "myKeyspace",
      "tableName": "myTable2",
      "resourceArn": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/catalog/table/myTable2"
    }
  ]
}
```

2. Verifica che Amazon Keyspaces sia un servizio disponibile per la creazione di endpoint VPC nella regione corrente. AWS Il comando è mostrato in grassetto, seguito dall'output di esempio.

```
aws ec2 describe-vpc-endpoint-services

{
  "ServiceNames": [
    "com.amazonaws.us-east-1.cassandra",
    "com.amazonaws.us-east-1.cassandra-fips"
    "api.aws.us-east-1.cassandra-streams"
  ]
}
```

Se Amazon Keyspaces è uno dei servizi disponibili nell'output del comando, puoi procedere con la creazione di un endpoint VPC.

3. Per connetterti ad Amazon Keyspaces utilizzando endpoint dual-stack IPv6 abilitati, verifica che il tuo VPC supporti e configura le sottoreti con il supporto. IPv6 IPv6 Per aggiungere IPv6 supporto a un VPC esistente che attualmente supporta solo VPC IPv4, consulta il [IPv6 supporto per il tuo VPC nella Amazon VPC User Guide](#);
4. Determinare l'identificatore VPC.

```
aws ec2 describe-vpcs

{
  "Vpcs": [
    {
      "OwnerId": "111122223333",
      "InstanceTenancy": "default",
      "Ipv6CidrBlockAssociationSet": [
        {
          "AssociationId": "vpc-cidr-assoc-0000aaa0a00a00aa0",
          "Ipv6CidrBlock": "2600:1f18:e19:7d00::/56",
          "Ipv6CidrBlockState": {
            "State": "associated"
          },
          "NetworkBorderGroup": "us-east-1",
          "Ipv6Pool": "Amazon",
          "Ipv6AddressAttribute": "public",
          "IpSource": "amazon"
        }
      ],
    }
  ],
}
```

```

        "CidrBlockAssociationSet": [
            {
                "AssociationId": "vpc-cidr-assoc-00a0000a",
                "CidrBlock": "111.11.0.0/16",
                "CidrBlockState": {
                    "State": "associated"
                }
            }
        ],
        "IsDefault": true,
        "BlockPublicAccessStates": {
            "InternetGatewayBlockMode": "off"
        },
        "VpcId": "vpc-a1234bcd",
        "State": "available",
        "CidrBlock": "111.11.0.0/16",
        "DhcpOptionsId": "dopt-a00aaaaa"
    }
]
}

```

Nell'output di esempio, l'ID VPC è `vpc-a1234bcd`.

5. Utilizza un filtro per raccogliere dettagli sulle sottoreti del VPC.

```

aws ec2 describe-subnets --filters "Name=vpc-id,Values=vpc-a1234bcd"

{
    "Subnets": [
        {
            "AvailabilityZoneId": "use1-az1",
            "MapCustomerOwnedIpOnLaunch": false,
            "OwnerId": "111122223333",
            "AssignIpv6AddressOnCreation": false,
            "Ipv6CidrBlockAssociationSet": [
                {
                    "AssociationId": "subnet-cidr-assoc-05d75732736740283",
                    "Ipv6CidrBlock": "*****",
                    "Ipv6CidrBlockState": {
                        "State": "associated"
                    },
                    "Ipv6AddressAttribute": "public",
                    "IpSource": "amazon"
                }
            ]
        }
    ]
}

```

```

    ],
    "SubnetArn": "arn:aws:ec2:us-east-1:111122223333:subnet/
subnet-70b24b16",
    "EnableDns64": false,
    "Ipv6Native": false,
    "PrivateDnsNameOptionsOnLaunch": {
        "HostnameType": "ip-name",
        "EnableResourceNameDnsARecord": false,
        "EnableResourceNameDnsAAAARecord": false
    },
    "BlockPublicAccessStates": {
        "InternetGatewayBlockMode": "off"
    },
    "SubnetId": "subnet-70b24b16",
    "State": "available",
    "VpcId": "vpc-a1234bcd",
    "CidrBlock": "*****/20",
    "AvailableIpAddressCount": 4096,
    "AvailabilityZone": "us-east-1a",
    "DefaultForAz": true,
    "MapPublicIpOnLaunch": true
},
{
    "AvailabilityZoneId": "use1-az2",
    "MapCustomerOwnedIpOnLaunch": false,
    "OwnerId": "111122223333",
    "AssignIpv6AddressOnCreation": false,
    "Ipv6CidrBlockAssociationSet": [
        {
            "AssociationId": "subnet-cidr-assoc-0ec6fb253e05b17eb",
            "Ipv6CidrBlock": "*****",
            "Ipv6CidrBlockState": {
                "State": "associated"
            },
            "Ipv6AddressAttribute": "public",
            "IpSource": "amazon"
        }
    ],
    "SubnetArn": "arn:aws:ec2:us-east-1:111122223333:subnet/subnet-
c63ffbe7",
    "EnableDns64": false,
    "Ipv6Native": false,
    "PrivateDnsNameOptionsOnLaunch": {
        "HostnameType": "ip-name",

```

```

        "EnableResourceNameDnsARecord": false,
        "EnableResourceNameDnsAAAARecord": false
    },
    "BlockPublicAccessStates": {
        "InternetGatewayBlockMode": "off"
    },
    "SubnetId": "subnet-c63ffbe7",
    "State": "available",
    "VpcId": "vpc-a1234bcd",
    "CidrBlock": "*****/20",
    "AvailableIpAddressCount": 4087,
    "AvailabilityZone": "us-east-1b",
    "DefaultForAz": true,
    "MapPublicIpOnLaunch": true
    }
]
}

```

Nell'output di esempio, sono disponibili due IDs sottoreti: e. subnet-70b24b16 subnet-c63ffbe7

6. Crea l'endpoint VPC. Per il parametro `--vpc-id`, specificare l'ID VPC della fase precedente. Per il `--subnet-ids` parametro, specificate la sottorete IDs del passaggio precedente. Utilizzate il `--vpc-endpoint-type` parametro per definire l'endpoint come interfaccia. Per creare un endpoint dual-stack, usa. `--ip-address-type dualstack` Per ulteriori informazioni sul comando, vedere nel Command Reference [create-vpc-endpoint](#).AWS CLI

```

aws ec2 create-vpc-endpoint \
  --vpc-endpoint-type Interface \
  --vpc-id vpc-a1234bcd \
  --ip-address-type dualstack \
  --service-name com.amazonaws.us-east-1.cassandra \
  --subnet-ids subnet-70b24b16 subnet-c63ffbe7

{
  "VpcEndpoint": {
    "VpcEndpointId": "vpce-0000000abc111d2ef3",
    "VpcEndpointType": "Interface",
    "VpcId": "vpc-a1234bcd",
    "ServiceName": "com.amazonaws.us-east-1.cassandra",
    "State": "pending",
    "RouteTableIds": [],
    "SubnetIds": [

```



```

        "subnet-70b24b16",
        "subnet-c63ffbe7"
    ],
    "Groups": [
        {
            "GroupId": "sg-0123456789",
            "GroupName": "default"
        }
    ],
    "IpAddressType": "dualstack",
    "DnsOptions": {
        "DnsRecordIpType": "dualstack"
    },
    "PrivateDnsEnabled": true,
    "RequesterManaged": false,
    "NetworkInterfaceIds": [
        "eni-08cd525f72ea6f1fa",
        "eni-07b1f6c895169d8fb"
    ],
    "DnsEntries": [
        {
            "DnsName": "vpce-0000000000-1234567.cassandra.us-
east-1.vpce.amazonaws.com",
            "HostedZoneId": "Z7HUB22UULQXV"
        },
        {
            "DnsName": "vpce-0000000000-1234567-us-east-1a.cassandra.us-
east-1.vpce.amazonaws.com",
            "HostedZoneId": "Z7HUB22UULQXV"
        },
        {
            "DnsName": "cassandra.us-east-1.amazonaws.com",
            "HostedZoneId": "ZONEIDPENDING"
        },
        {
            "DnsName": "cassandra.us-east-1.api.aws",
            "HostedZoneId": "ZONEIDPENDING"
        }
    ],
    "CreationTimestamp": "2025-09-19T15:19:19.266000+00:00",
    "OwnerId": "111122223333",
    "ServiceRegion": "us-east-1"
}

```

```
}
```

Passaggio 4: Configurare le autorizzazioni per la connessione agli endpoint VPC

Le procedure in questa fase mostrano come configurare regole e autorizzazioni per l'utilizzo dell'endpoint VPC con Amazon Keyspaces.

Per configurare una regola in entrata per il nuovo endpoint per consentire il traffico TCP in entrata

1. Nella console Amazon VPC, nel pannello a sinistra, scegli Endpoints e scegli l'endpoint creato nel passaggio precedente.
2. Scegli Gestisci gruppi di sicurezza, quindi scegli il gruppo di sicurezza associato a questo endpoint.
3. Scegli Regole in entrata, quindi scegli Modifica regole in entrata.
4. Aggiungi una regola in entrata con Digita come CQLSH/CASSANDRA. Questo imposta automaticamente l'intervallo di porte su 9142.
5. Per salvare la nuova regola in entrata, scegli Salva regole.

Per configurare le autorizzazioni utente IAM

1. Verifica che l'utente IAM utilizzato per connettersi ad Amazon Keyspaces disponga delle autorizzazioni appropriate. In AWS Identity and Access Management (IAM), puoi utilizzare la policy AWS gestita AmazonKeyspacesReadOnlyAccess per concedere all'utente IAM l'accesso in lettura ad Amazon Keyspaces.
 - a. Accedi Console di gestione AWS e apri la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
 - b. Nel pannello di controllo della console IAM, seleziona Utenti, quindi scegli l'utente IAM dall'elenco.
 - c. Nella pagina Summary (Riepilogo), scegli Add permissions (Aggiungi autorizzazioni).
 - d. Scegli Attach existing policies directly (Collega direttamente le policy esistenti).
 - e. Dall'elenco delle politiche, scegli AmazonKeyspacesReadOnlyAccess, quindi scegli Avanti: revisione.
 - f. Scegli Add Permissions (Aggiungi autorizzazioni).

2. Verifica di poter accedere ad Amazon Keyspaces tramite l'endpoint VPC.

```
aws keyspaces list-tables --keyspace-name 'my_keyspace'
```

Se lo desideri, puoi provare altri AWS CLI comandi per Amazon Keyspaces. Per ulteriori informazioni, consulta la sezione relativa alle [informazioni di riferimento ai comandi di AWS CLI](#).

Note

Le autorizzazioni minime richieste a un utente o ruolo IAM per accedere ad Amazon Keyspaces sono le autorizzazioni di lettura per la tabella di sistema, come illustrato nella seguente policy. Per ulteriori informazioni sulle autorizzazioni basate su policy, consulta [the section called “Esempi di policy basate sull’identità”](#)

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cassandra:Select"
      ],
      "Resource": [
        "arn:aws:cassandra:us-east-1:111122223333:/keyspace/system*"
      ]
    }
  ]
}
```

3. Concedi all'utente IAM l'accesso in lettura all' EC2 istanza Amazon con il VPC.

Quando utilizzi Amazon Keyspaces con endpoint VPC, devi concedere all'utente o al ruolo IAM che accede ad Amazon Keyspaces le autorizzazioni di sola lettura per l'istanza Amazon e il VPC per raccogliere i dati degli endpoint EC2 e delle interfacce di rete. Amazon Keyspaces memorizza queste informazioni nella `system.peers` tabella e le utilizza per gestire le connessioni.

 Note

Le policy gestite `AmazonKeyspacesFullAccess` includono `AmazonKeyspacesReadOnlyAccess_v2` le autorizzazioni necessarie per consentire ad Amazon Keyspaces di accedere all'istanza EC2 Amazon per leggere informazioni sugli endpoint VPC di interfaccia disponibili.

- a. Accedi Console di gestione AWS e apri la console IAM all'indirizzo. <https://console.aws.amazon.com/iam/>
- b. Nella dashboard della console IAM, scegli Policies.
- c. Scegli Crea policy, quindi scegli la scheda JSON.
- d. Copia la seguente politica e scegli Avanti: Tag.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ListVPCEndpoints",
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeVpcEndpoints"
      ],
      "Resource": "*"
    }
  ]
}
```

- e. Scegli Avanti: revisione, inserisci `keyspacesVPCendpoint` il nome della politica e scegli Crea politica.
- f. Nel pannello di controllo della console IAM, seleziona Utenti, quindi scegli l'utente IAM dall'elenco.
- g. Nella pagina Summary (Riepilogo), scegli Add permissions (Aggiungi autorizzazioni).
- h. Scegli Attach existing policies directly (Collega direttamente le policy esistenti).
- i. Dall'elenco delle politiche, scegli gli spazi chiave `VPCendpoint`, quindi scegli Avanti: revisione.

- j. Scegli Add Permissions (Aggiungi autorizzazioni).
4. Per verificare che la `system.peers` tabella Amazon Keyspaces venga aggiornata con le informazioni VPC, esegui la seguente query dalla tua istanza Amazon utilizzando. EC2 `cqlsh`. Se non l'hai ancora installata `cqlsh` sulla tua EC2 istanza Amazon nella fase 2, segui le istruzioni riportate in [the section called “Utilizzo di cqlsh-expansion”](#).

```
SELECT * FROM system.peers;
```

L'output restituisce nodi con indirizzi IPv6 IP privati, a seconda della configurazione del VPC e della sottorete nella regione. AWS

```
peer                                | data_center | host_id
      | preferred_ip                | rack      | release_version
| rpc_address                    | schema_version |
tokens
-----+-----
+-----+-----+-----+-----+
+-----+-----+-----+-----+
+-----+-----+-----+-----+
+-----+-----+-----+-----+
+-----+-----+-----+-----+
      2600:1111:2222:3333:283b:8e6:d04f | us-east-1 | dddddd-7a22-3582-
a73d-49338a686a53 | 2600:1111:2222:3333:283b:8e6:d04f | us-east-1 |
      3.11.2 | 2600:1111:2222:3333:283b:8e6:d04f | 05deae2d-6405-494d-a965-
c0e5836bcb3c | {'85070591730234615865843651857942052863'}
      2600:1111:2222:4444:7d26:5a09:1b44 | us-east-1 | 66666666-035d-37ef-
a247-19a6a867ab09 | 2600:1111:2222:4444:7d26:5a09:1b44 | us-east-1 |
      3.11.2 | 2600:1111:2222:4444:7d26:5a09:1b44 | 05deae2d-6405-494d-a965-
c0e5836bcb3c | {'170141183460469231731687303715884105726'}
```

Note

È necessario utilizzare una `cqlsh` connessione ad Amazon Keyspaces per confermare che l'endpoint VPC sia stato configurato correttamente. Se utilizzi il tuo ambiente locale o l'editor CQL di Amazon Keyspaces in Console di gestione AWS, la connessione passa automaticamente attraverso l'endpoint pubblico anziché l'endpoint VPC. Se vedi nove indirizzi IP, queste sono le voci che Amazon Keyspaces scrive automaticamente nella `system.peers` tabella per le connessioni endpoint pubbliche.

Fase 5: configura il monitoraggio con CloudWatch

Questo passaggio mostra come utilizzare Amazon per CloudWatch monitorare la connessione degli endpoint VPC ad Amazon Keyspaces.

AWS PrivateLink pubblica punti dati CloudWatch sugli endpoint dell'interfaccia. Puoi utilizzare le metriche per verificare che le prestazioni del sistema siano quelle previste. Lo spazio dei nomi `AWS/PrivateLinkEndpoints` in CloudWatch include le metriche per gli endpoint dell'interfaccia. Per ulteriori informazioni, consulta le [CloudWatch metriche](#) contenute nella Guida. `AWS PrivateLink`

Per creare una CloudWatch dashboard con metriche degli endpoint VPC

1. Apri la console all' CloudWatch indirizzo. <https://console.aws.amazon.com/cloudwatch/>
2. Nel pannello di navigazione seleziona Dashboards (Pannelli di controllo). Quindi scegli Crea dashboard. Inserisci un nome per la dashboard e scegli Crea.
3. In Aggiungi widget, scegli Numero.
4. In Metriche, scegli `PrivateLinkEndpointsAWS/`.
5. Scegli il tipo di endpoint, il nome del servizio, l'ID endpoint VPC, l'ID VPC.
6. Seleziona le metriche **ActiveConnections** e **NewConnections** scegli Crea widget.
7. Salva la dashboard.

La `ActiveConnections` metrica è definita come il numero di connessioni attive simultanee che l'endpoint ha ricevuto nell'ultimo periodo di un minuto. La `NewConnections` metrica è definita come il numero di nuove connessioni stabilite tramite l'endpoint nell'ultimo periodo di un minuto.

Per ulteriori informazioni sulla creazione di dashboard, consulta [Create dashboard](#) nella Guida per l'utente. CloudWatch

Passaggio 6: (Facoltativo) Procedure consigliate per configurare la dimensione del pool di connessioni per l'applicazione

In questa sezione, descriviamo come determinare la dimensione ideale del pool di connessioni in base ai requisiti di velocità di trasmissione delle query dell'applicazione.

Amazon Keyspaces consente un massimo di 3.000 query CQL al secondo per connessione TCP. Quindi non c'è praticamente alcun limite al numero di connessioni che un driver può stabilire con

Amazon Keyspaces. Tuttavia, ti consigliamo di adattare la dimensione del pool di connessioni ai requisiti della tua applicazione e di considerare gli endpoint disponibili quando utilizzi Amazon Keyspaces con connessioni endpoint VPC.

La dimensione del pool di connessioni viene configurata nel driver client. Ad esempio, in base a un pool locale di 2 dimensioni e a un endpoint di interfaccia VPC creato in 3 zone di disponibilità, il driver stabilisce 6 connessioni per l'interrogazione (7 in totale, inclusa una connessione di controllo). Utilizzando queste 6 connessioni, è possibile supportare un massimo di 18.000 query CQL al secondo.

Se l'applicazione deve supportare 40.000 query CQL al secondo, procedi a ritroso partendo dal numero di query necessarie per determinare la dimensione del pool di connessioni richiesta. Per supportare 40.000 query CQL al secondo, è necessario configurare la dimensione del pool locale in modo che sia almeno 5, il che supporta un minimo di 45.000 query CQL al secondo.

È possibile monitorare se si supera la quota per il numero massimo di operazioni al secondo, per connessione utilizzando la metrica nel namespace. `PerConnectionRequestRateExceeded` CloudWatch AWS/Cassandra La `PerConnectionRequestRateExceeded` metrica mostra il numero di richieste ad Amazon Keyspaces che superano la quota per la frequenza di richieste per connessione.

Gli esempi di codice in questo passaggio mostrano come stimare e configurare il pool di connessioni quando si utilizzano gli endpoint VPC di interfaccia.

Java

È possibile configurare il numero di connessioni per pool nel driver Java. Per un esempio completo di connessione al driver di un client Java, vedere [the section called “Utilizzo di un driver client Java Cassandra”](#).

All'avvio del driver client, viene stabilita innanzitutto la connessione di controllo per le attività amministrative, ad esempio le modifiche allo schema e alla topologia. Quindi vengono create le connessioni aggiuntive.

Nell'esempio seguente, la configurazione del driver delle dimensioni del pool locale è specificata come 2. Se l'endpoint VPC viene creato su 3 sottoreti all'interno del VPC, il risultato è `NewConnections 7` ingressi per l'endpoint dell'interfaccia, come mostrato CloudWatch nella formula seguente.

```
NewConnections = 3 (VPC subnet endpoints created across) * 2 (pool size) + 1
( control connection)
```

```
datastax-java-driver {

    basic.contact-points = [ "cassandra.us-east-1.amazonaws.com:9142"]
    advanced.auth-provider{
        class = PlainTextAuthProvider
        username = "ServiceUserName"
        password = "ServicePassword"
    }
    basic.load-balancing-policy {
        local-datacenter = "us-east-1"
        slow-replica-avoidance = false
    }

    advanced.ssl-engine-factory {
        class = DefaultSslEngineFactory
        truststore-path = "./src/main/resources/cassandra_truststore.jks"
        truststore-password = "my_password"
        hostname-validation = false
    }
    advanced.connection {
        pool.local.size = 2
    }
}
```

Se il numero di connessioni attive non corrisponde alla dimensione del pool configurato (aggregazione tra sottoreti) + 1 connessione di controllo, qualcosa impedisce la creazione delle connessioni.

Node.js

È possibile configurare il numero di connessioni per pool nel driver Node.js. Per un esempio completo di connessione al driver client Node.js, vedere [the section called “Utilizzo di un driver client Cassandra Node.js”](#).

L'esempio di codice riportato di seguito utilizza il *keyspaces-bundle.pem* file per il certificato digitale. Per ulteriori informazioni, consulta [the section called “Prima di iniziare”](#).

Per il seguente esempio di codice, la configurazione del driver della dimensione del pool locale è specificata come 1. Se l'endpoint VPC viene creato su 4 sottoreti all'interno del VPC, si ottengono

NewConnections 5 ingressi per l'endpoint dell'interfaccia, come mostrato CloudWatch nella formula seguente.

```
NewConnections = 4 (VPC subnet endpoints created across) * 1 (pool size) + 1
( control connection)
```

```
const cassandra = require('cassandra-driver');
const fs = require('fs');
const types = cassandra.types;
const auth = new cassandra.auth.PlainTextAuthProvider('ServiceUserName',
  'ServicePassword');
const sslOptions1 = {
  ca: [
    fs.readFileSync('/home/ec2-user/keyspaces-bundle.pem',
  'utf-8')],
  host: 'cassandra.us-east-1.amazonaws.com',
  rejectUnauthorized: true
};
const client = new cassandra.Client({
  contactPoints: ['cassandra.us-east-1.amazonaws.com'],
  localDataCenter: 'us-east-1',
  pooling: { coreConnectionsPerHost: { [types.distance.local]:
1 } },
  consistency: types.consistencies.localQuorum,
  queryOptions: { isIdempotent: true },
  authProvider: auth,
  sslOptions: sslOptions1,
  protocolOptions: { port: 9142 }
});
```

Fase 7: (Facoltativo) Pulizia

Se desideri eliminare le risorse che hai creato in questo tutorial, segui queste procedure.

Per rimuovere l'endpoint VPC per Amazon Keyspaces

1. Accedi alla tua EC2 istanza Amazon.
2. Determina l'ID dell'endpoint VPC utilizzato per Amazon Keyspaces. Se si omettono i grep parametri, vengono visualizzate le informazioni sugli endpoint VPC per tutti i servizi.

```
aws ec2 describe-vpc-endpoint-services | grep ServiceName | grep cassandra

{
  "VpcEndpoint": {
    "PolicyDocument": "{\"Version\":\"2000-00-00\",\"Statement\":[{\"Effect\":\"Allow\",\"Principal\":\"*\",\"Action\":\"*\",\"Resource\":\"*\"}]}\",
    "VpcId": "vpc-a1234bcd",
    "State": "available",
    "ServiceName": "com.amazonaws.us-east-1.cassandra",
    "RouteTableIds": [],
    "VpcEndpointId": "vpce-1a23b4c5",
    "CreationTimestamp": "2025-07-26T22:00:14Z"
  }
}
```

Nell'output di esempio, l'ID endpoint VPC è **vpce-1a23b4c5**. Assicurati di sostituire questo valore con il tuo.

3. Eliminare l'endpoint VPC.

```
aws ec2 delete-vpc-endpoints --vpc-endpoint-ids vpce-1a23b4c5

{
  "Unsuccessful": []
}
```

L'array vuoto [] indica la riuscita dell'operazione (non ci sono state richieste non riuscite).

Per terminare la tua istanza Amazon EC2

1. Apri la EC2 console Amazon all'indirizzo <https://console.aws.amazon.com/ec2/>.
2. Nel riquadro di navigazione, scegliere Instances (Istanze).
3. Scegli la tua EC2 istanza Amazon.
4. Scegli Azioni, scegli Stato dell'istanza, quindi scegli Termina.
5. Nella finestra di conferma scegli Sì, termina.

Tutorial: Integrazione con Apache Spark per importare o esportare dati

Apache Spark è un motore open source per l'analisi dei dati su larga scala. Apache Spark ti consente di eseguire analisi sui dati archiviati in Amazon Keyspaces in modo più efficiente. Puoi anche utilizzare Amazon Keyspaces per fornire alle applicazioni un accesso coerente e in single-digit-millisecond lettura ai dati di analisi di Spark. L'open source Spark Cassandra Connector semplifica la lettura e la scrittura di dati tra Amazon Keyspaces e Spark.

Il supporto di Amazon Keyspaces per Spark Cassandra Connector semplifica l'esecuzione dei carichi di lavoro Cassandra nelle pipeline di analisi basate su Spark utilizzando un servizio di database completamente gestito e senza server. Con Amazon Keyspaces, non devi preoccuparti che Spark possa competere per le stesse risorse infrastrutturali sottostanti delle tue tabelle. Le tabelle di Amazon Keyspaces si ridimensionano automaticamente verso l'alto e verso il basso in base al traffico dell'applicazione.

Il seguente tutorial illustra i passaggi e le best practice necessari per leggere e scrivere dati su Amazon Keyspaces utilizzando il connettore Spark Cassandra. Il tutorial dimostra come migrare i dati su Amazon Keyspaces caricando dati da un file con Spark Cassandra Connector e scrivendoli in una tabella Amazon Keyspaces. Quindi, il tutorial mostra come leggere i dati da Amazon Keyspaces utilizzando il connettore Spark Cassandra. Lo faresti per eseguire i carichi di lavoro di Cassandra in pipeline di analisi basate su Spark.

Argomenti

- [Prerequisiti per stabilire connessioni ad Amazon Keyspaces con Spark Cassandra Connector](#)
- [Passaggio 1: configura Amazon Keyspaces per l'integrazione con il connettore Apache Cassandra Spark](#)
- [Fase 2: Configurazione del connettore Apache Cassandra Spark](#)
- [Fase 3: Creare il file di configurazione dell'applicazione](#)
- [Fase 4: Preparare i dati di origine e la tabella di destinazione in Amazon Keyspaces](#)
- [Passaggio 5: scrivere e leggere i dati di Amazon Keyspaces utilizzando il connettore Apache Cassandra Spark](#)
- [Risoluzione degli errori più comuni durante l'utilizzo del connettore Spark Cassandra con Amazon Keyspaces](#)

Prerequisiti per stabilire connessioni ad Amazon Keyspaces con Spark Cassandra Connector

Prima di connetterti ad Amazon Keyspaces con Spark Cassandra Connector, devi assicurarti di aver installato quanto segue. La compatibilità di Amazon Keyspaces con il connettore Spark Cassandra è stata testata con le seguenti versioni consigliate:

- Java versione 8
- Scala 2.12
- Spark 3.4
- Cassandra Connector 2.5 e versioni successive
- Driver Cassandra 4.12

1. Per installare Scala, segui le istruzioni su <https://www.scala-lang.org/download/scala2.html>
2. Per installare Spark 3.4.1, segui questo esempio.

```
curl -o spark-3.4.1-bin-hadoop3.tgz -k https://d1cdn.apache.org/spark/spark-3.4.1/spark-3.4.1-bin-hadoop3.tgz

# now to untar
tar -zxvf spark-3.4.1-bin-hadoop3.tgz

# set this variable.
export SPARK_HOME=$PWD/spark-3.4.1-bin-hadoop3
````
```

### Passaggio 1: configura Amazon Keyspaces per l'integrazione con il connettore Apache Cassandra Spark

In questo passaggio, confermi che il partizionatore del tuo account è compatibile con Apache Spark Connector e configuri le autorizzazioni IAM richieste. Le seguenti best practice consentono di fornire una capacità sufficiente read/write per la tabella.

1. Verifica che il `Murmur3Partitioner` partizionatore sia il partizionatore predefinito per il tuo account. Questo partizionatore è compatibile con lo Spark Cassandra Connector. Per ulteriori

informazioni sui partizionatori e su come modificarli, consulta [the section called “Lavorare con i partizionatori”](#)

2. Configura le autorizzazioni IAM per Amazon Keyspaces, utilizzando gli endpoint VPC dell'interfaccia, con Apache Spark.

- Assegna read/write l'accesso alla tabella utenti e l'accesso in lettura alle tabelle di sistema, come mostrato nell'esempio di policy IAM riportato di seguito.
- [È necessario compilare la tabella system.peers con gli endpoint VPC di interfaccia disponibili per i client che accedono ad Amazon Keyspaces con Spark tramite endpoint VPC.](#)

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "cassandra:Select",
 "cassandra:Modify"
],
 "Resource": [
 "arn:aws:cassandra:us-east-1:111122223333:/keyspace/mykeyspace/table/mytable",
 "arn:aws:cassandra:us-east-1:111122223333:/keyspace/system*"
]
 },
 {
 "Sid": "ListVPCEndpoints",
 "Effect": "Allow",
 "Action": [
 "ec2:DescribeNetworkInterfaces",
 "ec2:DescribeVpcEndpoints"
],
 "Resource": "*"
 }
]
}
```

3. Prendi in considerazione le seguenti best practice per configurare una capacità di throughput di lettura/scrittura sufficiente per la tabella Amazon Keyspaces per supportare il traffico proveniente dallo Spark Cassandra Connector.

- Inizia a utilizzare la capacità on-demand per testare lo scenario.
- Per ottimizzare il costo del throughput delle tabelle per gli ambienti di produzione, utilizza un limitatore di velocità per il traffico proveniente dal connettore e configura la tabella in modo che utilizzi la capacità assegnata con scalabilità automatica. Per ulteriori informazioni, consulta [the section called “Gestisci la capacità di throughput con la scalabilità automatica”](#).
- È possibile utilizzare un limitatore di velocità fisso fornito con il driver Cassandra. [Nel repository degli esempi sono disponibili alcuni limitatori di velocità personalizzati per Amazon Keyspaces.AWS](#)
- Per ulteriori informazioni sulla gestione della capacità, consulta. [the section called “Configura le modalità di capacità di lettura/scrittura”](#)

## Fase 2: Configurazione del connettore Apache Cassandra Spark

Apache Spark è una piattaforma di calcolo generica che puoi configurare in diversi modi. Per configurare Spark e Spark Cassandra Connector per l'integrazione con Amazon Keyspaces, ti consigliamo di iniziare con le impostazioni di configurazione minime descritte nella sezione seguente, per poi aumentarle in un secondo momento in base al tuo carico di lavoro.

- Crea partizioni Spark di dimensioni inferiori a 8. MBs

In Spark, le partizioni rappresentano un blocco atomico di dati che può essere eseguito in parallelo. Quando scrivi dati su Amazon Keyspaces con Spark Cassandra Connector, più piccola è la partizione Spark, minore è la quantità di record che l'attività scriverà. Se un'attività Spark rileva più errori, fallisce una volta esaurito il numero di tentativi designato. Per evitare di ripetere operazioni di grandi dimensioni e rielaborare una grande quantità di dati, mantieni piccole le dimensioni della partizione Spark.

- Utilizza un numero ridotto di scritture simultanee per esecutore con un numero elevato di tentativi.

Amazon Keyspaces restituisce errori di capacità insufficiente ai driver Cassandra come timeout operativi. Non puoi risolvere i timeout causati da una capacità insufficiente modificando la durata del timeout configurata, perché Spark Cassandra Connector tenta di riprovare le richieste in modo trasparente utilizzando il `MultipleRetryPolicy`. Per garantire che i nuovi tentativi non sovraccarichino il pool di connessioni del driver, utilizza un numero ridotto di scritture simultanee per esecutore con un numero elevato di tentativi. Il seguente frammento di codice ne è un esempio.

```
spark.cassandra.query.retry.count = 500
```

```
spark.cassandra.output.concurrent.writes = 3
```

- Suddividi il throughput totale e distribuiscilo su più sessioni di Cassandra.
- Il Cassandra Spark Connector crea una sessione per ogni esecutore Spark. Pensate a questa sessione come all'unità di scala per determinare il throughput richiesto e il numero di connessioni richieste.
- Quando definite il numero di core per executor e il numero di core per task, iniziate con un valore basso e aumentatelo in base alle esigenze.
- Imposta gli errori delle attività Spark per consentire l'elaborazione in caso di errori temporanei. Dopo aver acquisito familiarità con le caratteristiche e i requisiti di traffico dell'applicazione, ti consigliamo di `spark.task.maxFailures` impostare un valore limitato.
- Ad esempio, la seguente configurazione può gestire due attività simultanee per esecutore, per sessione:

```
spark.executor.instances = configurable -> number of executors for the session.
spark.executor.cores = 2 -> Number of cores per executor.
spark.task.cpus = 1 -> Number of cores per task.
spark.task.maxFailures = -1
```

- Disattiva il raggruppamento in batch.
- Ti consigliamo di disattivare il batch per migliorare i modelli di accesso casuale. Il seguente frammento di codice ne è un esempio.

```
spark.cassandra.output.batch.size.rows = 1 (Default = None)
spark.cassandra.output.batch.grouping.key = none (Default = Partition)
spark.cassandra.output.batch.grouping.buffer.size = 100 (Default = 1000)
```

- **SPARK\_LOCAL\_DIRS** Impostato su un disco locale veloce con spazio sufficiente.
- Per impostazione predefinita, Spark salva i file di output delle mappe e i set di dati distribuiti resilienti (RDDs) in una cartella. `/tmp` A seconda della configurazione dell'host Spark, ciò può comportare la perdita di spazio sugli errori di stile del dispositivo.
- Per impostare la variabile di `SPARK_LOCAL_DIRS` ambiente su una directory chiamata `example/spark-dir`, puoi usare il seguente comando.

```
export SPARK_LOCAL_DIRS=/example/spark-dir
```

## Fase 3: Creare il file di configurazione dell'applicazione

Per utilizzare il connettore open source Spark Cassandra con Amazon Keyspaces, devi fornire un file di configurazione dell'applicazione che contenga le impostazioni necessarie per la connessione con il driver Java. DataStax Puoi utilizzare credenziali specifiche del servizio o il plug-in SigV4 per connetterti.

Se non l'hai già fatto, devi convertire il certificato digitale utilizzato per creare la connessione TLS in un file TrustStore. Puoi seguire i passaggi dettagliati riportati nel tutorial sulla [the section called “Prima di iniziare”](#) connessione dei driver Java. Prendi nota del percorso e della password del file TrustStore perché hai bisogno di queste informazioni quando crei il file di configurazione dell'applicazione.

### Connect con l'autenticazione SigV4

Questa sezione mostra un `application.conf` file di esempio che puoi usare quando ti connetti con AWS le credenziali e il plugin SigV4. Se non l'avete già fatto, dovete generare le vostre chiavi di accesso IAM (un ID della chiave di accesso e una chiave di accesso segreta) e salvarle nel file di AWS configurazione o come variabili di ambiente. Per istruzioni dettagliate, vedi [the section called “Credenziali richieste per l'autenticazione AWS”](#).

Nell'esempio seguente, sostituisci il percorso del file TrustStore e sostituisci la password.

```
datastax-java-driver {
 basic.contact-points = ["cassandra.us-east-1.amazonaws.com:9142"]
 basic.load-balancing-policy {
 class = DefaultLoadBalancingPolicy
 local-datacenter = us-east-1
 slow-replica-avoidance = false
 }
 basic.request {
 consistency = LOCAL_QUORUM
 }
 advanced {
 auth-provider = {
 class = software.aws.mcs.auth.SigV4AuthProvider
 aws-region = us-east-1
 }
 ssl-engine-factory {
 class = DefaultSslEngineFactory
 truststore-path = "path_to_file/cassandra_truststore.jks"
```



```
 truststore-password = "password"
 hostname-validation=false
 }
}
 advanced.connection.pool.local.size = 3
}
```

Aggiorna e salva questo file di configurazione come `/home/user1/application.conf`. I seguenti esempi utilizzano questo percorso.

## Connect con credenziali specifiche del servizio

Questa sezione mostra un `application.conf` file di esempio che puoi utilizzare per connetterti con credenziali specifiche del servizio. Se non l'hai già fatto, devi generare credenziali specifiche per il servizio per Amazon Keyspaces. Per istruzioni dettagliate, vedi [the section called “Crea credenziali specifiche per il servizio”](#).

Nell'esempio seguente, sostituisci `username` e `password` con le tue credenziali. Inoltre, sostituisci il percorso del file TrustStore e sostituisci la password.

```
datastax-java-driver {
 basic.contact-points = ["cassandra.us-east-1.amazonaws.com:9142"]
 basic.load-balancing-policy {
 class = DefaultLoadBalancingPolicy
 local-datacenter = us-east-1
 }
 basic.request {
 consistency = LOCAL_QUORUM
 }
 advanced {
 auth-provider = {
 class = PlainTextAuthProvider
 username = "username"
 password = "password"
 aws-region = "us-east-1"
 }
 ssl-engine-factory {
 class = DefaultSslEngineFactory
 truststore-path = "path_to_file/cassandra_truststore.jks"
 truststore-password = "password"
 hostname-validation=false
 }
 metadata = {
```

```
 schema {
 token-map.enabled = true
 }
 }
}
```

Aggiorna e salva questo file di configurazione `/home/user1/application.conf` da utilizzare con l'esempio di codice.

## Connect con una tariffa fissa

Per imporre una tariffa fissa per ogni esecutore Spark, puoi definire un Request throttler. Questo acceleratore di richieste limita la frequenza delle richieste al secondo. Lo Spark Cassandra Connector implementa una sessione Cassandra per executor. L'utilizzo della formula seguente può aiutarti a ottenere un throughput costante rispetto a una tabella.

```
max-request-per-second * numberOfExecutors = total throughput against a table
```

È possibile aggiungere questo esempio al file di configurazione dell'applicazione creato in precedenza.

```
datastax-java-driver {
 advanced.throttler {
 class = RateLimitingRequestThrottler

 max-requests-per-second = 3000
 max-queue-size = 30000
 drain-interval = 1 millisecond
 }
}
```

## Fase 4: Preparare i dati di origine e la tabella di destinazione in Amazon Keyspaces

In questo passaggio, crei un file sorgente con dati di esempio e una tabella Amazon Keyspaces.

1. Crea il file sorgente. Puoi scegliere una delle seguenti opzioni:
  - Per questo tutorial, si utilizza un file con valori separati da virgole (CSV) con il nome `keyspaces_sample_table.csv` come file di origine per la migrazione dei dati. Il

file di esempio fornito contiene alcune righe di dati per una tabella con lo stesso nome.

`book_awards`

- Scaricate il file CSV di esempio (`keyspaces_sample_table.csv`) contenuto nel seguente file di archivio [samplemigration.zip](#). Decomprimi l'archivio e prendi nota del percorso verso `keyspaces_sample_table.csv`
- Se desideri scrivere dati in Amazon Keyspaces utilizzando il tuo file CSV, assicurati che i dati siano randomizzati. I dati letti direttamente da un database o esportati in file flat vengono in genere ordinati in base alla partizione e alla chiave primaria. L'importazione di dati ordinati in Amazon Keyspaces può causare la scrittura in segmenti più piccoli di partizioni Amazon Keyspaces, con conseguente distribuzione del traffico non uniforme. Ciò può comportare un rallentamento delle prestazioni e tassi di errore più elevati.

Al contrario, la randomizzazione dei dati aiuta a sfruttare le funzionalità di bilanciamento del carico integrate di Amazon Keyspaces distribuendo il traffico tra le partizioni in modo più uniforme. Esistono vari strumenti che puoi utilizzare per la randomizzazione dei dati. Per un esempio che utilizza lo strumento open source [Shuf](#), consulta [the section called “Fase 2: Preparare i dati”](#) il tutorial sulla migrazione dei dati. Di seguito è riportato un esempio che mostra come mescolare i dati in un file. `DataFrame`

```
import org.apache.spark.sql.functions.randval
shuffledDF = dataframe.orderBy(rand())
```

## 2. Crea lo spazio chiave e la tabella di destinazione in Amazon Keyspaces.

- a. Connect ad Amazon Keyspaces utilizzando `cqlsh-expansion` Per le istruzioni `cqlsh-expansion` di installazione, consulta [the section called “Utilizzo di cqlsh-expansion”](#).

Sostituisci l'endpoint del servizio nell'esempio seguente con il tuo valore.

```
cqlsh-expansion cassandra.us-east-1.amazonaws.com 9142 --ssl
```

- b. Create un nuovo keyspace con il nome `catalog` mostrato nell'esempio seguente.

```
CREATE KEYSPACE catalog WITH REPLICATION = {'class': 'SingleRegionStrategy'};
```

- c. Dopo che il nuovo keyspace ha lo stato di disponibile, utilizzate il codice seguente per creare la tabella di destinazione. `book_awards` Per ulteriori informazioni sulla creazione asincrona

di risorse e su come verificare se una risorsa è disponibile, consulta [the section called “Controlla lo stato di creazione del keyspace”](#)

```
CREATE TABLE catalog.book_awards (
 year int,
 award text,
 rank int,
 category text,
 book_title text,
 author text,
 publisher text,
 PRIMARY KEY ((year, award), category, rank)
);
```

## Passaggio 5: scrivere e leggere i dati di Amazon Keyspaces utilizzando il connettore Apache Cassandra Spark

In questo passaggio, inizierai caricando i dati dal file di esempio in un file DataFrame con lo Spark Cassandra Connector. Successivamente, scrivi i dati da DataFrame nella tabella Amazon Keyspaces. Puoi anche utilizzare questa parte in modo indipendente, ad esempio per migrare i dati in una tabella Amazon Keyspaces. Infine, leggi i dati dalla tua tabella in una DataFrame utilizzando lo Spark Cassandra Connector. Puoi anche utilizzare questa parte in modo indipendente, ad esempio per leggere i dati da una tabella Amazon Keyspaces per eseguire analisi dei dati con Apache Spark.

1. Avvia Spark Shell come mostrato nell'esempio seguente. Nota che questo esempio utilizza l'autenticazione SigV4.

```
./spark-shell --files application.conf --conf
spark.cassandra.connection.config.profile.path=application.conf
--packages software.aws.mcs:aws-sigv4-auth-cassandra-java-driver-
plugin:4.0.5,com.datastax.spark:spark-cassandra-connector_2.12:3.1.0 --conf
spark.sql.extensions=com.datastax.spark.connector.CassandraSparkExtensions
```

2. Importa lo Spark Cassandra Connector con il seguente codice.

```
import org.apache.spark.sql.cassandra._
```

3. Per leggere i dati dal file CSV e archivarli in un fileDataFrame, puoi utilizzare il seguente esempio di codice.

```
var df =
 spark.read.option("header","true").option("inferSchema","true").csv("keyspaces_sample_tabl
```

È possibile visualizzare il risultato con il seguente comando.

```
scala> df.show();
```

L'output dovrebbe essere simile a questo.

```
+-----+-----+-----+-----+-----+-----+
+-----+
| award|year| category|rank| author| book_title|
|publisher|
+-----+-----+-----+-----+-----+-----+
+-----+
|Kwesi Manu Prize|2020| Fiction| 1| Akua Mansa| Where did you go?|
SomePublisher|
|Kwesi Manu Prize|2020| Fiction| 2| John Stiles| Yesterday|
Example Books|
|Kwesi Manu Prize|2020| Fiction| 3| Nikki Wolf|Moving to the Cha...|
AnyPublisher|
| Wolf|2020|Non-Fiction| 1| Wang Xiulan| History of Ideas|
Example Books|
| Wolf|2020|Non-Fiction| 2|Ana Carolina Silva| Science Today|
SomePublisher|
| Wolf|2020|Non-Fiction| 3| Shirley Rodriguez|The Future of Sea...|
AnyPublisher|
| Richard Roe|2020| Fiction| 1|Alejandro Rosalez| Long Summer|
SomePublisher|
| Richard Roe|2020| Fiction| 2| Arnav Desai| The Key|
Example Books|
| Richard Roe|2020| Fiction| 3| Mateo Jackson| Inside the Whale|
AnyPublisher|
+-----+-----+-----+-----+-----+-----+
+-----+
```

È possibile confermare lo schema dei dati nel DataFrame modo illustrato nell'esempio seguente.

```
scala> df.printSchema
```

L'output dovrebbe essere simile a questo.

```
root
|-- award: string (nullable = true)
|-- year: integer (nullable = true)
|-- category: string (nullable = true)
|-- rank: integer (nullable = true)
|-- author: string (nullable = true)
|-- book_title: string (nullable = true)
|-- publisher: string (nullable = true)
```

4. Usa il comando seguente per scrivere i dati nella DataFrame tabella Amazon Keyspaces.

```
df.write.cassandraFormat("book_awards", "catalog").mode("APPEND").save()
```

5. Per confermare che i dati sono stati salvati, puoi rileggerli in un dataframe, come mostrato nell'esempio seguente.

```
var newDf = spark.read.cassandraFormat("book_awards", "catalog").load()
```

Quindi puoi mostrare i dati che ora sono contenuti nel dataframe.

```
scala> newDf.show()
```

L'output di quel comando dovrebbe assomigliare a questo.

```
+-----+-----+-----+-----+
+----+----+
| book_title| author| award| category|
|publisher|rank|year|
+-----+-----+-----+-----+
+----+----+
| Long Summer| Alejandro Rosalez| Richard Roe| Fiction|
SomePublisher| 1|2020|
| History of Ideas| Wang Xiulan| Wolf|Non-Fiction|Example
Books| 1|2020|
| Where did you go?| Akua Mansa|Kwesi Manu Prize| Fiction|
SomePublisher| 1|2020|
| Inside the Whale| Mateo Jackson| Richard Roe| Fiction|
AnyPublisher| 3|2020|
```

```

| Yesterday| John Stiles|Kwesi Manu Prize| Fiction|Example
Books| 2|2020|
|Moving to the Cha...| Nikki Wolf|Kwesi Manu Prize| Fiction|
AnyPublisher| 3|2020|
|The Future of Sea...| Shirley Rodriguez| Wolf|Non-Fiction|
AnyPublisher| 3|2020|
| Science Today|Ana Carolina Silva| Wolf|Non-Fiction|
SomePublisher| 2|2020|
| The Key| Arnav Desai| Richard Roe| Fiction|Example
Books| 2|2020|
+-----+-----+-----+-----+-----+
+----+----+

```

## Risoluzione degli errori più comuni durante l'utilizzo del connettore Spark Cassandra con Amazon Keyspaces

Se utilizzi Amazon Virtual Private Cloud e ti connetti ad Amazon Keyspaces, gli errori più comuni riscontrati durante l'utilizzo del connettore Spark sono causati dai seguenti problemi di configurazione.

- L'utente o il ruolo IAM utilizzato nel VPC non dispone delle autorizzazioni necessarie per accedere alla tabella `system.peers` in Amazon Keyspaces. Per ulteriori informazioni, consulta [the section called “Inserimento delle voci della `system.peers` tabella con informazioni sugli endpoint VPC dell'interfaccia”](#).
- L'utente o il ruolo IAM non dispone delle read/write autorizzazioni richieste per la tabella degli utenti e l'accesso in lettura alle tabelle di sistema in Amazon Keyspaces. Per ulteriori informazioni, consulta [the section called “Fase 1: Configurazione di Amazon Keyspaces”](#).
- La configurazione del driver Java non disabilita la verifica del nome host durante la creazione della connessione SSL/TLS. Per alcuni esempi, consulta [the section called “Fase 2: Configurare il driver”](#).

Per una procedura dettagliata per la risoluzione dei problemi di connessione, consulta. [the section called “Errori di connessione agli endpoint VPC”](#)

Inoltre, puoi utilizzare i CloudWatch parametri di Amazon per aiutarti a risolvere i problemi relativi alla configurazione del connettore Spark Cassandra in Amazon Keyspaces. Per ulteriori informazioni sull'utilizzo di Amazon Keyspaces con CloudWatch, consulta. [the section called “Monitoraggio con CloudWatch”](#)

La sezione seguente descrive le metriche più utili da osservare quando utilizzi lo Spark Cassandra Connector.

### PerConnectionRequestRateExceeded

Amazon Keyspaces ha una quota di 3.000 richieste al secondo, per connessione. Ogni esecutore Spark stabilisce una connessione con Amazon Keyspaces. L'esecuzione di più tentativi può esaurire la quota di richieste per connessione. Se superi questa quota, Amazon Keyspaces emette un `PerConnectionRequestRateExceeded` parametro in CloudWatch.

Se vedi `PerConnectionRequestRateExceeded` eventi presenti insieme ad altri errori di sistema o utente, è probabile che Spark stia eseguendo più tentativi oltre il numero di richieste assegnato per connessione.

Se vedi `PerConnectionRequestRateExceeded` eventi senza altri errori, potrebbe essere necessario aumentare il numero di connessioni nelle impostazioni del driver per consentire una maggiore velocità di trasmissione, oppure potresti dover aumentare il numero di executor nel tuo job Spark.

### StoragePartitionThroughputCapacityExceeded

Amazon Keyspaces ha una quota di 1.000 WCUs o WRUs al secondo/3.000 RCUs o RRUs al secondo, per partizione. Se visualizzi `StoragePartitionThroughputCapacityExceeded` CloudWatch eventi, ciò potrebbe indicare che i dati non sono randomizzati durante il caricamento. Per esempi su come mescolare i dati, vedi [the section called “Fase 4: Preparare i dati di origine e la tabella di destinazione”](#)

## Errori e avvertenze comuni

Se utilizzi Amazon Virtual Private Cloud e ti connetti ad Amazon Keyspaces, il driver Cassandra potrebbe emettere un messaggio di avviso relativo al nodo di controllo stesso nella tabella `system.peers`. Per ulteriori informazioni, consulta [the section called “Errori e avvertenze comuni”](#). Puoi tranquillamente ignorare questo avviso.



# Tutorial: Connect da un'applicazione containerizzata ospitata su Amazon Elastic Kubernetes Service

Questo tutorial illustra i passaggi necessari per configurare un cluster Amazon Elastic Kubernetes Service (Amazon EKS) per ospitare un'applicazione containerizzata che si connette ad Amazon Keyspaces utilizzando l'autenticazione SigV4.

Amazon EKS è un servizio gestito che elimina la necessità di installare, utilizzare e mantenere il proprio piano di controllo Kubernetes. [Kubernetes](#) è un sistema open source che consente di automatizzare la gestione, il dimensionamento e l'implementazione di applicazioni distribuite in container.

Il tutorial fornisce step-by-step indicazioni per configurare, creare e distribuire un'applicazione Java containerizzata su Amazon EKS. Nell'ultimo passaggio esegui l'applicazione per scrivere dati su una tabella Amazon Keyspaces.

## Argomenti

- [Prerequisiti per la connessione da Amazon EKS ad Amazon Keyspaces](#)
- [Fase 1: configurare il cluster Amazon EKS e configurare le autorizzazioni IAM](#)
- [Passaggio 2: configura l'applicazione](#)
- [Passaggio 3: crea l'immagine dell'applicazione e carica il file Docker nel tuo repository Amazon ECR](#)
- [Fase 4: Implementa l'applicazione su Amazon EKS e scrivi i dati nella tabella](#)
- [Fase 5: Pulizia \(facoltativa\)](#)

## Prerequisiti per la connessione da Amazon EKS ad Amazon Keyspaces

Crea le seguenti AWS risorse prima di iniziare con il tutorial

1. Prima di iniziare questo tutorial, segui le istruzioni di AWS configurazione contenute in [Accesso ad Amazon Keyspaces \(per Apache Cassandra\)](#). Questi passaggi includono la registrazione AWS e la creazione di un principale AWS Identity and Access Management (IAM) con accesso ad Amazon Keyspaces.
2. Crea uno spazio di chiavi Amazon Keyspaces con il nome `aws` e una tabella con il nome `su user` cui scrivere dall'applicazione containerizzata in esecuzione in Amazon EKS più avanti in questo tutorial. Puoi farlo con o utilizzando AWS CLI `cqlsh`

## AWS CLI

```
aws keyspaces create-keyspace --keyspace-name 'aws'
```

Per confermare che il keyspace è stato creato, puoi usare il seguente comando.

```
aws keyspaces list-keyspaces
```

Per creare la tabella, puoi usare il seguente comando.

```
aws keyspaces create-table --keyspace-name 'aws' --table-name 'user' --schema-
definition 'allColumns=[
 {name=username,type=text}, {name=fname,type=text},
{name=last_update_date,type=timestamp},{name=lname,type=text}],
partitionKeys=[{name=username}]'
```

Per confermare che la tabella è stata creata, è possibile utilizzare il seguente comando.

```
aws keyspaces list-tables --keyspace-name 'aws'
```

Per ulteriori informazioni, consulta [creare keyspace](#) e [creare tabella](#) nel AWS CLI Command Reference.

## cqlsh

```
CREATE KEYSPACE aws WITH replication = {'class': 'SimpleStrategy',
 'replication_factor': '3'} AND durable_writes = true;
CREATE TABLE aws.user (
 username text PRIMARY KEY,
 fname text,
 last_update_date timestamp,
 lname text
);
```

Per verificare che la tabella sia stata creata, è possibile utilizzare la seguente dichiarazione.

```
SELECT * FROM system_schema.tables WHERE keyspace_name = "aws";
```

La tua tabella dovrebbe essere elencata nell'output di questa dichiarazione. Nota che può esserci un ritardo nella creazione della tabella. Per ulteriori informazioni, consulta [the section called “CREATE TABLE”](#).

3. Crea un cluster Amazon EKS con un tipo di nodo Fargate - Linux. Fargate è un motore di elaborazione serverless che consente di distribuire Kubernetes Pods senza gestire le istanze Amazon Amazon. EC2 Per seguire questo tutorial senza dover aggiornare il nome del cluster in tutti i comandi di esempio, crea un cluster con lo stesso nome `my-eks-cluster` seguendo le istruzioni in [Getting started with Amazon EKS](#), `eksctl` nella Amazon EKS User Guide. Una volta creato il cluster, verifica che i nodi e i due Pod predefiniti siano funzionanti e integri. Puoi farlo con il seguente comando.

```
kubectl get pods -A -o wide
```

Dovresti vedere qualcosa di simile a questo risultato.

| NAMESPACE   | NAME                                              | READY | STATUS  | RESTARTS  | AGE    | IP        |
|-------------|---------------------------------------------------|-------|---------|-----------|--------|-----------|
| NODE        |                                                   |       |         | NOMINATED | NODE   | READINESS |
| GATES       |                                                   |       |         |           |        |           |
| kube-system | coredns-1234567890-abcde                          | 1/1   | Running | 0         | 18m    |           |
| 192.0.2.0   | fargate-ip-192-0-2-0.region-code.compute.internal |       |         |           | <none> |           |
| <none>      |                                                   |       |         |           |        |           |
| kube-system | coredns-1234567890-12345                          | 1/1   | Running | 0         | 18m    |           |
| 192.0.2.1   | fargate-ip-192-0-2-1.region-code.compute.internal |       |         |           | <none> |           |
| <none>      |                                                   |       |         |           |        |           |

4. Installazione di Docker. Per istruzioni su come installare Docker su un' EC2 istanza Amazon, consulta [Install Docker](#) nella Amazon Elastic Container Registry User Guide.

Docker è disponibile per diversi sistemi operativi, compresa la maggior parte delle distribuzioni Linux, ad esempio Ubuntu, e persino per macOS e Windows. Per ulteriori informazioni sull'installazione di Docker sul tuo specifico sistema operativo, consulta la [guida all'installazione di Docker](#).

5. Crea un repository Amazon ECR. Amazon ECR è un servizio di registro delle immagini dei container AWS gestito che puoi utilizzare con la tua CLI preferita per inviare, estrarre e gestire immagini Docker. Per ulteriori informazioni sui repository Amazon ECR, consulta la [Amazon Elastic Container Registry User Guide](#). Puoi utilizzare il seguente comando per creare un repository con il nome `my-ecr-repository`

```
aws ecr create-repository --repository-name my-ecr-repository
```

Dopo aver completato i passaggi preliminari, procedere a [the section called “Fase 1: configurare il cluster Amazon EKS”](#)

## Fase 1: configurare il cluster Amazon EKS e configurare le autorizzazioni IAM

Configura il cluster Amazon EKS e crea le risorse IAM necessarie per consentire a un account di servizio Amazon EKS di connettersi alla tabella Amazon Keyspaces.

1. Crea un provider Open ID Connect (OIDC) per il cluster Amazon EKS. Ciò è necessario per utilizzare i ruoli IAM per gli account di servizio. Per ulteriori informazioni sui provider OIDC e su come crearli, consulta [Creating an IAM OIDC provider for your cluster nella](#) Amazon EKS User Guide.

- a. Creare un provider di identità OIDC IAM per il cluster con il comando seguente. Questo esempio presuppone che il nome del cluster sia `my-eks-cluster`. Se hai un cluster con un nome diverso, ricordati di aggiornare il nome in tutti i comandi futuri.

```
eksctl utils associate-iam-oidc-provider --cluster my-eks-cluster --approve
```

- b. Conferma che il provider di identità OIDC sia stato registrato con IAM con il seguente comando.

```
aws iam list-open-id-connect-providers --region us-east-1
```

L'output dovrebbe essere simile a questo. Prendi nota dell'Amazon Resource Name (ARN) dell'OIDC, ne avrai bisogno nella fase successiva quando crei una policy di fiducia per l'account di servizio.

```
{
 "OpenIDConnectProviderList": [
 ..
 {
 "Arn": "arn:aws:iam::111122223333:oidc-provider/oidc.eks.us-
east-1.amazonaws.com/id/EXAMPLED539D4633E53DE1B71EXAMPLE"
```

```
 }
]
}
```

2. Crea un account di servizio per il cluster Amazon EKS. Gli account di servizio forniscono un'identità per i processi eseguiti in un Pod. Un Pod è l'oggetto Kubernetes più piccolo e semplice che puoi utilizzare per distribuire un'applicazione containerizzata. Successivamente, crea un ruolo IAM che l'account di servizio possa assumere per ottenere le autorizzazioni per le risorse. Puoi accedere a qualsiasi AWS servizio da un Pod che è stato configurato per utilizzare un account di servizio che può assumere un ruolo IAM con autorizzazioni di accesso a quel servizio.
  - a. Crea un nuovo spazio dei nomi per l'account del servizio. Un namespace aiuta a isolare le risorse del cluster create per questo tutorial. È possibile creare un nuovo spazio dei nomi utilizzando il seguente comando.

```
kubectl create namespace my-eks-namespace
```

- b. Per utilizzare uno spazio dei nomi personalizzato, è necessario associarlo a un profilo Fargate. Il codice seguente ne è un esempio.

```
eksctl create fargateprofile \
 --cluster my-eks-cluster \
 --name my-fargate-profile \
 --namespace my-eks-namespace \
 --labels *=*
```

- c. Crea un account di servizio con il nome *my-eks-serviceaccount* nello spazio dei nomi *my-eks-namespace* per il tuo cluster Amazon EKS utilizzando il seguente comando.

```
cat >my-serviceaccount.yaml <<EOF
apiVersion: v1
kind: ServiceAccount
metadata:
 name: my-eks-serviceaccount
 namespace: my-eks-namespace
EOF
kubectl apply -f my-serviceaccount.yaml
```

- d. Esegui il comando seguente per creare un file di policy di fiducia che indichi al ruolo IAM di considerare attendibile il tuo account di servizio. Questa relazione di fiducia è necessaria

prima che un preside possa assumere un ruolo. È necessario apportare le seguenti modifiche al file:

- Per laPrincipal, inserisci l'ARN che IAM ha restituito al `list-open-id-connect-providers` comando. L'ARN contiene il numero di conto e la regione.
- Nella condition dichiarazione, sostituisci l'id Regione AWS e l'ID OIDC.
- Verificate che il nome e lo spazio dei nomi dell'account del servizio siano corretti.

È necessario allegare il file delle policy di fiducia nel passaggio successivo quando si crea il ruolo IAM.

```
cat >trust-relationship.json <<EOF
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Principal": {
 "Federated": "arn:aws:iam::111122223333:oidc-provider/
oidc.eks.us-east-1.amazonaws.com/id/EXAMPLED539D4633E53DE1B71EXAMPLE"
 },
 "Action": "sts:AssumeRoleWithWebIdentity",
 "Condition": {
 "StringEquals": {
 "oidc.eks.us-east-1.amazonaws.com/
id/EXAMPLED539D4633E53DE1B71EXAMPLE:sub": "system:serviceaccount:my-eks-namespace:my-eks-serviceaccount",
 "oidc.eks.us-east-1.amazonaws.com/
id/EXAMPLED539D4633E53DE1B71EXAMPLE:aud": "sts.amazonaws.com"
 }
 }
 }
]
}
EOF
```

Facoltativo: puoi anche aggiungere più voci nelle StringLike condizioni StringEquals or per consentire a più account di servizio o namespace di assumere il ruolo. Per consentire al tuo account di servizio di assumere un ruolo IAM in un altro AWS account, consulta [Autorizzazioni IAM per più account](#) nella Amazon EKS User Guide.

3. Crea un ruolo IAM con il nome da assumere `my-iam-role` per l'account del servizio Amazon EKS. Allega al ruolo il file di policy di fiducia creato nell'ultimo passaggio. La policy di fiducia specifica l'account di servizio e il provider OIDC di cui il ruolo IAM può fidarsi.

```
aws iam create-role --role-name my-iam-role --assume-role-policy-document file://trust-relationship.json --description "EKS service account role"
```

4. Assegna le autorizzazioni del ruolo IAM ad Amazon Keyspaces allegando una policy di accesso.
  - a. Allega una policy di accesso per definire le azioni che il ruolo IAM può eseguire su risorse Amazon Keyspaces specifiche. Per questo tutorial utilizziamo la policy AWS gestita `AmazonKeyspacesFullAccess`, perché la nostra applicazione scriverà dati nella tua tabella Amazon Keyspaces. Come best practice, tuttavia, si consiglia di creare politiche di accesso personalizzate che implementino il principio dei privilegi minimi. Per ulteriori informazioni, consulta [the section called “Come funziona Amazon Keyspaces con IAM”](#).

```
aws iam attach-role-policy --role-name my-iam-role --policy-arn=arn:aws:iam::aws:policy/AmazonKeyspacesFullAccess
```

Conferma che la policy è stata associata correttamente al ruolo IAM con la seguente dichiarazione.

```
aws iam list-attached-role-policies --role-name my-iam-role
```

L'output dovrebbe essere simile a questo.

```
{
 "AttachedPolicies": [
 {
 "PolicyName": "AmazonKeyspacesFullAccess",
 "PolicyArn": "arn:aws:iam::aws:policy/AmazonKeyspacesFullAccess"
 }
]
}
```

- b. Annota l'account del servizio con l'Amazon Resource Name (ARN) del ruolo IAM che può assumere. Assicurati di aggiornare l'ARN del ruolo con l'ID del tuo account.

```
kubectl annotate serviceaccount -n my-eks-namespace my-eks-serviceaccount
eks.amazonaws.com/role-arn=arn:aws:iam::111122223333:role/my-iam-role
```

5. Verifica che il ruolo IAM e l'account di servizio siano configurati correttamente.
  - a. Verifica che la policy di fiducia del ruolo IAM sia configurata correttamente con la seguente dichiarazione.

```
aws iam get-role --role-name my-iam-role --query Role.AssumeRolePolicyDocument
```

L'output dovrebbe essere simile a questo.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Principal": {
 "Federated": "arn:aws:iam::111122223333:oidc-provider/
oidc.eks.us-east-1.amazonaws.com/id/EXAMPLED539D4633E53DE1B71EXAMPLE"
 },
 "Action": "sts:AssumeRoleWithWebIdentity",
 "Condition": {
 "StringEquals": {
 "oidc.eks.us-east-1/id/
EXAMPLED539D4633E53DE1B71EXAMPLE:aud": "sts.amazonaws.com",
 "oidc.eks.us-east-1.amazonaws.com/id/
EXAMPLED539D4633E53DE1B71EXAMPLE:sub": "system:serviceaccount:my-eks-
namespace:my-eks-serviceaccount"
 }
 }
 }
]
}
```

- b. Verifica che l'account del servizio Amazon EKS sia annotato con il ruolo IAM.

```
kubectl describe serviceaccount my-eks-serviceaccount -n my-eks-namespace
```

L'output dovrebbe essere simile a questo.



```
Name: my-eks-serviceaccount
Namespace:my-eks-namespace
Labels: <none>
Annotations: eks.amazonaws.com/role-arn: arn:aws:iam::111122223333:role/my-iam-
role
Image pull secrets: <none>
Mountable secrets: <none>
Tokens: <none>
[...]
```

Dopo aver creato l'account di servizio Amazon EKS, il ruolo IAM e configurato le relazioni e le autorizzazioni richieste, procedi [the section called “Fase 2: Configurare l'applicazione”](#) con.

## Passaggio 2: configura l'applicazione

In questa fase crei la tua applicazione che si connette ad Amazon Keyspaces utilizzando il plugin SigV4. [Puoi visualizzare e scaricare l'applicazione Java di esempio dal repository di codice di esempio Amazon Keyspaces su Github](#). Oppure puoi continuare utilizzando la tua applicazione, assicurandoti di completare tutti i passaggi di configurazione.

Configura la tua applicazione e aggiungi le dipendenze richieste.

1. È possibile scaricare l'applicazione Java di esempio clonando il repository Github utilizzando il seguente comando.

```
git clone https://github.com/aws-samples/amazon-keyspaces-examples.git
```

2. Dopo aver scaricato il repository Github, decomprimi il file scaricato e vai alla directory del file.  
`resources application.conf`

### a. Configurazione dell'applicazione

In questo passaggio si configura il plugin di autenticazione SigV4. È possibile utilizzare il seguente esempio nella propria applicazione. Se non l'avete già fatto, dovete generare le vostre chiavi di accesso IAM (un ID di chiave di accesso e una chiave di accesso segreta) e salvarle nel file di AWS configurazione o come variabili di ambiente. Per istruzioni dettagliate, vedi [the section called “Credenziali richieste per l'autenticazione AWS”](#). Aggiorna la AWS regione e l'endpoint di servizio per Amazon Keyspaces in base alle esigenze. Per ulteriori endpoint di servizio, consulta [the section called “Endpoint di servizio”](#)

Sostituisci la posizione del truststore, il nome del truststore e la password del truststore con i tuoi.

```
datastax-java-driver {
 basic.contact-points = ["cassandra.us-east-1.amazonaws.com:9142"]
 basic.load-balancing-policy.local-datacenter = "us-east-1"
 advanced.auth-provider {
 class = software.aws.mcs.auth.SigV4AuthProvider
 aws-region = "us-east-1"
 }
 advanced.ssl-engine-factory {
 class = DefaultSslEngineFactory
 truststore-path = "truststore_locationtruststore_name.jks"
 truststore-password = "truststore_password;"
 }
}
```

- b. Aggiungi la dipendenza del modulo STS.

Ciò aggiunge la possibilità di utilizzare un `WebIdentityTokenCredentialsProvider` che restituisce le AWS credenziali che l'applicazione deve fornire in modo che l'account del servizio possa assumere il ruolo IAM. È possibile eseguire questa operazione in base all'esempio seguente.

```
<dependency>
 <groupId>com.amazonaws</groupId>
 <artifactId>aws-java-sdk-sts</artifactId>
 <version>1.11.717</version>
</dependency>
```

- c. Aggiungi la dipendenza SigV4.

Questo pacchetto implementa il plug-in di autenticazione SigV4 necessario per l'autenticazione su Amazon Keyspaces.

```
<dependency>
 <groupId>software.aws.mcs</groupId>
 <artifactId>aws-sigv4-auth-cassandra-java-driver-plugin</
artifactId>
 <version>4.0.3</version>
</dependency>
```

### 3. Aggiungi una dipendenza per la registrazione.

Senza registri, la risoluzione dei problemi di connessione è impossibile. In questo tutorial, utilizziamo `slf4j` come framework di registrazione e lo utilizziamo `logback.xml` per memorizzare l'output del registro. Abbiamo impostato il livello di registrazione su `per debug` stabilire la connessione. È possibile utilizzare l'esempio seguente per aggiungere la dipendenza.

```
<dependency>
 <groupId>org.slf4j</groupId>
 <artifactId>slf4j-api</artifactId>
 <version>2.0.5</version>
</dependency>
```

È possibile utilizzare il seguente frammento di codice per configurare la registrazione.

```
<configuration>
 <appender name="STDOUT" class="ch.qos.logback.core.ConsoleAppender">

 <encoder>
 <pattern>%d{HH:mm:ss.SSS} [%thread] %-5level %logger{36} - %msg%n</
pattern>
 </encoder>
 </appender>

 <root level="debug">
 <appender-ref ref="STDOUT" />
 </root>
</configuration>
```

#### Note

Il debug livello è necessario per esaminare gli errori di connessione. Dopo esserti connesso correttamente ad Amazon Keyspaces dalla tua applicazione, puoi modificare il livello di registrazione in base alle tue esigenze. `warning`

## Passaggio 3: crea l'immagine dell'applicazione e carica il file Docker nel tuo repository Amazon ECR

In questo passaggio, compila l'applicazione di esempio, crea un'immagine Docker e invia l'immagine al tuo repository Amazon ECR.

Crea la tua applicazione, crea un'immagine Docker e inviala ad Amazon Elastic Container Registry

1. Imposta le variabili di ambiente per la build che definisce la tua Regione AWS. Sostituisci le regioni negli esempi con le tue.

```
export CASSANDRA_HOST=cassandra.us-east-1.amazonaws.com:9142
export CASSANDRA_DC=us-east-1
```

2. Compila la tua applicazione con Apache Maven versione 3.6.3 o successiva usando il seguente comando.

```
mvn clean install
```

Questo crea un JAR file con tutte le dipendenze incluse nella directory. target

3. Recupera l'URI del repository ECR necessario per il passaggio successivo con il seguente comando. Assicurati di aggiornare la regione con quella che hai utilizzato.

```
aws ecr describe-repositories --region us-east-1
```

L'output dovrebbe essere simile a quello dell'esempio seguente.

```
"repositories": [
 {
 "repositoryArn": "arn:aws:ecr:us-east-1:111122223333:repository/my-ecr-
repository",
 "registryId": "111122223333",
 "repositoryName": "my-ecr-repository",
 "repositoryUri": "111122223333.dkr.ecr.us-east-1.amazonaws.com/my-ecr-repository",
 "createdAt": "2023-11-02T03:46:34+00:00",
 "imageTagMutability": "MUTABLE",
 "imageScanningConfiguration": {
 "scanOnPush": false
 },
 "encryptionConfiguration": {
```

```
"encryptionType": "AES256"
}
},
```

4. Dalla directory principale dell'applicazione, crea l'immagine Docker utilizzando l'URI del repository dell'ultimo passaggio. Modifica il file Docker secondo necessità. Nel comando build, assicurati di sostituire l'ID dell'account e di Regione AWS impostarlo sulla regione in cui si trova il repository `my-ecr-repository` Amazon ECR.

```
docker build -t 111122223333.dkr.ecr.us-east-1.amazonaws.com/my-ecr-
repository:latest .
```

5. Recupera un token di autenticazione per inviare l'immagine Docker ad Amazon ECR. Puoi farlo con il seguente comando.

```
aws ecr get-login-password --region us-east-1 | docker login --username AWS --
password-stdin 111122223333.dkr.ecr.us-east-1.amazonaws.com
```

6. Innanzitutto, verifica la presenza di immagini esistenti nel tuo repository Amazon ECR. Utilizzare il seguente comando.

```
aws ecr describe-images --repository-name my-ecr-repository --region us-east-1
```

Quindi, invia l'immagine Docker al repository. Utilizzare il seguente comando.

```
docker push 111122223333.dkr.ecr.us-east-1.amazonaws.com/my-ecr-repository:latest
```

## Fase 4: Implementa l'applicazione su Amazon EKS e scrivi i dati nella tabella

In questa fase del tutorial, configuri la distribuzione di Amazon EKS per la tua applicazione e confermi che l'applicazione è in esecuzione e può connettersi ad Amazon Keyspaces.

Per distribuire un'applicazione su Amazon EKS, devi configurare tutte le impostazioni pertinenti in un file chiamato `deployment.yaml`. Questo file viene quindi utilizzato da Amazon EKS per distribuire l'applicazione. I metadati del file devono contenere le seguenti informazioni:

- Nome dell'applicazione: il nome dell'applicazione. Per questo tutorial, usiamo `my-keyspaces-app`.

- Namespace Kubernetes, lo spazio dei nomi del cluster Amazon EKS. Per questo tutorial, utilizziamo `my-eks-namespace`
- Account del servizio Amazon EKS: nome dell'account del servizio Amazon EKS. Per questo tutorial, utilizziamo `my-eks-serviceaccount`.
- image name il nome dell'immagine dell'applicazione. Per questo tutorial, usiamo `my-keyspaces-app`.
- URI dell'immagine: l'URI dell'immagine Docker di Amazon ECR.
- AWS ID account: l'ID del tuo AWS account.
- Ruolo IAM ARN l'ARN del ruolo IAM creato per l'account di servizio da assumere. Per questo tutorial, utilizziamo `my-iam-role`
- Regione AWS del cluster Amazon EKS in Regione AWS cui hai creato il cluster Amazon EKS.

In questa fase, distribuisce ed esegue l'applicazione che si connette ad Amazon Keyspaces e scrive i dati nella tabella.

1. Configura il file `deployment.yaml`. È necessario sostituire i seguenti valori:

- `name`
- `namespace`
- `serviceAccountName`
- `image`
- `AWS_ROLE_ARN` value
- Il Regione AWS pin `CASSANDRA_HOST`
- `AWS_REGION`

È possibile utilizzare il seguente file come esempio.

```
apiVersion: apps/v1
kind: Deployment
metadata:
 name: my-keyspaces-app
 namespace: my-eks-namespace
spec:
 replicas: 1
 selector:
```

```

matchLabels:
 app: my-keyspaces-app
template:
 metadata:
 labels:
 app: my-keyspaces-app
 spec:
 serviceAccountName: my-eks-serviceaccount
 containers:
 - name: my-keyspaces-app
 image: 111122223333.dkr.ecr.us-east-1.amazonaws.com/my-ecr-repository:latest
 ports:
 - containerPort: 8080
 env:
 - name: CASSANDRA_HOST
 value: "cassandra.us-east-1.amazonaws.com:9142"
 - name: CASSANDRA_DC
 value: "us-east-1"
 - name: AWS_WEB_IDENTITY_TOKEN_FILE
 value: /var/run/secrets/eks.amazonaws.com/serviceaccount/token
 - name: AWS_ROLE_ARN
 value: "arn:aws:iam::111122223333:role/my-iam-role"
 - name: AWS_REGION
 value: "us-east-1"

```

## 2. Implementa deployment.yaml.

```
kubectl apply -f deployment.yaml
```

L'output dovrebbe essere simile a questo.

```
deployment.apps/my-keyspaces-app created
```

## 3. Verifica lo stato del Pod nel tuo namespace del cluster Amazon EKS.

```
kubectl get pods -n my-eks-namespace
```

L'output dovrebbe essere simile a quello di questo esempio.

| NAME | READY | STATUS | RESTARTS | AGE |
|------|-------|--------|----------|-----|
|------|-------|--------|----------|-----|

```
my-keyspaces-app-123abcde4f-g5hij 1/1 Running 0 75s
```

Per ulteriori dettagli, è possibile utilizzare il seguente comando.

```
kubectl describe pod my-keyspaces-app-123abcde4f-g5hij -n my-eks-namespace
```

```
Name: my-keyspaces-app-123abcde4f-g5hij
Namespace: my-eks-namespace
Priority: 2000001000
Priority Class Name: system-node-critical
Service Account: my-eks-serviceaccount
Node: fargate-ip-192-168-102-209.ec2.internal/192.168.102.209
Start Time: Thu, 23 Nov 2023 12:15:43 +0000
Labels: app=my-keyspaces-app
 eks.amazonaws.com/fargate-profile=my-fargate-profile
 pod-template-hash=6c56fccc56
Annotations: CapacityProvisioned: 0.25vCPU 0.5GB
 Logging: LoggingDisabled: LOGGING_CONFIGMAP_NOT_FOUND
Status: Running
IP: 192.168.102.209
IPs:
 IP: 192.168.102.209
Controlled By: ReplicaSet/my-keyspaces-app-6c56fccc56
Containers:
 my-keyspaces-app:
 Container ID: containerd://41ff7811d33ae4bc398755800abdc132335d51d74f218ba81da0700a6f8c67b
 Image: 111122223333.dkr.ecr.us-east-1.amazonaws.com/
my_eks_repository:latest
 Image ID: 111122223333.dkr.ecr.us-east-1.amazonaws.com/
my_eks_repository@sha256:fd3c6430fc5251661efce99741c72c1b4b03061474940200d0524b84a951439c
 Port: 8080/TCP
 Host Port: 0/TCP
 State: Running
 Started: Thu, 23 Nov 2023 12:15:19 +0000
 Finished: Thu, 23 Nov 2023 12:16:17 +0000
 Ready: True
 Restart Count: 1
 Environment:
 CASSANDRA_HOST: cassandra.us-east-1.amazonaws.com:9142
 CASSANDRA_DC: us-east-1
```



```

 AWS_WEB_IDENTITY_TOKEN_FILE: /var/run/secrets/eks.amazonaws.com/
serviceaccount/token
 AWS_ROLE_ARN: arn:aws:iam::111122223333:role/my-iam-role
 AWS_REGION: us-east-1
 AWS_STS_REGIONAL_ENDPOINTS: regional
Mounts:
 /var/run/secrets/eks.amazonaws.com/serviceaccount from aws-iam-token (ro)
 /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-fssbf (ro)
Conditions:
 Type Status
 Initialized True
 Ready True
 ContainersReady True
 PodScheduled True
Volumes:
 aws-iam-token:
 Type: Projected (a volume that contains injected data from
multiple sources)
 TokenExpirationSeconds: 86400
 kube-api-access-fssbf:
 Type: Projected (a volume that contains injected data from
multiple sources)
 TokenExpirationSeconds: 3607
 ConfigMapName: kube-root-ca.crt
 ConfigMapOptional: <nil>
 DownwardAPI: true
QoS Class: BestEffort
Node-Selectors: <none>
Tolerations: node.kubernetes.io/not-ready:NoExecute op=Exists for
300s
 node.kubernetes.io/unreachable:NoExecute op=Exists for
300s
Events:
 Type Reason Age From Message
 ---- -
 Warning LoggingDisabled 2m13s fargate-scheduler Disabled logging
because aws-logging configmap was not found. configmap "aws-logging" not found
 Normal Scheduled 89s fargate-scheduler Successfully
assigned my-eks-namespace/my-keyspaces-app-6c56fccc56-mgs2m to fargate-
ip-192-168-102-209.ec2.internal
 Normal Pulled 75s kubelet Successfully pulled image "111122223333.dkr.ecr.us-east-1.amazonaws.com/
my_eks_repository:latest" in 13.027s (13.027s including waiting)

```

```

Normal Pulling 54s (x2 over 88s) kubelet Pulling image
"111122223333.dkr.ecr.us-east-1.amazonaws.com/my_eks_repository:latest"
Normal Created 54s (x2 over 75s) kubelet Created container
my-keyspaces-app
Normal Pulled 54s kubelet
Successfully pulled image "111122223333.dkr.ecr.us-east-1.amazonaws.com/
my_eks_repository:latest" in 222ms (222ms including waiting)
Normal Started 53s (x2 over 75s) kubelet Started container
my-keyspaces-app

```

- Controlla i log del Pod per confermare che l'applicazione è in esecuzione e può connettersi alla tabella Amazon Keyspaces. Puoi farlo con il seguente comando. Assicurati di sostituire il nome della tua distribuzione.

```
kubectl logs -f my-keyspaces-app-123abcde4f-g5hij -n my-eks-namespace
```

Dovresti essere in grado di visualizzare le voci di registro dell'applicazione che confermano la connessione ad Amazon Keyspaces come nell'esempio seguente.

```

2:47:20.553 [s0-admin-0] DEBUG c.d.o.d.i.c.metadata.MetadataManager
- [s0] Adding initial contact points [Node(endPoint=cassandra.us-
east-1.amazonaws.com/1.222.333.44:9142, hostId=null, hashCode=e750d92)]
22:47:20.562 [s0-admin-1] DEBUG c.d.o.d.i.c.c.ControlConnection - [s0] Initializing
with event types [SCHEMA_CHANGE, STATUS_CHANGE, TOPOLOGY_CHANGE]
22:47:20.564 [s0-admin-1] DEBUG c.d.o.d.i.core.context.EventBus - [s0] Registering
com.datastax.oss.driver.internal.core.metadata.LoadBalancingPolicyWrapper$$Lambda
$812/0x00000000801105e88@769afb95 for class
com.datastax.oss.driver.internal.core.metadata.NodeStateEvent
22:47:20.566 [s0-admin-1] DEBUG c.d.o.d.i.c.c.ControlConnection -
[s0] Trying to establish a connection to Node(endPoint=cassandra.us-
east-1.amazonaws.com/1.222.333.44:9142, hostId=null, hashCode=e750d92)

```

- Esegui la seguente query CQL sulla tabella Amazon Keyspaces per confermare che una riga di dati sia stata scritta nella tabella:

```
SELECT * from aws.user;
```

Verrà visualizzato l'output seguente:

```

fname | lname | username | last_update_date
-----+-----+-----+-----

```

```
random | k | test | 2023-12-07 13:58:31.57+0000
```

## Fase 5: Pulizia (facoltativa)

Segui questi passaggi per rimuovere tutte le risorse create in questo tutorial.

Rimuovi le risorse create in questo tutorial

1. Elimina la tua distribuzione. A tale scopo, è possibile utilizzare il seguente comando.

```
kubectl delete deployment my-keyspaces-app -n my-eks-namespace
```

2. Elimina il cluster Amazon EKS e tutti i Pod in esso contenuti. Ciò elimina anche le risorse correlate come l'account di servizio e il provider di identità OIDC. A tale scopo, è possibile utilizzare il seguente comando.

```
eksctl delete cluster --name my-eks-cluster --region us-east-1
```

3. Elimina il ruolo IAM utilizzato per l'account di servizio Amazon EKS con autorizzazioni di accesso ad Amazon Keyspaces. Innanzitutto, devi rimuovere la policy gestita associata al ruolo.

```
aws iam detach-role-policy --role-name my-iam-role --policy-arn
arn:aws:iam::aws:policy/AmazonKeyspacesFullAccess
```

È quindi possibile eliminare il ruolo utilizzando il seguente comando.

```
aws iam delete-role --role-name my-iam-role
```

Per ulteriori informazioni, consulta [Eliminazione di un ruolo IAM \(AWS CLI\)](#) nella Guida per l'utente IAM.

4. Elimina il repository Amazon ECR, incluse tutte le immagini in esso archiviate. Puoi farlo usando il seguente comando.

```
aws ecr delete-repository \
 --repository-name my-ecr-repository \
 --force \
 --region us-east-1
```

Nota che il `force` flag è necessario per eliminare un repository che contiene immagini. Per eliminare prima la tua immagine, puoi farlo usando il seguente comando.

```
aws ecr batch-delete-image \
 --repository-name my-ecr-repository \
 --image-ids imageTag=latest \
 --region us-east-1
```

Per ulteriori informazioni, consulta [Eliminare un'immagine](#) nella Amazon Elastic Container Registry User Guide.

5. Elimina lo spazio chiave e la tabella Amazon Keyspaces. L'eliminazione dello spazio chiave elimina automaticamente tutte le tabelle in quel keyspace. A tale scopo, è possibile utilizzare una delle seguenti opzioni.

#### AWS CLI

```
aws keyspaces delete-keyspace --keyspace-name 'aws'
```

Per confermare che il keyspace è stato eliminato, puoi usare il seguente comando.

```
aws keyspaces list-keyspaces
```

Per eliminare prima la tabella, puoi usare il seguente comando.

```
aws keyspaces delete-table --keyspace-name 'aws' --table-name 'user'
```

Per confermare che la tabella è stata eliminata, è possibile utilizzare il seguente comando.

```
aws keyspaces list-tables --keyspace-name 'aws'
```

Per ulteriori informazioni, consulta [delete keyspace](#) ed [delete table](#) nel AWS CLI Command Reference.

#### cqlsh

```
DROP KEYSPACE IF EXISTS "aws";
```

Per verificare che i tuoi keyspaces siano stati eliminati, puoi usare la seguente dichiarazione.

```
SELECT * FROM system_schema.keyspaces ;
```

Il tuo keyspace non dovrebbe essere elencato nell'output di questa istruzione. Nota che può esserci un ritardo prima che gli spazi chiave vengano eliminati. Per ulteriori informazioni, consulta [the section called “RILASCIA LO SPAZIO CHIAVE”](#).

Per eliminare prima la tabella, puoi usare il seguente comando.

```
DROP TABLE "aws.user"
```

Per confermare che la tabella è stata eliminata, è possibile utilizzare il seguente comando.

```
SELECT * FROM system_schema.tables WHERE keyspace_name = "aws";
```

La tua tabella non dovrebbe essere elencata nell'output di questa dichiarazione. Nota che può esserci un ritardo prima che la tabella venga eliminata. Per ulteriori informazioni, consulta [the section called “DROP TABLE”](#).

## Tutorial: esporta una tabella Amazon Keyspaces in Amazon S3 utilizzando AWS Glue

Questo tutorial mostra come esportare una tabella Amazon Keyspaces in un bucket Amazon S3 utilizzando AWS Glue. Per questo tutorial, molti passaggi manuali sono automatizzati utilizzando script di shell disponibili nel repository Github di Amazon [Keyspaces](#). Utilizzando questo processo, puoi esportare i dati di Amazon Keyspaces in Amazon S3 senza dover configurare un cluster Spark.

### Argomenti

- [Prerequisiti per l'esportazione di dati da Amazon Keyspaces ad Amazon S3](#)
- [Fase 1: creare il bucket Amazon S3, scaricare gli strumenti necessari e configurare l'ambiente](#)
- [Fase 2: configurare il AWS Glue job che esporta la tabella Amazon Keyspaces](#)
- [Fase 3: Esegui il AWS Glue processo per esportare la tabella Amazon Keyspaces nel bucket Amazon S3 dal AWS CLI](#)
- [Fase 4: \(Facoltativo\) Creare un trigger per pianificare il processo di esportazione](#)

- [Fase 5: \(Facoltativo\) Pulizia](#)

## Prerequisiti per l'esportazione di dati da Amazon Keyspaces ad Amazon S3

Conferma i seguenti prerequisiti e crea le risorse Amazon Keyspaces prima di iniziare con il tutorial

1. Prima di iniziare questo tutorial, segui le istruzioni di AWS configurazione riportate in. [Accesso ad Amazon Keyspaces \(per Apache Cassandra\)](#) Questi passaggi includono la registrazione AWS e la creazione di un principale AWS Identity and Access Management (IAM) con accesso ad Amazon Keyspaces.
2. Gli script di questo tutorial utilizzano le tue credenziali e per impostazione predefinita Regione AWS vengono archiviati in una posizione nota. Per ulteriori informazioni, consulta [the section called “Gestione delle chiavi di accesso”](#).

L'esempio seguente mostra come memorizzare i valori richiesti come variabili di ambiente per l'utente predefinito.

```
$ export AWS_ACCESS_KEY_ID=AKIAIOSFODNN7EXAMPLE
$ export AWS_SECRET_ACCESS_KEY=wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
$ export AWS_DEFAULT_REGION=us-east-1
```

3. Per eseguire gli script di questo tutorial, è necessario che sul computer siano installati i seguenti software e strumenti:
  - [Java](#)
  - [Apache Maven](#)
  - [Git](#)
  - [AWS CLI](#)

Questo tutorial è stato testato con AWS CLI 2, Java 17.0.13 e Apache Maven 3.8.7.

4. È necessaria una tabella Amazon Keyspaces con dati di esempio da esportare più avanti in questo tutorial. Puoi usare la tua tabella Amazon Keyspaces o creare una tabella di esempio seguendo i passaggi del tutorial. [Nozioni di base](#)
  - a. Per installare `cqlsh-expansion`, segui i passaggi indicati in [the section called “Utilizzo di cqlsh-expansion”](#).

- b. Verifica che il `Murmur3Partitioner` partizionatore sia il partizionatore predefinito per il tuo account. Questo partizionatore è compatibile con Apache Spark Cassandra Connector e con AWS Glue. Per ulteriori informazioni sui partizionatori, vedere [the section called “Lavorare con i partizionatori”](#)

Per modificare il partizionatore del tuo account, puoi utilizzare la seguente dichiarazione.

```
SELECT partitioner FROM system.local;

UPDATE system.local set
partitioner='org.apache.cassandra.dht.Murmur3Partitioner' where key='local';
```

- c. Per creare uno spazio di chiavi Amazon Keyspaces, segui i passaggi indicati. [the section called “Crea uno spazio chiave”](#)
- d. Per creare la tabella Amazon Keyspaces, segui i passaggi indicati. [the section called “Creare una tabella”](#)
- e. Per caricare dati di esempio nella tabella da esportare in Amazon S3, segui i passaggi indicati. [the section called “Crea”](#)

Dopo aver completato i passaggi preliminari, procedi a [the section called “Fase 1: creare il bucket Amazon S3, scaricare gli strumenti e configurare l'ambiente”](#)

## Fase 1: creare il bucket Amazon S3, scaricare gli strumenti necessari e configurare l'ambiente

In questa fase, scarichi gli strumenti esterni e crei e configuri le AWS risorse necessarie per la soluzione di esportazione automatica dei dati di una tabella Amazon Keyspaces in un bucket Amazon S3 utilizzando un processo AWS Glue. [Per eseguire tutte queste attività in modo efficiente, eseguiamo uno script di shell con il nome `setup-connector.sh` disponibile su Github.](#)

Lo script `setup-connector.sh` automatizza i seguenti passaggi.

1. Crea un bucket Amazon S3 utilizzando CloudFormation. Questo bucket memorizza il jar e i file di configurazione scaricati, nonché i dati della tabella esportati.
2. Crea un ruolo IAM utilizzando CloudFormation. AWS Glue i job utilizzano questo ruolo per accedere ad Amazon Keyspaces e Amazon S3.
3. Scarica il [connettore Apache Spark Cassandra](#) e lo carica nel bucket Amazon S3.

4. Scarica il [plug-in di autenticazione SigV4](#) e lo carica nel bucket Amazon S3.
5. Scarica le [estensioni Apache Spark e le](#) carica nel bucket Amazon S3.
6. Scarica la [Keyspaces Retry Policy](#) da Github, compila il codice utilizzando Maven e carica l'output nel bucket Amazon S3.
7. Carica il **keyspaces-application.conf**file nel bucket Amazon S3.

Usa lo script di **setup-connector.sh** shell per automatizzare i passaggi di installazione e configurazione.

1. Copia i file dal repository [aws-glue](#) su Github sul tuo computer locale. Questa directory contiene lo script della shell e altri file richiesti.
2. Esegui lo script della shell `setup-connector.sh`. È possibile specificare i seguenti tre parametri opzionali.
  1. **SETUP\_STACKNAME**— Questo è il nome dello CloudFormation stack utilizzato per creare le AWS risorse.
  2. **S3\_BUCKET\_NAME**— Questo è il nome del bucket Amazon S3.
  3. **GLUE\_SERVICE\_ROLE\_NAME**— Questo è il nome del ruolo di servizio IAM AWS Glue utilizzato per eseguire processi che si connettono ad Amazon Keyspaces e Amazon S3.

È possibile utilizzare il seguente comando per eseguire lo script di shell, fornire i tre parametri con i seguenti nomi.

```
./setup-connector.sh cfn-setup s3-keyspaces iam-export-role
```

Per confermare che il bucket è stato creato, è possibile utilizzare il seguente AWS CLI comando.

```
aws s3 ls s3://s3-keyspaces
```

L'output del comando dovrebbe essere simile a questo.

```
PRE conf/
PRE jars/
```



Per confermare che il ruolo IAM è stato creato e per esaminarne i dettagli, puoi utilizzare la seguente AWS CLI dichiarazione.

```
aws iam get-role --role-name "iam-export-role"
```

```
{
 "Role": {
 "Path": "/",
 "RoleName": "iam-export-role",
 "RoleId": "AKIAIOSFODNN7EXAMPLE",
 "Arn": "arn:aws:iam::111122223333:role/iam-export-role",
 "CreateDate": "2025-01-28T16:09:03+00:00",
 "AssumeRolePolicyDocument": {
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Principal": {
 "Service": "glue.amazonaws.com"
 },
 "Action": "sts:AssumeRole"
 }
]
 },
 "Description": "AWS Glue service role to import and export data from Amazon Keyspaces",
 "MaxSessionDuration": 3600,
 "RoleLastUsed": {
 "LastUsedDate": "2025-01-29T12:03:54+00:00",
 "Region": "us-east-1"
 }
 }
}
```

Se il processo di CloudFormation stack fallisce, puoi rivedere le informazioni dettagliate sull'errore relativo allo stack fallito nella CloudFormation console.

Dopo aver creato il bucket Amazon S3 contenente tutti gli script e gli strumenti e aver configurato il ruolo IAM, procedi con. [the section called “Fase 2: Configurare il lavoro AWS Glue ”](#)

## Fase 2: configurare il AWS Glue job che esporta la tabella Amazon Keyspaces

Nella seconda fase del tutorial, usi lo script `setup-export.sh` disponibile su [Github](#) per creare e configurare il AWS Glue job che si connette ad Amazon Keyspaces utilizzando il plug-in SigV4 e quindi esporta la tabella specificata nel bucket Amazon S3 creato nel passaggio precedente. L'utilizzo dello script consente di esportare dati da Amazon Keyspaces senza configurare un cluster Apache Spark.

Crea un AWS Glue processo per esportare una tabella Amazon Keyspaces in un bucket Amazon S3.

- In questo passaggio, esegui lo script di `setup-export.sh` shell che si trova nella `export-to-s3/` directory da utilizzare CloudFormation per creare e configurare il AWS Glue processo di esportazione. Lo script accetta i seguenti parametri.

`PARENT_STACK_NAME, EXPORT_STACK_NAME, KEYSPACE_NAME, TABLE_NAME, S3_URI, FORMAT`

- `PARENT_STACK_NAME`— Il nome dello CloudFormation stack creato nel passaggio precedente.
- `EXPORT_STACK_NAME`— Il nome dello CloudFormation stack che crea il processo di AWS Glue esportazione.
- `KEYSPACE_NAME` e `TABLE_NAME` — Il nome completo dello spazio delle chiavi e della tabella da esportare. Per questo tutorial, lo usiamo `catalog.book_awards`, ma puoi sostituirlo con il tuo nome di tabella completo.
- `S3_URI`— L'URI opzionale del bucket Amazon S3. L'impostazione predefinita è il bucket Amazon S3 dello stack principale.
- `FORMAT`— Il formato di dati opzionale. Il valore predefinito è `parquet`. Per questo tutorial, per facilitare il caricamento e la trasformazione dei dati, utilizziamo l'impostazione predefinita.

È possibile utilizzare il seguente comando come esempio.

```
setup-export.sh cfn-setup cfn-glue catalog book_awards
```

Per confermare che il lavoro è stato creato, è possibile utilizzare la seguente istruzione.

```
aws glue list-jobs
```

L'output dell'istruzione dovrebbe essere simile a questo.

```
{
 "JobNames": [
 "AmazonKeyspacesExportToS3-cfn-setup-cfn-glue"
]
}
```

Per vedere i dettagli del lavoro, puoi usare il seguente comando.

```
aws glue get-job --job-name AmazonKeyspacesExportToS3-cfn-setup-cfn-glue
```

L'output del comando mostra tutti i dettagli del lavoro. Ciò include gli argomenti predefiniti che è possibile sovrascrivere durante l'esecuzione del processo.

```
{
 "Job": {
 "Name": "AmazonKeyspacesExportToS3-cfn-setup-cfn-glue",
 "JobMode": "SCRIPT",
 "JobRunQueuingEnabled": false,
 "Description": "export to s3",
 "Role": "iam-export-role",
 "CreatedOn": "2025-01-30T15:53:30.765000+00:00",
 "LastModifiedOn": "2025-01-30T15:53:30.765000+00:00",
 "ExecutionProperty": {
 "MaxConcurrentRuns": 1
 },
 "Command": {
 "Name": "glueetl",
 "ScriptLocation": "s3://s3-keyspaces/scripts/cfn-setup-cfn-glue-export.scala",
 "PythonVersion": "3"
 },
 "DefaultArguments": {
 "--write-shuffle-spills-to-s3": "true",
 "--S3_URI": "s3://s3-keyspaces",
 "--TempDir": "s3://s3-keyspaces/shuffle-space/export-sample/"
 }
 }
}
```

```

 "--extra-jars": "s3://s3-keyspaces/jars/spark-cassandra-
connector-assembly_2.12-3.1.0.jar,s3://s3-keyspaces/jars/aws-sigv4-auth-
cassandra-java-driver-plugin-4.0.9-shaded.jar,s3://s3-keyspaces/jars/spark-
extension_2.12-2.8.0-3.4.jar,s3://s3-keyspaces/jars/amazon-keyspaces-helpers-1.0-
SNAPSHOT.jar",
 "--class": "GlueApp",
 "--user-jars-first": "true",
 "--enable-metrics": "true",
 "--enable-spark-ui": "true",
 "--KEYSPACE_NAME": "catalog",
 "--spark-event-logs-path": "s3://s3-keyspaces/spark-logs/",
 "--enable-continuous-cloudwatch-log": "true",
 "--write-shuffle-files-to-s3": "true",
 "--FORMAT": "parquet",
 "--TABLE_NAME": "book_awards",
 "--job-language": "scala",
 "--extra-files": "s3://s3-keyspaces/conf/keyspaces-application.conf",
 "--DRIVER_CONF": "keyspaces-application.conf"
 },
 "MaxRetries": 0,
 "AllocatedCapacity": 4,
 "Timeout": 2880,
 "MaxCapacity": 4.0,
 "WorkerType": "G.2X",
 "NumberOfWorkers": 2,
 "GlueVersion": "3.0"
}
}

```

Se il processo di CloudFormation stack fallisce, puoi esaminare gli errori relativi allo stack fallito nella console. CloudFormation Puoi rivedere i dettagli del processo di esportazione nella AWS Glue console selezionando i lavori ETL nel menu a sinistra.

Dopo aver confermato i dettagli del processo di AWS Glue esportazione, procedi con l'esecuzione del processo per esportare i dati dalla tabella Amazon Keyspaces. [the section called “Fase 3: Esegui il processo di esportazione AWS Glue dal AWS CLI”](#)

## Fase 3: Esegui il AWS Glue processo per esportare la tabella Amazon Keyspaces nel bucket Amazon S3 dal AWS CLI

In questo passaggio, usi AWS CLI per eseguire il AWS Glue job creato nel passaggio precedente per esportare una tabella Amazon Keyspaces nel tuo bucket in Amazon S3.

Esegui il processo di esportazione dal AWS CLI

1. Nell'esempio seguente, il AWS CLI comando esegue il processo creato nel passaggio precedente.

```
aws glue start-job-run --job-name AmazonKeyspacesExportToS3-cfn-setup-cfn-glue
```

- È possibile sovrascrivere qualsiasi parametro del AWS Glue processo, inclusi gli argomenti predefiniti del AWS CLI comando. Per sovrascrivere gli argomenti predefiniti del job, ad esempio keyspace o table name, potete passarli come argomenti. Per un elenco completo degli argomenti, [start-job-run](#) consultate AWS Glue Command Line Reference.

Il comando seguente esegue il processo di AWS Glue esportazione, ma sostituisce il numero di AWS Glue lavoratori, il tipo di lavoratore e il nome della tabella.

```
aws glue start-job-run --job-name AmazonKeyspacesExportToS3-cfn-setup-cfn-glue \
 --number-of-workers 8 --worker-type G.2X \
 --arguments '{"--TABLE_NAME": "my_table"}'
```

2. Verifica che la tabella sia stata esportata nel bucket Amazon S3. In base alle dimensioni della tabella, l'operazione può richiedere del tempo. Al termine del processo di esportazione, è possibile visualizzare le seguenti cartelle nel bucket utilizzando il comando example.

```
aws s3 ls s3://s3-keyspaces
```

L'output mostra la seguente struttura nel bucket.

```
PRE conf/
PRE export/
PRE jars/
PRE scripts/
PRE spark-logs/
```

I tuoi file si troveranno nella seguente struttura di cartelle sotto `export`, `data/time` i valori mostreranno i tuoi valori.

```
\----- export
 \----- keyspace_name
 \----- table_name
 \----- snapshot
 \----- year=2025
 \----- month=01
 \----- day=02
 \----- hour=09
 \----- minute=22
 \--- YOUR DATA HERE
```

Per pianificare il AWS Glue lavoro appena eseguito manualmente, procedi con [the section called “Fase 4: \(Facoltativo\) Pianificare il processo di esportazione”](#).

## Fase 4: (Facoltativo) Creare un trigger per pianificare il processo di esportazione

Per eseguire regolarmente il processo di esportazione creato nel passaggio precedente, puoi creare un trigger pianificato. Per ulteriori informazioni, consulta [AWS Glue i trigger](#) nella Guida per gli AWS Glue sviluppatori.

Pianifica un lavoro AWS Glue

1. Il AWS CLI comando seguente è un esempio di un semplice trigger con il nome `KeyspacesExportWeeklyTrigger` che esegue il AWS Glue processo con lo stesso nome `AmazonKeyspacesExportToS3-cfn-setup-cfn-glue` una volta alla settimana il lunedì alle 12:00 UTC.

```
aws glue create-trigger \
 --name KeyspacesExportWeeklyTrigger \
 --type SCHEDULED \
 --schedule "cron(0 12 ? * MON *)" \
 --start-on-creation \
 --actions '[{
 "JobName": "AmazonKeyspacesExportToS3-cfn-setup-cfn-glue"
```

```
}}'
```

- Per sovrascrivere le impostazioni predefinite del processo pianificato, è possibile passarle come argomenti. In questo esempio sostituiamo il nome dello spazio chiave, il nome della tabella, il numero di lavoratori e il tipo di lavoratore passandoli come argomenti. Il comando seguente ne è un esempio.

```
aws glue create-trigger \
 --name KeyspacesExportWeeklyTrigger \
 --type SCHEDULED \
 --schedule "cron(0 12 ? * MON *)" \
 --start-on-creation \
 --actions '[{
 "JobName": "AmazonKeyspacesExportToS3-cfn-setup-cfn-glue",
 "Arguments": {
 "--number-of-workers": "8",
 "--worker-type": "G.2X",
 "--table_name": "my_table",
 "--keyspace_name": "my_keyspace"
 }
}]'
```

2. Per confermare che il trigger è stato creato, utilizzare il comando seguente.

```
aws glue list-triggers
```

L'output del comando dovrebbe essere simile a questo.

```
{
 "TriggerNames": [
 "KeyspacesExportWeeklyTrigger"
]
}
```

Per ripulire le AWS risorse create in questo tutorial, procedi [alla sezione chiamata “Fase 5: Pulizia \(facoltativa\)”](#).

## Fase 5: (Facoltativo) Pulizia

Segui questi passaggi per rimuovere tutte le AWS risorse create in questo tutorial.

## Rimuovi le risorse create in questo tutorial

1. Elimina il secondo CloudFormation stack creato in questo tutorial. Questo rimuove il AWS Glue job e il trigger creati in questo tutorial. A tale scopo, è possibile usare il comando riportato di seguito.

```
aws cloudformation delete-stack --stack-name cfn-glue
```

2. Elimina il bucket Amazon S3 insieme a tutti i dati in esso archiviati. A tale scopo, è possibile utilizzare il seguente comando.

```
aws s3 rm s3://s3-keyspaces --recursive
```

3. Elimina il primo stack creato in questo tutorial. Ciò elimina il ruolo IAM e le autorizzazioni associate create in questo tutorial. È possibile utilizzare il comando seguente come esempio.

```
aws cloudformation delete-stack --stack-name cfn-setup
```

4. Elimina lo spazio chiave e la tabella Amazon Keyspaces. L'eliminazione dello spazio chiave elimina automaticamente tutte le tabelle in quel keyspace. A tale scopo, è possibile utilizzare una delle seguenti opzioni.

### AWS CLI

```
aws keyspaces delete-keyspace --keyspace-name 'aws'
```

Per confermare che il keyspace è stato eliminato, puoi usare il seguente comando.

```
aws keyspaces list-keyspaces
```

Per eliminare prima la tabella, puoi usare il seguente comando.

```
aws keyspaces delete-table --keyspace-name 'aws' --table-name 'user'
```

Per confermare che la tabella è stata eliminata, è possibile utilizzare il seguente comando.

```
aws keyspaces list-tables --keyspace-name 'aws'
```



Per ulteriori informazioni, consulta [delete keyspace](#) ed [delete table](#) nel AWS CLI Command Reference.

cqlsh

```
DROP KEYSPACE IF EXISTS "aws";
```

Per verificare che i tuoi keyspace siano stati eliminati, puoi usare la seguente dichiarazione.

```
SELECT * FROM system_schema.keyspaces ;
```

Il tuo keyspace non dovrebbe essere elencato nell'output di questa istruzione. Nota che può esserci un ritardo prima che gli spazi chiave vengano eliminati. Per ulteriori informazioni, consulta [the section called “RILASCIA LO SPAZIO CHIAVE”](#).

Per eliminare prima la tabella, puoi usare il seguente comando.

```
DROP TABLE "aws.user"
```

Per confermare che la tabella è stata eliminata, è possibile utilizzare il seguente comando.

```
SELECT * FROM system_schema.tables WHERE keyspace_name = "aws";
```

La tua tabella non dovrebbe essere elencata nell'output di questa dichiarazione. Nota che può esserci un ritardo prima che la tabella venga eliminata. Per ulteriori informazioni, consulta [the section called “DROP TABLE”](#).

# Gestione delle risorse serverless in Amazon Keyspaces (per Apache Cassandra)

Amazon Keyspaces (per Apache Cassandra) è serverless. Invece di distribuire, gestire e mantenere le risorse di storage e di calcolo per il carico di lavoro attraverso i nodi di un cluster, Amazon Keyspaces alloca le risorse di storage e di velocità effettiva di lettura/scrittura direttamente alle tabelle.

Amazon Keyspaces fornisce automaticamente lo storage in base ai dati archiviati nelle tabelle. Aumenta e riduce lo storage man mano che scrivi, aggiorni ed elimini i dati e paghi solo per lo storage che utilizzi. I dati vengono replicati su più [zone di disponibilità](#) per un'elevata disponibilità. Amazon Keyspaces monitora continuamente le dimensioni delle tabelle per determinare i costi di storage. Per ulteriori informazioni su come Amazon Keyspaces calcola la dimensione fatturabile dei dati, consulta [the section called “Stima della dimensione delle righe”](#)

Questo capitolo tratta gli aspetti chiave della gestione delle risorse in Amazon Keyspaces.

- **Stima della dimensione delle righe:** per stimare la dimensione codificata delle righe in Amazon Keyspaces, considera fattori come i metadati delle chiavi di partizione, i metadati delle colonne di clustering, gli identificatori di colonna, i tipi di dati e i metadati delle righe. Questa dimensione di riga codificata viene utilizzata per la fatturazione, la gestione delle quote e la pianificazione della capacità di throughput assegnata.
- **Stima del consumo di capacità:** questa sezione contiene esempi di come stimare il consumo di capacità di lettura e scrittura per scenari comuni come query con intervallo, query con limiti, scansioni di tabelle, transazioni leggere, colonne statiche e tabelle multiregionali. Puoi utilizzare Amazon CloudWatch per monitorare l'utilizzo effettivo della capacità. Per ulteriori informazioni sul monitoraggio con CloudWatch, consulta [the section called “Monitoraggio con CloudWatch”](#).
- **Configura le modalità di capacità di lettura/scrittura:** puoi scegliere tra due modalità di capacità per l'elaborazione di letture e scritture sulle tabelle:
  - **Modalità on demand (impostazione predefinita):** paga per richiesta per la velocità effettiva di lettura e scrittura. Amazon Keyspaces può scalare istantaneamente la capacità fino a qualsiasi livello di traffico raggiunto in precedenza.
  - **Modalità di provisioning:** specifica in anticipo il numero richiesto di unità con capacità di lettura e scrittura. Questa modalità aiuta a mantenere prestazioni di throughput prevedibili.

- Gestisci la capacità di throughput con la scalabilità automatica: per le tabelle con provisioning, è possibile abilitare la scalabilità automatica per regolare automaticamente la capacità di throughput in base al traffico effettivo delle applicazioni. Amazon Keyspaces utilizza il tracciamento del target per aumentare o diminuire la capacità fornita, mantenendo l'utilizzo all'obiettivo specificato.
- Usa la capacità di burst in modo efficace: Amazon Keyspaces fornisce capacità di burst riservando una parte del throughput inutilizzato per gestire i picchi di traffico. Questa flessibilità consente occasionali picchi di attività superiori al throughput previsto.

Per risolvere gli errori di capacità, vedere. [the section called “Errori di capacità serverless”](#)

## Argomenti

- [Stima della dimensione delle righe in Amazon Keyspaces](#)
- [Stima del consumo di capacità per la velocità effettiva di lettura e scrittura in Amazon Keyspaces](#)
- [Configura le modalità di capacità di lettura/scrittura in Amazon Keyspaces](#)
- [Gestisci automaticamente la capacità di throughput con la scalabilità automatica di Amazon Keyspaces](#)
- [Usa la capacità burst in modo efficace in Amazon Keyspaces](#)

## Stima della dimensione delle righe in Amazon Keyspaces

Amazon Keyspaces fornisce uno storage completamente gestito che offre prestazioni di lettura e scrittura a una cifra di millisecondi e archivia i dati in modo duraturo su più zone di disponibilità. AWS Amazon Keyspaces allega i metadati a tutte le righe e le colonne chiave primarie per supportare un accesso efficiente ai dati e un'elevata disponibilità.

Questo argomento fornisce dettagli su come stimare la dimensione codificata delle righe in Amazon Keyspaces. La dimensione codificata delle righe viene utilizzata per il calcolo della fattura e dell'utilizzo della quota. È inoltre possibile utilizzare la dimensione delle righe codificate per stimare i requisiti di capacità di throughput assegnati per le tabelle.

Per calcolare la dimensione codificata delle righe in Amazon Keyspaces, puoi utilizzare le seguenti linee guida.

## Argomenti

- [Stima la dimensione codificata delle colonne](#)

- [Stima la dimensione codificata dei valori dei dati in base al tipo di dati](#)
- [Considera l'impatto delle funzionalità di Amazon Keyspaces sulla dimensione delle righe](#)
- [Scegliete la formula giusta per calcolare la dimensione codificata di una riga](#)
- [Esempio di calcolo della dimensione delle righe](#)

## Stima la dimensione codificata delle colonne

Questa sezione mostra come stimare la dimensione codificata delle colonne in Amazon Keyspaces.

- Colonne regolari: per le colonne normali, che sono colonne che non sono chiavi primarie, colonne di clustering o STATIC colonne, usa la dimensione grezza dei dati della cella in base al [tipo di dati](#) e aggiungi i metadati richiesti. I tipi di dati e alcune differenze chiave nel modo in cui Amazon Keyspaces archivia i valori dei tipi di dati e i metadati sono elencati nella sezione successiva.
- Colonne chiave di partizione: le chiavi di partizione possono contenere fino a 2048 byte di dati. Ogni colonna chiave nella chiave di partizione richiede fino a 3 byte di metadati. Quando si calcola la dimensione della riga, è necessario presupporre che ogni colonna della chiave di partizione utilizzi tutti i 3 byte di metadati.
- Colonne di clustering: le colonne di clustering possono memorizzare fino a 850 byte di dati. Oltre alla dimensione del valore dei dati, ogni colonna di clustering richiede fino al 20% della dimensione del valore dei dati per i metadati. Quando si calcola la dimensione della riga, è necessario aggiungere 1 byte di metadati per ogni 5 byte di valore dei dati della colonna di clustering.

### Note

Per supportare interrogazioni efficienti e indicizzazione integrata, Amazon Keyspaces memorizza due volte il valore dei dati di ogni chiave di partizione e colonna chiave di clustering.

- Nomi di colonna: lo spazio richiesto per ogni nome di colonna viene memorizzato utilizzando un identificatore di colonna e aggiunto a ciascun valore di dati memorizzato nella colonna. Il valore di archiviazione dell'identificatore di colonna dipende dal numero complessivo di colonne della tabella:
  - 1—62 colonne: 1 byte
  - 63-124 colonne: 2 byte
  - 125-186 colonne: 3 byte

Per ogni 62 colonne aggiuntive aggiungere 1 byte. Tieni presente che in Amazon Keyspaces, è possibile modificare fino a 225 colonne regolari con una singola istruzione INSERT or UPDATE. Per ulteriori informazioni, consulta [the section called “Quote di servizio Amazon Keyspaces”](#).

## Stima la dimensione codificata dei valori dei dati in base al tipo di dati

Questa sezione mostra come stimare la dimensione codificata di diversi tipi di dati in Amazon Keyspaces.

- **Tipi di stringa:** Cassandra ASCII e i tipi di dati VARCHAR string sono tutti archiviati in Amazon Keyspaces utilizzando Unicode con codifica binaria UTF-8. TEXT La dimensione di una stringa in Amazon Keyspaces è uguale al numero di byte con codifica UTF-8.
- **Tipi numerici:** Cassandra INT, BIGINT, SMALLINT, TINYINT, e VARINT i tipi di dati vengono archiviati in Amazon Keyspaces come valori di dati a lunghezza variabile, con un massimo di 38 cifre significative. Gli zero iniziali e finali vengono tagliati. La dimensione di ognuno di questi tipi di dati è di circa 1 byte per due cifre significative + 1 byte.
- **Tipo di blob:** A BLOB in Amazon Keyspaces viene archiviato con la lunghezza in byte non elaborata del valore.
- **Tipo booleano:** la dimensione di uno o più Boolean valori è di 1 byte. Null
- **Tipi di raccolta:** colonna che memorizza tipi di dati di raccolta come LIST o MAP richiede 3 byte di metadati, indipendentemente dal contenuto. La dimensione di un LIST or MAP è (id della colonna) + sum (dimensione degli elementi annidati) + (3 byte). La dimensione di un LIST or vuoto MAP è (id di colonna) + (3 byte). Ogni singolo LIST MAP elemento richiede anche 1 byte di metadati.
- **Tipi definiti dall'utente:** un [tipo definito dall'utente \(UDT\)](#) richiede 3 byte per i metadati, indipendentemente dal contenuto. Per ogni elemento UDT, Amazon Keyspaces richiede 1 byte aggiuntivo di metadati.

Per calcolare la dimensione codificata di un UDT, inizia con `field name` e `field value` per i campi di un UDT:

- **nome di campo:** ogni nome di campo dell'UDT di primo livello viene memorizzato utilizzando un identificatore. Il valore di archiviazione dell'identificatore dipende dal numero complessivo di campi nell'UDT di primo livello e può variare tra 1 e 3 byte:
  - 1—62 campi: 1 byte
  - 63—124 campi: 2 byte

- 125— numero massimo di campi: 3 byte
- valore del campo: i byte necessari per memorizzare i valori dei campi dell'UDT di primo livello dipendono dal tipo di dati archiviati:
  - Tipo di dati scalare: i byte necessari per l'archiviazione sono gli stessi dello stesso tipo di dati archiviato in una colonna normale.
  - UDT congelato: per ogni UDT annidato congelato, l'UDT nidificato ha le stesse dimensioni del protocollo binario CQL. Per un UDT nidificato, vengono memorizzati 4 byte per ogni campo (compresi i campi vuoti) e il valore del campo archiviato è il formato di serializzazione del protocollo binario CQL del valore del campo.
- Collezioni Frozen:
  - LIST e SET: per un ambiente congelato LIST o annidatoSET, vengono archiviati 4 byte per ogni elemento della raccolta più il formato di serializzazione del protocollo binario CQL del valore della raccolta.
  - MAP — Per un blocco congelato annidatoMAP, ogni coppia chiave-valore ha i seguenti requisiti di archiviazione:
    - Per ogni chiave, allocare 4 byte, quindi aggiungere il formato di serializzazione del protocollo binario CQL della chiave.
    - Per ogni valore allocare 4 byte, quindi aggiungere il formato di serializzazione del protocollo binario CQL del valore.
- Parola chiave FROZEN: per le raccolte congelate annidate all'interno di raccolte congelate, Amazon Keyspaces non richiede byte aggiuntivi per i metadati.
- Parola chiave STATICA: i dati STATIC delle colonne non vengono conteggiati ai fini della dimensione massima delle righe di 1 MB. Per calcolare la dimensione dei dati delle colonne statiche, vedi [the section called “Calcola la dimensione statica delle colonne per partizione logica”](#).

## Considera l'impatto delle funzionalità di Amazon Keyspaces sulla dimensione delle righe

Questa sezione mostra come le funzionalità di Amazon Keyspaces influiscono sulla dimensione codificata di una riga.

- Timestamp lato client: i timestamp lato client vengono memorizzati per ogni colonna di ogni riga quando la funzionalità è attivata. Questi timestamp occupano circa 20-40 byte (a seconda dei dati)

e contribuiscono ai costi di archiviazione e velocità effettiva della riga. Per ulteriori informazioni sui timestamp lato client, vedere. [the section called “Timestamp lato client”](#)

- Time to Live (TTL): i metadati TTL occupano circa 8 byte per riga quando la funzionalità è attivata. Inoltre, i metadati TTL vengono archiviati per ogni colonna di ogni riga. I metadati TTL occupano circa 8 byte per ogni colonna che memorizza un tipo di dati scalare o una raccolta congelata. Se la colonna memorizza un tipo di dati di raccolta che non è bloccato, per ogni elemento della raccolta TTL richiede circa 8 byte aggiuntivi per i metadati. Per una colonna che memorizza un tipo di dati di raccolta quando il TTL è abilitato, puoi utilizzare la formula seguente.

```
total encoded size of column = (column id) + sum (nested elements + collection
 metadata (1 byte) + TTL metadata (8 bytes)) + collection column metadata (3 bytes)
```

I metadati TTL contribuiscono al costo di archiviazione e velocità effettiva della riga. Per ulteriori informazioni su TTL, consulta. [the section called “Fai scadere i dati con Time to Live”](#)

## Scegliete la formula giusta per calcolare la dimensione codificata di una riga

Questa sezione mostra le diverse formule che puoi utilizzare per stimare i requisiti di storage o di capacità di throughput per una riga di dati in Amazon Keyspaces.

La dimensione totale codificata di una riga di dati può essere stimata in base a una delle seguenti formule, in base al tuo obiettivo:

- Capacità di trasmissione: per stimare la dimensione codificata di una riga per valutare le unità di read/write richiesta richieste (: RRUs/WRUs) or read/write capacity units (RCUs/WCUs)

```
total encoded size of row = partition key columns + clustering columns + regular
 columns
```

- Dimensione di archiviazione: per stimare la dimensione codificata di una riga e prevederla `BillableTableSizeInBytes`, aggiungi i metadati richiesti per l'archiviazione della riga:

```
total encoded size of row = partition key columns + clustering columns + regular
 columns + row metadata (100 bytes)
```

**⚠ Important**

Tutti i metadati delle colonne, ad esempio gli ID delle colonne, i metadati delle chiavi di partizione, i metadati delle colonne di clustering, nonché i timestamp, il TTL e i metadati di riga sul lato client, vengono conteggiati ai fini della dimensione massima delle righe di 1 MB.

## Esempio di calcolo della dimensione delle righe

Considerate il seguente esempio di tabella in cui tutte le colonne sono di tipo intero. La tabella ha due colonne chiave di partizione, due colonne di clustering e una colonna normale. Poiché questa tabella ha cinque colonne, lo spazio richiesto per l'identificatore del nome della colonna è di 1 byte.

```
CREATE TABLE mykeyspace.mytable(pk_col1 int, pk_col2 int, ck_col1 int, ck_col2 int,
 reg_col1 int, primary key((pk_col1, pk_col2),ck_col1, ck_col2));
```

In questo esempio, calcoliamo la dimensione dei dati quando scriviamo una riga nella tabella, come illustrato nella seguente dichiarazione:

```
INSERT INTO mykeyspace.mytable (pk_col1, pk_col2, ck_col1, ck_col2, reg_col1)
values(1,2,3,4,5);
```

Per stimare i byte totali richiesti da questa operazione di scrittura, è possibile utilizzare i seguenti passaggi.

1. Calcola la dimensione di una colonna chiave di partizione aggiungendo i byte per il tipo di dati archiviato nella colonna e i byte dei metadati. Ripeti questa operazione per tutte le colonne chiave di partizione.

- a. Calcola la dimensione della prima colonna della chiave di partizione (pk\_col1):

```
(2 bytes for the integer data type) x 2 + 1 byte for the column id + 3 bytes
for partition key metadata = 8 bytes
```

- b. Calcola la dimensione della seconda colonna della chiave di partizione (pk\_col2):

```
(2 bytes for the integer data type) x 2 + 1 byte for the column id + 3 bytes
for partition key metadata = 8 bytes
```



- c. Aggiungi entrambe le colonne per ottenere la dimensione totale stimata delle colonne chiave di partizione:

```
8 bytes + 8 bytes = 16 bytes for the partition key columns
```

2. Calcola la dimensione della colonna di clustering aggiungendo i byte per il tipo di dati archiviato nella colonna e i byte dei metadati. Ripeti l'operazione per tutte le colonne di clustering.

- a. Calcola la dimensione della prima colonna della colonna di clustering (ck\_col1):

```
(2 bytes for the integer data type) x 2 + 20% of the data value (2 bytes) for
clustering column metadata + 1 byte for the column id = 6 bytes
```

- b. Calcola la dimensione della seconda colonna della colonna di clustering (ck\_col2):

```
(2 bytes for the integer data type) x 2 + 20% of the data value (2 bytes) for
clustering column metadata + 1 byte for the column id = 6 bytes
```

- c. Aggiungi entrambe le colonne per ottenere la dimensione totale stimata delle colonne di clustering:

```
6 bytes + 6 bytes = 12 bytes for the clustering columns
```

3. Aggiungi la dimensione delle colonne normali. In questo esempio abbiamo solo una colonna che memorizza un numero intero a una cifra, che richiede 2 byte con 1 byte per l'ID della colonna.
4. Infine, per ottenere la dimensione totale delle righe codificate, sommate i byte per tutte le colonne. Per stimare la dimensione fatturabile per l'archiviazione, aggiungi i 100 byte aggiuntivi per i metadati delle righe:

```
16 bytes for the partition key columns + 12 bytes for clustering columns + 3 bytes
for the regular column + 100 bytes for row metadata = 131 bytes.
```

Per scoprire come monitorare le risorse serverless con Amazon CloudWatch, consulta [the section called "Monitoraggio con CloudWatch"](#).

# Stima del consumo di capacità per la velocità effettiva di lettura e scrittura in Amazon Keyspaces

Quando leggi o scrivi dati in Amazon Keyspaces, la quantità di read/write request units (RRUs/WRUs) or read/write capacity units (RCUs/WCUs) consumati dalla query dipende dalla quantità totale di dati che Amazon Keyspaces deve elaborare per eseguire la query. In alcuni casi, i dati restituiti al client possono essere un sottoinsieme dei dati che Amazon Keyspaces ha dovuto leggere per elaborare la query. Per le scritture condizionali, Amazon Keyspaces consuma capacità di scrittura anche se il controllo condizionale fallisce.

Per stimare la quantità totale di dati elaborati per una richiesta, devi considerare la dimensione codificata di una riga e il numero totale di righe. Questo argomento illustra alcuni esempi di scenari e modelli di accesso comuni per mostrare come Amazon Keyspaces elabora le query e come ciò influisca sul consumo di capacità. Puoi seguire gli esempi per stimare i requisiti di capacità delle tue tabelle e utilizzare Amazon CloudWatch per osservare il consumo di capacità di lettura e scrittura per questi casi d'uso.

Per informazioni su come calcolare la dimensione codificata delle righe in Amazon Keyspaces, consulta [the section called “Stima della dimensione delle righe”](#)

## Argomenti

- [Stima il consumo di capacità delle query di intervallo in Amazon Keyspaces](#)
- [Stima il consumo di capacità di lettura delle query limite](#)
- [Stima il consumo di capacità di lettura delle scansioni delle tabelle](#)
- [Stima del consumo di capacità delle transazioni leggere in Amazon Keyspaces](#)
- [Stima del consumo di capacità per le colonne statiche in Amazon Keyspaces](#)
- [Stima e fornisci la capacità per una tabella multiregionale in Amazon Keyspaces](#)
- [Stima del consumo di capacità di lettura e scrittura con Amazon CloudWatch in Amazon Keyspaces](#)

## Stima il consumo di capacità delle query di intervallo in Amazon Keyspaces

Per esaminare il consumo di capacità di lettura di una query con intervallo, utilizziamo la seguente tabella di esempio che utilizza la modalità di capacità su richiesta.

```
pk1 | pk2 | pk3 | ck1 | ck2 | ck3 | value
```

```

-----+-----+-----+-----+-----+-----+-----+-----
a | b | 1 | a | b | 50 | <any value that results in a row size larger than 4KB>
a | b | 1 | a | b | 60 | value_1
a | b | 1 | a | b | 70 | <any value that results in a row size larger than 4KB>

```

Ora esegui la seguente query su questa tabella.

```

SELECT * FROM amazon_keyspaces.example_table_1 WHERE pk1='a' AND pk2='b' AND pk3=1 AND
ck1='a' AND ck2='b' AND ck3 > 50 AND ck3 < 70;

```

Riceverai il seguente set di risultati dalla query e l'operazione di lettura eseguita da Amazon Keyspaces consuma 2 RRUs in LOCAL\_QUORUM modalità di coerenza.

```

pk1 | pk2 | pk3 | ck1 | ck2 | ck3 | value
-----+-----+-----+-----+-----+-----+-----
a | b | 1 | a | b | 60 | value_1

```

Amazon Keyspaces ne consuma 2 RRUs per valutare le righe con i valori `ck3=60` ed `ck3=70` elaborare la query. Tuttavia, Amazon Keyspaces restituisce solo la riga in cui la WHERE condizione specificata nella query è vera, ovvero la riga con valore. `ck3=60` Per valutare l'intervallo specificato nella query, Amazon Keyspaces legge la riga che corrisponde al limite superiore dell'intervallo, in questo caso `ck3 = 70`, ma non restituisce quella riga nel risultato. Il consumo di capacità di lettura si basa sui dati letti durante l'elaborazione della query, non sui dati restituiti.

## Stima il consumo di capacità di lettura delle query limite

Durante l'elaborazione di una query che utilizza la LIMIT clausola, Amazon Keyspaces legge le righe fino alla dimensione massima della pagina quando cerca di soddisfare la condizione specificata nella query. Se Amazon Keyspaces non riesce a trovare dati corrispondenti sufficienti che soddisfino il LIMIT valore nella prima pagina, potrebbero essere necessarie una o più chiamate impaginate. Per continuare le letture nella pagina successiva, puoi utilizzare un token di impaginazione. La dimensione predefinita della pagina è 1 MB. Per ridurre la capacità di lettura quando si utilizzano le LIMIT clausole, è possibile ridurre le dimensioni della pagina. Per ulteriori informazioni sull'impaginazione, vedere. [the section called “Impagina i risultati”](#)

Per un esempio, diamo un'occhiata alla seguente query.

```

SELECT * FROM my_table WHERE partition_key=1234 LIMIT 1;

```

Se non imposti la dimensione della pagina, Amazon Keyspaces legge 1 MB di dati anche se ti restituisce solo 1 riga. Per fare in modo che Amazon Keyspaces legga solo una riga, puoi impostare la dimensione della pagina su 1 per questa query. In questo caso, Amazon Keyspaces leggerebbe solo una riga, a condizione che non ci siano righe scadute in base a Time-to-live impostazioni o timestamp lato client.

Il `PAGE SIZE` parametro determina quante righe Amazon Keyspaces scansiona dal disco per ogni richiesta, non quante righe Amazon Keyspaces restituisce al client. Amazon Keyspaces applica i filtri che fornisci, ad esempio la disuguaglianza su colonne non chiave o `LIMIT` dopo la scansione dei dati su disco. Se non lo imposti esplicitamente `PAGE SIZE`, Amazon Keyspaces legge fino a 1 MB di dati prima di applicare i filtri. Ad esempio, se utilizzi `LIMIT 1 Without SpecifyPAGE SIZE`, Amazon Keyspaces potrebbe leggere migliaia di righe dal disco prima di applicare la clausola limite e restituire solo una riga.

Per evitare un eccesso di lettura, riduci il numero di righe `PAGE SIZE` che Amazon Keyspaces scansiona per ogni recupero. Ad esempio, se lo definisci `LIMIT 5` nella tua query, imposta un valore compreso tra 5 e 10 in modo che Amazon Keyspaces scansioni solo 5-10 righe per ogni chiamata impaginata. `PAGE SIZE` Puoi modificare questo numero per ridurre il numero di recuperi. Per limiti superiori alla dimensione della pagina, Amazon Keyspaces mantiene il conteggio totale dei risultati con lo stato di impaginazione. Nel caso `LIMIT` di 10.000 righe, Amazon Keyspaces può recuperare questi risultati in due pagine da 5.000 righe ciascuna. Il limite di 1 MB è il limite superiore per qualsiasi set di dimensioni di pagina.

## Stima il consumo di capacità di lettura delle scansioni delle tabelle

Le interrogazioni che generano scansioni complete della tabella, ad esempio le query che utilizzano l'`ALLOW FILTERING` opzione, sono un altro esempio di query che elaborano più letture di quelle restituite come risultati. Inoltre, il consumo di capacità di lettura si basa sui dati letti, non sui dati restituiti.

Per l'esempio di scansione delle tabelle, utilizziamo la seguente tabella di esempio in modalità di capacità su richiesta.

```
pk | ck | value
---+---+-----
pk | 10 | <any value that results in a row size larger than 4KB>
pk | 20 | value_1
pk | 30 | <any value that results in a row size larger than 4KB>
```

Amazon Keyspaces crea una tabella in modalità di capacità su richiesta con quattro partizioni per impostazione predefinita. In questa tabella di esempio, tutti i dati sono archiviati in una partizione e le tre partizioni rimanenti sono vuote.

Ora esegui la seguente query sulla tabella.

```
SELECT * from amazon_keyspaces.example_table_2;
```

Questa query genera un'operazione di scansione della tabella in cui Amazon Keyspaces analizza tutte e quattro le partizioni della tabella e ne consuma 6 in modalità di coerenza. RRUs LOCAL\_QUORUM Innanzitutto, Amazon Keyspaces ne consuma 3 RRUs per leggere le tre righe con. `pk= 'pk'` Quindi, Amazon Keyspaces utilizza i 3 componenti aggiuntivi RRUs per la scansione delle tre partizioni vuote della tabella. Poiché questa query genera una scansione della tabella, Amazon Keyspaces analizza tutte le partizioni della tabella, incluse le partizioni senza dati.

## Stima del consumo di capacità delle transazioni leggere in Amazon Keyspaces

Le transazioni leggere (LWT) consentono di eseguire operazioni di scrittura condizionali sui dati della tabella. Le operazioni di aggiornamento condizionale sono utili per l'inserimento, l'aggiornamento e l'eliminazione dei record in base a condizioni che valutano lo stato corrente.

In Amazon Keyspaces, tutte le operazioni di scrittura richiedono la coerenza LOCAL\_QUORUM e non sono previsti costi aggiuntivi per l'utilizzo. LWTs La differenza LWTs è che quando viene eseguito un controllo delle condizioni LWTFALSE, Amazon Keyspaces utilizza unità di capacità di scrittura WCUs () o unità di richiesta di scrittura (). WRUs Il numero di WCUs/WRUs consumati dipende dalla dimensione della riga.

Ad esempio, se la dimensione della riga è 2 KB, la scrittura condizionale non riuscita consuma due WCUs/. WRUs Se la riga non esiste attualmente nella tabella, l'operazione consuma una/. WCUs WRUs

Per determinare il numero di richieste che hanno provocato errori nel controllo delle condizioni, puoi monitorare la `ConditionalCheckFailed` metrica in CloudWatch

### Stima i costi LWT per le tabelle con Time to Live (TTL)

LWTs può richiedere unità di capacità di lettura aggiuntive (RCUs) o unità di richiesta di lettura (RRUs) per tabelle configurate con TTL che non utilizzano timestamp lato client. Quando si utilizzano

Le IF EXISTS risultanti dal controllo delle condizioni delle IF NOT EXISTS parole chiave FALSE, vengono consumate le seguenti unità di capacità:

- RCUs/RRUs – If the row exists, the RCUs/RRUs consumati si basano sulla dimensione della riga esistente.
- RCUs/RRUs – If the row doesn't exist, a single RCU/RRU viene consumato.

Se la condizione valutata determina un'operazione di scrittura riuscita, WCUs/WRUs vengono utilizzati in base alla dimensione della nuova riga.

## Stima del consumo di capacità per le colonne statiche in Amazon Keyspaces

In una tabella Amazon Keyspaces con colonne di clustering, puoi utilizzare la STATIC parola chiave per creare una colonna statica. Il valore memorizzato in una colonna statica è condiviso tra tutte le righe di una partizione logica. Quando aggiorni il valore di questa colonna, Amazon Keyspaces applica automaticamente la modifica a tutte le righe della partizione.

Questa sezione descrive come calcolare la dimensione codificata dei dati quando scrivi su colonne statiche. Questo processo viene gestito separatamente dal processo di scrittura dei dati nelle colonne non statiche di una riga. Oltre alle quote di dimensione per i dati statici, le operazioni di lettura e scrittura sulle colonne statiche influiscono anche sulla capacità di misurazione e di velocità effettiva delle tabelle in modo indipendente. Per le differenze funzionali con Apache Cassandra quando si utilizzano colonne statiche e risultati di lettura con intervalli di pagine, vedere [the section called "Paginazione"](#)

### Argomenti

- [Calcola la dimensione statica delle colonne per partizione logica in Amazon Keyspaces](#)
- [Stima dei requisiti di capacità di throughput per le operazioni di lettura/scrittura su dati statici in Amazon Keyspaces](#)

## Calcola la dimensione statica delle colonne per partizione logica in Amazon Keyspaces

Questa sezione fornisce dettagli su come stimare la dimensione codificata delle colonne statiche in Amazon Keyspaces. La dimensione codificata viene utilizzata per il calcolo della bolletta e dell'utilizzo delle quote. È inoltre necessario utilizzare la dimensione codificata quando si calcolano i requisiti di

capacità di throughput assegnati per le tabelle. Per calcolare la dimensione codificata delle colonne statiche in Amazon Keyspaces, puoi utilizzare le seguenti linee guida.

- Le chiavi di partizione possono contenere fino a 2048 byte di dati. Ogni colonna chiave nella chiave di partizione richiede fino a 3 byte di metadati. Questi byte di metadati vengono conteggiati ai fini della quota di dimensione dei dati statici di 1 MB per partizione. Quando si calcola la dimensione dei dati statici, è necessario presupporre che ogni colonna chiave di partizione utilizzi tutti i 3 byte di metadati.
- Utilizza la dimensione grezza dei valori dei dati della colonna statica in base al tipo di dati. Per ulteriori informazioni sui tipi di dati, consulta [the section called “Tipi di dati”](#).
- Aggiungi 104 byte alla dimensione dei dati statici per i metadati.
- Le colonne di clustering e le colonne con chiavi non primarie regolari non vengono conteggiate ai fini della dimensione dei dati statici. Per informazioni su come stimare la dimensione dei dati non statici all'interno delle righe, consulta [the section called “Stima della dimensione delle righe”](#)

La dimensione totale codificata di una colonna statica si basa sulla seguente formula:

```
partition key columns + static columns + metadata = total encoded size of static data
```

Considerate il seguente esempio di tabella in cui tutte le colonne sono di tipo intero. La tabella ha due colonne chiave di partizione, due colonne di clustering, una colonna normale e una colonna statica.

```
CREATE TABLE mykeyspace.mytable(pk_col1 int, pk_col2 int, ck_col1 int, ck_col2
int, reg_col1 int, static_col1 int static, primary key((pk_col1, pk_col2),ck_col1,
ck_col2));
```

In questo esempio, calcoliamo la dimensione dei dati statici della seguente dichiarazione:

```
INSERT INTO mykeyspace.mytable (pk_col1, pk_col2, static_col1) values(1,2,6);
```

Per stimare i byte totali richiesti da questa operazione di scrittura, è possibile utilizzare i seguenti passaggi.

1. Calcola la dimensione di una colonna chiave di partizione aggiungendo i byte per il tipo di dati archiviato nella colonna e i byte dei metadati. Ripeti questa operazione per tutte le colonne chiave di partizione.
  - a. Calcola la dimensione della prima colonna della chiave di partizione (pk\_col1):

```
4 bytes for the integer data type + 3 bytes for partition key metadata = 7
bytes
```

- b. Calcola la dimensione della seconda colonna della chiave di partizione (pk\_col2):

```
4 bytes for the integer data type + 3 bytes for partition key metadata = 7
bytes
```

- c. Aggiungi entrambe le colonne per ottenere la dimensione totale stimata delle colonne chiave di partizione:

```
7 bytes + 7 bytes = 14 bytes for the partition key columns
```

2. Aggiungi la dimensione delle colonne statiche. In questo esempio, abbiamo solo una colonna statica che memorizza un numero intero (che richiede 4 byte).
3. Infine, per ottenere la dimensione totale codificata dei dati delle colonne statiche, sommate i byte per le colonne chiave primarie e le colonne statiche e aggiungete i 104 byte aggiuntivi per i metadati:

```
14 bytes for the partition key columns + 4 bytes for the static column + 104 bytes
for metadata = 122 bytes.
```

Puoi anche aggiornare dati statici e non statici con la stessa istruzione. Per stimare la dimensione totale dell'operazione di scrittura, è necessario innanzitutto calcolare la dimensione dell'aggiornamento dei dati non statici. Quindi calcola la dimensione dell'aggiornamento della riga come mostrato nell'esempio in [the section called “Stima della dimensione delle righe”](#) e aggiungi i risultati.

In questo caso, è possibile scrivere un totale di 2 MB: 1 MB è la quota massima per la dimensione delle righe e 1 MB è la quota per la dimensione massima dei dati statici per partizione logica.

Per calcolare la dimensione totale di un aggiornamento di dati statici e non statici nella stessa istruzione, è possibile utilizzare la formula seguente:

```
(partition key columns + static columns + metadata = total encoded size of static data)
+ (partition key columns + clustering columns + regular columns + row metadata = total
encoded size of row)
= total encoded size of data written
```



Considerate il seguente esempio di tabella in cui tutte le colonne sono di tipo intero. La tabella ha due colonne chiave di partizione, due colonne di clustering, una colonna normale e una colonna statica.

```
CREATE TABLE mykeyspace.mytable(pk_col1 int, pk_col2 int, ck_col1 int, ck_col2
int, reg_col1 int, static_col1 int static, primary key((pk_col1, pk_col2),ck_col1,
ck_col2));
```

In questo esempio, calcoliamo la dimensione dei dati quando scriviamo una riga nella tabella, come mostrato nella seguente dichiarazione:

```
INSERT INTO mykeyspace.mytable (pk_col1, pk_col2, ck_col1, ck_col2, reg_col1,
static_col1) values(2,3,4,5,6,7);
```

Per stimare i byte totali richiesti da questa operazione di scrittura, è possibile utilizzare i seguenti passaggi.

1. Calcola la dimensione totale codificata dei dati statici come mostrato in precedenza. In questo esempio, sono 122 byte.
2. Aggiungi la dimensione della dimensione totale codificata della riga in base all'aggiornamento dei dati non statici, seguendo la procedura descritta in [the section called “Stima della dimensione delle righe”](#). In questo esempio, la dimensione totale dell'aggiornamento della riga è di 134 byte.

```
122 bytes for static data + 134 bytes for nonstatic data = 256 bytes.
```

## Stima dei requisiti di capacità di throughput per le operazioni di lettura/scrittura su dati statici in Amazon Keyspaces

I dati statici sono associati alle partizioni logiche in Cassandra, non a singole righe. Le partizioni logiche in Amazon Keyspaces possono avere dimensioni praticamente illimitate se si estendono su più partizioni di storage fisiche. Di conseguenza, i contatori di Amazon Keyspaces scrivono le operazioni su dati statici e non statici separatamente. Inoltre, le scritture che includono dati statici e non statici richiedono operazioni sottostanti aggiuntive per garantire la coerenza dei dati.

Se si esegue un'operazione di scrittura mista di dati statici e non statici, si ottengono due operazioni di scrittura separate, una per dati non statici e una per dati statici. Ciò vale sia per le modalità di capacità di lettura/scrittura su richiesta che per quelle predisposte.

L'esempio seguente fornisce dettagli su come stimare le unità di capacità di lettura (RCUs) e le unità di capacità di scrittura (WCUs) richieste quando si calcolano i requisiti di capacità di throughput assegnata per le tabelle in Amazon Keyspaces con colonne statiche. Puoi stimare la capacità necessaria alla tua tabella per elaborare scritture che includono dati statici e non statici utilizzando la seguente formula:

```
2 x WCUs required for nonstatic data + 2 x WCUs required for static data
```

Ad esempio, se l'applicazione scrive 27 KBs di dati al secondo e ogni scrittura include 25,5 KBs di dati non statici e 1,5 KBs di dati statici, la tabella ne richiede 56 WCUs ( $2 \times 26 \text{ WCUs} + 2 \times 2$ ). WCUs

Amazon Keyspaces misura le letture di dati statici e non statici allo stesso modo delle letture di più righe. Di conseguenza, il prezzo della lettura di dati statici e non statici nella stessa operazione si basa sulla dimensione aggregata dei dati elaborati per eseguire la lettura.

Per scoprire come monitorare le risorse serverless con Amazon CloudWatch, consulta [the section called "Monitoraggio con CloudWatch"](#).

## Stima e fornisci la capacità per una tabella multiregionale in Amazon Keyspaces

È possibile configurare la capacità di throughput di una tabella multiregionale in due modi:

- modalità di capacità su richiesta, misurata in unità di richiesta di scrittura (WRUs)
- Modalità di capacità fornita con scalabilità automatica, misurata in unità di capacità di scrittura (WCUs)

È possibile utilizzare la modalità di capacità assegnata con scalabilità automatica o la modalità di capacità su richiesta per garantire che una tabella multiarea abbia una capacità sufficiente per eseguire scritture replicate su tutti. Regioni AWS

### Note

La modifica della modalità di capacità della tabella in una delle regioni modifica la modalità di capacità per tutte le repliche.

Per impostazione predefinita, Amazon Keyspaces utilizza la modalità on-demand per tabelle multiregionali. Con la modalità on demand, non è necessario specificare la velocità effettiva di lettura e scrittura che si prevede che l'applicazione esegua. Amazon Keyspaces si adatta istantaneamente ai tuoi carichi di lavoro man mano che aumentano o diminuiscono fino a qualsiasi livello di traffico raggiunto in precedenza. Se il livello di traffico di un carico di lavoro raggiunge un nuovo picco, Amazon Keyspaces si adatta rapidamente per adattarsi al carico di lavoro.

Se scegli la modalità di capacità assegnata per una tabella, devi configurare il numero di unità di capacità di lettura (RCUs) e unità di capacità di scrittura (WCUs) al secondo richieste dall'applicazione.

Per pianificare le esigenze di capacità di throughput di una tabella multiregionale, è innanzitutto necessario stimare il numero di velocità WCUs al secondo necessario per ciascuna regione. Quindi aggiungi le scritture da tutte le regioni in cui viene replicata la tabella e utilizza la somma per fornire la capacità per ciascuna regione. Ciò è necessario perché ogni scrittura eseguita in una regione deve essere ripetuta anche in ogni regione di replica.

Se la tabella non ha una capacità sufficiente per gestire le scritture da tutte le regioni, si verificheranno delle eccezioni di capacità. Inoltre, aumenteranno i tempi di attesa per la replica interregionale.

Ad esempio, se disponi di una tabella multiregionale in cui sono previste 5 scritture al secondo negli Stati Uniti orientali (Virginia settentrionale), 10 scritture al secondo negli Stati Uniti orientali (Ohio) e 5 scritture al secondo in Europa (Irlanda), dovresti aspettarti che la tabella consumi 20 scritture WCUs in ciascuna regione: Stati Uniti orientali (Virginia settentrionale), Stati Uniti orientali (Ohio) ed Europa (Irlanda). Ciò significa che in questo esempio è necessario fornire 20 WCUs per ciascuna replica della tabella. Puoi monitorare il consumo di capacità della tua tabella utilizzando Amazon CloudWatch. Per ulteriori informazioni, consulta [the section called “Monitoraggio con CloudWatch”](#).

Ogni scrittura viene fatturata come 1 WCU, quindi in questo esempio vedresti un totale di 60 WCUs fatturate. Per ulteriori informazioni sui prezzi, consulta i prezzi di [Amazon Keyspaces \(per Apache Cassandra\)](#).

Per ulteriori informazioni sulla capacità fornita con la [the section called “Gestisci la capacità di throughput con la scalabilità automatica”](#) scalabilità automatica di Amazon Keyspaces, consulta.

 Note

Se una tabella è in esecuzione in modalità di capacità fornita con scalabilità automatica, la capacità di scrittura assegnata può fluttuare all'interno delle impostazioni di ridimensionamento automatico per ciascuna regione.

## Stima del consumo di capacità di lettura e scrittura con Amazon CloudWatch in Amazon Keyspaces

Per stimare e monitorare il consumo di capacità di lettura e scrittura, puoi utilizzare una CloudWatch dashboard. Per ulteriori informazioni sui parametri disponibili per Amazon Keyspaces, consulta [the section called “Parametri e dimensioni”](#)

Per monitorare le unità di capacità di lettura e scrittura utilizzate da una dichiarazione specifica con CloudWatch, puoi seguire questi passaggi.

1. Crea una nuova tabella con dati di esempio
2. Configura un CloudWatch pannello di controllo Amazon Keyspaces per la tabella. Per iniziare, puoi utilizzare un modello di dashboard disponibile su [Github](#).
3. Esegui l'istruzione CQL, ad esempio utilizzando l'ALLOW FILTERING opzione, e controlla le unità di capacità di lettura utilizzate per la scansione completa della tabella nella dashboard.

## Configura le modalità di capacità di lettura/scrittura in Amazon Keyspaces

Amazon Keyspaces offre due modalità di capacità di lettura/scrittura per l'elaborazione di letture e scritture sulle tabelle:

- Su richiesta (impostazione predefinita)
- Assegnata

La modalità di capacità di lettura/scrittura scelta controlla il modo in cui ti viene addebitato il throughput di lettura e scrittura e come viene gestita la capacità di throughput delle tabelle.

### Argomenti

- [Configura la modalità di capacità su richiesta](#)
- [Configura la modalità di capacità di throughput assegnata](#)
- [Visualizza la modalità di capacità di una tabella in Amazon Keyspaces](#)
- [Cambia la modalità di capacità](#)
- [Pre-riscalda una nuova tabella per la modalità di capacità su richiesta in Amazon Keyspaces](#)
- [Pre-riscalda una tabella esistente per la modalità di capacità su richiesta in Amazon Keyspaces](#)

## Configura la modalità di capacità su richiesta

La modalità di capacità on demand di Amazon Keyspaces (per Apache Cassandra) è un'opzione di fatturazione flessibile in grado di soddisfare migliaia di richieste al secondo senza pianificazione della capacità. Questa opzione offre pay-per-request prezzi per le richieste di lettura e scrittura in modo da pagare solo per ciò che usi.

Quando scegli la modalità on-demand, Amazon Keyspaces può scalare istantaneamente la capacità di throughput della tabella fino a qualsiasi livello di traffico raggiunto in precedenza, per poi ridurla quando il traffico delle applicazioni diminuisce. Se il livello di traffico di un carico di lavoro raggiunge un nuovo picco, il servizio si adatta rapidamente per aumentare la capacità di throughput della tabella. Puoi abilitare la modalità di capacità su richiesta sia per le tabelle nuove che per quelle esistenti.

La modalità su richiesta è una buona opzione se si verifica una delle seguenti condizioni:

- Si creano nuove tabelle con carichi di lavoro non noti.
- Si dispone di un traffico delle applicazioni non prevedibile.
- Si preferisce la facilità di pagamento per l'utilizzo effettivo.

Per iniziare con la modalità on demand, puoi creare una nuova tabella o aggiornare una tabella esistente per utilizzare la modalità di capacità su richiesta utilizzando la console o con alcune righe di codice Cassandra Query Language (CQL). Per ulteriori informazioni, consulta [the section called "Tabelle"](#).

### Argomenti

- [Unità di richiesta di lettura e unità di richiesta di scrittura](#)
- [Proprietà di dimensionamento e picco di traffico](#)

- [Velocità di trasmissione effettiva iniziale per la modalità di capacità on demand](#)

## Unità di richiesta di lettura e unità di richiesta di scrittura

Con le tabelle in modalità di capacità on demand, non è necessario specificare in anticipo la velocità di lettura e scrittura che si prevede venga utilizzata dall'applicazione. Amazon Keyspaces ti addebita le spese di lettura e scrittura eseguite sulle tabelle in termini di unità di richiesta di lettura (RRUs) e unità di richiesta di scrittura (WRUs).

- Una RRU rappresenta una richiesta di LOCAL\_QUORUM lettura o due richieste di LOCAL\_ONE lettura per una riga di dimensioni fino a 4 KB. Se è necessario leggere una riga di dimensioni superiori a 4 KB, l'operazione di lettura utilizza elementi aggiuntivi RRUs. Il numero totale di elementi RRUs richiesti dipende dalla dimensione della riga e dal fatto che si desideri utilizzare LOCAL\_QUORUM o LOCAL\_ONE leggere la coerenza. Ad esempio, per leggere una riga da 8 KB sono necessarie 2 unità che RRUs utilizzano la coerenza di LOCAL\_QUORUM lettura e 1 RRU se si sceglie la coerenza di LOCAL\_ONE lettura.
- Una WRU rappresenta una scrittura per una riga di dimensioni fino a 1 KB. Tutte le scritture utilizzano LOCAL\_QUORUM la coerenza e non sono previsti costi aggiuntivi per l'utilizzo di transazioni leggere (LWTs). Se è necessario scrivere una riga più grande di 1 KB, l'operazione di scrittura utilizza elementi aggiuntivi WRUs. Il numero totale di WRUs elementi richiesti dipende dalla dimensione della riga. Ad esempio, se la dimensione della riga è di 2 KB, ne occorrono 2 WRUs per eseguire una richiesta di scrittura.

Per informazioni sui livelli di coerenza supportati, vedere [the section called “Livelli di coerenza di Cassandra supportati”](#).

## Proprietà di dimensionamento e picco di traffico

Le tabelle Amazon Keyspaces che utilizzano la modalità di capacità su richiesta si adattano automaticamente al volume di traffico dell'applicazione. La modalità di capacità on demand adatta automaticamente fino al doppio del precedente picco di traffico su una tabella. Ad esempio, lo schema di traffico dell'applicazione potrebbe variare tra 5.000 e 10.000 LOCAL\_QUORUM letture al secondo, dove 10.000 letture al secondo era il picco di traffico precedente.

Con questo modello, la modalità di capacità su richiesta supporta istantaneamente un traffico sostenuto fino a 20.000 letture al secondo. Se l'applicazione registra un traffico di 20.000 letture

al secondo, tale picco diventa il nuovo picco precedente, che consente al traffico successivo di raggiungere fino a 40.000 letture al secondo.

Se hai bisogno di più del doppio del picco precedente su una tabella, Amazon Keyspaces alloca automaticamente più capacità all'aumentare del volume di traffico. Questo aiuta a garantire che la tabella abbia una capacità di throughput sufficiente per elaborare le richieste aggiuntive. Tuttavia, è possibile che si verifichino errori di capacità di throughput insufficiente se si supera il doppio del picco precedente entro 30 minuti.

Ad esempio, supponete che lo schema di traffico dell'applicazione vari tra 5.000 e 10.000 letture al secondo con elevata coerenza, dove 20.000 letture al secondo rappresentavano il picco di traffico raggiunto in precedenza. In questo caso, il servizio consiglia di distanziare la crescita del traffico su almeno 30 minuti prima di aumentare fino a 40.000 letture al secondo.

Per informazioni su come stimare il consumo di capacità di lettura e scrittura di una tabella, consulta [the section called “Stima del consumo di capacità”](#)

Per ulteriori informazioni sulle quote predefinite per il tuo account e su come aumentarle, consulta [Quote](#).

## Velocità di trasmissione effettiva iniziale per la modalità di capacità on demand

Se crei una nuova tabella con la modalità di capacità su richiesta abilitata o passi una tabella esistente alla modalità di capacità su richiesta per la prima volta, la tabella presenta le seguenti impostazioni di picco precedenti, anche se in precedenza non serviva traffico utilizzando la modalità di capacità su richiesta:

- Tabella appena creata con modalità di capacità su richiesta: il picco precedente era di 2.000 e 6.000. WRUs RRUs Puoi guidare immediatamente fino al doppio del picco precedente. In questo modo, le tabelle on demand appena create possono servire fino a 4.000 WRUs e 12.000. RRUs
- Tabella esistente passata alla modalità di capacità su richiesta: il picco precedente è la metà del precedente WCUs ed è stato RCUs predisposto per la tabella o le impostazioni di una tabella appena creata con la modalità di capacità su richiesta, a seconda di quale sia la più alta.

## Configura la modalità di capacità di throughput assegnata

Se si sceglie la modalità di capacità di throughput provisioning, si specifica il numero di letture e scritture al secondo necessarie per l'applicazione. Questo ti aiuta a gestire l'utilizzo di Amazon

Keyspaces in modo che rimanga pari o inferiore a una frequenza di richieste definita per mantenere la prevedibilità. Per ulteriori informazioni sulla scalabilità automatica per il throughput assegnato, consulta [the section called “Gestisci la capacità di throughput con la scalabilità automatica”](#)

La modalità Provisioned Throughput Capacity è una buona opzione se si verifica una delle seguenti condizioni:

- Si dispone di un traffico delle applicazioni prevedibile.
- Si eseguono applicazioni il cui traffico è costante o aumenta gradualmente.
- È possibile prevedere i requisiti di capacità.

## Unità di capacità in lettura e unità di capacità in scrittura

Per le tabelle delle modalità di capacità di throughput assegnate, si specifica la capacità di throughput in termini di unità di capacità di lettura (RCUs) e unità di capacità di scrittura (WCUs):

- Una RCU rappresenta una LOCAL\_QUORUM lettura al secondo, o due LOCAL\_ONE letture al secondo, per una riga di dimensioni fino a 4 KB. Se è necessario leggere una riga di dimensioni superiori a 4 KB, l'operazione di lettura utilizza elementi aggiuntivi. RCUs

Il numero totale di elementi RCUs richiesti dipende dalla dimensione della riga e dal fatto che tu voglia LOCAL\_QUORUM o LOCAL\_ONE legga. Ad esempio, se la dimensione della riga è di 8 KB, ne occorrono 2 RCUs per supportare una LOCAL\_QUORUM lettura al secondo e 1 RCU se si sceglie la modalità di lettura. LOCAL\_ONE

- Una WCU rappresenta una scrittura al secondo per una riga di dimensioni fino a 1 KB. Tutte le scritture utilizzano LOCAL\_QUORUM la coerenza e non sono previsti costi aggiuntivi per l'utilizzo di transazioni leggere (LWTs). Se è necessario scrivere una riga più grande di 1 KB, l'operazione di scrittura utilizza elementi aggiuntivi WCUs.

Il numero totale di WCUs elementi richiesti dipende dalla dimensione della riga. Ad esempio, se la dimensione della riga è di 2 KB, ne occorrono 2 WCUs per supportare una richiesta di scrittura al secondo. Per ulteriori informazioni su come stimare il consumo di capacità di lettura e scrittura di una tabella, vedere [the section called “Stima del consumo di capacità”](#).

Se l'applicazione legge o scrive righe più grandi (fino alla dimensione massima delle righe di Amazon Keyspaces di 1 MB), consuma più unità di capacità. Per ulteriori informazioni su come stimare la dimensione delle righe, consulta [the section called “Stima della dimensione delle righe”](#) Si supponga,



ad esempio, di creare una tabella predisposta con 6 RCUs e 6. WCUs Con queste impostazioni, la tua applicazione può eseguire le seguenti operazioni:

- Esegue LOCAL\_QUORUM letture fino a 24 KB al secondo (4 KB × 6). RCUs
- Esegui LOCAL\_ONE letture fino a 48 KB al secondo (il doppio della velocità di lettura).
- Scrivi fino a 6 KB al secondo (1 KB × 6 WCUs).

Il throughput assegnato è la quantità massima di capacità di throughput che un'applicazione può consumare da una tabella. Se l'applicazione supera la capacità di throughput assegnata, è possibile che si verifichino errori di capacità insufficiente.

Ad esempio, una richiesta di lettura che non ha una capacità di throughput sufficiente ha esito negativo con un'`Read_Timeout` eccezione e viene registrata nella metrica. `ReadThrottleEvents`  
Una richiesta di scrittura che non ha una capacità di throughput sufficiente ha esito negativo con un'`Write_Timeout` eccezione e viene registrata nella metrica. `WriteThrottleEvents`

Puoi utilizzare Amazon CloudWatch per monitorare le metriche di throughput fornite ed effettive e gli eventi di capacità insufficiente. Per ulteriori informazioni su questi parametri, consulta [the section called “Parametri e dimensioni”](#).

#### Note

Errori ripetuti dovuti a una capacità insufficiente possono portare a eccezioni specifiche dei driver lato client, ad esempio il DataStax driver Java non funziona con un. `NoHostAvailableException`

Per modificare le impostazioni della capacità di throughput per le tabelle, è possibile utilizzare l'istruzione Console di gestione AWS o l'`ALTER TABLE` istruzione using CQL. Per ulteriori informazioni, vedere. [the section called “ALTER TABLE”](#)

Per ulteriori informazioni sulle quote predefinite per il tuo account e su come aumentarle, consulta. [Quote](#)

## Visualizza la modalità di capacità di una tabella in Amazon Keyspaces

Puoi interrogare la tabella di sistema nello spazio delle chiavi di sistema Amazon Keyspaces per esaminare le informazioni sulla modalità di capacità relative a una tabella. Puoi anche verificare

se una tabella utilizza la modalità di capacità di throughput su richiesta o fornita. Se la tabella è configurata con la modalità di capacità di throughput assegnata, è possibile visualizzare la capacità di throughput fornita per la tabella.

È inoltre possibile utilizzare la AWS CLI per visualizzare la modalità di capacità di una tabella.

Per modificare la velocità effettiva assegnata a una tabella, vedere. [the section called “Cambia la modalità di capacità”](#)

## Cassandra Query Language (CQL)

### Esempio

```
SELECT * from system_schema_mcs.tables where keyspace_name = 'mykeyspace' and
table_name = 'mytable';
```

Una tabella configurata con la modalità di capacità su richiesta restituisce quanto segue.

```
{
 "capacity_mode":{
 "last_update_to_pay_per_request_timestamp":"1579551547603",
 "throughput_mode":"PAY_PER_REQUEST"
 }
}
```

Una tabella configurata con la modalità di capacità di throughput assegnata restituisce quanto segue.

```
{
 "capacity_mode":{
 "last_update_to_pay_per_request_timestamp":"1579048006000",
 "read_capacity_units":"5000",
 "throughput_mode":"PROVISIONED",
 "write_capacity_units":"6000"
 }
}
```

Il `last_update_to_pay_per_request_timestamp` valore viene misurato in millisecondi.

## CLI

Visualizza la modalità di capacità di throughput di una tabella utilizzando il AWS CLI

```
aws keyspaces get-table --keyspace-name myKeyspace --table-name myTable
```

L'output del comando può essere simile a questo per una tabella in modalità di capacità fornita.

```
"capacitySpecification": {
 "throughputMode": "PROVISIONED",
 "readCapacityUnits": 4000,
 "writeCapacityUnits": 2000
}
```

L'output di una tabella in modalità su richiesta è simile al seguente.

```
"capacitySpecification": {
 "throughputMode": "PAY_PER_REQUEST",
 "lastUpdateToPayPerRequestTimestamp": "2024-10-03T10:48:19.092000+00:00"
}
```

## Cambia la modalità di capacità

Quando passi una tabella dalla modalità di capacità fornita alla modalità di capacità su richiesta, Amazon Keyspaces apporta diverse modifiche alla struttura della tabella e delle partizioni. Questo processo può richiedere alcuni minuti. Durante il periodo di passaggio, la tabella offre un throughput coerente con gli importi di WCU e RCU precedentemente forniti.

Quando si passa dalla modalità di capacità su richiesta alla modalità di capacità fornita, la tabella offre un throughput coerente con il picco precedente raggiunto quando la tabella era impostata sulla modalità di capacità su richiesta.

I seguenti periodi di attesa si applicano quando si cambia modalità di capacità:

- È possibile passare da una tabella appena creata in modalità on demand alla modalità di capacità fornita in qualsiasi momento. Tuttavia, è possibile tornare alla modalità on demand solo 24 ore dopo il timestamp di creazione della tabella.
- È possibile passare da una tabella esistente in modalità on-demand alla modalità di capacità assegnata in qualsiasi momento. Tuttavia, è possibile passare dalla modalità di capacità fornita a quella su richiesta solo una volta nell'arco di 24 ore.

## Cassandra Query Language (CQL)

Modifica la modalità di capacità di throughput di una tabella utilizzando CQL

1. Per modificare la modalità di capacità di una tabella, è `PROVISIONED` necessario configurare le unità di capacità di lettura e scrittura in base ai valori di picco previsti del carico di lavoro. La seguente dichiarazione ne è un esempio. È inoltre possibile eseguire questa istruzione per regolare la capacità di lettura o le unità di capacità di scrittura della tabella.

```
ALTER TABLE catalog.book_awards WITH CUSTOM_PROPERTIES={'capacity_mode':
{'throughput_mode': 'PROVISIONED', 'read_capacity_units': 6000,
 'write_capacity_units': 3000}};
```

Per configurare la modalità di capacità fornita con auto-scaling, vedere. [the section called “Configura il ridimensionamento automatico su una tabella esistente”](#)

2. Per modificare la modalità di capacità di una tabella in modalità on demand, imposta la modalità di throughput su. `PAY_PER_REQUEST` La seguente dichiarazione ne è un esempio.

```
ALTER TABLE catalog.book_awards WITH CUSTOM_PROPERTIES={'capacity_mode':
{'throughput_mode': 'PAY_PER_REQUEST'}};
```

3. È possibile utilizzare la seguente istruzione per confermare la modalità di capacità della tabella.

```
SELECT * from system_schema.tables where keyspace_name = 'catalog' and
table_name = 'book_awards';
```

Una tabella configurata con la modalità di capacità su richiesta restituisce quanto segue.

```
{
 "capacity_mode":{
 "last_update_to_pay_per_request_timestamp":"1727952499092",
 "throughput_mode":"PAY_PER_REQUEST"
 }
}
```

Il `last_update_to_pay_per_request_timestamp` valore viene misurato in millisecondi.

## CLI

Modificate la modalità di capacità di trasmissione di una tabella utilizzando il AWS CLI

1. Per modificare la modalità di capacità della tabella, è PROVISIONED necessario configurare le unità di capacità di lettura e scrittura in base ai valori di picco previsti del carico di lavoro. Il comando seguente ne è un esempio. È inoltre possibile eseguire questo comando per regolare la capacità di lettura o le unità di capacità di scrittura della tabella.

```
aws keyspaces update-table --keyspace-name catalog --table-name book_awards
 \--capacity-specification
 throughputMode=PROVISIONED,readCapacityUnits=6000,writeCapacityUnits=3000
```

Per configurare la modalità di capacità fornita con auto-scaling, vedere. [the section called “Configura il ridimensionamento automatico su una tabella esistente”](#)

2. Per modificare la modalità di capacità di una tabella in modalità on demand, è necessario impostare la modalità di throughput su. PAY\_PER\_REQUEST La seguente dichiarazione ne è un esempio.

```
aws keyspaces update-table --keyspace-name catalog --table-name book_awards
 \--capacity-specification
 throughputMode=PAY_PER_REQUEST
```

3. È possibile utilizzare il comando seguente per esaminare la modalità di capacità configurata per una tabella.

```
aws keyspaces get-table --keyspace-name catalog --table-name book_awards
```

L'output di una tabella in modalità su richiesta ha il seguente aspetto.

```
"capacitySpecification": {
 "throughputMode": "PAY_PER_REQUEST",
 "lastUpdateToPayPerRequestTimestamp": "2024-10-03T10:48:19.092000+00:00"
}
```

## Pre-riscalda una nuova tabella per la modalità di capacità su richiesta in Amazon Keyspaces

Amazon Keyspaces ridimensiona automaticamente le partizioni di storage in base al throughput, ma per nuove tabelle o nuovi picchi di throughput, l'allocazione delle partizioni di storage richieste può richiedere più tempo. Per garantire che le tabelle in modalità on-demand e con capacità fornita dispongano di partizioni di storage sufficienti a supportare un aumento improvviso del throughput, puoi preriscaldare una tabella nuova o esistente.

Uno scenario comune per il preriscaldamento di una nuova tabella è la migrazione di dati da un altro database, che potrebbe richiedere il caricamento di terabyte di dati in un breve periodo di tempo.

Per le tabelle su richiesta, Amazon Keyspaces alloca automaticamente più capacità all'aumentare del volume di traffico. Le nuove tabelle su richiesta possono supportare fino a 4.000 scritture al secondo e 12.000 letture fortemente coerenti o 24.000 letture eventualmente coerenti al secondo. Una tabella su richiesta aumenta il traffico in base alla velocità effettiva registrata in precedenza nel tempo.

Se si prevede un picco di capacità di picco superiore alle impostazioni per le nuove tabelle, è possibile preriscaldare la tabella fino alla capacità massima del picco previsto.

Per preriscaldare una nuova tabella per la modalità di capacità su richiesta in Amazon Keyspaces, puoi seguire questi passaggi. Per preriscaldare una tabella esistente, consulta [the section called “Preriscalda una tabella esistente per aumentarne la capacità su richiesta”](#)

Prima di iniziare, controlla le [quote relative all'account e alla tabella per verificare se](#) la modalità provisioning è attiva e modificala se necessario.

### Console

Come preriscaldare un nuovo tavolo per la modalità di capacità su richiesta

1. [Accedi a e apri Console di gestione AWS la console Amazon Keyspaces a casa](https://console.aws.amazon.com/keyspaces/)<https://console.aws.amazon.com/keyspaces/>.
2. Nel pannello di navigazione, scegli Tabelle, quindi seleziona Crea tabella.
3. Nella pagina Crea tabella, nella sezione Dettagli della tabella, seleziona un keyspace e fornisci un nome per la nuova tabella.
4. Nella sezione Colonne, crea lo schema per la tua tabella.
5. Nella sezione Chiave primaria, definisci la chiave primaria della tabella e seleziona le colonne di clustering opzionali.

6. Nella sezione Impostazioni tabella, scegli Personalizza impostazioni.
7. Continua con le impostazioni della capacità di lettura/scrittura.
8. Per la modalità Capacity, scegli Provisioned.
9. Nella sezione Capacità di lettura, deselecta Scala automaticamente.

Imposta le unità di capacità fornite della tabella sul valore di picco previsto.

10. Nella sezione Capacità di scrittura, scegli le stesse impostazioni definite nel passaggio precedente per la capacità di lettura o configura i valori della capacità manualmente.
11. Scegliere Create table (Crea tabella). La tabella viene creata con le impostazioni di capacità specificate.
12. Quando lo stato della tabella diventa Attivo, puoi passare alla modalità di capacità su richiesta.

## Cassandra Query Language (CQL)

Preriscalda una nuova tabella per la modalità on-demand utilizzando CQL

1. Crea una nuova tabella in modalità provisioning e specifica la capacità di picco prevista per le letture e le scritture per la nuova tabella. La seguente dichiarazione ne è un esempio.

```
CREATE TABLE catalog.book_awards (
 year int,
 award text,
 rank int,
 category text,
 book_title text,
 author text,
 publisher text,
 PRIMARY KEY ((year, award), category, rank))
WITH CUSTOM_PROPERTIES = {
 'capacity_mode': {
 'throughput_mode': 'PROVISIONED',
 'read_capacity_units': 18000,
 'write_capacity_units': 6000
 }
};
```

2. Conferma lo stato della tabella. È possibile utilizzare la seguente dichiarazione.

```
SELECT keyspace_name, table_name, status FROM system_schema_mcs.tables WHERE
keyspace_name = 'catalog' AND table_name = 'book_awards';
```

| keyspace_name | table_name  | status |
|---------------|-------------|--------|
| catalog       | book_awards | ACTIVE |

(1 rows)

- Quando lo stato della tabella è impostato su **ACTIVE**, è possibile utilizzare la seguente istruzione per modificare la modalità di capacità della tabella in modalità su richiesta impostando la modalità di trasmissione su **PAY\_PER\_REQUEST**. La seguente dichiarazione ne è un esempio.

```
ALTER TABLE catalog.book_awards WITH CUSTOM_PROPERTIES={'capacity_mode':
{'throughput_mode': 'PAY_PER_REQUEST'}};
```

- È possibile utilizzare la seguente istruzione per confermare che la tabella è ora in modalità su richiesta e visualizzare lo stato della tabella.

```
SELECT * from system_schema_mcs.tables where keyspace_name = 'catalog' and
table_name = 'book_awards';
```

## CLI

Preriscaldare una nuova tabella per la modalità di capacità su richiesta utilizzando il AWS CLI

- Crea una nuova tabella in modalità provisioning e specifica i valori di capacità di picco previsti per le letture e le scritture per la nuova tabella. La seguente dichiarazione ne è un esempio.

```
aws keyspaces create-table --keyspace-name catalog --table-name book_awards
 \--schema-definition
 'allColumns=[{name=pk,type=int},{name=ck,type=int}],partitionKeys=[{name=pk},
 {name=ck}]'
 \--capacity-specification
 throughputMode=PROVISIONED,readCapacityUnits=18000,writeCapacityUnits=6000
```

- Conferma lo stato della tabella. È possibile utilizzare la seguente dichiarazione.



```
aws keyspaces get-table --keyspace-name catalog --table-name book_awards
```

3. Quando la tabella è attiva e la capacità è stata fornita, è possibile modificare la tabella in modalità su richiesta. Di seguito è riportato un esempio di ciò.

```
aws keyspaces update-table --keyspace-name catalog --table-name book_awards --
capacity-specification throughputMode=PAY_PER_REQUEST
```

4. È possibile utilizzare la seguente istruzione per confermare che la tabella è ora in modalità su richiesta e visualizzare lo stato della tabella.

```
aws keyspaces get-table --keyspace-name catalog --table-name book_awards
```

Quando la tabella è attiva in modalità di capacità su richiesta, è pronta a gestire una capacità di throughput simile a quella precedente in modalità di capacità fornita.

## Pre-riscalda una tabella esistente per la modalità di capacità su richiesta in Amazon Keyspaces

Amazon Keyspaces ridimensiona automaticamente le partizioni di storage in base al throughput, ma per nuove tabelle o nuovi picchi di throughput, l'allocazione delle partizioni di storage richieste può richiedere più tempo. Per garantire che le tabelle in modalità on-demand e con capacità fornita dispongano di partizioni di storage sufficienti a supportare un aumento improvviso del throughput, puoi preriscaldare una tabella nuova o esistente.

Se si prevede un picco di capacità di picco del tavolo pari al doppio rispetto al picco precedente entro gli stessi 30 minuti, è possibile preriscaldare il tavolo fino alla capacità massima del picco previsto.

Per preriscaldare una tabella on-demand esistente in Amazon Keyspaces, puoi seguire questi passaggi. Per preriscaldare un nuovo tavolo, consulta [the section called “Preriscalda un nuovo tavolo per sfruttare la capacità su richiesta”](#)

Prima di iniziare, controlla le [quote relative all'account e alla tabella per verificare se](#) la modalità provisioning è attiva e modificala se necessario.

Quindi esamina i [periodi di attesa](#) richiesti tra le diverse modalità di capacità. Tieni presente che fino a quando la tabella non tornerà in modalità on-demand, dovrai sostenere i costi relativi alla capacità fornita.

## Console

Come preriscaldare una tabella esistente in modalità on-demand

1. [Accedi a e apri Console di gestione AWS la console Amazon Keyspaces a casa](https://console.aws.amazon.com/keyspaces/)<https://console.aws.amazon.com/keyspaces/>.
2. Scegli la tabella su cui vuoi lavorare e vai alla scheda Capacità.
3. Nella sezione Impostazioni di capacità, scegli Modifica.
4. In modalità Capacity, imposta la tabella in modalità Provisioned capacity.
5. Nella sezione Capacità di lettura, deseleziona Scala automaticamente.

Imposta le unità di capacità fornite della tabella sul valore di picco previsto.

6. Nella sezione Capacità di scrittura, scegli le stesse impostazioni definite nel passaggio precedente per la capacità di lettura o configura i valori della capacità manualmente.
7. Una volta definite le impostazioni della capacità assegnata, scegliete Salva. Dopo aver salvato le modifiche, lo stato della tabella viene visualizzato come Aggiornamento in corso... fino a quando non viene fornita la capacità. Si noti che per le tabelle di grandi dimensioni, il processo di preriscaldamento può richiedere del tempo, poiché i dati devono essere suddivisi tra le partizioni. Durante questo periodo, puoi continuare ad accedere alla tabella e aspettarti che la capacità di picco precedentemente configurata sia disponibile.
8. Quando lo stato della tabella diventa Attivo, puoi riportare la tabella alla modalità di capacità su richiesta.

## Cassandra Query Language (CQL)

Preriscalda una tabella esistente per la modalità su richiesta utilizzando CQL

1. Cambia la modalità di capacità della tabella PROVISIONED e configura la capacità di lettura e la capacità di scrittura in base ai valori di picco previsti.

```
ALTER TABLE catalog.book_awards WITH CUSTOM_PROPERTIES={'capacity_mode':
{'throughput_mode': 'PROVISIONED', 'read_capacity_units': 18000,
'write_capacity_units': 6000}};
```

2. Conferma che la tabella sia attiva. La seguente dichiarazione ne è un esempio.

```
SELECT * from system_schema_mcs.tables where keyspace_name = 'catalog' and
table_name = 'book_awards';
```

- Quando lo stato della tabella è **ACTIVE**, è possibile utilizzare la seguente istruzione per modificare la modalità di capacità della tabella in modalità su richiesta impostando la modalità di trasmissione su **PAY\_PER\_REQUEST**. La seguente dichiarazione ne è un esempio.

```
ALTER TABLE catalog.book_awards WITH CUSTOM_PROPERTIES={'capacity_mode':
{'throughput_mode': 'PAY_PER_REQUEST'}};
```

- È possibile utilizzare la seguente istruzione per confermare che la tabella è ora in modalità su richiesta e visualizzare lo stato della tabella.

```
SELECT * from system_schema_mcs.tables where keyspace_name = 'catalog' and
table_name = 'book_awards';
```

## CLI

Preriscaldare una tabella esistente per la modalità su richiesta utilizzando il AWS CLI

- Cambia la modalità di capacità della tabella **PROVISIONED** e configura la capacità di lettura e la capacità di scrittura in base ai valori di picco previsti. Il comando seguente ne è un esempio.

```
aws keyspaces update-table --keyspace-name catalog --table-name book_awards
 \--capacity-specification
 throughputMode=PROVISIONED,readCapacityUnits=18000,writeCapacityUnits=6000
```

- Verificate che lo stato della tabella sia attivo e che la capacità sia stata fornita. È possibile utilizzare la seguente dichiarazione.

```
aws keyspaces get-table --keyspace-name catalog --table-name book_awards
```

- Quando lo stato della tabella è impostato **ACTIVE** e la capacità è stata fornita, è possibile utilizzare la seguente istruzione per modificare la modalità di capacità della tabella in modalità su richiesta impostando la modalità di trasmissione su **PAY\_PER\_REQUEST**. La seguente dichiarazione ne è un esempio.

```
aws keyspaces update-table --keyspace-name catalog --table-name book_awards
 \--capacity-specification
 throughputMode=PAY_PER_REQUEST
```

4. È possibile utilizzare la seguente istruzione per confermare che la tabella è ora in modalità su richiesta e visualizzare lo stato della tabella.

```
aws keyspaces get-table --keyspace-name catalog --table-name book_awards
```

Quando la tabella è attiva in modalità di capacità su richiesta, è pronta a gestire una capacità di throughput simile a quella precedente in modalità di capacità fornita.

## Gestisci automaticamente la capacità di throughput con la scalabilità automatica di Amazon Keyspaces

Molti carichi di lavoro dei database sono ciclici per natura o sono difficili da prevedere in anticipo. Ad esempio, considera un'app di social network in cui la maggior parte degli utenti sono attivi durante le ore diurne. Il database deve essere in grado di gestire l'attività diurna e ciò non è necessario per gli stessi livelli di throughput durante la notte.

Un altro esempio potrebbe essere una nuova app di gioco per dispositivi mobili che sta registrando una nuova adozione. Se il gioco diventasse molto popolare, potrebbe superare le risorse di database disponibili, con conseguente rallentamento delle prestazioni e insoddisfazione dei clienti. Questi tipi di carichi di lavoro richiedono spesso l'intervento manuale per aumentare o diminuire le risorse di database in risposta alla variazione dei livelli di utilizzo.

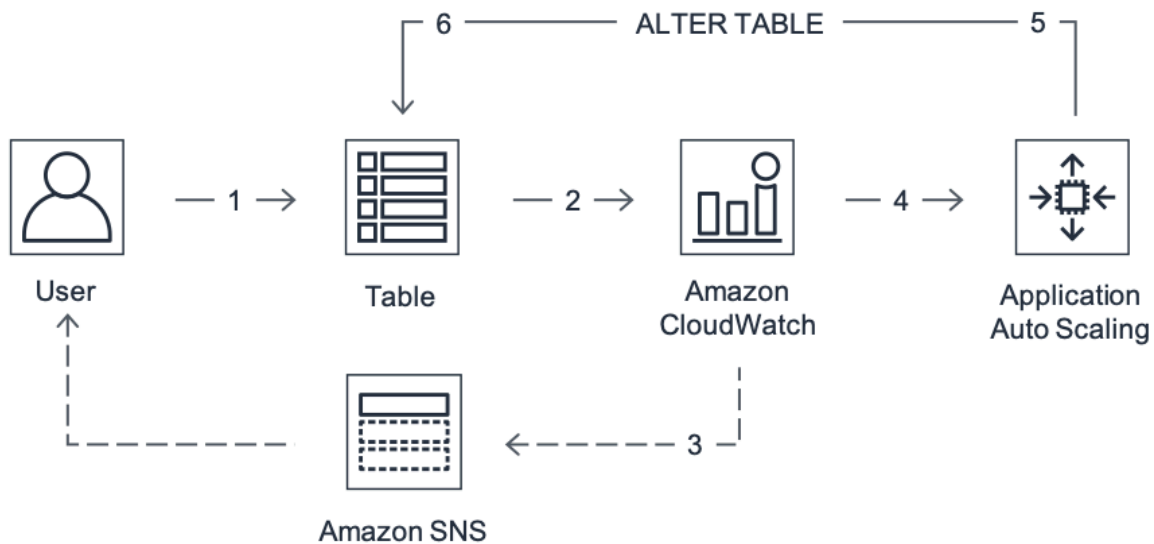
Amazon Keyspaces (per Apache Cassandra) ti aiuta a fornire la capacità di throughput in modo efficiente per carichi di lavoro variabili regolando automaticamente la capacità di throughput in risposta al traffico effettivo delle applicazioni. Amazon Keyspaces utilizza il servizio Application Auto Scaling per aumentare e diminuire la capacità di lettura e scrittura di una tabella per tuo conto. Per ulteriori informazioni su Application Auto Scaling, consultate la [Application Auto Scaling](#) User Guide.

### Note

Per iniziare rapidamente a usare la scalabilità automatica di Amazon Keyspaces, consulta [the section called “Configurazione e aggiornamento delle politiche di scalabilità automatica”](#)

## Come funziona il ridimensionamento automatico di Amazon Keyspaces

Il diagramma seguente fornisce una panoramica di alto livello di come il ridimensionamento automatico di Amazon Keyspaces gestisce la capacità di throughput per una tabella.



Per abilitare il ridimensionamento automatico per una tabella, crei una politica di ridimensionamento. La politica di scalabilità specifica se si desidera scalare la capacità di lettura o di scrittura (o entrambe) e le impostazioni dell'unità di capacità minima e massima assegnata per la tabella.

La politica di scalabilità definisce anche l'utilizzo previsto. L'utilizzo obiettivo è il rapporto tra le unità di capacità consumate e le unità di capacità fornite in un determinato momento, espresso in percentuale. Il ridimensionamento automatico utilizza un algoritmo di tracciamento del target per regolare il throughput assegnato della tabella verso l'alto o verso il basso in risposta ai carichi di lavoro effettivi. Lo fa in modo che l'utilizzo effettivo della capacità rimanga pari o vicino all'utilizzo previsto.

È possibile impostare i valori di utilizzo target di scalabilità automatica tra il 20 e il 90 per cento per la capacità di lettura e scrittura. Il tasso di utilizzo target predefinito è del 70 per cento. È possibile impostare l'utilizzo target su una percentuale inferiore se il traffico cambia rapidamente e si desidera che la capacità inizi a crescere prima. È inoltre possibile impostare il tasso di utilizzo target su un tasso più elevato se il traffico delle applicazioni cambia più lentamente e si desidera ridurre il costo del throughput.

Per ulteriori informazioni sulle politiche di scalabilità, consulta [Target tracking scaling policy for Application Auto Scaling nella Application Auto Scaling User Guide](#).

Quando crei una politica di scalabilità, Amazon Keyspaces crea due coppie di allarmi CloudWatch Amazon per tuo conto. Ogni coppia rappresenta i limiti superiore e inferiore per le impostazioni di throughput assegnate e consumate. Questi CloudWatch allarmi vengono attivati quando l'utilizzo effettivo della tabella si discosta dall'utilizzo previsto per un periodo di tempo prolungato. Per ulteriori informazioni su Amazon CloudWatch, consulta la [Amazon CloudWatch User Guide](#).

Quando viene attivato uno degli CloudWatch allarmi, Amazon Simple Notification Service (Amazon SNS) ti invia una notifica (se l'hai abilitata). L' CloudWatch allarme richiama quindi Application Auto Scaling per valutare la politica di scalabilità. Questo a sua volta invia una richiesta Alter Table ad Amazon Keyspaces per aumentare o diminuire la capacità fornita della tabella, a seconda dei casi. Per ulteriori informazioni sulle notifiche di Amazon SNS, consulta [Configurazione delle notifiche Amazon SNS](#).

Amazon Keyspaces elabora la richiesta Alter Table aumentando (o diminuendo) la capacità di throughput assegnata alla tabella in modo che si avvicini all'utilizzo previsto.

#### Note

La scalabilità automatica di Amazon Keyspaces modifica le impostazioni di throughput assegnate solo quando il carico di lavoro effettivo rimane elevato (o ridotto) per un periodo prolungato di diversi minuti. L'algoritmo di monitoraggio obiettivi di cerca di tenere l'utilizzo di destinazione pari o vicino al valore scelto a lungo termine. I picchi di attività improvvisi e di breve durata sono soddisfatti dalla capacità di ottimizzazione integrata della tabella.

## Come funziona la scalabilità automatica per le tabelle multiregionali

Per garantire che ci sia sempre una capacità di lettura e scrittura sufficiente per tutte le repliche di tabella in una tabella multiregionale in modalità Regioni AWS di capacità fornita, ti consigliamo di configurare la scalabilità automatica di Amazon Keyspaces.

Quando si utilizza una tabella multiarea in modalità provisioning con ridimensionamento automatico, non è possibile disabilitare il ridimensionamento automatico per una singola replica di tabella. Ma puoi regolare le impostazioni di ridimensionamento automatico di lettura della tabella per diverse regioni. Ad esempio, è possibile specificare diverse impostazioni di capacità di lettura e di ridimensionamento automatico di lettura per ogni regione in cui viene replicata la tabella.

Le impostazioni di lettura automatica configurate per una replica di tabella in una regione specificata sovrascrivono le impostazioni generali di ridimensionamento automatico della tabella. La capacità

di scrittura, tuttavia, deve rimanere sincronizzata su tutte le repliche delle tabelle per garantire una capacità sufficiente per replicare le scritture in tutte le regioni.

La scalabilità automatica di Amazon Keyspaces aggiorna in modo indipendente la capacità fornita della tabella in ciascuna regione in Regione AWS base all'utilizzo in quella regione. Di conseguenza, la capacità fornita in ciascuna regione per una tabella multiarea potrebbe essere diversa quando la scalabilità automatica è attiva.

Puoi configurare le impostazioni di ridimensionamento automatico di una tabella multiregionale e delle relative repliche utilizzando la console Amazon Keyspaces, l'API o CQL. AWS CLI Per ulteriori informazioni su come creare e aggiornare le impostazioni di ridimensionamento automatico per le tabelle multiregionali, vedere. [the section called “Aggiornamento della capacità assegnata e delle impostazioni di ridimensionamento automatico per una tabella multiregionale”](#)

#### Note

Se utilizzi la scalabilità automatica per tabelle multiregionali, devi sempre utilizzare le operazioni dell'API Amazon Keyspaces per configurare le impostazioni di auto scaling. Se utilizzi direttamente le operazioni dell'API Application Auto Scaling per configurare le impostazioni di auto scaling, non hai la possibilità Regioni AWS di specificare la tabella Multi-regione. Ciò può comportare configurazioni non supportate.

## Note per l'utilizzo

Prima di iniziare a utilizzare il ridimensionamento automatico di Amazon Keyspaces, è necessario tenere presente quanto segue:

- Il ridimensionamento automatico di Amazon Keyspaces non è disponibile nella regione del Medio Oriente (Emirati Arabi Uniti).
- La scalabilità automatica di Amazon Keyspaces può aumentare la capacità di lettura o scrittura ogni volta che è necessario, in conformità con la tua politica di scalabilità. Tutte le quote di Amazon Keyspaces rimangono in vigore, come descritto in. [Quote](#)
- Il ridimensionamento automatico di Amazon Keyspaces non impedisce di modificare manualmente le impostazioni di throughput assegnate. Queste regolazioni manuali non influiscono sugli CloudWatch allarmi esistenti associati alla politica di scalabilità.
- Se utilizzi la console per creare una tabella con capacità di throughput assegnata, la scalabilità automatica di Amazon Keyspaces è abilitata per impostazione predefinita. Puoi modificare le

impostazioni di ridimensionamento automatico in qualsiasi momento. Per ulteriori informazioni, consulta [the section called “Disattiva la scalabilità automatica di Amazon Keyspaces per una tabella”](#).

- Se utilizzi CloudFormation per creare policy di scalabilità, dovresti gestire le policy di scalabilità in CloudFormation modo che lo stack sia sincronizzato con il modello di stack. Se modifichi le politiche di scalabilità da Amazon Keyspaces, queste verranno sovrascritte con i valori originali del modello di stack quando CloudFormation lo stack viene reimpostato.
- Se utilizzi la scalabilità automatica di Amazon Keyspaces CloudTrail per monitorare la scalabilità automatica di Amazon Keyspaces, potresti ricevere avvisi per le chiamate effettuate da Application Auto Scaling come parte del processo di convalida della configurazione. Puoi filtrare questi avvisi utilizzando il `invokedBy` campo che contiene questi controlli di convalida. `application-autoscaling.amazonaws.com`

## Configura e aggiorna le politiche di scalabilità automatica di Amazon Keyspaces

Puoi utilizzare la console, CQL o AWS Command Line Interface (AWS CLI) per configurare la scalabilità automatica di Amazon Keyspaces per tabelle nuove ed esistenti. Puoi anche modificare le impostazioni di ridimensionamento automatico o disabilitare il ridimensionamento automatico.

Per funzionalità più avanzate come l'impostazione dei tempi di cooldown scale-in e scale-out, ti consigliamo di utilizzare CQL o il per gestire le politiche di scalabilità di AWS CLI Amazon Keyspaces.

### Argomenti

- [Configura le autorizzazioni per la scalabilità automatica di Amazon Keyspaces](#)
- [Crea una nuova tabella con ridimensionamento automatico](#)
- [Configura il ridimensionamento automatico su una tabella esistente](#)
- [Visualizza la configurazione di scalabilità automatica di Amazon Keyspaces della tua tabella](#)
- [Disattiva la scalabilità automatica di Amazon Keyspaces per una tabella](#)
- [Visualizza l'attività di scalabilità automatica per una tabella Amazon Keyspaces in Amazon CloudWatch](#)



## Configura le autorizzazioni per la scalabilità automatica di Amazon Keyspaces

Per iniziare, verifica che il responsabile disponga delle autorizzazioni appropriate per creare e gestire le impostazioni di ridimensionamento automatico. In AWS Identity and Access Management (IAM), la policy AWS gestita `AmazonKeyspacesFullAccess` è necessaria per gestire le policy di scalabilità di Amazon Keyspaces.

### Important

`application-autoscaling`: \*sono necessarie autorizzazioni per disabilitare il ridimensionamento automatico su una tabella. È necessario disattivare il ridimensionamento automatico per una tabella prima di poterla eliminare.

Per configurare un utente o un ruolo IAM per l'accesso alla console Amazon Keyspaces e la scalabilità automatica di Amazon Keyspaces, aggiungi la seguente policy.

Per allegare la policy **AmazonKeyspacesFullAccess**

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
2. Nella dashboard della console IAM, scegli Utenti, quindi scegli il tuo utente o ruolo IAM dall'elenco.
3. Nella pagina Summary (Riepilogo), scegli Add permissions (Aggiungi autorizzazioni).
4. Scegli Attach existing policies directly (Collega direttamente le policy esistenti).
5. Dall'elenco delle politiche, scegli `AmazonKeyspacesFullAccess`, quindi scegli Avanti: revisione.
6. Scegli Add Permissions (Aggiungi autorizzazioni).

## Crea una nuova tabella con ridimensionamento automatico

Quando crei una nuova tabella Amazon Keyspaces, puoi abilitare automaticamente la scalabilità automatica per la capacità di scrittura o lettura della tabella. Ciò consente ad Amazon Keyspaces di contattare Application Auto Scaling per tuo conto per registrare la tabella come destinazione scalabile e regolare la capacità di scrittura o lettura fornita.

Per ulteriori informazioni su come creare una tabella multiregionale e configurare diverse impostazioni di ridimensionamento automatico per le repliche di tabelle, vedere [the section called “Crea una tabella multiregionale in modalità provisioning”](#)

#### Note

La scalabilità automatica di Amazon Keyspaces richiede la presenza di un ruolo collegato al servizio (AWSServiceRoleForApplicationAutoScaling\_CassandraTable) che esegue azioni di ridimensionamento automatico per tuo conto. Questo ruolo viene creato automaticamente per te. Per ulteriori informazioni, consulta [the section called “Uso di ruoli collegati ai servizi”](#).

## Console

Crea una nuova tabella con ridimensionamento automatico abilitato utilizzando la console

1. [Accedi a e apri Console di gestione AWS la console Amazon Keyspaces a casahttps://console.aws.amazon.com/keyspaces/.](https://console.aws.amazon.com/keyspaces/)
2. Nel pannello di navigazione, scegli Tabelle, quindi seleziona Crea tabella.
3. Nella pagina Crea tabella, nella sezione Dettagli della tabella, seleziona un keyspace e fornisci un nome per la nuova tabella.
4. Nella sezione Colonne, crea lo schema per la tua tabella.
5. Nella sezione Chiave primaria, definisci la chiave primaria della tabella e seleziona le colonne di clustering opzionali.
6. Nella sezione Impostazioni tabella, scegli Personalizza impostazioni.
7. Continua con le impostazioni della capacità di lettura/scrittura.
8. Per la modalità Capacity, scegli Provisioned.
9. Nella sezione Capacità di lettura, conferma che l'opzione Scale automaticamente sia selezionata.

In questo passaggio, si selezionano le unità di capacità di lettura minima e massima per la tabella, nonché l'utilizzo previsto.

- Unità di capacità minima: immettete il valore per il livello minimo di velocità effettiva che la tabella deve essere sempre pronta a supportare. Il valore deve essere compreso tra 1 e la

quota massima di throughput al secondo prevista per l'account (40.000 per impostazione predefinita).

- Unità di capacità massima: inserisci la quantità massima di velocità effettiva che desideri fornire per la tabella. Il valore deve essere compreso tra 1 e la quota massima di throughput al secondo prevista per l'account (40.000 per impostazione predefinita).
- Utilizzo obiettivo: inserisci un tasso di utilizzo obiettivo compreso tra il 20% e il 90%. Quando il traffico supera il tasso di utilizzo target definito, la capacità viene automaticamente aumentata. Quando il traffico scende al di sotto dell'obiettivo definito, viene nuovamente ridimensionato automaticamente.

#### Note

Per ulteriori informazioni sulle quote predefinite per il tuo account e su come aumentarle, consulta. [Quote](#)

10. Nella sezione Capacità di scrittura, scegli le stesse impostazioni definite nel passaggio precedente per la capacità di lettura o configura i valori della capacità manualmente.
11. Scegliere Create table (Crea tabella). La tabella viene creata con i parametri di ridimensionamento automatico specificati.

## Cassandra Query Language (CQL)

Crea una nuova tabella con la scalabilità automatica di Amazon Keyspaces tramite CQL

Per configurare le impostazioni di auto scaling per una tabella a livello di codice, usi l'AUTOSCALING\_SETTINGSistruzione che contiene i parametri per la scalabilità automatica di Amazon Keyspaces. I parametri definiscono le condizioni che spingono Amazon Keyspaces a modificare il throughput assegnato alla tabella e quali azioni opzionali aggiuntive intraprendere. In questo esempio, definisci le impostazioni di ridimensionamento automatico per mytable.

La policy contiene i seguenti elementi:

- AUTOSCALING\_SETTINGS— Indica se Amazon Keyspaces è autorizzato a modificare la capacità di throughput per tuo conto. Sono richiesti i seguenti valori:
  - `provisioned_write_capacity_autoscaling_update`:
  - `minimum_units`

- `maximum_units`
- `provisioned_read_capacity_autoscaling_update`:
  - `minimum_units`
  - `maximum_units`
- `scaling_policy`— Amazon Keyspaces supporta la politica di tracciamento degli obiettivi. Per definire la politica di tracciamento degli obiettivi, configuri i seguenti parametri.
  - `target_value`— La scalabilità automatica di Amazon Keyspaces assicura che il rapporto tra capacità consumata e capacità fornita rimanga pari o vicino a questo valore. `target_value` viene definito in percentuale.
  - `disableScaleIn`: (Facoltativo) A boolean che specifica se `scale-in` è disabilitato o abilitato per la tabella. Questo parametro è disabilitato per impostazione predefinita. Per attivarlo `scale-in`, imposta il booleano valore su `FALSE`. Ciò significa che la capacità viene automaticamente ridotta per una tabella per conto dell'utente.
  - `scale_out_cooldown`— Un'attività di scalabilità orizzontale aumenta la velocità effettiva assegnata alla tabella. Per aggiungere un periodo di recupero per le attività con scalabilità orizzontale, specifica un valore, in secondi, per `scale_out_cooldown`. Se non specifichi un valore, il valore predefinito è 0. Per ulteriori informazioni sul tracciamento del target e sui periodi di cooldown, consulta [Target Tracking Scaling Policies](#) nella Application Auto Scaling User Guide.
  - `scale_in_cooldown`— Un'attività scalabile riduce il throughput assegnato alla tabella. Per aggiungere un periodo di recupero per le attività scalabili, specifica un valore, in secondi, per `scale_in_cooldown`. Se non specifichi un valore, il valore predefinito è 0. Per ulteriori informazioni sul tracciamento del target e sui periodi di cooldown, consulta [Target Tracking Scaling Policies](#) nella Application Auto Scaling User Guide.

 Note

Per capire meglio come funziona `target_value`, supponiamo di avere una tabella con un'impostazione di velocità effettiva assegnata pari a 200 unità di capacità in scrittura. Si decide di creare una policy di dimensionamento per questa tabella, con un `target_value` del 70%.

Si supponga ora di iniziare a guidare il traffico di scrittura verso la tabella in modo che la velocità effettiva di scrittura sia di 150 unità di capacità. Il consumed-to-provisioned rapporto è ora (150/ 200), ovvero il 75 per cento. Questo rapporto supera l'obiettivo, quindi

la scalabilità automatica aumenta la capacità di scrittura assegnata a 215 in modo che il rapporto sia (150/ 215), ovvero il 69,77 percento, il più vicino `target_value` possibile, ma senza superarlo.

Per `mytable`, hai impostato la capacità di lettura e scrittura al `TargetValue` 50%. La scalabilità automatica di Amazon Keyspaces regola il throughput assegnato dalla tabella entro un intervallo di 5-10 unità di capacità in modo che il rapporto rimanga pari o vicino al `consumed-to-provisioned` 50%. Per quanto riguarda la capacità di lettura, si impostano i valori per e fino a 60 secondi.

`ScaleOutCooldown` `ScaleInCooldown`

Puoi utilizzare la seguente istruzione per creare una nuova tabella Amazon Keyspaces con la scalabilità automatica abilitata.

```
CREATE TABLE mykeyspace.mytable(pk int, ck int, PRIMARY KEY (pk, ck))
WITH CUSTOM_PROPERTIES = {
 'capacity_mode': {
 'throughput_mode': 'PROVISIONED',
 'read_capacity_units': 1,
 'write_capacity_units': 1
 }
} AND AUTOSCALING_SETTINGS = {
 'provisioned_write_capacity_autoscaling_update': {
 'maximum_units': 10,
 'minimum_units': 5,
 'scaling_policy': {
 'target_tracking_scaling_policy_configuration': {
 'target_value': 50
 }
 }
 },
 'provisioned_read_capacity_autoscaling_update': {
 'maximum_units': 10,
 'minimum_units': 5,
 'scaling_policy': {
 'target_tracking_scaling_policy_configuration': {
 'target_value': 50,
 'scale_in_cooldown': 60,
 'scale_out_cooldown': 60
 }
 }
 }
}
```

```
};
```

## CLI

Crea una nuova tabella con il ridimensionamento automatico di Amazon Keyspaces utilizzando AWS CLI

Per configurare le impostazioni di auto scaling per una tabella a livello di codice, usi l'`autoScalingSpecification` che definisce i parametri per la scalabilità automatica di Amazon Keyspaces. I parametri definiscono le condizioni che spingono Amazon Keyspaces a modificare il throughput assegnato alla tabella e quali azioni opzionali aggiuntive intraprendere. In questo esempio, definisci le impostazioni di ridimensionamento automatico per `mytable`.

La policy contiene i seguenti elementi:

- `autoScalingSpecification`— Indica se Amazon Keyspaces è autorizzato a regolare la capacità di trasmissione per tuo conto. È possibile abilitare la scalabilità automatica per la capacità di lettura e scrittura separatamente. Quindi è necessario specificare i seguenti parametri per `autoScalingSpecification`:
  - `writeCapacityAutoScaling`— Le unità con capacità di scrittura massima e minima.
  - `readCapacityAutoScaling`— Le unità con capacità di lettura massima e minima.
  - `scalingPolicy`— Amazon Keyspaces supporta la politica di tracciamento degli obiettivi. Per definire la politica di tracciamento degli obiettivi, configuri i seguenti parametri.
    - `targetValue`— La scalabilità automatica di Amazon Keyspaces assicura che il rapporto tra capacità consumata e capacità fornita rimanga pari o vicino a questo valore. `targetValue` viene definito in percentuale.
    - `disableScaleIn`: (Facoltativo) A boolean che specifica se `scale-in` è disabilitato o abilitato per la tabella. Questo parametro è disabilitato per impostazione predefinita. Per attivarlo `scale-in`, imposta il boolean valore su `FALSE`. Ciò significa che la capacità viene automaticamente ridotta per una tabella per conto dell'utente.
    - `scaleOutCooldown`— Un'attività di scalabilità orizzontale aumenta la velocità effettiva assegnata alla tabella. Per aggiungere un periodo di recupero per le attività con scalabilità orizzontale, specifica un valore, in secondi, per. `ScaleOutCooldown` Il valore predefinito è 0. Per ulteriori informazioni sul tracciamento del target e sui periodi di cooldown, consulta [Target Tracking Scaling Policies](#) nella Application Auto Scaling User Guide.
    - `scaleInCooldown`— Un'attività scalabile riduce il throughput assegnato alla tabella. Per aggiungere un periodo di recupero per le attività scalabili, specifica un valore, in

secondi, per. `ScaleInCooldown` Il valore predefinito è 0. Per ulteriori informazioni sul tracciamento del target e sui periodi di cooldown, consulta [Target Tracking Scaling Policies](#) nella *Application Auto Scaling User Guide*.

#### Note

Per capire meglio come funziona `TargetValue`, supponiamo di avere una tabella con un'impostazione di velocità effettiva assegnata pari a 200 unità di capacità in scrittura. Si decide di creare una policy di dimensionamento per questa tabella, con un `TargetValue` del 70%.

Si supponga ora di iniziare a guidare il traffico di scrittura verso la tabella in modo che la velocità effettiva di scrittura sia di 150 unità di capacità. Il `consumed-to-provisioned` rapporto è ora (150/ 200), ovvero il 75 per cento. Questo rapporto supera l'obiettivo, quindi la scalabilità automatica aumenta la capacità di scrittura assegnata a 215 in modo che il rapporto sia (150/ 215), ovvero il 69,77 per cento, il più vicino `TargetValue` possibile, ma senza superarlo.

Per `mytable`, hai impostato la capacità di lettura e scrittura al `TargetValue` 50%. La scalabilità automatica di Amazon Keyspaces regola il throughput assegnato dalla tabella entro un intervallo di 5-10 unità di capacità in modo che il rapporto rimanga pari o vicino al `consumed-to-provisioned` 50%. Per quanto riguarda la capacità di lettura, si impostano i valori per e fino a 60 secondi.

`ScaleOutCooldown` `ScaleInCooldown`

Quando si creano tabelle con impostazioni di ridimensionamento automatico complesse, è utile caricare le impostazioni di ridimensionamento automatico da un file JSON. Per l'esempio seguente, puoi scaricare il file JSON di esempio da [auto-scaling.zip](#) ed estrarlo `auto-scaling.json`, prendendo nota del percorso del file. In questo esempio, il file JSON si trova nella directory corrente. Per le diverse opzioni relative al percorso dei file, consultate [Come caricare i parametri da un file](#).

```
aws keyspaces create-table --keyspace-name mykeyspace --table-name mytable
 \ --schema-definition 'allColumns=[{name=pk,type=int},
{name=ck,type=int}],partitionKeys=[{name=pk},{name=ck}]'
 \ --capacity-specification
throughputMode=PROVISIONED,readCapacityUnits=1,writeCapacityUnits=1
 \ --auto-scaling-specification file://auto-scaling.json
```

## Configura il ridimensionamento automatico su una tabella esistente

Puoi aggiornare una tabella Amazon Keyspaces esistente per attivare la scalabilità automatica per la capacità di scrittura o lettura della tabella. Se stai aggiornando una tabella che è attualmente in modalità di capacità su richiesta, devi prima modificare la modalità di capacità della tabella in modalità di capacità fornita.

Per ulteriori informazioni su come aggiornare le impostazioni di ridimensionamento automatico per una tabella multiregionale, vedere. [the section called “Aggiornamento della capacità assegnata e delle impostazioni di ridimensionamento automatico per una tabella multiregionale”](#)

La scalabilità automatica di Amazon Keyspaces richiede la presenza di un ruolo collegato al servizio (AWSServiceRoleForApplicationAutoScaling\_CassandraTable) che esegue azioni di ridimensionamento automatico per tuo conto. Questo ruolo viene creato automaticamente per te. Per ulteriori informazioni, consulta [the section called “Uso di ruoli collegati ai servizi”](#).

### Console

Configura il ridimensionamento automatico di Amazon Keyspaces per una tabella esistente

1. [Accedi a e apri Console di gestione AWS la console Amazon Keyspaces a casahttps://console.aws.amazon.com/keyspaces/.](https://console.aws.amazon.com/keyspaces/)
2. Scegli la tabella su cui vuoi lavorare e vai alla scheda Capacità.
3. Nella sezione Impostazioni di capacità, scegli Modifica.
4. In modalità Capacity, assicurati che la tabella utilizzi la modalità Provisioned capacity.
5. Seleziona Scale automaticamente e consulta il passaggio 6 [the section called “Crea una nuova tabella con ridimensionamento automatico”](#) per modificare la capacità di lettura e scrittura.
6. Una volta definite le impostazioni di ridimensionamento automatico, scegli Salva.

### Cassandra Query Language (CQL)

Configura una tabella esistente con la scalabilità automatica di Amazon Keyspaces tramite CQL

Puoi utilizzare l'ALTER TABLEistruzione per una tabella Amazon Keyspaces esistente per configurare la scalabilità automatica per la capacità di scrittura o lettura della tabella. Se stai aggiornando una tabella che è attualmente in modalità di capacità su richiesta, devi



`capacity_mode` impostarla su `provisioned`. Se la tabella è già in modalità di capacità assegnata, questo campo può essere omissso.

Nell'esempio seguente, l'istruzione aggiorna la tabella `mytable`, che è in modalità di capacità su richiesta. L'istruzione modifica la modalità di capacità della tabella in modalità provisioning con la scalabilità automatica abilitata.

La capacità di scrittura è configurata nell'intervallo di 5-10 unità di capacità con un valore target del 50%. La capacità di lettura è inoltre configurata nell'intervallo di 5-10 unità di capacità con un valore obiettivo del 50%. Per la capacità di lettura, si impostano i valori per `scale_out_cooldown` e `scale_in_cooldown` fino a 60 secondi.

```
ALTER TABLE mykeyspace.mytable
WITH CUSTOM_PROPERTIES = {
 'capacity_mode': {
 'throughput_mode': 'PROVISIONED',
 'read_capacity_units': 1,
 'write_capacity_units': 1
 }
} AND AUTOSCALING_SETTINGS = {
 'provisioned_write_capacity_autoscaling_update': {
 'maximum_units': 10,
 'minimum_units': 5,
 'scaling_policy': {
 'target_tracking_scaling_policy_configuration': {
 'target_value': 50
 }
 }
 },
 'provisioned_read_capacity_autoscaling_update': {
 'maximum_units': 10,
 'minimum_units': 5,
 'scaling_policy': {
 'target_tracking_scaling_policy_configuration': {
 'target_value': 50,
 'scale_in_cooldown': 60,
 'scale_out_cooldown': 60
 }
 }
 }
};
```

## CLI

Configura una tabella esistente con il ridimensionamento automatico di Amazon Keyspaces utilizzando AWS CLI

Per una tabella Amazon Keyspaces esistente, puoi attivare la scalabilità automatica per la capacità di scrittura o lettura della tabella utilizzando l'operazione. `UpdateTable`

Puoi utilizzare il seguente comando per attivare la scalabilità automatica di Amazon Keyspaces per una tabella esistente. Le impostazioni di ridimensionamento automatico per la tabella vengono caricate da un file JSON. Per l'esempio seguente, puoi scaricare il file JSON di esempio da [auto-scaling.zip](#) ed estrarlo `auto-scaling.json`, prendendo nota del percorso del file. In questo esempio, il file JSON si trova nella directory corrente. Per le diverse opzioni relative al percorso dei file, consultate [Come caricare i parametri da un file](#).

Per ulteriori informazioni sulle impostazioni di ridimensionamento automatico utilizzate nell'esempio seguente, vedere [the section called “Crea una nuova tabella con ridimensionamento automatico”](#).

```
aws keyspaces update-table --keyspace-name mykeyspace --table-name mytable
 \ --capacity-specification
 throughputMode=PROVISIONED,readCapacityUnits=1,writeCapacityUnits=1
 \ --auto-scaling-specification file://auto-scaling.json
```

## Visualizza la configurazione di scalabilità automatica di Amazon Keyspaces della tua tabella

Puoi utilizzare la console, CQL o il AWS CLI per visualizzare e aggiornare le impostazioni di ridimensionamento automatico di Amazon Keyspaces di una tabella.

### Console

Visualizza le impostazioni di ridimensionamento automatico utilizzando la console

1. Scegli la tabella che desideri visualizzare e vai alla scheda Capacità.
2. Nella sezione Impostazioni di capacità, scegli Modifica. Ora puoi modificare le impostazioni nelle sezioni Capacità di lettura o Capacità di scrittura. Per ulteriori informazioni su queste impostazioni, consultare [the section called “Crea una nuova tabella con ridimensionamento automatico”](#).

## Cassandra Query Language (CQL)

Visualizza la politica di scalabilità automatica di Amazon Keyspaces della tua tabella utilizzando CQL

Per visualizzare i dettagli della configurazione di ridimensionamento automatico di una tabella, utilizzare il comando seguente.

```
SELECT * FROM system_schema_mcs.autoscaling WHERE keyspace_name = 'mykeyspace' AND
table_name = 'mytable';
```

L'output di questo comando è simile al seguente.

```
keyspace_name | table_name | provisioned_read_capacity_autoscaling_update
|
provisioned_write_capacity_autoscaling_update
-----+-----
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
mykeyspace | mytable | {'minimum_units': 5, 'maximum_units':
10, 'scaling_policy': {'target_tracking_scaling_policy_configuration':
{'scale_out_cooldown': 60, 'disable_scale_in': false, 'target_value':
50, 'scale_in_cooldown': 60}}} | {'minimum_units': 5, 'maximum_units':
10, 'scaling_policy': {'target_tracking_scaling_policy_configuration':
{'scale_out_cooldown': 0, 'disable_scale_in': false, 'target_value': 50,
'scale_in_cooldown': 0}}}
```

## CLI

Visualizza la politica di scalabilità automatica di Amazon Keyspaces della tua tabella utilizzando il AWS CLI

Per visualizzare la configurazione di ridimensionamento automatico di una tabella, è possibile utilizzare l'get-table-auto-scaling-settingsoperazione. Il seguente comando CLI ne è un esempio.

```
aws keyspaces get-table-auto-scaling-settings --keyspace-name mykeyspace --table-
name mytable
```

L'output di questo comando è simile al seguente.

```
{
 "keyspaceName": "mykeyspace",
 "tableName": "mytable",
 "resourceArn": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/mykeyspace/
table/mytable",
 "autoScalingSpecification": {
 "writeCapacityAutoScaling": {
 "autoScalingDisabled": false,
 "minimumUnits": 5,
 "maximumUnits": 10,
 "scalingPolicy": {
 "targetTrackingScalingPolicyConfiguration": {
 "disableScaleIn": false,
 "scaleInCooldown": 0,
 "scaleOutCooldown": 0,
 "targetValue": 50.0
 }
 }
 },
 "readCapacityAutoScaling": {
 "autoScalingDisabled": false,
 "minimumUnits": 5,
 "maximumUnits": 10,
 "scalingPolicy": {
 "targetTrackingScalingPolicyConfiguration": {
 "disableScaleIn": false,
 "scaleInCooldown": 60,
 "scaleOutCooldown": 60,
 "targetValue": 50.0
 }
 }
 }
 }
}
```

## Disattiva la scalabilità automatica di Amazon Keyspaces per una tabella

Puoi disattivare la scalabilità automatica di Amazon Keyspaces per la tua tabella in qualsiasi momento. Se non hai più bisogno di ridimensionare la capacità di lettura o scrittura della tabella, dovresti prendere in considerazione la possibilità di disattivare la scalabilità automatica in modo che

Amazon Keyspaces non continui a modificare le impostazioni della capacità di lettura o scrittura della tabella. Puoi aggiornare la tabella utilizzando la console, CQL o. AWS CLI

La disattivazione del ridimensionamento automatico elimina anche gli CloudWatch allarmi creati per tuo conto.

Per eliminare il ruolo collegato al servizio utilizzato da Application Auto Scaling per accedere alla tabella Amazon Keyspaces, segui la procedura riportata di seguito. [the section called “Eliminazione di un ruolo collegato al servizio per Amazon Keyspaces”](#)

#### Note

Per eliminare il ruolo collegato al servizio utilizzato da Application Auto Scaling, è necessario disabilitare il ridimensionamento automatico su tutte le tabelle dell'account in tutto. Regioni AWS

## Console

Disattiva il ridimensionamento automatico di Amazon Keyspaces per la tua tabella utilizzando la console

Utilizzo della console Amazon Keyspaces

1. [Accedi a e apri Console di gestione AWS la console Amazon Keyspaces a casahttps://console.aws.amazon.com/keyspaces/.](https://console.aws.amazon.com/keyspaces/)
2. Scegli la tabella che desideri aggiornare e vai alla scheda Capacità.
3. Nella sezione Impostazioni di capacità, scegli Modifica.
4. Per disabilitare il ridimensionamento automatico di Amazon Keyspaces, deseleziona la casella di controllo Scala automaticamente. La disabilitazione del ridimensionamento automatico annulla la registrazione della tabella come destinazione scalabile con Application Auto Scaling.

## Cassandra Query Language (CQL)

Disattiva il ridimensionamento automatico di Amazon Keyspaces per la tua tabella utilizzando CQL

L'istruzione seguente disattiva la scalabilità automatica per la capacità di scrittura della tabella `mytable`.

```
ALTER TABLE mykeyspace.mytable
WITH AUTOSCALING_SETTINGS = {
 'provisioned_write_capacity_autoscaling_update': {
 'autoscaling_disabled': true
 }
};
```

## CLI

Disattiva il ridimensionamento automatico di Amazon Keyspaces per la tua tabella utilizzando il AWS CLI

Il comando seguente disattiva il ridimensionamento automatico per la capacità di lettura della tabella. Elimina anche gli CloudWatch allarmi che sono stati creati per tuo conto.

```
aws keyspaces update-table --keyspace-name mykeyspace --table-name mytable
 \ --auto-scaling-specification
 readCapacityAutoScaling={autoScalingDisabled=true}
```

## Visualizza l'attività di scalabilità automatica per una tabella Amazon Keyspaces in Amazon CloudWatch

Puoi monitorare il modo in cui la scalabilità automatica di Amazon Keyspaces utilizza le risorse utilizzando Amazon CloudWatch, che genera metriche sull'utilizzo e sulle prestazioni. Segui i passaggi indicati nella [Guida per l'Application Auto Scaling utente](#) per creare una dashboard. CloudWatch

## Usa la capacità burst in modo efficace in Amazon Keyspaces

Amazon Keyspaces offre una certa flessibilità nel provisioning del throughput per partizione fornendo capacità burst. Ogni volta che non utilizzi appieno il throughput di una partizione, Amazon Keyspaces riserva una parte di quella capacità inutilizzata per successivi aumenti di throughput per gestire i picchi di utilizzo.

Amazon Keyspaces conserva attualmente fino a 5 minuti (300 secondi) di capacità di lettura e scrittura inutilizzata. Durante un'attività occasionale di lettura o scrittura, queste unità di capacità

aggiuntiva possono essere consumate rapidamente, anche più velocemente della capacità di throughput al secondo prevista per la tabella.

Amazon Keyspaces può inoltre consumare capacità burst per la manutenzione in background e altre attività senza preavviso.

Tieni presente che tali dettagli della capacità di ottimizzazione in futuro potrebbero cambiare.

# Utilizzo delle funzionalità di Amazon Keyspaces (per Apache Cassandra)

Questo capitolo fornisce dettagli sull'utilizzo di Amazon Keyspaces e varie funzionalità del database, ad esempio backup e ripristino, Time to Live e replica multiregionale.

- **Time to Live:** Amazon Keyspaces fa scadere automaticamente i dati delle tabelle in base al valore Time to Live impostato. Scopri come configurare il TTL e come utilizzarlo nelle tue tabelle.
- **PITR:** proteggi le tue tabelle Amazon Keyspaces da operazioni di scrittura o cancellazione accidentali creando backup continui dei dati delle tabelle. Scopri come configurare PITR sulle tue tabelle e come ripristinare una tabella in un momento specifico o come ripristinare una tabella che è stata eliminata accidentalmente.
- **Utilizzo di tabelle multiregionali:** le tabelle multiregionali in Amazon Keyspaces devono avere una capacità di throughput di scrittura configurata in modalità di capacità on-demand o fornita con scalabilità automatica. Pianifica le esigenze di capacità di throughput stimando le unità di capacità di scrittura richieste (WCUs) per ogni regione e fornisci la somma delle scritture provenienti da tutte le regioni per garantire una capacità sufficiente per le scritture replicate.
- **Amazon Keyspaces change data capture (CDC):** Amazon Keyspaces CDC trasmette eventi di modifica record a livello di riga dalla tua tabella quasi in tempo reale. Scopri come utilizzare la Kinesis Client Library (KCL) per consumare ed elaborare i dati dai flussi CDC di Amazon Keyspaces.
- **Query e impaginazione:** Amazon Keyspaces supporta funzionalità di interrogazione avanzate come l'utilizzo IN dell'operatore con SELECT istruzioni, l'ordinamento dei risultati e l'impaginazione automatica di set di risultati di grandi dimensioni. ORDER BY Questa sezione spiega come Amazon Keyspaces elabora queste query e fornisce esempi.
- **Partizionatori:** Amazon Keyspaces fornisce tre partizionatori: `Murmur3Partitioner` (impostazione predefinita) e `RandomPartitioner` `DefaultPartitioner` Puoi modificare il partizionatore per regione a livello di account utilizzando Console di gestione AWS o Cassandra Query Language (CQL).
- **Timestamp lato client:** i timestamp lato client sono timestamp compatibili con Cassandra che Amazon Keyspaces mantiene per ogni cella della tabella. Utilizza i timestamp lato client per la risoluzione dei conflitti e per consentire all'applicazione client di determinare l'ordine delle scritture.
- **Tipi definiti dall'utente (UDTs):** con UDTs puoi definire strutture di dati nelle tue applicazioni che rappresentano gerarchie di dati del mondo reale.



- **Etichettatura delle risorse:** puoi etichettare le risorse di Amazon Keyspaces come keyspace e tabelle utilizzando i tag. I tag aiutano a classificare le risorse, consentono il monitoraggio dei costi e consentono di configurare il controllo degli accessi in base ai tag. Questa sezione descrive le restrizioni, le operazioni e le best practice relative all'etichettatura per Amazon Keyspaces.
- **CloudFormation modelli:** ti CloudFormation aiuta a modellare e configurare gli spazi chiave e le tabelle di Amazon Keyspaces in modo da dedicare meno tempo alla creazione e alla gestione delle risorse e dell'infrastruttura.

## Argomenti

- [Spazi chiave di sistema in Amazon Keyspaces](#)
- [Tipi definiti dall'utente \(UDTs\) in Amazon Keyspaces](#)
- [Utilizzo delle query CQL in Amazon Keyspaces](#)
- [Utilizzo di flussi di acquisizione dati di modifica \(CDC\) in Amazon Keyspaces](#)
- [Utilizzo dei partizionatori in Amazon Keyspaces](#)
- [Timestamp lato client in Amazon Keyspaces](#)
- [Replica multiregionale per Amazon Keyspaces \(per Apache Cassandra\)](#)
- [Backup e ripristino dei dati con point-in-time ripristino per Amazon Keyspaces](#)
- [Fai scadere i dati con Time to Live \(TTL\) per Amazon Keyspaces \(per Apache Cassandra\)](#)
- [Utilizzo di questo servizio con un SDK AWS](#)
- [Utilizzo di tag ed etichette per le risorse Amazon Keyspaces](#)
- [Crea risorse Amazon Keyspaces con AWS CloudFormation](#)
- [Utilizzo di NoSQL Workbench con Amazon Keyspaces \(per Apache Cassandra\)](#)

## Spazi chiave di sistema in Amazon Keyspaces

Questa sezione fornisce dettagli sull'utilizzo degli spazi chiave di sistema in Amazon Keyspaces (per Apache Cassandra).

Amazon Keyspaces utilizza quattro keyspace di sistema:

- `system`
- `system_schema`
- `system_schema_mcs`

- `system_multiregion_info`

Le seguenti sezioni forniscono dettagli sugli spazi chiave di sistema e sulle tabelle di sistema supportati in Amazon Keyspaces.

## system

Questo è un keyspace di Cassandra. Amazon Keyspaces utilizza le seguenti tabelle.

| Nomi delle tabelle | Nomi delle colonne                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Commenti                                                                                                                                                                                                                                                                                                |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>local</code> | <code>key</code> , <code>bootstrap</code> , <code>ped</code> , <code>broadcast</code> , <code>_address</code> , <code>cluster_name</code> , <code>cql_version</code> , <code>data_center</code> , <code>gossip_generation</code> , <code>host_id</code> , <code>listen_address</code> , <code>native_protocol_version</code> , <code>partitioner</code> , <code>rack</code> , <code>release_version</code> , <code>rpc_address</code> , <code>schema_version</code> , <code>thrift_version</code> , <code>tokens</code> , <code>truncated_at</code> | Informazioni sul keyspace locale.                                                                                                                                                                                                                                                                       |
| <code>peers</code> | <code>peer</code> , <code>data_center</code> , <code>host_id</code> , <code>preferred_ip</code> , <code>rack</code> , <code>release_version</code> , <code>rpc_address</code> , <code>schema_version</code> , <code>tokens</code>                                                                                                                                                                                                                                                                                                                   | Interroga questa tabella per vedere gli endpoint disponibili. Ad esempio, se ti connetti tramite un endpoint pubblico, vedrai un elenco di nove indirizzi IP disponibili. Se ti connetti tramite un endpoint FIPS, viene visualizzato un elenco di tre indirizzi IP. Se ti connetti tramite un endpoint |

| Nomi delle tabelle  | Nomi delle colonne                                                                                | Commenti                                                                                                                                                                                                                                                          |
|---------------------|---------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                     |                                                                                                   | AWS PrivateLink VPC, vedrai l'elenco degli indirizzi IP che hai configurato. Per ulteriori informazioni, consulta <a href="#">the section called “Inserimento delle voci della system.peers tabella con informazioni sugli endpoint VPC dell'interfaccia”</a> .   |
| size_estimates      | keyspace_name,<br>table_name, range_start, range_end,<br>mean_partition_size,<br>partitions_count | Questa tabella definisce la dimensione totale e il numero di partizioni per ogni intervallo o di token per ogni tabella. Ciò è necessario per l'Apache Cassandra Spark Connector , che utilizza la dimensione stimata della partizione per distribuire il lavoro. |
| prepared_statements | prepared_id,<br>logged_keyspace,<br>query_string                                                  | Questa tabella contiene informazioni sulle interrogazioni salvate.                                                                                                                                                                                                |

## system\_schema

Questo è un keyspace di Cassandra. Amazon Keyspaces utilizza le seguenti tabelle.

| Nomi delle tabelle | Nomi delle colonne                               | Commenti                                     |
|--------------------|--------------------------------------------------|----------------------------------------------|
| keyspaces          | keyspace_name,<br>durable_writes,<br>replication | Informazioni su uno spazio chiave specifico. |

| Nomi delle tabelle | Nomi delle colonne                                                                                                                                                                                                                                                                                                    | Commenti                                                      |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------|
| tables             | keyspace_name, table_name, bloom_filter_fp_chance, caching, comment, compaction, compression, crc_check_chance, dclocal_read_repair_chance, default_time_to_live, extensions, flags, gc_grace_seconds, id, max_index_interval, memtable_flush_period_in_ms, min_index_interval, read_repair_chance, speculative_retry | Informazioni su una tabella specifica.                        |
| types              | keyspace_name, type_name, field_names, field_types                                                                                                                                                                                                                                                                    | Informazioni su un tipo specifico definito dall'utente (UDT). |
| columns            | keyspace_name, table_name, column_name, clustering_order, column_name_bytes, kind, position, type                                                                                                                                                                                                                     | Informazioni su una colonna specifica.                        |

## system\_schema\_mcs

Si tratta di uno spazio di chiavi Amazon Keyspaces che memorizza informazioni su impostazioni specifiche di AWS Amazon Keyspaces.

| Nomi delle tabelle | Nomi delle colonne                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Commenti                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| keyspaces          | keyspace_name,<br>durable_writes,<br>replication                                                                                                                                                                                                                                                                                                                                                                                                                                         | Interroga questa tabella per scoprire a livello di codice se è stato creato un keyspace. Per ulteriori informazioni, consulta <a href="#">the section called “Controlla lo stato di creazione del keyspace”</a> .                                                                                                                                                                                                                                                                                                                                                                         |
| tables             | keyspace_name,<br>creation_time,<br>speculative_retry,<br>cdc, gc_grace_<br>seconds, crc_check_<br>chance, min_index_<br>interval, bloom_fil<br>ter_fp_chance,<br>flags, custom_pr<br>operties, dclocal_r<br>ead_repair_chance,<br>table_name, caching,<br>default_time_to_li<br>ve, read_repa<br>ir_chance, max_index_<br>interval, extension<br>s, compaction,<br>comment, id, compressi<br>on, memtable_<br>flush_period_in_ms<br>, cdc_specification,<br>latest_stream_arn<br>status | <p>Interroga questa tabella per scoprire lo stato di una tabella specifica. Per ulteriori informazioni, consulta <a href="#">the section called “Controlla lo stato di creazione della tabella”</a>.</p> <p>Puoi anche interrogare questa tabella per elencare le impostazioni specifiche di Amazon Keyspaces e archiviate come. custom_properties Per esempio:</p> <ul style="list-style-type: none"> <li>• capacity_mode</li> <li>• cdc_specification</li> <li>• client_side_timestamps</li> <li>• encryption_specification</li> <li>• point_in_time_recovery</li> <li>• ttl</li> </ul> |

| Nomi delle tabelle          | Nomi delle colonne                                                                                                                                                                                                                                                                    | Commenti                                                                                                                                                                                                                                                                                                 |
|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>tables_history</code> | <code>keyspace_name,</code><br><code>table_name,</code> <code>event_time,</code> <code>creation_time,</code><br><code>custom_properties,</code><br><code>event</code>                                                                                                                 | Interroga questa tabella per conoscere le modifiche allo schema per una tabella specifica.                                                                                                                                                                                                               |
| <code>columns</code>        | <code>keyspace_name,</code><br><code>table_name,</code> <code>column_name,</code> <code>clustering_order,</code> <code>column_name_bytes,</code> <code>kind,</code><br><code>position,</code> <code>type</code>                                                                       | Questa tabella è identica alla tabella Cassandra nel <code>system_schema</code> keyspace.                                                                                                                                                                                                                |
| <code>tags</code>           | <code>resource_id,</code><br><code>keyspace_name,</code><br><code>resource_name,</code><br><code>resource_type,</code> <code>tags</code>                                                                                                                                              | Interroga questa tabella per scoprire se un keyspace contiene dei tag. Per ulteriori informazioni, consulta <a href="#">the section called “Visualizza i tag della tabella”</a> .                                                                                                                        |
| <code>types</code>          | <code>keyspace_name,</code><br><code>type_name,</code> <code>field_names,</code> <code>field_types,</code><br><code>max_nesting_depth,</code><br><code>last_modified_timestamp,</code> <code>status,</code><br><code>direct_referring_tables,</code> <code>direct_parent_types</code> | Interroga questa tabella per trovare informazioni sui tipi definiti dall'utente ()UDTs. Ad esempio, potete interrogare questa tabella per elencare tutto UDTs per un determinato keyspace. Per ulteriori informazioni, consulta <a href="#">the section called “Tipi definiti dall'utente () UDTs”</a> . |

| Nomi delle tabelle | Nomi delle colonne                                                                                               | Commenti                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|--------------------|------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| autoscaling        | keyspace_name,<br>table_name, provisioned_read_capacity_update,<br>provisioned_write_capacity_autoscaling_update | Interroga questa tabella per ottenere le impostazioni di ridimensionamento automatico di una tabella fornita. Tieni presente che queste impostazioni non saranno disponibili finché la tabella non sarà attiva. Per interrogare questa tabella, è necessario specificare <code>keyspace_name</code> e <code>table_name</code> nella WHERE clausola. Per ulteriori informazioni, consulta <a href="#">the section called “Visualizza la configurazione di scalabilità automatica di Amazon Keyspaces della tua tabella”</a> . |

## system\_multiregion\_info

Si tratta di uno spazio di chiavi Amazon Keyspaces che memorizza informazioni sulla replica multiregionale.

| Nomi delle tabelle | Nomi delle colonne                           | Commenti                                                                                                                                                                                                                                         |
|--------------------|----------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| tables             | keyspace_name,<br>table_name, region, status | Questa tabella contiene informazioni sulle tabelle con più aree geografiche, ad esempio la Regioni AWS modalità di replica della tabella e lo stato della tabella. Puoi anche interrogare questa tabella per elencare le impostazioni specifiche |

| Nomi delle tabelle     | Nomi delle colonne                                                                                                      | Commenti                                                                                                                                                                                                                                                                                                                                                                                                                     |
|------------------------|-------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                        |                                                                                                                         | <p>e di Amazon Keyspaces archiviate come. <code>custom_properties</code> Per esempio:</p> <ul style="list-style-type: none"> <li>• <code>capacity_mode</code></li> </ul> <p>Per interrogare questa tabella, devi specificare <code>keyspace_name</code> e <code>table_name</code> nella WHERE clausola. Per ulteriori informazioni, consulta <a href="#">the section called “Crea uno spazio chiave multiregionale”</a>.</p> |
| <code>keyspaces</code> | <code>keyspace_name</code> ,<br><code>region</code> , <code>status</code> ,<br><code>tables_replication_progress</code> | <p>Questa tabella contiene informazioni sullo stato di avanzamento di un'ALTER KEYSPACE operazione che aggiunge una replica a un keyspace, ad esempio quante tabelle sono già state create nella nuova regione e quante tabelle sono ancora in corso. Per alcuni esempi, vedere. <a href="#">the section called “Controlla l'avanzamento della replica”</a></p>                                                              |



| Nomi delle tabelle | Nomi delle colonne                                                                                                                                                    | Commenti                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| autoscaling        | keyspace_name,<br>table_name, provisioned_read_capacity_autoscaling_update,<br>provisioned_write_capacity_autoscaling_update, region                                  | Interroga questa tabella per ottenere le impostazioni di ridimensionamento automatico di una tabella con provisioning multiregionale. Tieni presente che queste impostazioni non saranno disponibili finché la tabella non sarà attiva. Per interrogare questa tabella, è necessario specificare <code>keyspace_name</code> e <code>table_name</code> nella WHERE clausola. Per ulteriori informazioni, consulta <a href="#">the section called “Aggiornamento della capacità assegnata e delle impostazioni di ridimensionamento automatico per una tabella multiregionale”</a> . |
| types              | keyspace_name,<br>type_name, field_names, field_types,<br>max_nesting_depth, last_modified_timestamp, status,<br>direct_referring_tables, direct_parent_types, region | Interroga questa tabella per trovare informazioni sui tipi definiti dall'utente (UDTs) negli spazi chiave multiregionali. Ad esempio, potete interrogare questa tabella per elencare tutte le repliche delle tabelle e le rispettive AWS regioni utilizzate UDTs per un determinato spazio di chiavi. Per ulteriori informazioni, consulta <a href="#">the section called “Tipi definiti dall'utente () UDTs”</a> .                                                                                                                                                                |

# Tipi definiti dall'utente (UDTs) in Amazon Keyspaces

Un tipo definito dall'utente (UDT) è un raggruppamento di campi e tipi di dati che puoi utilizzare per definire una singola colonna in Amazon Keyspaces. I tipi di dati validi UDTs sono tutti i tipi di dati Cassandra supportati, comprese le raccolte e altri UDTs che hai già creato nello stesso spazio chiave. Per ulteriori informazioni sui tipi di dati Cassandra supportati, consulta [the section called “Supporto per i tipi di dati Cassandra”](#)

Puoi utilizzare i tipi definiti dall'utente (UDTs) in Amazon Keyspaces per organizzare i dati in modo più efficiente. Ad esempio, puoi creare UDTs con raccolte annidate che ti consentono di implementare una modellazione dei dati più complessa nelle tue applicazioni. Puoi anche usare la parola chiave `frozen` per definire UDTs

UDTs sono legati a uno spazio chiave e disponibili per tutte le tabelle e UDTs nello stesso spazio chiave. È possibile creare spazi chiave UDTs a regione singola e multiarea.

È possibile creare nuove tabelle o modificare tabelle esistenti e aggiungere nuove colonne che utilizzano un UDT. Per creare un UDT con un UDT nidificato, quest'ultimo deve essere congelato.

Per verificare quanti UDTs sono supportati per keyspace, i livelli di nidificazione supportati e altri valori e quote predefiniti correlati, vedi UDTs [the section called “Quote e valori predefiniti per i tipi definiti dall'utente \(UDTs\) in Amazon Keyspaces”](#)

Per informazioni su come calcolare la dimensione codificata di, vedere UDTs [the section called “Stima la dimensione codificata dei valori dei dati in base al tipo di dati”](#)

Per ulteriori informazioni sulla sintassi CQL, vedere [the section called “Tipi”](#)

Per ulteriori informazioni sul UDTs ripristino point-in-time, vedere [the section called “PITR e UDTs”](#)

## Argomenti

- [Configura le autorizzazioni per lavorare con tipi definiti dall'utente \(UDTs\) in Amazon Keyspaces](#)
- [Crea un tipo definito dall'utente \(UDT\) in Amazon Keyspaces](#)
- [Visualizza i tipi definiti dall'utente \(UDTs\) in Amazon Keyspaces](#)
- [Eliminare un tipo definito dall'utente \(UDT\) in Amazon Keyspaces](#)

## Configura le autorizzazioni per lavorare con tipi definiti dall'utente (UDTs) in Amazon Keyspaces

Come le tabelle, UDTs sono legate a uno spazio chiave specifico. Ma a differenza delle tabelle, non è possibile definire direttamente le autorizzazioni per UDTs. UDTs non sono considerate risorse in AWS e non hanno identificatori univoci nel formato di Amazon Resource Name (ARN). Invece, per concedere a un IAM le autorizzazioni principali per eseguire azioni specifiche su un UDT, devi definire le autorizzazioni per lo spazio chiave a cui è associato l'UDT. Per utilizzare spazi chiave multiregionali UDTs, sono necessarie autorizzazioni aggiuntive.

Per poter creare, visualizzare o eliminare, il principale UDTs, ad esempio un utente o un ruolo IAM, necessita delle stesse autorizzazioni necessarie per eseguire la stessa azione sullo spazio chiave a cui è associato l'UDT.

Per ulteriori informazioni su AWS Identity and Access Management, vedere. [the section called “AWS Identity and Access Management”](#)

### Autorizzazioni per creare un UDT

Per creare un UDT in uno spazio chiave a regione singola, il principale necessita delle Create autorizzazioni per lo spazio chiave.

La seguente politica IAM ne è un esempio.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "cassandra:Create",
 "Resource": [
 "arn:aws:cassandra:us-east-1:111122223333:/keyspace/my_keyspace/"
]
 }
]
}
```

Per creare un UDT in uno spazio chiave multiregionale, oltre alle autorizzazioni, il principale necessita anche Create delle autorizzazioni per l'azione CreateMultiRegionResource per lo spazio chiave specificato.

La seguente politica IAM ne è un esempio.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": ["cassandra:Create", "cassandra:CreateMultiRegionResource"],
 "Resource": [
 "arn:aws:cassandra:us-east-1:111122223333:/keyspace/my_keyspace/"
]
 }
]
}
```

## Autorizzazioni per visualizzare un UDT

Per visualizzare o elencare UDTs in uno spazio chiave a regione singola, il principale necessita delle autorizzazioni di lettura per lo spazio chiave di sistema. Per ulteriori informazioni, consulta [the section called “system\\_schema\\_mcs”](#).

La seguente politica IAM ne è un esempio.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "cassandra:Select",
 "Resource": [
 "arn:aws:cassandra:us-east-1:111122223333:/keyspace/system*"
]
 }
]
}
```

Per visualizzare o UDTs elencare uno spazio chiave multiregionale, il principale necessita delle autorizzazioni per le azioni SELECT e SelectMultiRegionResource per lo spazio chiave di sistema. Per ulteriori informazioni, consulta [the section called “system\\_multiregion\\_info”](#).

La seguente politica IAM ne è un esempio.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": ["cassandra:Select", "cassandra:SelectMultiRegionResource"],
 "Resource": [
 "arn:aws:cassandra:us-east-1:111122223333:/keyspace/system*"
]
 }
]
}
```

## Autorizzazioni per eliminare un UDT

Per eliminare un UDT da uno spazio chiave a regione singola, il principale necessita delle autorizzazioni per l'Dropazione relativa allo spazio chiave specificato.

La seguente politica IAM ne è un esempio.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "cassandra:Drop",
 "Resource": [
 "arn:aws:cassandra:us-east-1:111122223333:/keyspace/my_keyspace/"
]
 }
]
}
```

Per eliminare un UDT da uno spazio chiave multiregionale, il principale necessita delle autorizzazioni per l'Dropazione e per l'azione per lo spazio chiave DropMultiRegionResource specificato.

La seguente politica IAM ne è un esempio.

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

```

 {
 "Effect": "Allow",
 "Action": ["cassandra:Drop", "cassandra:DropMultiRegionResource"],
 "Resource": [
 "arn:aws:cassandra:us-east-1:111122223333:/keyspace/my_keyspace/"
]
 }
]
}

```

## Crea un tipo definito dall'utente (UDT) in Amazon Keyspaces

Per creare un UDT in uno spazio chiave a regione singola, è possibile utilizzare l'`CREATE TYPE`istruzione in CQL, il `create-type` comando con o la console. AWS CLI

I nomi UDT devono contenere al massimo 48 caratteri, devono iniziare con un carattere alfabetico e possono contenere solo caratteri alfanumerici e trattini bassi. Amazon Keyspaces converte automaticamente i caratteri maiuscoli in caratteri minuscoli.

In alternativa, puoi dichiarare un nome UDT tra virgolette. Quando si dichiara un nome UDT tra virgolette, Amazon Keyspaces mantiene le lettere maiuscole e consente l'uso di caratteri speciali.

Puoi anche utilizzare le virgolette come parte del nome quando crei l'UDT, ma devi evitare ogni carattere di virgolette doppie con un carattere di virgolette doppie aggiuntivo.

La tabella seguente mostra esempi di nomi UDT consentiti. La prima colonna mostra come inserire il nome quando si crea il tipo, la seconda colonna mostra come Amazon Keyspaces formatta il nome internamente. Amazon Keyspaces si aspetta il nome formattato per operazioni come `GetType`.

| Nome inserito | Nome formattato | Nota                                                                                                                |
|---------------|-----------------|---------------------------------------------------------------------------------------------------------------------|
| MY_UDT        | my_udt          | Senza virgolette, Amazon Keyspaces converte tutti i caratteri maiuscoli in minuscoli.                               |
| "MY_UDT"      | MY_UDT          | Con le virgolette, Amazon Keyspaces rispetta i caratteri maiuscoli e rimuove le virgolette dal nome formattato.     |
| "1234"        | 1234            | Con le virgolette, il nome può iniziare con un numero e Amazon Keyspaces rimuove le virgolette dal nome formattato. |

| Nome inserito                    | Nome formattato                | Nota                                                                                                                      |
|----------------------------------|--------------------------------|---------------------------------------------------------------------------------------------------------------------------|
| "Special_<br>Ch@r@cter<br>s<>!!" | Special_C<br>h@r@cters<br><>!! | Con le virgolette, il nome può contenere caratteri speciali e Amazon Keyspaces rimuove le virgolette dal nome formattato. |
| "nested"<br>""""quote<br>s"      | nested""<br>quotes             | Amazon Keyspaces rimuove le virgolette esterne e le virgolette di escape dal nome formattato.                             |

## Console

Crea un tipo definito dall'utente (UDT) con la console Amazon Keyspaces

1. [Accedi a e apri Console di gestione AWS la console Amazon Keyspaces a casahttps://console.aws.amazon.com/keyspaces/.](https://console.aws.amazon.com/keyspaces/)
2. Nel riquadro di navigazione, scegli Keyspaces, quindi scegli uno spazio chiave dall'elenco.
3. Scegliere la scheda UDTs.
4. Scegli Crea UDT
5. In Dettagli UDT, inserisci il nome dell'UDT. Nei campi UDT si definisce lo schema dell'UDT.
6. Per finire, scegli Crea UDT.

## Cassandra Query Language (CQL)

Crea un tipo definito dall'utente (UDT) con CQL

In questo esempio creiamo una nuova versione della tabella dei premi dei libri utilizzata in [the section called “Creare una tabella”](#). In questa tabella, memorizziamo tutti i premi che un autore riceve per un determinato libro. Ne creiamo due UDTs che sono annidati e contengono informazioni sul libro che ha ricevuto un premio.

1. Crea uno spazio chiave con il nome. catalog

```
CREATE KEYSPACE catalog WITH REPLICATION = {'class': 'SingleRegionStrategy'};
```

2. Crea il primo tipo. Questo tipo memorizza i codici BISAC, che vengono utilizzati per definire il genere dei libri. Un codice BISAC è composto da un codice alfanumerico e da un massimo di quattro aree tematiche.

```
CREATE TYPE catalog.bisac (
 bisac_code text,
 subject1 text,
 subject2 text,
 subject3 text,
 subject4 text
);
```

3. Crea un secondo tipo per i premi letterari che utilizzi il primo UDT. L'UDT annidato deve essere congelato.

```
CREATE TYPE catalog.book (
 award_title text,
 book_title text,
 publication_date date,
 page_count int,
 ISBN text,
 genre FROZEN <bisac>
);
```

4. Crea una tabella con una colonna per il nome dell'autore e utilizza un tipo di elenco per i premi del libro. Nota che l'UDT utilizzato nell'elenco deve essere congelato.

```
CREATE TABLE catalog.authors (
 author_name text PRIMARY KEY,
 awards list <FROZEN <book>>
);
```

5. In questo passaggio inseriamo una riga di dati nella nuova tabella.

```
CONSISTENCY LOCAL_QUORUM;
```

```
INSERT INTO catalog.authors (author_name, awards) VALUES (
 'John Stiles' ,
 [{
 award_title: 'Wolf',
 book_title: 'Yesterday',
```



```

 publication_date: '2020-10-10',
 page_count: 345,
 isbn: '026204630X',
 genre: { bisac_code: 'FIC014090', subject1: 'FICTION', subject2:
'Historical', subject3: '20th Century', subject4: 'Post-World War II'
 },
 {award_title: 'Richard Roe',
 book_title: 'Who ate the cake?',
 publication_date: '2019-05-13',
 page_count: 193,
 isbn: '9780262046305',
 genre: { bisac_code: 'FIC022130', subject1: 'FICTION', subject2: 'Mystery &
Detective', subject3: 'Cozy', subject4: 'Culinary'
 }
]
};

```

6. Nell'ultimo passaggio leggiamo i dati della tabella.

```
SELECT * FROM catalog.authors;
```

L'output del comando dovrebbe essere simile a questo.

```

author_name | awards

+-----+-----
John Stiles | [{award_title: 'Wolf', book_title: 'Yesterday', publication_date:
2020-10-10, page_count: 345, isbn: '026204630X', genre: {bisac_code:
'FIC014090', subject1: 'FICTION', subject2: 'Historical', subject3: '20th
Century', subject4: 'Post-World War II'}}, {award_title: 'Richard Roe',
book_title: 'Who ate the cake?', publication_date: 2019-05-13, page_count: 193,
isbn: '9780262046305', genre: {bisac_code: 'FIC022130', subject1: 'FICTION',
subject2: 'Mystery & Detective', subject3: 'Cozy', subject4: 'Culinary'}}]

(1 rows)

```

Per ulteriori informazioni sulla sintassi CQL, vedere. [the section called “CREATE TYPE”](#)

## CLI

Creare un tipo definito dall'utente (UDT) con AWS CLI

1. Per creare un tipo è possibile utilizzare la seguente sintassi.

```
aws keyspaces create-type
--keyspace-name 'my_keyspace'
--type-name 'my_udt'
--field-definitions
 '['
 {"name" : "field1", "type" : "int"},
 {"name" : "field2", "type" : "text"}
 ']
```

2. L'output di quel comando è simile a quello di questo esempio. Nota che typeName restituisce il nome formattato dell'UDT.

```
{
 "keyspaceArn": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/
my_keyspace/",
 "typeName": "my_udt"
}
```

## Visualizza i tipi definiti dall'utente (UDTs) in Amazon Keyspaces

Per visualizzare o elencare tutto UDTs in uno spazio chiave a regione singola, è possibile interrogare la tabella `system_schema_mcs.types` nello spazio chiave di sistema utilizzando un'istruzione in CQL o utilizzare `list-type` i comandi `get-type` and `con`, o la AWS CLI console.

Per entrambe le opzioni, il principale IAM necessita delle autorizzazioni di lettura per lo spazio chiave del sistema. Per ulteriori informazioni, consulta [the section called “Configurare le autorizzazioni”](#).

### Console

Visualizza i tipi definiti dall'utente (UDT) con la console Amazon Keyspaces

1. [Accedi a e apri Console di gestione AWS la console Amazon Keyspaces a casahttps://console.aws.amazon.com/keyspaces/](https://console.aws.amazon.com/keyspaces/).
2. Nel riquadro di navigazione, scegli Keyspaces, quindi scegli uno spazio chiave dall'elenco.
3. Scegli la UDTsscheda per rivedere l'elenco di tutti gli elementi UDTs presenti nel keyspace.
4. Per esaminare un UDT in dettaglio, scegliete un UDT dall'elenco.

5. Nella scheda Schema puoi rivedere lo schema. Nella scheda Usato in puoi vedere se questo UDT è usato in tabelle o altro UDTs. Nota che puoi eliminare solo quelli UDTs che non sono utilizzati da tabelle o altro UDTs.

## Cassandra Query Language (CQL)

Visualizza i tipi definiti dall'utente (UDTs) di uno spazio chiave a regione singola con CQL

1. Per vedere i tipi disponibili in un determinato spazio chiave, puoi usare la seguente istruzione.

```
SELECT type_name
FROM system_schema_mcs.types
WHERE keyspace_name = 'my_keyspace';
```

2. Per visualizzare i dettagli su un tipo specifico, è possibile utilizzare la seguente dichiarazione.

```
SELECT
 keyspace_name,
 type_name,
 field_names,
 field_types,
 max_nesting_depth,
 last_modified_timestamp,
 status,
 direct_referring_tables,
 direct_parent_types
FROM system_schema_mcs.types
WHERE keyspace_name = 'my_keyspace' AND type_name = 'my_udt';
```

3. Puoi elencare tutto UDTs ciò che esiste nell'account utilizzandoDESC TYPE.

```
DESC TYPES;

Keyspace my_keyspace

my_udt1 my_udt2

Keyspace my_keyspace2

my_udt1
```

4. È possibile elencare tutto UDTs nel keyspace attualmente selezionato utilizzando `DESC TYPE`.

```
USE my_keyspace;
my_keyspace DESC TYPES;

my_udt1 my_udt2
```

5. Per elencare tutto UDTs in uno spazio chiave multiregionale, è possibile interrogare la tabella di sistema `types` nello spazio chiave. `system_multiregion_info` La seguente query ne è un esempio.

```
SELECT keyspace_name, type_name, region, status FROM
 system_multiregion_info.types WHERE keyspace_name = 'mykeyspace' AND table_name
 = 'mytable';
```

L'output di questo comando è simile al seguente.

| keyspace_name | table_name | region         | status |
|---------------|------------|----------------|--------|
| mykeyspace    | mytable    | us-east-1      | ACTIVE |
| mykeyspace    | mytable    | ap-southeast-1 | ACTIVE |
| mykeyspace    | mytable    | eu-west-1      | ACTIVE |

## CLI

Visualizza i tipi definiti dall'utente (UDTs) con AWS CLI

1. Per elencare i tipi disponibili in un keyspace, puoi usare il `list-types` comando.

```
aws keyspaces list-types
--keyspace-name 'my_keyspace'
```

L'output di quel comando è simile a quello di questo esempio.

```
{
 "types": [
 "my_udt",
 "parent_udt"
]
}
```

```
}
```

2. Per visualizzare i dettagli su un determinato tipo puoi usare il `get-type` comando.

```
aws keyspaces get-type
--type-name 'my_udt'
--keyspace-name 'my_keyspace'
```

L'output di questo comando è simile a quello di questo esempio.

```
{
 "keyspaceName": "my_keyspace",
 "typeName": "my_udt",
 "fieldDefinitions": [
 {
 "name": "a",
 "type": "int"
 },
 {
 "name": "b",
 "type": "text"
 }
],
 "lastModifiedTimestamp": 1721328225776,
 "maxNestingDepth": 3
 "status": "ACTIVE",
 "directReferringTables": [],
 "directParentTypes": [
 "parent_udt"
],
 "keyspaceArn": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/
my_keyspace/"
}
```

## Eliminare un tipo definito dall'utente (UDT) in Amazon Keyspaces

Per eliminare un UDT in un keyspace, è possibile utilizzare l'`DROP TYPE`istruzione in CQL, il `delete-type` comando con o la console AWS CLI.

## Console

Elimina un tipo definito dall'utente (UDT) con la console Amazon Keyspaces

1. [Accedi a e apri Console di gestione AWS la console Amazon Keyspaces a casa](https://console.aws.amazon.com/keyspaces/)<https://console.aws.amazon.com/keyspaces/>.
2. Nel riquadro di navigazione, scegli Keyspaces, quindi scegli uno spazio chiave dall'elenco.
3. Scegliere la scheda UDTs.
4. Scegli l'UDT che desideri eliminare. In Usato in puoi confermare che il tipo che desideri eliminare non è attualmente utilizzato da una tabella o da un altro UDT.
5. Scegli Elimina sopra il riepilogo.
6. Digita Delete nella finestra di dialogo che appare e scegli Elimina UDT.

## Cassandra Query Language (CQL)

Elimina un tipo definito dall'utente (UDT) con CQL

- Per eliminare un tipo, è possibile utilizzare la seguente istruzione.

```
DROP TYPE my_keyspace.my_udt;
```

Per ulteriori informazioni sulla sintassi CQL, vedere. [the section called “DROP TYPE”](#)

## CLI

Eliminare un tipo definito dall'utente (UDT) con AWS CLI

1. Per eliminare un tipo, è possibile utilizzare il seguente comando.

```
aws keyspaces delete-type
--keyspace-name 'my_keyspace'
--type-name 'my_udt'
```

2. L'output del comando è simile a quello di questo esempio.

```
{
 "keyspaceArn": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/
my_keyspace/",
```

```
"typeName": "my_udt"
}
```

## Utilizzo delle query CQL in Amazon Keyspaces

Questa sezione fornisce un'introduzione all'utilizzo delle query in Amazon Keyspaces (per Apache Cassandra). Le istruzioni CQL disponibili per interrogare, trasformare e gestire i dati sono `SELECT`, `INSERT`, `UPDATE`, `DELETE`. Negli argomenti seguenti vengono descritte alcune delle opzioni più complesse disponibili quando si lavora con le query. Per la sintassi completa del linguaggio con esempi, vedere [the section called "Istruzioni DML"](#).

### Argomenti

- [Usa l'`IN` operatore con l'`SELECT` istruzione in una query in Amazon Keyspaces](#)
- [Usa le istruzioni batch in Amazon Keyspaces](#)
- [Ordina i risultati con `ORDER BY` in Amazon Keyspaces](#)
- [Impaginare i risultati in Amazon Keyspaces](#)

## Usa l'`IN` operatore con l'`SELECT` istruzione in una query in Amazon Keyspaces

### SELEZIONA IN

È possibile interrogare i dati dalle tabelle utilizzando l'`SELECT` istruzione, che legge una o più colonne per una o più righe in una tabella e restituisce un set di risultati contenente le righe corrispondenti alla richiesta. Un'`SELECT` istruzione contiene un'istruzione `select_clause` che determina quali colonne leggere e restituire nel set di risultati. La clausola può contenere istruzioni per trasformare i dati prima di restituirli. La `WHERE` clausola facoltativa specifica quali righe devono essere interrogate ed è composta da relazioni sulle colonne che fanno parte della chiave primaria. Amazon Keyspaces supporta la `IN` parola chiave nella clausola `WHERE`. Questa sezione utilizza esempi per mostrare come Amazon Keyspaces elabora `SELECT` le istruzioni con la `IN` parola chiave.

Questo esempio dimostra come Amazon Keyspaces suddivide `SELECT` l'istruzione con `IN` la parola chiave in sottoquery. In questo esempio utilizziamo una tabella con il nome `my_keyspace.customers`. La tabella ha una colonna chiave primari `department_id`, due colonne `sales_region_id` di clustering e e una colonna che contiene il nome del cliente nella `customer_name` colonna. `sales_representative_id`

```
SELECT * FROM my_keyspace.customers;
```

| department_id | sales_region_id | sales_representative_id | customer_name |
|---------------|-----------------|-------------------------|---------------|
| 0             | 0               | 0                       | a             |
| 0             | 0               | 1                       | b             |
| 0             | 1               | 0                       | c             |
| 0             | 1               | 1                       | d             |
| 1             | 0               | 0                       | e             |
| 1             | 0               | 1                       | f             |
| 1             | 1               | 0                       | g             |
| 1             | 1               | 1                       | h             |

Utilizzando questa tabella, è possibile eseguire la seguente SELECT istruzione per trovare i clienti nei reparti e nelle aree di vendita a cui si è interessati con la IN parola chiave contenuta nella WHERE clausola. La seguente dichiarazione ne è un esempio.

```
SELECT * FROM my_keyspace.customers WHERE department_id IN (0, 1) AND sales_region_id IN (0, 1);
```

Amazon Keyspaces divide questa dichiarazione in quattro sottoquery, come illustrato nell'output seguente.

```
SELECT * FROM my_keyspace.customers WHERE department_id = 0 AND sales_region_id = 0;
```

| department_id | sales_region_id | sales_representative_id | customer_name |
|---------------|-----------------|-------------------------|---------------|
| 0             | 0               | 0                       | a             |
| 0             | 0               | 1                       | b             |

```
SELECT * FROM my_keyspace.customers WHERE department_id = 0 AND sales_region_id = 1;
```

| department_id | sales_region_id | sales_representative_id | customer_name |
|---------------|-----------------|-------------------------|---------------|
| 0             | 1               | 0                       | c             |
| 0             | 1               | 1                       | d             |

```
SELECT * FROM my_keyspace.customers WHERE department_id = 1 AND sales_region_id = 0;
```

| department_id | sales_region_id | sales_representative_id | customer_name |
|---------------|-----------------|-------------------------|---------------|
| 1             | 0               | 0                       | e             |



```

1 | 0 | 1 | f
SELECT * FROM my_keyspace.customers WHERE department_id = 1 AND sales_region_id = 1;

department_id | sales_region_id | sales_representative_id | customer_name
-----+-----+-----+-----
1 | 1 | 0 | g
1 | 1 | 1 | h

```

Quando viene utilizzata la IN parola chiave, Amazon Keyspaces impagina automaticamente i risultati in uno dei seguenti casi:

- Dopo l'elaborazione di ogni decima sottoquery.
- Dopo aver elaborato 1 MB di IO logico.
- Se hai configurato aPAGE SIZE, Amazon Keyspaces impagina dopo aver letto il numero di query da elaborare in base al set. PAGE SIZE
- Quando usi la LIMIT parola chiave per ridurre il numero di righe restituite, Amazon Keyspaces impagina dopo aver letto il numero di query da elaborare in base al set. LIMIT

La tabella seguente viene utilizzata per illustrare questo aspetto con un esempio.

Per ulteriori informazioni sull'impaginazione, vedere. [the section called “Impagina i risultati”](#)

```

SELECT * FROM my_keyspace.customers;

department_id | sales_region_id | sales_representative_id | customer_name
-----+-----+-----+-----
2 | 0 | 0 | g
2 | 1 | 1 | h
2 | 2 | 2 | i
0 | 0 | 0 | a
0 | 1 | 1 | b
0 | 2 | 2 | c
1 | 0 | 0 | d
1 | 1 | 1 | e
1 | 2 | 2 | f
3 | 0 | 0 | j
3 | 1 | 1 | k
3 | 2 | 2 | l

```

È possibile eseguire la seguente istruzione in questa tabella per vedere come funziona l'impaginazione.

```
SELECT * FROM my_keyspace.customers WHERE department_id IN (0, 1, 2, 3) AND
sales_region_id IN (0, 1, 2) AND sales_representative_id IN (0, 1);
```

Amazon Keyspaces elabora questa dichiarazione come 24 sottoquery, poiché la cardinalità del prodotto cartesiano di tutti i IN termini contenuti in questa query è 24.

| department_id | sales_region_id | sales_representative_id | customer_name |
|---------------|-----------------|-------------------------|---------------|
| 0             | 0               | 0                       | a             |
| 0             | 1               | 1                       | b             |
| 1             | 0               | 0                       | d             |
| 1             | 1               | 1                       | e             |

---MORE---

| department_id | sales_region_id | sales_representative_id | customer_name |
|---------------|-----------------|-------------------------|---------------|
| 2             | 0               | 0                       | g             |
| 2             | 1               | 1                       | h             |
| 3             | 0               | 0                       | j             |

---MORE---

| department_id | sales_region_id | sales_representative_id | customer_name |
|---------------|-----------------|-------------------------|---------------|
| 3             | 1               | 1                       | k             |

Questo esempio mostra come utilizzare la ORDER BY clausola in un'istruzione con la parola chiave SELECT IN

```
SELECT * FROM my_keyspace.customers WHERE department_id IN (3, 2, 1) ORDER BY
sales_region_id DESC;
```

| department_id | sales_region_id | sales_representative_id | customer_name |
|---------------|-----------------|-------------------------|---------------|
| 3             | 2               | 2                       | l             |
| 3             | 1               | 1                       | k             |
| 3             | 0               | 0                       | j             |
| 2             | 2               | 2                       | i             |
| 2             | 1               | 1                       | h             |
| 2             | 0               | 0                       | g             |

|   |  |   |  |   |  |   |
|---|--|---|--|---|--|---|
| 1 |  | 2 |  | 2 |  | f |
| 1 |  | 1 |  | 1 |  | e |
| 1 |  | 0 |  | 0 |  | d |

Le sottoquery vengono elaborate nell'ordine in cui le colonne della chiave di partizione e della chiave di clustering vengono presentate nella query. Nell'esempio seguente, le sottoquery per il valore della chiave di partizione» 2 «vengono elaborate per prime, seguite dalle sottoquery per il valore della chiave di partizione» 3 «e» 1». I risultati di una determinata sottoquery vengono ordinati in base alla clausola di ordinamento della query, se presente, o all'ordine di clustering della tabella definito durante la creazione della tabella.

```
SELECT * FROM my_keyspace.customers WHERE department_id IN (2, 3, 1) ORDER BY
sales_region_id DESC;
```

| department_id           |  | sales_region_id |  | sales_representative_id |  | customer_name |
|-------------------------|--|-----------------|--|-------------------------|--|---------------|
| -----+-----+-----+----- |  |                 |  |                         |  |               |
| 2                       |  | 2               |  | 2                       |  | i             |
| 2                       |  | 1               |  | 1                       |  | h             |
| 2                       |  | 0               |  | 0                       |  | g             |
| 3                       |  | 2               |  | 2                       |  | l             |
| 3                       |  | 1               |  | 1                       |  | k             |
| 3                       |  | 0               |  | 0                       |  | j             |
| 1                       |  | 2               |  | 2                       |  | f             |
| 1                       |  | 1               |  | 1                       |  | e             |
| 1                       |  | 0               |  | 0                       |  | d             |

## Usa le istruzioni batch in Amazon Keyspaces

Puoi combinare più INSERT DELETE operazioni e in un'UPDATEunica dichiarazione. BATCH LOGGEDi batch sono l'impostazione predefinita.

```
batch_statement ::= BEGIN [UNLOGGED] BATCH
 [USING update_parameter(AND update_parameter)*]
 modification_statement (';' modification_statement)*
 APPLY BATCH
modification_statement ::= insert_statement | update_statement | delete_statement
```

Quando si esegue un'istruzione batch, il driver combina tutte le istruzioni del batch in un'unica operazione batch.

Per decidere quale tipo di operazione batch utilizzare, è possibile prendere in considerazione le seguenti linee guida.

Utilizzate i batch registrati quando:

- Avete bisogno di garanzie atomiche sulle transazioni.
- Latenze leggermente più elevate sono un compromesso accettabile.

Utilizza batch non registrati quando:

- È necessario ottimizzare le operazioni a partizione singola.
- Vuoi ridurre il sovraccarico della rete.
- Hai requisiti di throughput elevati.

Per informazioni sulle quote delle dichiarazioni in batch, vedere. [Quote](#)

## Batch non registrati

Con i batch non registrati, Amazon Keyspaces elabora più operazioni come un'unica richiesta senza mantenere un registro batch. Con un'operazione batch non registrata, è possibile che alcune azioni abbiano successo mentre altre falliscano. I batch non registrati sono utili quando si desidera:

- Ottimizza le operazioni all'interno di una singola partizione.
- Riduci il traffico di rete raggruppando le richieste correlate.

La sintassi per un batch non registrato è simile a quella di un batch registrato, con l'aggiunta della parola chiave. UNLOGGED

```
BEGIN UNLOGGED BATCH
 INSERT INTO users (id, firstname, lastname) VALUES (1, 'John', 'Doe');
 INSERT INTO users (id, firstname, lastname) VALUES (2, 'Jane', 'Smith');
APPLY BATCH;
```

## Batch registrati

Un batch registrato combina più azioni di scrittura in un'unica operazione atomica. Quando si esegue un batch registrato:

- Tutte le azioni hanno successo insieme o falliscono insieme.

- L'operazione è sincrona e idempotente.
- Puoi scrivere su più tabelle Amazon Keyspaces, purché si trovino nello stesso AWS account e. Regione AWS

I batch registrati possono avere latenze leggermente superiori. Per applicazioni ad alto rendimento, prendi in considerazione l'utilizzo di batch non registrati.

Non sono previsti costi aggiuntivi per l'utilizzo di batch registrati in Amazon Keyspaces. Paghi solo per le scritture che fanno parte delle tue operazioni in batch. Amazon Keyspaces esegue due scritture sottostanti per ogni riga del batch: una per preparare la riga per il batch e una per eseguire il commit del batch. Quando pianifichi la capacità per le tabelle che utilizzano batch registrati, ricorda che ogni riga di un batch richiede una capacità doppia rispetto a un'operazione di scrittura standard. Ad esempio, se l'applicazione esegue un batch registrato al secondo con tre righe da 1 KB, è necessario fornire sei unità di capacità di scrittura (WCUs) rispetto alle sole tre WCUs per singole scritture o batch non registrati.

Per informazioni sui prezzi, consulta i prezzi di [Amazon Keyspaces \(per Apache Cassandra\)](#).

Le migliori pratiche per le operazioni in batch

Prendi in considerazione le seguenti pratiche consigliate per l'utilizzo di operazioni batch di Amazon Keyspaces.

- Abilita il ridimensionamento automatico per assicurarti di disporre di una capacità di throughput sufficiente per consentire alle tabelle di gestire le operazioni in batch e i requisiti di throughput aggiuntivi dei batch registrati.
- Utilizzate operazioni singole o batch non registrati quando le operazioni possono essere eseguite indipendentemente senza influire sulla correttezza dell'applicazione.
- Progetta l'applicazione in modo da ridurre al minimo gli aggiornamenti simultanei alle stesse righe, poiché le operazioni batch simultanee possono entrare in conflitto e fallire.
- Per l'ingestione di grandi quantità di dati ad alta velocità senza requisiti di atomicità, utilizzate operazioni di scrittura individuali o batch non registrati.

Coerenza e concorrenza

Amazon Keyspaces applica le seguenti regole di coerenza e concorrenza per i batch registrati:

- Tutte le operazioni in batch utilizzano il livello di coerenza. LOCAL\_QUORUM

- I batch simultanei che interessano righe diverse possono essere eseguiti contemporaneamente.
- Le DELETE operazioni INSERT simultanee UPDATE o sulle righe coinvolte in un batch in corso hanno esito negativo a causa di un conflitto.

## Operatori e condizioni supportati

### Operatori WHERE di clausola supportati:

- Uguaglianza (=)

### Operatori non supportati:

- Operatori di intervallo (>, <, >=, <=)
- Operatore IN
- Operatore LIKE
- Operatore BETWEEN

### Non supportato nei batch registrati:

- Dichiarazioni multiple che riguardano la stessa riga
- Controoperazioni
- Range cancella

## Condizioni di errore delle istruzioni batch registrate

### Un'operazione batch registrata può fallire in uno dei seguenti casi:

- Le espressioni condizionali (come IF NOT EXISTS o IF) restituiscono false.
- Una o più operazioni contengono parametri non validi.
- La richiesta è in conflitto con un'altra operazione batch in esecuzione sulle stesse righe.
- La tabella non dispone di una capacità predisposta sufficiente.
- Una riga supera il limite massimo di dimensione.
- Il formato dei dati di input non è valido.

## Istruzioni Batch e replica in più regioni

### Nelle distribuzioni in più regioni:

- Le operazioni della regione di origine sono sincrone e atomiche.

- Le operazioni della regione di destinazione sono asincrone.
- È garantita la replica di tutte le operazioni in batch, ma potrebbe non essere possibile mantenere l'isolamento durante l'applicazione.

### Monitora le operazioni in batch

Puoi monitorare le operazioni in batch utilizzando i CloudWatch parametri di Amazon per tenere traccia di prestazioni, errori e modelli di utilizzo. Amazon Keyspaces fornisce i seguenti CloudWatch parametri per il monitoraggio delle operazioni batch per tabella:

- `SuccessfulRequestCount`— Tieni traccia delle operazioni batch riuscite.
- `Latency`— Misura le prestazioni delle operazioni in batch.
- `ConsumedWriteCapacityUnits`— Monitora il consumo di capacità delle operazioni in batch.

Per ulteriori informazioni, consulta [the section called “Parametri di Amazon Keyspaces”](#).

Oltre ai CloudWatch parametri, puoi utilizzarli AWS CloudTrail per registrare tutte le azioni dell'API Amazon Keyspaces. Ogni azione API nel batch viene registrata, CloudTrail semplificando il monitoraggio e la verifica delle operazioni batch nelle tabelle Amazon Keyspaces.

### Esempi di operazioni in batch

Di seguito è riportato un esempio di istruzione batch di base registrata.

```
BEGIN BATCH
 INSERT INTO users (id, firstname, lastname) VALUES (1, 'John', 'Doe');
 INSERT INTO users (id, firstname, lastname) VALUES (2, 'Jane', 'Smith');
APPLY BATCH;
```

Questo è un esempio di batch che include DELETE istruzioni INSERTUPDATE, e.

```
BEGIN BATCH
 INSERT INTO users (id, firstname, lastname) VALUES (1, 'John', 'Doe');
 UPDATE users SET firstname = 'Johnny' WHERE id = 2;
 DELETE FROM users WHERE id = 3;
APPLY BATCH;
```

Questo è un esempio di batch che utilizza timestamp lato client.

```
BEGIN BATCH
 INSERT INTO users (id, firstname, lastname) VALUES (1, 'John', 'Stiles') USING
 TIMESTAMP 1669069624;
 INSERT INTO users (id, firstname, lastname) VALUES (2, 'Jane', 'Doe') USING
 TIMESTAMP 1669069624;
APPLY BATCH;

BEGIN BATCH
 UPDATE users USING TIMESTAMP 1669069624 SET firstname = 'Carlos' WHERE id = 1;
 UPDATE users USING TIMESTAMP 1669069624 SET firstname = 'Diego' WHERE id = 2;
APPLY BATCH;
```

Questo è un esempio di batch condizionale.

```
BEGIN BATCH
 INSERT INTO users (id, firstname, lastname) VALUES (1, 'Jane', 'Doe') IF NOT
 EXISTS;
 INSERT INTO users (id, firstname, lastname) VALUES (2, 'John', 'Doe') IF NOT
 EXISTS;
APPLY BATCH;

BEGIN BATCH
 UPDATE users SET lastname = 'Stiles' WHERE id = 1 IF lastname = 'Doe';
 UPDATE users SET lastname = 'Stiles' WHERE id = 2 IF lastname = 'Doe';
APPLY BATCH;
```

Questo è un esempio di batch che utilizza Time to Live (TTL).

```
BEGIN BATCH
 INSERT INTO users (id, firstname, lastname) VALUES (1, 'John', 'Doe') USING TTL
 3600;
 INSERT INTO users (id, firstname, lastname) VALUES (2, 'Jane', 'Smith') USING TTL
 7200;
APPLY BATCH;
```

Questo è un esempio di istruzione batch che aggiorna più tabelle.

```
BEGIN BATCH
 INSERT INTO users (id, firstname) VALUES (1, 'John');
 INSERT INTO user_emails (user_id, email) VALUES (1, 'john@example.com');
```



```
APPLY BATCH;
```

Questo è un esempio di operazione batch che utilizza tipi definiti dall'utente (UDTs). L'esempio presuppone che l'`addressUDT` esista.

```
BEGIN BATCH
 INSERT INTO users (id, firstname, address)
 VALUES (1, 'John', {street: '123 Main St', city: 'NYC', zip: '10001'});
 INSERT INTO users (id, firstname, address)
 VALUES (2, 'Jane', {street: '456 Oak Ave', city: 'LA', zip: '90210'});
APPLY BATCH;

BEGIN BATCH
 UPDATE users SET address.zip = '10002' WHERE id = 1;
 UPDATE users SET address.city = 'Boston' WHERE id = 2;
APPLY BATCH;
```

## Ordina i risultati con **ORDER BY** in Amazon Keyspaces

La `ORDER BY` clausola specifica l'ordinamento dei risultati restituiti in un'istruzione. `SELECT` L'istruzione accetta un elenco di nomi di colonna come argomenti e per ogni colonna è possibile specificare l'ordinamento dei dati. È possibile specificare solo colonne di raggruppamento nelle clausole di ordinamento, le colonne non raggruppate non sono consentite.

Le due opzioni di ordinamento disponibili per i risultati restituiti sono l'ordinamento crescente e l'ASCordinamento decrescente. `DESC`

```
SELECT * FROM my_keyspace.my_table ORDER BY (col1 ASC, col2 DESC, col3 ASC);
```

| col1 | col2 | col3 |
|------|------|------|
| 0    | 6    | a    |
| 1    | 5    | b    |
| 2    | 4    | c    |
| 3    | 3    | d    |
| 4    | 2    | e    |
| 5    | 1    | f    |
| 6    | 0    | g    |

```
SELECT * FROM my_keyspace.my_table ORDER BY (col1 DESC, col2 ASC, col3 DESC);
```

| col1 | col2 | col3 |
|------|------|------|
| 6    | 0    | g    |
| 5    | 1    | f    |
| 4    | 2    | e    |
| 3    | 3    | d    |
| 2    | 4    | c    |
| 1    | 5    | b    |
| 0    | 6    | a    |

Se non si specifica il criterio di ordinamento nell'istruzione di query, viene utilizzato l'ordinamento predefinito della colonna di clustering.

I possibili criteri di ordinamento che è possibile utilizzare in una clausola di ordinamento dipendono dall'ordinamento assegnato a ciascuna colonna di raggruppamento al momento della creazione della tabella. I risultati delle query possono essere ordinati solo nell'ordine definito per tutte le colonne di clustering al momento della creazione della tabella o nell'ordine inverso del criterio di ordinamento definito. Non sono consentite altre combinazioni possibili.

Ad esempio, se la tabella `CLUSTERING ORDER` è (col1 ASC, col2 DESC, col3 ASC), i parametri validi per `ORDER BY` sono (col1 ASC, col2 DESC, col3 ASC) o (col1 DESC, col2 ASC, col3 DESC). Per ulteriori informazioni su, vedere sotto. `CLUSTERING ORDER table_options` [the section called “CREATE TABLE”](#)

## Impaginare i risultati in Amazon Keyspaces

Amazon Keyspaces impagina automaticamente i risultati delle `SELECT` istruzioni quando i dati letti per elaborare l'`SELECT`istruzione superano 1 MB. Con l'impaginazione, i risultati dell'`SELECT`istruzione sono suddivisi in «pagine» di dati di dimensioni pari o inferiori a 1 MB.

Un'applicazione può elaborare la prima pagina dei risultati, quindi la seconda pagina e così via. I client devono sempre verificare la presenza di token di impaginazione durante l'elaborazione di `SELECT` query che restituiscono più righe.

Se un client fornisce un messaggio `PAGE SIZE` che richiede la lettura di più di 1 MB di dati, Amazon Keyspaces suddivide automaticamente i risultati in più pagine in base agli incrementi di lettura dei dati di 1 MB.

Ad esempio, se la dimensione media di una riga è di 100 KB e si specifica un valore `PAGE SIZE` di 20, Amazon Keyspaces impagina automaticamente i dati dopo aver letto 10 righe (1000 KB di dati letti).

Poiché Amazon Keyspaces impagina i risultati in base al numero di righe lette per elaborare una richiesta e non al numero di righe restituite nel set di risultati, alcune pagine potrebbero non contenere alcuna riga se si eseguono query filtrate.

Ad esempio, se `PAGE SIZE` imposti su 10 e Keyspaces valuta 30 righe per elaborare la tua query `SELECT`, Amazon Keyspaces restituirà tre pagine. Se solo un sottoinsieme di righe corrisponde alla tua query, alcune pagine potrebbero avere meno di 10 righe. Per un esempio di come le `LIMIT` query possono influire sulla capacità `PAGE SIZE` di lettura, vedere. [the section called “Stima il consumo di capacità di lettura delle query limite”](#)

Per un confronto con l'impaginazione di Apache Cassandra, vedere. [the section called “Paginazione”](#)

## Utilizzo di flussi di acquisizione dati di modifica (CDC) in Amazon Keyspaces

Amazon Keyspaces change data capture (CDC) registra gli eventi di modifica a livello di riga da una tabella Amazon Keyspaces in tempo quasi reale.

Amazon Keyspaces CDC consente casi d'uso basati sugli eventi come l'IoT industriale e il rilevamento delle frodi, nonché casi d'uso di elaborazione dei dati come la ricerca di testo completo e l'archiviazione dei dati. Gli eventi di modifica che Amazon Keyspaces CDC acquisisce negli stream possono essere utilizzati da applicazioni downstream che svolgono funzioni aziendali critiche come analisi dei dati, ricerca di testo, formazione/inferenza ML e backup continui dei dati per l'archiviazione. Ad esempio, puoi trasferire i dati di flusso a servizi di AWS analisi e archiviazione come Amazon OpenSearch Service, Amazon Redshift e Amazon S3 per un'ulteriore elaborazione.

Amazon Keyspaces CDC offre record di modifiche ordinati nel tempo e deduplicati per le tabelle, con ridimensionamento automatico della velocità di trasmissione dei dati e tempi di conservazione fino a 24 ore.

Gli stream CDC di Amazon Keyspaces sono completamente serverless e non è necessario gestire l'infrastruttura di dati per registrare gli eventi di cambiamento. Inoltre, Amazon Keyspaces CDC non consuma alcuna capacità di tabella né per l'elaborazione né per lo storage. Per ulteriori informazioni, consulta [the section called “Come funzionano gli stream CDC”](#).

Puoi utilizzare l'API Amazon Keyspaces Streams per creare applicazioni che utilizzano flussi Amazon Keyspaces CDC e agire in base ai contenuti. Per gli endpoint disponibili, consulta. [the section called “Accedi agli endpoint di flusso CDC”](#)

Per un elenco completo di tutte le operazioni disponibili per Amazon Keyspaces nell'API Streams, consulta Amazon [Keyspaces Streams API Reference](#).

## Argomenti

- [Come funzionano i flussi di change data capture \(CDC\) in Amazon Keyspaces](#)
- [Come utilizzare i flussi di acquisizione dei dati di modifica \(CDC\) in Amazon Keyspaces](#)

# Come funzionano i flussi di change data capture (CDC) in Amazon Keyspaces

Questa sezione fornisce una panoramica di come funzionano i flussi di change data capture (CDC) in Amazon Keyspaces.

Amazon Keyspaces change data capture (CDC) registra una sequenza ordinata di modifiche a livello di riga nelle tabelle di Amazon Keyspaces e archivia queste informazioni in un registro chiamato stream per un massimo di 24 ore. Ogni modifica a livello di riga genera un nuovo record CDC che contiene le informazioni della colonna chiave principale e gli stati «prima» e «dopo» della riga, comprese tutte le colonne. Le applicazioni possono accedere allo stream e visualizzare le mutazioni quasi in tempo reale.

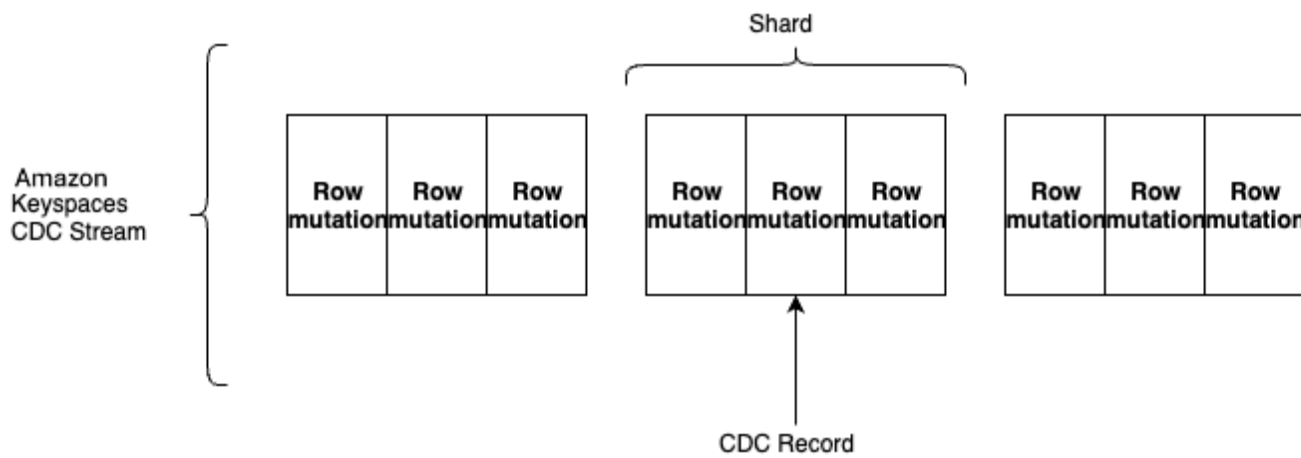
Quando abiliti CDC sulla tua tabella, Amazon Keyspaces crea un nuovo flusso CDC e inizia a raccogliere informazioni su ogni modifica nella tabella. Lo stream CDC ha un Amazon Resource Name (ARN) con il seguente formato:

```
arn:${Partition}:cassandra:{Region}:${Account}:/keyspace/${keyspaceName}/table/
${tableName}/stream/${streamLabel}
```

Puoi selezionare il tipo di informazioni o il tipo di visualizzazione che lo stream CDC raccoglie per ogni record quando abiliti per la prima volta lo stream CDC. Non puoi modificare il tipo di visualizzazione dello stream in seguito. Amazon Keyspaces supporta i seguenti tipi di visualizzazione:

- **NEW\_AND\_OLD\_IMAGES**— Cattura le versioni della riga prima e dopo la mutazione. Questa è l'impostazione predefinita.
- **NEW\_IMAGE**— Cattura la versione della riga dopo la mutazione.
- **OLD\_IMAGE**— Cattura la versione della riga prima della mutazione.
- **KEYS\_ONLY**— Cattura le chiavi di partizione e di clustering della riga che è stata modificata.

Ogni stream CDC è composto da record. Ogni record rappresenta una modifica di una singola riga in una tabella Amazon Keyspaces. I record sono organizzati logicamente in gruppi noti come shard. Questi gruppi sono organizzati logicamente per intervalli della chiave primaria (combinazione di chiavi di partizione, intervalli di chiavi di clustering) e sono un costrutto interno di Amazon Keyspaces. Ogni frammento funge da contenitore per più record e contiene le informazioni necessarie per l'accesso e l'iterazione di questi record.



A ogni record CDC viene assegnato un numero di sequenza, che riflette l'ordine in cui il record è stato pubblicato all'interno dello shard. È garantito che il numero di sequenza sia crescente e unico all'interno di ogni frammento.

Amazon Keyspaces crea ed elimina automaticamente gli shard. In base ai carichi di traffico, Amazon Keyspaces può anche dividere o unire gli shard nel tempo. Ad esempio, Amazon Keyspaces può dividere uno shard in più nuovi shard o unire gli shard in un nuovo shard singolo. Amazon Keyspaces APIs pubblica le informazioni sullo shard e sul flusso CDC per consentire alle applicazioni che utilizzano di elaborare i record nell'ordine corretto accedendo all'intero diagramma di derivazione di uno shard.

Amazon Keyspaces CDC si basa sui seguenti principi su cui puoi fare affidamento durante la creazione della tua applicazione:

- Ogni record di mutazione a livello di riga appare esattamente una volta nello stream CDC.
- Quando si consumano i frammenti in ordine di discendenza, ogni record di mutazione a livello di riga appare nella stessa sequenza dell'ordine di mutazione effettivo sulla chiave primaria.

## Argomenti

- [Come funziona la conservazione dei dati per gli stream CDC in Amazon Keyspaces](#)

- [Come funziona la scadenza dei dati Time to Live \(TTL\) con i flussi CDC in Amazon Keyspaces](#)
- [Come funzionano le operazioni in batch per gli stream CDC in Amazon Keyspaces](#)
- [Come funzionano le colonne statiche nei flussi CDC in Amazon Keyspaces](#)
- [Come funziona la crittografia a riposo per gli stream CDC in Amazon Keyspaces](#)
- [Come funziona la replica multiregionale per i flussi CDC in Amazon Keyspaces](#)
- [Stream CDC e integrazione con i servizi AWS](#)

## Come funziona la conservazione dei dati per gli stream CDC in Amazon Keyspaces

Amazon Keyspaces conserva i record nello stream CDC per un periodo di 24 ore. Non puoi modificare il periodo di conservazione. Se disabiliti CDC su una tabella, i dati nello stream continuano a essere leggibili per 24 ore. Trascorso questo periodo, i dati scadono e i record vengono eliminati automaticamente.

## Come funziona la scadenza dei dati Time to Live (TTL) con i flussi CDC in Amazon Keyspaces

Amazon Keyspaces mostra l'ora di scadenza a livello e a column/cell livello di riga in un campo di metadati chiamato `expirationTime` nei record delle modifiche CDC. Quando Amazon Keyspaces TTL rileva la scadenza di una cella, CDC crea un nuovo record di modifica che mostra TTL come origine della modifica. Per ulteriori informazioni sul TTL, consulta [the section called “Fai scadere i dati con Time to Live”](#)

## Come funzionano le operazioni in batch per gli stream CDC in Amazon Keyspaces

Le operazioni in batch sono suddivise internamente in singole modifiche a livello di riga. Amazon Keyspaces conserva tutti i record all'interno dei flussi CDC a livello di riga, anche se la modifica è avvenuta in un'operazione in batch. Amazon Keyspaces mantiene l'ordine dei record all'interno del flusso CDC nella stessa sequenza dell'ordine di mutazione che si è verificato a livello di riga o sulla chiave primaria.

## Come funzionano le colonne statiche nei flussi CDC in Amazon Keyspaces

I valori statici delle colonne sono condivisi tra tutte le righe di una partizione in Cassandra. A causa di questo comportamento, Amazon Keyspaces acquisisce tutti gli aggiornamenti di una colonna statica come record separato nel flusso CDC. I seguenti esempi riassumono il comportamento delle mutazioni statiche delle colonne:

- Quando viene aggiornata solo la colonna statica, lo stream CDC contiene una modifica di riga per la colonna statica come unica colonna della riga.
- Quando una riga viene aggiornata senza alcuna modifica alla colonna statica, lo stream CDC contiene una modifica di riga che contiene tutte le colonne tranne la colonna statica.
- Quando una riga viene aggiornata insieme alla colonna statica, lo stream CDC contiene due modifiche di riga separate, una per la colonna statica e l'altra per il resto della riga.

## Come funziona la crittografia a riposo per gli stream CDC in Amazon Keyspaces

Per crittografare i dati inattivi nel log ordinato dal CDC, Amazon Keyspaces utilizza la stessa chiave di crittografia già utilizzata per la tabella. Per ulteriori informazioni sulla crittografia dei dati inattivi, consultare [the section called “Crittografia a riposo”](#).

## Come funziona la replica multiregionale per i flussi CDC in Amazon Keyspaces

Puoi abilitare e disabilitare i flussi CDC per le singole repliche di una tabella multiregionale utilizzando l'API o il comando CQL. `update-table ALTER TABLE A` causa della replica asincrona e della risoluzione dei conflitti, i flussi CDC per le tabelle multiregionali non sono coerenti tra loro. Regioni AWS Pertanto, i record che Amazon Keyspaces acquisisce nello stream potrebbero apparire in un ordine diverso in diverse regioni.

Per ulteriori informazioni sulla replica in più regioni, consulta. [the section called “Replica multi regione”](#)

## Stream CDC e integrazione con i servizi AWS

### Come utilizzare gli endpoint VPC per flussi CDC in Amazon Keyspaces

Puoi utilizzare gli endpoint VPC per accedere ai flussi CDC di Amazon Keyspaces. Per informazioni su come creare e accedere agli endpoint VPC per gli stream, consulta. [the section called “Utilizzo degli endpoint VPC dell'interfaccia con flussi”](#)

### Come CloudWatch funziona il monitoraggio con per gli stream CDC in Amazon Keyspaces

Puoi utilizzare Amazon CloudWatch per monitorare le chiamate API effettuate all'endpoint CDC di Amazon Keyspaces. Per ulteriori informazioni sui parametri disponibili, consulta. [the section called “Metriche per Amazon Keyspaces change data capture \(CDC\)”](#)

## Come CloudTrail funziona la registrazione con per i flussi CDC in Amazon Keyspaces

Amazon Keyspaces CDC è integrato con AWS CloudTrail un servizio che fornisce una registrazione delle azioni intraprese da un utente, un ruolo o un servizio AWS in Amazon Keyspaces. CloudTrail acquisisce le chiamate API Data Definition Language (DDL) e le chiamate API Data Manipulation Language (DML) per Amazon Keyspaces come eventi. Le chiamate acquisite includono chiamate dalla console Amazon Keyspaces e chiamate programmatiche alle operazioni dell'API Amazon Keyspaces.

Per ulteriori informazioni sugli eventi CDC acquisiti da, consulta. CloudTrail [the section called “Registrazione con CloudTrail”](#)

## Come funziona il tagging per gli stream CDC in Amazon Keyspaces

Gli stream CDC di Amazon Keyspaces sono una risorsa etichettabile. Puoi taggare uno stream quando crei una tabella a livello di codice utilizzando CQL, l'SDK o il. AWS CLI. Puoi anche taggare stream esistenti, eliminare tag o visualizzare i tag di uno stream. Per ulteriori informazioni, consulta [the section called “Aggiungi tag a keyspace, tabelle e stream”](#).

## Come utilizzare i flussi di acquisizione dei dati di modifica (CDC) in Amazon Keyspaces

### Argomenti

- [Configura le autorizzazioni per lavorare con i flussi CDC in Amazon Keyspaces](#)
- [Come accedere agli endpoint di streaming CDC in Amazon Keyspaces](#)
- [Abilita uno stream CDC durante la creazione di una nuova tabella in Amazon Keyspaces](#)
- [Abilita uno stream CDC per una tabella esistente in Amazon Keyspaces](#)
- [Disattiva uno stream CDC in Amazon Keyspaces](#)
- [Visualizza gli stream CDC in Amazon Keyspaces](#)
- [Accedi ai record negli stream CDC in Amazon Keyspaces](#)
- [Usa la Kinesis Client Library \(KCL\) per elaborare i flussi di Amazon Keyspaces](#)

## Configura le autorizzazioni per lavorare con i flussi CDC in Amazon Keyspaces

Per abilitare gli stream CDC, il principale, ad esempio un utente o un ruolo IAM, necessita delle seguenti autorizzazioni.



Per ulteriori informazioni su AWS Identity and Access Management, vedere [the section called “AWS Identity and Access Management”](#)

Autorizzazioni per abilitare uno stream CDC per una tabella

[Per abilitare uno stream CDC per una tabella Amazon Keyspaces, il principale necessita innanzitutto delle autorizzazioni per creare o modificare una tabella e in secondo luogo delle autorizzazioni per creare il ruolo CDC collegato al servizio. AWSService RoleForAmazonKeyspaces](#) Amazon Keyspaces utilizza il ruolo collegato al servizio per pubblicare i CloudWatch parametri nel tuo account per tuo conto.

La seguente policy IAM ne è un esempio.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "cassandra:Create",
 "cassandra:CreateMultiRegionResource",
 "cassandra:Alter",
 "cassandra:AlterMultiRegionResource"
],
 "Resource": [
 "arn:aws:cassandra:us-east-1:111122223333:/keyspace/my_keyspace/*",
 "arn:aws:cassandra:us-east-1:111122223333:/keyspace/system*"
]
 },
 {
 "Sid": "KeyspacesCDCServiceLinkedRole",
 "Effect": "Allow",
 "Action": "iam:CreateServiceLinkedRole",
 "Resource": "arn:aws:iam::*:role/aws-service-role/cassandra-streams.amazonaws.com/AWSServiceRoleForAmazonKeyspacesCDC",
 "Condition": {
 "StringLike": {
 "iam:AWSServiceName": "cassandra-streams.amazonaws.com"
 }
 }
 }
]
}
```

Per disabilitare uno stream, sono necessarie solo ALTER TABLE le autorizzazioni.

### Autorizzazioni per visualizzare uno stream CDC

Per visualizzare o elencare gli stream CDC, il principale necessita delle autorizzazioni di lettura per lo spazio delle chiavi di sistema. Per ulteriori informazioni, consulta [the section called “system\\_schema\\_mcs”](#).

La seguente politica IAM ne è un esempio.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "cassandra:Select",
 "Resource": [
 "arn:aws:cassandra:us-east-1:111122223333:/keyspace/system*"
]
 }
]
}
```

Per visualizzare o elencare gli stream CDC con l' AWS CLI API Amazon Keyspaces, il principale necessita di autorizzazioni aggiuntive per le azioni e. `cassandra:ListStreams`  
`cassandra:GetStream`

La seguente policy IAM ne è un esempio.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "cassandra:Select",
 "cassandra:ListStreams",
 "cassandra:GetStream"
],
 "Resource": "*"
 }
]
}
```

```
}
```

## Autorizzazioni per leggere uno stream CDC

Per leggere gli stream CDC, il principale necessita delle seguenti autorizzazioni.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "cassandra:GetStream",
 "cassandra:GetShardIterator",
 "cassandra:GetRecords"
],
 "Resource": [
 "arn:aws:cassandra:us-east-1:111122223333:/keyspace/my_keyspace/
table/my_table/stream/stream_label"
]
 }
]
}
```

## Autorizzazioni per elaborare gli stream CDC di Amazon Keyspaces con la Kinesis Client Library (KCL)

Per elaborare i flussi CDC di Amazon Keyspaces con KCL, il responsabile IAM necessita delle seguenti autorizzazioni.

- Amazon Keyspaces— Accesso in sola lettura a uno stream Amazon Keyspaces CDC specificato.
- DynamoDB— Autorizzazioni per creare shard lease tabelle, accesso in lettura e scrittura alle tabelle e accesso in lettura all'indice, come richiesto per l'elaborazione del flusso KCL.
- CloudWatch— Autorizzazioni a pubblicare dati metrici da flussi Amazon Keyspaces CDC elaborati con KCL nello spazio dei nomi dell'applicazione client KCL del tuo account. CloudWatch Per ulteriori informazioni sul monitoraggio, consulta [Monitorare la Kinesis Client Library con Amazon CloudWatch](#)

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

```

"Statement":[
 {
 "Effect":"Allow",
 "Action":[
 "cassandra:GetStream",
 "cassandra:GetShardIterator",
 "cassandra:GetRecords"
],
 "Resource":[
 "arn:aws:cassandra:us-east-1:111122223333:/keyspace/my_keyspace/
table/my_table/stream/stream_label"
]
 },
 {
 "Effect":"Allow",
 "Action":[
 "dynamodb:CreateTable",
 "dynamodb:DescribeTable",
 "dynamodb:UpdateTable",
 "dynamodb:GetItem",
 "dynamodb:UpdateItem",
 "dynamodb:PutItem",
 "dynamodb>DeleteItem",
 "dynamodb:Scan"
],
 "Resource":[
 "arn:aws:dynamodb:us-east-1:111122223333:table/KCL_APPLICATION_NAME"
]
 },
 {
 "Effect":"Allow",
 "Action":[
 "dynamodb:CreateTable",
 "dynamodb:DescribeTable",
 "dynamodb:GetItem",
 "dynamodb:UpdateItem",
 "dynamodb:PutItem",
 "dynamodb>DeleteItem",
 "dynamodb:Scan"
],
 "Resource":[
 "arn:aws:dynamodb:us-east-1:111122223333:table/KCL_APPLICATION_NAME-
WorkerMetricStats",

```

```
 "arn:aws:dynamodb:us-east-1:111122223333:table/KCL_APPLICATION_NAME-CoordinatorState"
],
 {
 "Effect": "Allow",
 "Action": [
 "dynamodb:Query"
],
 "Resource": [
 "arn:aws:dynamodb:us-east-1:111122223333:table/KCL_APPLICATION_NAME/index/*"
]
 },
 {
 "Effect": "Allow",
 "Action": [
 "cloudwatch:PutMetricData"
],
 "Resource": "*"
 }
]
 }
}
```

## Come accedere agli endpoint di streaming CDC in Amazon Keyspaces

Amazon Keyspaces mantiene [endpoint separati per keyspaces/tables e per i](#) flussi CDC in ognuno dei quali è disponibile Regione AWS Amazon Keyspaces. Per accedere a uno stream CDC, seleziona la regione della tabella e sostituisci il `cassandra` prefisso con `cassandra-streams` nel nome dell'endpoint, come mostrato nell'esempio seguente:

| Nome Regione          | Regione   | Endpoint per keyspace e tabelle     | Endpoint per stream                         |
|-----------------------|-----------|-------------------------------------|---------------------------------------------|
| US East (N. Virginia) | us-east-1 | <i>cassandra</i> .us-east-1.api.aws | <i>cassandra-streams</i> .us-east-1.api.aws |

La tabella seguente contiene un elenco completo degli endpoint pubblici disponibili per. Amazon Keyspaces change data capture streams Amazon Keyspaces CDC streams supporta entrambi IPv4

e. IPv6 Tutti gli endpoint pubblici, ad esempio `cassandra-streams.us-east-1.api.aws`, sono endpoint dual stack che possono essere configurati per e. IPv4 IPv6

| Nome della regione                                  | Regione    | Endpoint                                          | Protocollo |  |
|-----------------------------------------------------|------------|---------------------------------------------------|------------|--|
| US East (Ohio)                                      | us-east-2  | <code>cassandra-streams.us-east-2.api.aws</code>  | HTTPS      |  |
| US East (N. Virginia)                               | us-east-1  | <code>cassandra-streams.us-east-1.api.aws</code>  | HTTPS      |  |
| Stati Uniti occidentali (California settentrionale) | us-west-1  | <code>cassandra-streams.us-west-1.api.aws</code>  | HTTPS      |  |
| US West (Oregon)                                    | us-west-2  | <code>cassandra-streams.us-west-2.api.aws</code>  | HTTPS      |  |
| Africa (Cape Town)                                  | af-south-1 | <code>cassandra-streams.af-south-1.api.aws</code> | HTTPS      |  |
| Asia Pacifico (Hong Kong)                           | ap-east-1  | <code>cassandra-streams.ap-east-1.api.aws</code>  | HTTPS      |  |
| Asia Pacifico (Mumbai)                              | ap-south-1 | <code>cassandra-streams.ap-south-1.api.aws</code> | HTTPS      |  |

| Nome della regione        | Regione        | Endpoint                                 | Protocollo |  |
|---------------------------|----------------|------------------------------------------|------------|--|
| Asia Pacifico (Seoul)     | ap-northeast-2 | cassandra-streams.ap-northeast-2.api.aws | HTTPS      |  |
| Asia Pacifico (Singapore) | ap-southeast-1 | cassandra-streams.ap-southeast-1.api.aws | HTTPS      |  |
| Asia Pacifico (Sydney)    | ap-southeast-2 | cassandra-streams.ap-southeast-2.api.aws | HTTPS      |  |
| Asia Pacifico (Tokyo)     | ap-northeast-1 | cassandra-streams.ap-northeast-1.api.aws | HTTPS      |  |
| Canada (Centrale)         | ca-central-1   | cassandra-streams.ca-central-1.api.aws   | HTTPS      |  |
| Europa (Francoforte)      | eu-central-1   | cassandra-streams.eu-central-1.api.aws   | HTTPS      |  |
| Europa (Irlanda)          | eu-west-1      | cassandra-streams.eu-west-1.api.aws      | HTTPS      |  |
| Europa (Londra)           | eu-west-2      | cassandra-streams.eu-west-2.api.aws      | HTTPS      |  |
| Europa (Parigi)           | eu-west-3      | cassandra-streams.eu-west-3.api.aws      | HTTPS      |  |

| Nome della regione                     | Regione       | Endpoint                                | Protocollo |  |
|----------------------------------------|---------------|-----------------------------------------|------------|--|
| Europa (Stoccolma)                     | eu-north-1    | cassandra-streams.eu-north-1.api.aws    | HTTPS      |  |
| Medio Oriente (Bahrein)                | me-south-1    | cassandra-streams.me-south-1.api.aws    | HTTPS      |  |
| Medio Oriente (Emirati Arabi Uniti)    | me-central-1  | cassandra-streams.me-central-1.api.aws  | HTTPS      |  |
| Sud America (São Paulo)                | sa-east-1     | cassandra-streams.sa-east-1.api.aws     | HTTPS      |  |
| AWS GovCloud (Stati Uniti orientali)   | us-gov-east-1 | cassandra-streams.us-gov-east-1.api.aws | HTTPS      |  |
| AWS GovCloud (Stati Uniti occidentali) | us-gov-west-1 | cassandra-streams.us-gov-west-1.api.aws | HTTPS      |  |



## Abilita uno stream CDC durante la creazione di una nuova tabella in Amazon Keyspaces

Per abilitare uno stream CDC quando crei una tabella, puoi usare l'`CREATE TABLE` istruzione in CQL o il `create-table` comando con AWS CLI

Per ogni riga modificata nella tabella, Amazon Keyspaces può acquisire le seguenti modifiche in base a `view_type` quella `cdc_specification` selezionata:

- `NEW_AND_OLD_IMAGES`— entrambe le versioni della riga, prima e dopo la modifica. Questa è l'impostazione predefinita.
- `NEW_IMAGE`— la versione della riga dopo la modifica.
- `OLD_IMAGE`— la versione della riga prima della modifica.
- `KEYS_ONLY`— le chiavi di partizione e di clustering della riga che è stata modificata.

Per informazioni su come etichettare uno stream, consulta [the section called “Crea stream con tag con una nuova tabella”](#)

### Note

Amazon Keyspaces CDC richiede la presenza di un ruolo collegato al servizio (`AWSServiceRoleForAmazonKeyspacesCDC`) che pubblica i dati metrici provenienti dai flussi di Amazon Keyspaces CDC nel tuo account per tuo conto.

`"cloudwatch:namespace": "AWS/Cassandra"` CloudWatch Questo ruolo viene creato automaticamente per te. Per ulteriori informazioni, consulta [the section called “Stream CDC”](#).

## Cassandra Query Language (CQL)

Abilita uno stream CDC quando crei una tabella con CQL

1. 

```
CREATE TABLE mykeyspace.mytable (a text, b text, PRIMARY KEY(a))
WITH CUSTOM_PROPERTIES={'cdc_specification': {'view_type': 'NEW_IMAGE'}} AND CDC
= TRUE;
```
2. Per confermare le impostazioni dello stream, puoi usare la seguente dichiarazione.

```
SELECT keyspace_name, table_name, cdc, custom_properties FROM
system_schema_mcs.tables WHERE keyspace_name = 'mykeyspace' AND table_name =
'mytable';
```

L'output di tale dichiarazione dovrebbe essere simile a questo.

```
SELECT keyspace_name, table_name, cdc, custom_properties FROM
system_schema_mcs.tables WHERE keyspace_name = 'mykeyspace' AND table_name =
'mytable';keyspace_name | table_name | cdc | custom_properties
-----+-----+-----+-----
+-----+-----+-----+-----
mykeyspace | mytable | True | {'capacity_mode':
{'last_update_to_pay_per_request_timestamp': '1741383893782',
'throughput_mode': 'PAY_PER_REQUEST'}, 'cdc_specification':
{'latest_stream_arn': 'arn:aws:cassandra:us-east-1:111122223333:/keyspace/
mykeyspace/table/mytable/stream/2025-03-07T21:44:53.783', 'status': 'ENABLED',
'view_type': 'NEW_IMAGE'}, 'encryption_specification': {'encryption_type':
'AWS_OWNED_KMS_KEY'}, 'point_in_time_recovery': {'status': 'disabled'}}>
```

## CLI

Abilita uno stream CDC quando crei una tabella con AWS CLI

1. Per creare uno stream puoi usare la seguente sintassi.

```
aws keyspaces create-table \
--keyspace-name 'mykeyspace' \
--table-name 'mytable' \
--schema-definition 'allColumns=[{name=a,type=text},
{name=b,type=text}],partitionKeys=[{name=a}]' \
--cdc-specification status=ENABLED,viewType=NEW_IMAGE
```

2. L'output di quel comando mostra la create-table risposta standard ed è simile a questo esempio.

```
{ "resourceArn": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/mykeyspace/
table/mytable" }
```

## Abilita uno stream CDC per una tabella esistente in Amazon Keyspaces

Per abilitare uno stream CDC per una tabella esistente, puoi usare l'`ALTER TABLE` istruzione in CQL, il `update-table` comando con o puoi usare la AWS CLI console.

Per ogni riga modificata nella tabella, Amazon Keyspaces può acquisire le seguenti modifiche in base a `view_type` quella `cdc_specification` selezionata:

- `NEW_AND_OLD_IMAGES`— entrambe le versioni della riga, prima e dopo la modifica. Questa è l'impostazione predefinita.
- `NEW_IMAGE`— la versione della riga dopo la modifica.
- `OLD_IMAGE`— la versione della riga prima della modifica.
- `KEYS_ONLY`— le chiavi di partizione e di clustering della riga che è stata modificata.

Per informazioni su come etichettare uno stream, consulta [the section called “Aggiungi nuovi tag a uno stream”](#)

### Note

Amazon Keyspaces CDC richiede la presenza di un ruolo collegato al servizio (`AWSServiceRoleForAmazonKeyspacesCDC`) che pubblica i dati metrici provenienti dai flussi di Amazon Keyspaces CDC nel tuo account per tuo conto. `"cloudwatch:namespace": "AWS/Cassandra"` CloudWatch Questo ruolo viene creato automaticamente per te. Per ulteriori informazioni, consulta [the section called “Stream CDC”](#).

## Cassandra Query Language (CQL)

### Abilita uno stream (flusso CDC) con CQL

È possibile utilizzare `ALTER TABLE` per abilitare uno stream per una tabella esistente.

1. L'esempio seguente crea uno stream che acquisisce solo le modifiche alle chiavi di partizione e di clustering di una riga modificata.

```
ALTER TABLE mykeyspace.mytable
WITH cdc = TRUE
WITH CUSTOM_PROPERTIES={'cdc_specification': {'view_type': 'KEYS_ONLY'}};
```

2. Per verificare le impostazioni dello stream, è possibile utilizzare la seguente dichiarazione.

```
SELECT keyspace_name, table_name, cdc, custom_properties FROM
system_schema_mcs.tables WHERE keyspace_name = 'mykeyspace' AND table_name =
'mytable';
```

L'output dell'istruzione è simile a questo.

```
keyspace_name | table_name | cdc | custom_properties
-----+-----+-----+-----
+-----+-----+-----+-----
mykeyspace | mytable | True | {'capacity_mode':
{'last_update_to_pay_per_request_timestamp': '1741385897045',
'throughput_mode': 'PAY_PER_REQUEST'}, 'cdc_specification':
{'latest_stream_arn': 'arn:aws:cassandra:us-east-1:111122223333:/keyspace/
mykeyspace/table/mytable/stream/2025-03-07T22:20:10.454', 'status': 'ENABLED',
'view_type': 'KEYS_ONLY'}, 'encryption_specification': {'encryption_type':
'AWS_OWNED_KMS_KEY'}, 'point_in_time_recovery': {'status': 'disabled'}}
```

## CLI

### Crea uno stream CDC con il AWS CLI

1. Per creare uno stream per una tabella esistente è possibile utilizzare la seguente sintassi.

```
aws keyspaces update-table \
--keyspace-name 'mykeyspace' \
--table-name 'mytable' \
--cdc-specification status=ENABLED,viewType=NEW_AND_OLD_IMAGES
```

2. L'output di quel comando mostra la create-table risposta standard ed è simile a questo esempio.

```
{ "resourceArn": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/mykeyspace/
table/mytable" }
```

## Console

Abilita uno stream CDC con la console Amazon Keyspaces

1. [Accedi a e apri Console di gestione AWS la console Amazon Keyspaces a casahttps://console.aws.amazon.com/keyspaces/.](https://console.aws.amazon.com/keyspaces/)
2. Nel pannello di navigazione, scegli Tabelle, quindi scegli una tabella dall'elenco.
3. Scegli la scheda Streams.
4. Scegli Modifica per abilitare uno streaming.
5. Seleziona Attiva gli stream.
6. Scegli il tipo di visualizzazione dello stream. Sono disponibili le seguenti opzioni. Tieni presente che non puoi modificare il tipo di visualizzazione di uno stream dopo che è stato creato.
  - Immagini nuove e vecchie: Amazon Keyspaces acquisisce entrambe le versioni della riga, prima e dopo la modifica. Questa è l'impostazione predefinita.
  - Nuova immagine: Amazon Keyspaces acquisisce solo la versione della riga dopo la modifica.
  - Vecchia immagine: Amazon Keyspaces acquisisce solo la versione della riga prima della modifica.
  - Solo chiave primaria: Amazon Keyspaces acquisisce solo le colonne chiave di partizione e clustering della riga che è stata modificata.
7. Per finire, scegli Salva modifiche.

## Disattiva uno stream CDC in Amazon Keyspaces

Per disabilitare uno stream CDC in un keyspace, puoi usare l'`ALTER TABLE`istruzione in CQL, il `update-table` comando con o la console. AWS CLI

### Cassandra Query Language (CQL)

Disabilita uno stream (flusso CDC) con CQL

1. Per disabilitare uno stream, puoi usare la seguente dichiarazione.

```
ALTER TABLE mykeyspace.mytable
```

```
WITH cdc = FALSE;
```

2. Per confermare che lo stream è disabilitato, puoi usare la seguente dichiarazione.

```
SELECT keyspace_name, table_name, cdc, custom_properties FROM
system_schema_mcs.tables WHERE keyspace_name = 'mykeyspace' AND table_name =
'mytable';
```

L'output di questa dichiarazione è simile a questo.

```
keyspace_name | table_name | cdc | custom_properties
-----+-----+-----+-----
+-----+-----+-----+-----
mykeyspace | mytable | False | {'capacity_mode':
{'last_update_to_pay_per_request_timestamp': '1741385668642',
'throughput_mode': 'PAY_PER_REQUEST'}, 'encryption_specification':
{'encryption_type': 'AWS_OWNED_KMS_KEY'}, 'point_in_time_recovery': {'status':
'disabled'}}
```

## CLI

Disabilita uno stream (flusso CDC) con il AWS CLI

1. Per disabilitare uno stream, puoi usare il seguente comando.

```
aws keyspaces update-table \
--keyspace-name 'mykeyspace' \
--table-name 'mytable' \
--cdc-specification status=DISABLED
```

2. L'output del comando è simile a quello di questo esempio.

```
{
 "keyspaceArn": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/
my_keyspace/",
 "streamName": "my_stream"
}
```

## Console

Disattiva uno stream (stream CDC) con la console Amazon Keyspaces

1. [Accedi a e apri Console di gestione AWS la console Amazon Keyspaces a casahttps://console.aws.amazon.com/keyspaces/.](https://console.aws.amazon.com/keyspaces/)
2. Nel pannello di navigazione, scegli Tabelle, quindi scegli una tabella dall'elenco.
3. Scegli la scheda Streams.
4. Scegli Modifica.
5. Deseleziona Attiva gli stream.
6. Scegli Salva modifiche per disabilitare lo streaming.

## Visualizza gli stream CDC in Amazon Keyspaces

Per visualizzare o elencare tutti i flussi in keyspace, puoi interrogare la tabella `system_schema_mcs.streams` nello spazio delle chiavi di sistema utilizzando un'istruzione in CQL o utilizzare `list-stream` i comandi `get-stream` and con o la console. AWS CLI

Per le autorizzazioni richieste, consulta [the section called “Configurazione delle autorizzazioni”](#).

## Cassandra Query Language (CQL)

Visualizza gli stream CDC con CQL

- Per monitorare lo stato CDC della tabella, è possibile utilizzare la seguente dichiarazione.

```
SELECT custom_properties
FROM system_schema_mcs.tables
WHERE keyspace_name='my_keyspace' and table_name='my_table';
```

L'output del comando è simile a questo.

```
...
custom_properties

{'cdc_specification':{'status': 'Enabled', 'view_type': 'NEW_IMAGE',
 'latest_stream_arn': 'arn:aws:cassandra:us-east-1:111122223333:/keyspace/
my_keyspace/table/my_table/stream/stream_label'}}
...
```

## CLI

### Visualizza gli stream CDC con il AWS CLI

1. Questo esempio mostra come visualizzare le informazioni sullo stream per una tabella.

```
aws keyspaces get-table \
--keyspace-name 'my_keyspace' \
--table-name 'my_table'
```

L'output del comando è simile al seguente.

```
{
 "keyspaceName": "my_keyspace",
 "tableName": "my_table",
 ... Other fields ...,
 "latestStreamArn": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/
my_keyspace/table/my_table/stream/stream_label",
 "cdcSpecification": {
 "status": "ENABLED",
 "viewType": "NEW_AND_OLD_IMAGES"
 }
}
```

2. Puoi elencare tutti gli stream del tuo account in un determinato Regione AWS modo. Il comando seguente ne è un esempio.

```
aws keyspacesstreams list-streams --region us-east-1
```

L'output del comando potrebbe essere simile a questo.

```
{
 "Streams": [
 {
 "StreamArn": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/
ks_1/table/t1/stream/2023-05-11T21:21:33.291",
 "StreamLabel": "2023-05-11T21:21:33.291",
 "KeyspaceName": "ks_1",
 "TableName": "t1",
 },
 {
```



```

 "StreamArn": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/
ks_1/table/t2/stream/2023-05-11T21:21:33.291",
 "StreamLabel": "2023-05-11T21:21:33.291",
 "KeyspaceName": "ks_1"Create a keyspace with the name catalog. Note
 that streams are not supported in multi-Region
keyspaces.
 "TableName": "t2",
 },
 {
 "StreamArn": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/
ks_2/table/t1/stream/2023-05-11T21:21:33.291",
 "StreamLabel": "2023-05-11T21:21:33.291",
 "KeyspaceName": "ks_3"
 "TableName": "t1",
 }
]
}

```

3. È inoltre possibile elencare gli stream CDC per un determinato keyspace utilizzando i seguenti parametri.

```
aws keyspacesstreams list-streams --keyspace-name ks_1 --region us-east-1
```

L'output del comando è simile a questo.

```

{
 "Streams": [
 {
 "StreamArn": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/
ks_1/table/t1/stream/2023-05-11T21:21:33.291",
 "StreamLabel": "2023-05-11T21:21:33.291",
 "KeyspaceName": "ks_1"
 "TableName": "t1",
 },
 {
 "StreamArn": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/
ks_1/table/t2/stream/2023-05-11T21:21:33.291",
 "StreamLabel": "2023-05-11T21:21:33.291",
 "KeyspaceName": "ks_1"
 "TableName": "t2",
 }
]
}

```

```
}
```

4. È inoltre possibile elencare gli stream CDC per una determinata tabella utilizzando i seguenti parametri.

```
aws keyspacesstreams list-streams --keyspace-name ks_1 --table-name t2 --
region us-east-1
```

L'output del comando è simile a questo.

```
{
 "Streams": [
 {
 "StreamArn": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/
ks_1/table/t2/stream/2023-05-11T21:21:33.291",
 "StreamLabel": "2023-05-11T21:21:33.291",
 "KeyspaceName": "ks_1",
 "TableName": "t2",
 }
]
}
```

## Console

Visualizza gli stream CDC nella console Amazon Keyspaces

1. [Accedi a e apri Console di gestione AWS la console Amazon Keyspaces a casahttps://console.aws.amazon.com/keyspaces/](https://console.aws.amazon.com/keyspaces/).
2. Nel pannello di navigazione, scegli Tabelle, quindi scegli una tabella dall'elenco.
3. Scegli la scheda Streams per rivedere i dettagli dello stream.

## Accedi ai record negli stream CDC in Amazon Keyspaces

Per accedere ai record di uno stream, utilizzi l'API [Amazon Keyspaces Streams](#). La sezione seguente contiene esempi su come accedere ai record utilizzando AWS CLI

Per le autorizzazioni richieste, consulta [the section called “Configurazione delle autorizzazioni”](#).

## Accedere ai record di uno stream utilizzando il AWS CLI

1. Puoi utilizzare l'API Amazon Keyspaces Streams per accedere ai record delle modifiche dello stream. Per ulteriori informazioni, consulta [Amazon Keyspaces Streams API Reference](#). Per recuperare gli shard all'interno dello stream, puoi utilizzare l'`get-stream` API come mostrato nell'esempio seguente.

```
aws keyspacesstreams get-stream \
--stream-arn 'arn:aws:cassandra:us-east-1:111122223333:/keyspace/mykeyspace/table/
mytable/stream/STREAM_LABEL '
```

Di seguito è riportato un esempio di output.

```
{
 "StreamArn": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/mykeyspace/
table/mytable/stream/2023-05-11T21:21:33.291",
 "StreamStatus": "ENABLED",
 "StreamViewType": "NEW_AND_OLD_IMAGES",
 "CreationRequestDateTime": "<CREATION_TIME>",
 "KeyspaceName": "mykeyspace",
 "TableName": "mytable",
 "StreamLabel": "2023-05-11T21:21:33.291",
 "Shards": [
 {
 "SequenceNumberRange": {
 "EndingSequenceNumber": "<END_SEQUENCE_NUMBER>",
 "StartingSequenceNumber": "<START_SEQUENCE_NUMBER>"
 },
 "ShardId": "<SHARD_ID>"
 },
],
}
```

2. Per recuperare i record dallo stream, iniziate con un iteratore che vi fornisca il punto di partenza per l'accesso ai record. A tale scopo, puoi utilizzare gli shard all'interno del flusso CDC restituito dall'API nel passaggio precedente. Per raccogliere l'iteratore, puoi utilizzare l'API `get-shard-iterator`. Per questo esempio, si utilizza un iteratore di tipo `TRIM_HORIZON` che recupera dall'ultimo punto (o inizio) tagliato dello shard.

```
aws keyspacesstreams get-shard-iterator \

```

```
--stream-arn 'arn:aws:cassandra:us-east-1:111122223333:/keyspace/mykeyspace/table/
mytable/stream/STREAM_LABEL' \
--shard-id 'SHARD_ID' \
--shard-iterator-type 'TRIM_HORIZON'
```

L'output del comando è simile a quello dell'esempio seguente.

```
{
 "ShardIterator": "<SHARD_ITERATOR>"
}
```

3. Per recuperare i record CDC utilizzando l'get-recordsAPI, puoi utilizzare l'iteratore restituito nell'ultimo passaggio. Il comando seguente ne è un esempio.

```
aws keyspacesstreams get-records \
--shard-iterator 'SHARD_ITERATOR' \
--limit 100
```

## Usa la Kinesis Client Library (KCL) per elaborare i flussi di Amazon Keyspaces

Questo argomento descrive come utilizzare la Kinesis Client Library (KCL) per consumare ed elaborare i dati dai flussi di Amazon Keyspaces change data capture (CDC).

Invece di lavorare direttamente con l'API Amazon Keyspaces Streams, lavorare con la Kinesis Client Library (KCL) offre molti vantaggi, ad esempio:

- Monitoraggio del lignaggio degli shard e gestione degli iteratori integrati.
- Bilanciamento automatico del carico tra i lavoratori.
- Tolleranza agli errori e ripristino in caso di guasti dei lavoratori.
- Checkpointing per tenere traccia dell'avanzamento dell'elaborazione.
- Adattamento alle variazioni della capacità del flusso.
- Elaborazione distribuita semplificata per l'elaborazione dei record CDC.

La sezione seguente descrive perché e come utilizzare la Kinesis Client Library (KCL) per elaborare i flussi e fornisce un esempio per l'elaborazione di un flusso CDC Amazon Keyspaces con KCL.

Per informazioni sui prezzi, consulta i prezzi di [Amazon Keyspaces \(per Apache Cassandra\)](#).

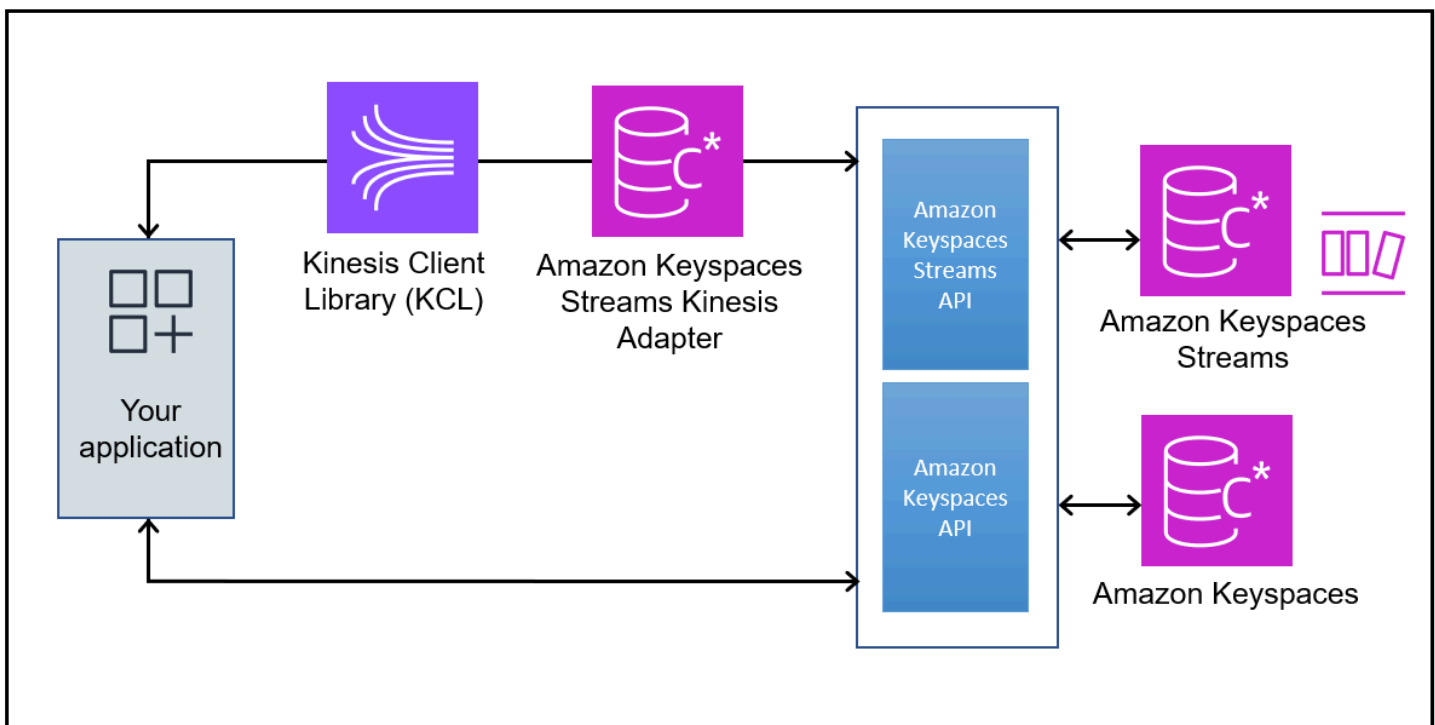
## Cos'è la Kinesis Client Library?

La Kinesis Client Library (KCL) è una libreria software Java autonoma progettata per semplificare il processo di consumo ed elaborazione dei dati dai flussi. KCL gestisce molte delle attività complesse associate all'elaborazione distribuita, consentendoti di concentrarti sull'implementazione della logica aziendale durante l'elaborazione dei dati di flusso. KCL gestisce attività come il bilanciamento del carico tra più lavoratori, la risposta agli errori dei lavoratori, il checkpoint dei record elaborati e la risposta alle variazioni del numero di shard nel flusso.

Per elaborare gli stream CDC di Amazon Keyspaces, puoi utilizzare i modelli di progettazione disponibili in KCL per lavorare con shard di stream e record di stream. KCL semplifica la codifica fornendo astrazioni utili sull'API Kinesis Data Streams di basso livello. Per ulteriori informazioni su KCL, consulta [Development consumer with KCL](#) nella Amazon Kinesis Data Streams Developer Guide.

Per scrivere applicazioni utilizzando KCL, si utilizza l'adattatore Kinesis di Amazon Keyspaces Streams. Il Kinesis Adapter implementa l'interfaccia Kinesis Data Streams in modo da poter utilizzare KCL per consumare ed elaborare i record dai flussi di Amazon Keyspaces. Per istruzioni su come configurare e installare l'adattatore Kinesis di Amazon Keyspaces Streams, visita il repository. [GitHub](#)

Il diagramma seguente mostra come queste librerie interagiscono tra loro.



KCL viene aggiornato frequentemente per incorporare versioni più recenti delle librerie sottostanti, miglioramenti della sicurezza e correzioni di bug. Ti consigliamo di utilizzare l'ultima versione di KCL per evitare problemi noti e beneficiare di tutti gli ultimi miglioramenti. Per trovare l'ultima versione di KCL, consulta l'archivio [GitHub KCL](#).

## Concetti KCL

Prima di implementare un'applicazione consumer utilizzando KCL, è necessario comprendere i seguenti concetti:

### Applicazione consumer KCL

Un'applicazione consumer KCL è un programma che elabora i dati da un flusso CDC di Amazon Keyspaces. Il KCL funge da intermediario tra il codice dell'applicazione consumer e lo stream CDC di Amazon Keyspaces.

### Lavoratore

Un worker è un'unità di esecuzione della tua applicazione consumer KCL che elabora i dati dal flusso CDC di Amazon Keyspaces. La tua applicazione può eseguire più worker distribuiti su più istanze.

### Processore di registrazione

Un record processor è la logica dell'applicazione che elabora i dati da uno shard nel flusso CDC di Amazon Keyspaces. Un processore di registrazione viene istanziato da un worker per ogni shard che gestisce.

### Leasing

Un leasing rappresenta la responsabilità del trattamento di uno shard. I lavoratori utilizzano i contratti di locazione per coordinare quale lavoratore sta elaborando quale frammento. KCL archivia i dati di leasing in una tabella in Amazon DynamoDB.

### Checkpoint

Un checkpoint è una registrazione della posizione nello shard fino alla quale l'elaboratore di dischi ha elaborato con successo i record. Il checkpointing consente all'applicazione di riprendere l'elaborazione dal punto in cui era stata interrotta se un lavoratore fallisce.

Con l'adattatore Amazon Keyspaces Kinesis, puoi iniziare a sviluppare utilizzando l'interfaccia KCL, con le chiamate API dirette senza problemi all'endpoint di streaming Amazon Keyspaces. Per un elenco degli endpoint disponibili, consulta [the section called “Accedi agli endpoint di flusso CDC”](#)

Quando si avvia l'applicazione, quest'ultima richiama la libreria KCL per creare un'istanza di un worker. È necessario fornire al lavoratore le informazioni di configurazione per l'applicazione, come il descrittore di flusso e AWS le credenziali, e il nome di una classe di processore di record fornita. Durante l'esecuzione del codice nel processore di record, il worker completa le seguenti attività:

- Si collega al flusso
- Enumera le partizioni all'interno del flusso
- Coordina le associazioni di shard con altri processi di lavoro (se presenti)
- Crea istanze di un elaboratore di record per ogni shard che gestisce
- Estrae i record di dati dal flusso
- Inserisce i record nell'elaboratore di record corrispondente
- Controlla i record elaborati
- Bilancia le associazioni tra shard e processi di lavoro quando il conteggio delle istanze del lavoro cambia
- Bilancia le associazioni tra partizioni e worker quando le partizioni sono suddivise

Implementa un'applicazione consumer KCL per gli stream CDC di Amazon Keyspaces

Questo argomento fornisce una step-by-step guida all'implementazione di un'applicazione consumer KCL per elaborare i flussi CDC di Amazon Keyspaces.

1. Prerequisiti: prima di iniziare, assicurati di avere:

- Una tabella Amazon Keyspaces con un flusso CDC
- Autorizzazioni IAM necessarie per consentire al principale IAM di accedere al flusso CDC di Amazon Keyspaces, creare e accedere a tabelle DynamoDB per l'elaborazione dei flussi KCL e autorizzazioni su cui pubblicare le metriche. CloudWatch Per ulteriori informazioni e un esempio di policy, consulta [the section called “Autorizzazioni per KCL”](#)
- Assicuratevi che nella configurazione locale siano impostate AWS credenziali valide. Per ulteriori informazioni, consulta [the section called “Gestione delle chiavi di accesso”](#).
- Java Development Kit (JDK) 8 o versione successiva
- Requisiti elencati nel [Readme](#) su Github.

2. In questo passaggio, aggiungi la dipendenza KCL al tuo progetto. Per Maven, aggiungi quanto segue al tuo pom.xml:

```
<dependencies>
 <dependency>
 <groupId>software.amazon.kinesis</groupId>
 <artifactId>amazon-kinesis-client</artifactId>
 <version>3.1.0</version>
 </dependency>
 <dependency>
 <groupId>software.amazon.keyspaces</groupId>
 <artifactId>keyspaces-streams-kinesis-adapter</artifactId>
 <version>1.0.0</version>
 </dependency>
</dependencies>
```

### Note

[Controlla sempre l'ultima versione di KCL nel repository KCL. GitHub](#)

### 3. Crea una classe factory che produca istanze di processori di registrazione:

```
import software.amazon.awssdk.services.keyspacesstreams.model.Record;
import
 software.amazon.keyspaces.streamsadapter.adapter.KeyspacesStreamsClientRecord;
import
 software.amazon.keyspaces.streamsadapter.model.KeyspacesStreamsProcessRecordsInput;
import
 software.amazon.keyspaces.streamsadapter.processor.KeyspacesStreamsShardRecordProcessor;
import software.amazon.kinesis.lifecycle.events.InitializationInput;
import software.amazon.kinesis.lifecycle.events.LeaseLostInput;
import software.amazon.kinesis.lifecycle.events.ShardEndedInput;
import software.amazon.kinesis.lifecycle.events.ShutdownRequestedInput;
import software.amazon.kinesis.processor.RecordProcessorCheckpoint;

public class RecordProcessor implements KeyspacesStreamsShardRecordProcessor {
 private String shardId;

 @Override
 public void initialize(InitializationInput initializationInput) {
 this.shardId = initializationInput.shardId();
 System.out.println("Initializing record processor for shard: " + shardId);
 }
}
```



```

@Override
public void processRecords(KeyspacesStreamsProcessRecordsInput
processRecordsInput) {
 try {
 for (KeyspacesStreamsClientRecord record :
processRecordsInput.records()) {
 Record keyspacesRecord = record.getRecord();
 System.out.println("Received record: " + keyspacesRecord);
 }

 if (!processRecordsInput.records().isEmpty()) {
 RecordProcessorCheckpointier checkpointier =
processRecordsInput.checkpointer();
 try {
 checkpointier.checkpoint();
 System.out.println("Checkpoint successful for shard: " +
shardId);
 } catch (Exception e) {
 System.out.println("Error while checkpointing for shard: " +
shardId + " " + e);
 }
 }
 } catch (Exception e) {
 System.out.println("Error processing records for shard: " + shardId + "
" + e);
 }
}

@Override
public void leaseLost(LeaseLostInput leaseLostInput) {
 System.out.println("Lease lost for shard: " + shardId);
}

@Override
public void shardEnded(ShardEndedInput shardEndedInput) {
 System.out.println("Shard ended: " + shardId);
 try {
 // This is required. Checkpoint at the end of the shard
 shardEndedInput.checkpointer().checkpoint();
 System.out.println("Final checkpoint successful for shard: " +
shardId);
 } catch (Exception e) {
 System.out.println("Error while final checkpointing for shard: " +
shardId + " " + e);
 }
}

```

```

 throw new RuntimeException("Error while final checkpointing", e);
 }
}

@Override
public void shutdownRequested(ShutdownRequestedInput shutdownRequestedInput) {
 System.out.println("Shutdown requested for shard " + shardId);
 try {
 shutdownRequestedInput.checkpointer().checkpoint();
 } catch (Exception e) {
 System.out.println("Error while checkpointing on shutdown for shard: "
+ shardId + " " + e);
 }
}
}

```

4. Create un record factory come illustrato nell'esempio seguente.

```

import software.amazon.kinesis.processor.ShardRecordProcessor;
import software.amazon.kinesis.processor.ShardRecordProcessorFactory;

import java.util.Queue;
import java.util.concurrent.ConcurrentLinkedQueue;

public class RecordProcessorFactory implements ShardRecordProcessorFactory {
 private final Queue<RecordProcessor> processors = new
ConcurrentLinkedQueue<>();

 @Override
 public ShardRecordProcessor shardRecordProcessor() {
 System.out.println("Creating new RecordProcessor");
 RecordProcessor processor = new RecordProcessor();
 processors.add(processor);
 return processor;
 }
}

```

5. In questo passaggio crei la classe base da configurare KCLv3 e l'adattatore Amazon Keyspaces.

```

import com.example.KCLExample.utils.RecordProcessorFactory;
import
software.amazon.keyspaces.streamsadapter.AmazonKeyspacesStreamsAdapterClient;
import software.amazon.keyspaces.streamsadapter.StreamsSchedulerFactory;

```

```
import java.util.Arrays;
import java.util.List;
import java.util.concurrent.ExecutionException;

import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.cloudwatch.CloudWatchAsyncClient;
import software.amazon.awssdk.services.dynamodb.DynamoDbAsyncClient;
import software.amazon.awssdk.services.dynamodb.model.DeleteTableRequest;
import software.amazon.awssdk.services.dynamodb.model.DeleteTableResponse;
import software.amazon.awssdk.services.keyspacesstreams.KeyspacesStreamsClient;
import software.amazon.awssdk.services.kinesis.KinesisAsyncClient;
import software.amazon.kinesis.common.ConfigsBuilder;
import software.amazon.kinesis.common.InitialPositionInStream;
import software.amazon.kinesis.common.InitialPositionInStreamExtended;
import software.amazon.kinesis.coordinator.CoordinatorConfig;
import software.amazon.kinesis.coordinator.Scheduler;
import software.amazon.kinesis.leases.LeaseManagementConfig;
import software.amazon.kinesis.processor.ProcessorConfig;
import software.amazon.kinesis.processor.StreamTracker;
import software.amazon.kinesis.retrieval.polling.PollingConfig;

public class KCLTestBase {

 protected KeyspacesStreamsClient streamsClient;
 protected KinesisAsyncClient adapterClient;
 protected DynamoDbAsyncClient dynamoDbAsyncClient;
 protected CloudWatchAsyncClient cloudWatchClient;
 protected Region region;
 protected RecordProcessorFactory recordProcessorFactory;
 protected Scheduler scheduler;
 protected Thread schedulerThread;

 public void baseSetUp() {
 recordProcessorFactory = new RecordProcessorFactory();
 setupKCLBase();
 }

 protected void setupKCLBase() {
 region = Region.US_EAST_1;

 streamsClient = KeyspacesStreamsClient.builder()
 .region(region)
 .build();
 adapterClient = new AmazonKeyspacesStreamsAdapterClient(
```

```
 streamsClient,
 region);
dynamoDbAsyncClient = DynamoDbAsyncClient.builder()
 .region(region)
 .build();
cloudWatchClient = CloudWatchAsyncClient.builder()
 .region(region)
 .build();
}

protected void startScheduler(Scheduler scheduler) {
 this.scheduler = scheduler;
 schedulerThread = new Thread(() -> scheduler.run());
 schedulerThread.start();
}

protected void shutdownScheduler() {
 if (scheduler != null) {
 scheduler.shutdown();
 try {
 schedulerThread.join(30000);
 } catch (InterruptedException e) {
 System.out.println("Error while shutting down scheduler " + e);
 }
 }
}

protected Scheduler createScheduler(String streamArn, String leaseTableName) {
 String workerId = "worker-" + System.currentTimeMillis();

 // Create ConfigsBuilder
 ConfigsBuilder configsBuilder = createConfigsBuilder(streamArn, workerId,
leaseTableName);

 // Configure retrieval config for polling
 PollingConfig pollingConfig = new PollingConfig(streamArn, adapterClient);

 // Create the Scheduler
 return StreamsSchedulerFactory.createScheduler(
 configsBuilder.checkpointConfig(),
 configsBuilder.coordinatorConfig(),
 configsBuilder.leaseManagementConfig(),
 configsBuilder.lifecycleConfig(),
 configsBuilder.metricsConfig(),
```

```

 configsBuilder.processorConfig(),

configsBuilder.retrievalConfig().retrievalSpecificConfig(pollingConfig),
 streamsClient,
 region
);
}

private ConfigsBuilder createConfigsBuilder(String streamArn, String workerId,
String leaseTableName) {
 ConfigsBuilder configsBuilder = new ConfigsBuilder(
 streamArn,
 leaseTableName,
 adapterClient,
 dynamoDbAsyncClient,
 cloudWatchClient,
 workerId,
 recordProcessorFactory);

 configureCoordinator(configsBuilder.coordinatorConfig());
 configureLeaseManagement(configsBuilder.leaseManagementConfig());
 configureProcessor(configsBuilder.processorConfig());
 configureStreamTracker(configsBuilder, streamArn);

 return configsBuilder;
}

private void configureCoordinator(CoordinatorConfig config) {
 config.skipShardSyncAtWorkerInitializationIfLeasesExist(true)
 .parentShardPollIntervalMillis(1000)
 .shardConsumerDispatchPollIntervalMillis(500);
}

private void configureLeaseManagement(LeaseManagementConfig config) {
 config.shardSyncIntervalMillis(0)
 .leasesRecoveryAuditorInconsistencyConfidenceThreshold(0)
 .leasesRecoveryAuditorExecutionFrequencyMillis(5000)
 .leaseAssignmentIntervalMillis(1000L);
}

private void configureProcessor(ProcessorConfig config) {
 config.callProcessRecordsEvenForEmptyRecordList(true);
}

```

```

 private void configureStreamTracker(ConfigsBuilder configsBuilder, String
streamArn) {
 StreamTracker streamTracker =
StreamsSchedulerFactory.createSingleStreamTracker(
 streamArn,

InitialPositionInStreamExtended.newInitialPosition(InitialPositionInStream.TRIM_HORIZON)
);
 configsBuilder.streamTracker(streamTracker);
 }

 public void deleteAllDdbTables(String baseTableName) {
 List<String> tablesToDelete = Arrays.asList(
 baseTableName,
 baseTableName + "-CoordinatorState",
 baseTableName + "-WorkerMetricStats"
);

 for (String tableName : tablesToDelete) {
 deleteTable(tableName);
 }
 }

 private void deleteTable(String tableName) {
 DeleteTableRequest deleteTableRequest = DeleteTableRequest.builder()
 .tableName(tableName)
 .build();

 try {
 DeleteTableResponse response =
dynamoDbAsyncClient.deleteTable(deleteTableRequest).get();
 System.out.println("Table deletion response " + response);
 } catch (InterruptedException | ExecutionException e) {
 System.out.println("Error deleting table: " + tableName + " " + e);
 }
 }
}

```

6. In questa fase implementerai la classe del processore di record per l'applicazione per avviare l'elaborazione degli eventi di modifica.

```
import software.amazon.kinesis.coordinator.Scheduler;
```

```
public class KCLTest {

 private static final int APP_RUNTIME_SECONDS = 1800;
 private static final int SLEEP_INTERNAL_MS = 60*1000;

 public static void main(String[] args) {
 KCLTestBase kclTestBase;

 kclTestBase = new KCLTestBase();
 kclTestBase.baseSetup();

 // Create and start scheduler
 String leaseTableName = generateUniqueApplicationName();

 // Update below to your Stream ARN
 String streamArn = "arn:aws:cassandra:us-east-1:759151643516:/keyspace/
cdc_sample_test/table/test_kcl_bool/stream/2025-07-01T15:52:57.529";
 Scheduler scheduler = kclTestBase.createScheduler(streamArn,
leaseTableName);
 kclTestBase.startScheduler(scheduler);

 // Wait for specified time before shutting down - KCL applications are
 // designed to run forever, however in this
 // example we will shut it down after APP_RUNTIME_SECONDS
 long startTime = System.currentTimeMillis();
 long endTime = startTime + (APP_RUNTIME_SECONDS * 1000);
 while (System.currentTimeMillis() < endTime) {
 try {
 // Print and sleep every minute
 Thread.sleep(SLEEP_INTERNAL_MS);
 System.out.println("Application is running");
 } catch (InterruptedException e) {
 System.out.println("Interrupted while waiting for records");
 Thread.currentThread().interrupt();
 break;
 }
 }

 // Stop the scheduler
 kclTestBase.shutdownScheduler();
 kclTestBase.deleteAllDdbTables(leaseTableName);
 }

 public static String generateUniqueApplicationName() {
```

```
String timestamp = String.valueOf(System.currentTimeMillis());
String randomString = java.util.UUID.randomUUID().toString().substring(0,
8);
return String.format("KCL-App-%s-%s", timestamp, randomString);
}
}
```

## Best practice

Segui queste best practice quando usi KCL con gli stream CDC di Amazon Keyspaces:

### Gestione errori

Implementa una solida gestione degli errori nel tuo processore di dischi per gestire le eccezioni con garbo. Prendi in considerazione l'implementazione della logica di ripetizione dei tentativi per i guasti transitori.

### Frequenza dei checkpoint

Bilancia la frequenza dei checkpoint per ridurre al minimo l'elaborazione duplicata garantendo al contempo un ragionevole monitoraggio dei progressi. Un checkpoint troppo frequente può influire sulle prestazioni, mentre un checkpoint troppo raro può portare a un maggior numero di ritrattamenti in caso di fallimento di un lavoratore.

### Scalabilità dei lavoratori

Ridimensiona il numero di lavoratori in base al numero di shard presenti nel tuo stream CDC. Un buon punto di partenza è avere un lavoratore per shard, ma potrebbe essere necessario adattarlo in base ai requisiti di elaborazione.

### Monitoraggio

Utilizzate CloudWatch le metriche fornite da KCL per monitorare lo stato e le prestazioni della vostra applicazione consumer. Le metriche chiave includono la latenza di elaborazione, l'età dei checkpoint e il numero di leasing.

### Test in corso

Testa a fondo la tua applicazione consumer, includendo scenari come guasti dei lavoratori, resharding degli stream e condizioni di carico variabili.



## Utilizzo di KCL con linguaggi non Java

Sebbene KCL sia principalmente una libreria Java, puoi utilizzarla con altri linguaggi di programmazione tramite MultiLangDaemon. MultiLangDaemon è un demone basato su Java che gestisce l'interazione tra il processore di record non Java e KCL.

KCL fornisce supporto per le seguenti lingue:

- Python
- Ruby
- Node.js
- .NET

[Per ulteriori informazioni sull'uso di KCL con linguaggi non Java, consulta la documentazione di KCL. MultiLangDaemon](#)

## Risoluzione dei problemi

Questa sezione fornisce soluzioni ai problemi più comuni che potresti riscontrare quando usi KCL con i flussi CDC di Amazon Keyspaces.

### Elaborazione lenta

Se la tua applicazione consumer elabora i record lentamente, considera:

- Aumentare il numero di istanze di lavoro
- Ottimizzazione della logica di elaborazione dei record
- Verifica della presenza di strozzature nei sistemi a valle

### Elaborazione duplicata

Se riscontri un'elaborazione duplicata dei record, controlla la logica del checkpoint. Assicurati di effettuare il checkpoint dopo aver elaborato correttamente i record.

### Fallimenti dei lavoratori

Se i lavoratori falliscono frequentemente, controlla:

- Limiti di risorse (CPU, memoria)
- Problemi di connettività di rete

- Problemi a livello di autorizzazioni

Problemi relativi ai tavoli di locazione

Se riscontri problemi con la tabella di leasing KCL:

- Verifica che la tua applicazione disponga delle autorizzazioni appropriate per accedere alla tabella Amazon Keyspaces
- Verifica che la tabella abbia un throughput assegnato sufficiente

## Utilizzo dei partizionatori in Amazon Keyspaces

In Apache Cassandra, i partizionatori controllano su quali nodi sono archiviati i dati nel cluster. I partizionatori creano un token numerico utilizzando un valore hash della chiave di partizione. Cassandra utilizza questo token per distribuire i dati tra i nodi. I client possono utilizzare questi token anche in SELECT operazioni e WHERE clausole per ottimizzare le operazioni di lettura e scrittura. Ad esempio, i client possono eseguire in modo efficiente query parallele su tabelle di grandi dimensioni specificando intervalli di token distinti da interrogare in ogni processo parallelo.

Amazon Keyspaces fornisce tre diversi partizionatori.

Murmur3 Partitioner (impostazione predefinita)

Compatibile con Apache CassandraMurmur3Partitioner. Murmur3PartitionerÈ il partizionatore Cassandra predefinito in Amazon Keyspaces e in Cassandra 1.2 e versioni successive.

RandomPartitioner

RandomPartitionerCompatibile con Apache Cassandra. RandomPartitionerÈ il partizionatore Cassandra predefinito per le versioni precedenti a Cassandra 1.2.

Partizionatore predefinito di Keyspaces

DefaultPartitionerrestituisce gli stessi risultati di token funzione di. RandomPartitioner

L'impostazione del partizionatore viene applicata per regione a livello di account. Ad esempio, se si modifica il partizionatore negli Stati Uniti orientali (Virginia settentrionale), la modifica viene applicata a tutte le tabelle dello stesso account in questa regione. È possibile modificare il partizionatore in tutta sicurezza in qualsiasi momento. Tieni presente che il completamento della modifica della configurazione richiede circa 10 minuti. Non è necessario ricaricare i dati di Amazon Keyspaces

quando si modifica l'impostazione del partizionatore. I client utilizzeranno automaticamente la nuova impostazione del partizionatore la prossima volta che si conatteranno.

## Come modificare il partizionatore in Amazon Keyspaces

È possibile modificare il partizionatore utilizzando Console di gestione AWS o Cassandra Query Language (CQL).

### Console di gestione AWS

Per modificare il partizionatore utilizzando la console Amazon Keyspaces

1. [Accedi a e apri Console di gestione AWS la console Amazon Keyspaces a casahttps://console.aws.amazon.com/keyspaces/](https://console.aws.amazon.com/keyspaces/).
2. Nel riquadro di navigazione, seleziona Configuration (Configurazione).
3. Nella pagina di configurazione, vai a Modifica partitioner.
4. Seleziona il partizionatore compatibile con la tua versione di Cassandra. La modifica del partizionatore richiede circa 10 minuti per essere applicata.

#### Note

Una volta completata la modifica della configurazione, devi disconnetterti e riconnetterti ad Amazon Keyspaces per richiedere l'utilizzo del nuovo partizionatore.

### Cassandra Query Language (CQL)

1. Per vedere quale partizionatore è configurato per l'account, puoi utilizzare la seguente query.

```
SELECT partitioner from system.local;
```

Se il partizionatore non è stato modificato, la query ha il seguente risultato.

```
partitioner

com.amazonaws.cassandra.DefaultPartitioner
```

2. Per aggiornare il partizionatore al Murmur3 partizionatore, è possibile utilizzare la seguente istruzione.

```
UPDATE system.local set
partitioner='org.apache.cassandra.dht.Murmur3Partitioner' where key='local';
```

3. Si noti che il completamento di questa modifica alla configurazione richiede circa 10 minuti. Per confermare che il partizionatore è stato impostato, è possibile eseguire nuovamente la SELECT query. Si noti che, a causa dell'eventuale coerenza di lettura, la risposta potrebbe non riflettere ancora i risultati della modifica del partizionatore recentemente completata. Se si ripete nuovamente l'SELECT operazione dopo poco tempo, la risposta dovrebbe restituire i dati più recenti.

```
SELECT partitioner from system.local;
```

#### Note

Devi disconnetterti e riconnetterti ad Amazon Keyspaces in modo che le richieste utilizzino il nuovo partizionatore.

## Timestamp lato client in Amazon Keyspaces

In Amazon Keyspaces, i timestamp lato client sono timestamp compatibili con Cassandra che vengono mantenuti per ogni cella della tabella. Puoi utilizzare i timestamp lato client per la risoluzione dei conflitti lasciando che le applicazioni client determinino l'ordine delle scritture. Ad esempio, quando i client di un'applicazione distribuita a livello globale aggiornano gli stessi dati, i timestamp lato client mantengono l'ordine in cui gli aggiornamenti sono stati effettuati sui client. Amazon Keyspaces utilizza questi timestamp per elaborare le scritture.

I timestamp lato client di Amazon Keyspaces sono completamente gestiti. Non è necessario gestire impostazioni di sistema di basso livello come le strategie di pulizia e compattazione.

Quando si eliminano i dati, le righe vengono contrassegnate per l'eliminazione con una lapide. Amazon Keyspaces rimuove automaticamente i dati archiviati (in genere entro 10 giorni) senza influire sulle prestazioni o sulla disponibilità delle applicazioni. I dati di Tombstoned non sono disponibili per le istruzioni DML (Data Manipulation Language). Man mano che si continuano a eseguire operazioni di lettura e scrittura su righe contenenti dati archiviati, i dati archiviati continuano a essere conteggiati ai fini dell'archiviazione, delle unità di capacità di lettura (L) e delle unità di capacità di scrittura (RCUs) fino a quando non vengono eliminati dallo storage. WCUs

Dopo aver attivato i timestamp lato client per una tabella, puoi specificare un timestamp con la clausola nella query CQL del Data Manipulation Language (DML). `USING TIMESTAMP` Per ulteriori informazioni, consulta [the section called “Utilizza i timestamp lato client nelle query”](#). Se non specifichi un timestamp nella tua query CQL, Amazon Keyspaces utilizza il timestamp passato dal driver del client. Se il driver client non fornisce timestamp, Amazon Keyspaces assegna automaticamente un timestamp a livello di cella, perché i timestamp non possono esserlo. `NULL` Per richiedere i timestamp, puoi utilizzare la funzione nell'istruzione DML. `WRITETIME`

Amazon Keyspaces non richiede costi aggiuntivi per l'attivazione dei timestamp lato client. Tuttavia, con i timestamp lato client puoi archiviare e scrivere dati aggiuntivi per ogni valore nella riga. Ciò può comportare un ulteriore utilizzo dello storage e, in alcuni casi, un ulteriore utilizzo del throughput. Per ulteriori informazioni sui prezzi di Amazon Keyspaces, consulta i prezzi di Amazon [Keyspaces \(per Apache Cassandra\)](#).

Quando i timestamp lato client sono attivati in Amazon Keyspaces, ogni colonna di ogni riga memorizza un timestamp. Questi timestamp occupano circa 20-40 byte (a seconda dei dati) e contribuiscono al costo di archiviazione e velocità effettiva della riga. Questi byte di metadati vengono conteggiati anche ai fini della quota di dimensione delle righe di 1 MB. Per determinare l'aumento complessivo dello spazio di archiviazione (per garantire che la dimensione delle righe rimanga inferiore a 1 MB), considera il numero di colonne della tabella e il numero di elementi di raccolta in ogni riga. Ad esempio, se una tabella ha 20 colonne, con ogni colonna che memorizza 40 byte di dati, la dimensione della riga aumenta da 800 byte a 1200 byte. Per ulteriori informazioni su come stimare la dimensione di una riga, vedere. [the section called “Stima della dimensione delle righe”](#) Oltre ai 400 byte aggiuntivi per l'archiviazione, in questo esempio, il numero di unità di capacità di scrittura (WCUs) consumate per scrittura aumenta da 1 WCU a 2. WCUs Per ulteriori informazioni su come calcolare la capacità di lettura e scrittura, vedere. [the section called “Configura le modalità di capacità di lettura/scrittura”](#)

Dopo aver attivato i timestamp lato client per una tabella, non è possibile disattivarla.

Per ulteriori informazioni su come utilizzare i timestamp lato client nelle query, consulta. [the section called “Utilizza i timestamp lato client nelle query”](#)

## Argomenti

- [In che modo i timestamp lato client di Amazon Keyspaces si integrano con i servizi AWS](#)
- [Crea una nuova tabella con timestamp lato client in Amazon Keyspaces](#)
- [Configura i timestamp lato client per una tabella in Amazon Keyspaces](#)
- [Usa i timestamp lato client nelle query in Amazon Keyspaces](#)

# In che modo i timestamp lato client di Amazon Keyspaces si integrano con i servizi AWS

La seguente metrica di timestamp lato client è disponibile in Amazon CloudWatch per consentire il monitoraggio continuo.

- **SystemReconciliationDeletes**— Il numero di operazioni di eliminazione necessarie per rimuovere i dati archiviati.

Per ulteriori informazioni su come monitorare le CloudWatch metriche, consulta [the section called “Monitoraggio con CloudWatch”](#)

Quando lo usi CloudFormation, puoi abilitare i timestamp lato client durante la creazione di una tabella Amazon Keyspaces. Per ulteriori informazioni, consulta la [AWS CloudFormation Guida per l'utente di](#).

## Crea una nuova tabella con timestamp lato client in Amazon Keyspaces

Segui questi esempi per creare una nuova tabella Amazon Keyspaces con timestamp lato client abilitati utilizzando Amazon Console di gestione AWS Keyspaces, Cassandra Query Language (CQL) o AWS Command Line Interface

### Console

Crea una nuova tabella con timestamp lato client (console)

1. [Accedi a e apri Console di gestione AWS la console Amazon Keyspaces a casahttps://console.aws.amazon.com/keyspaces/.](https://console.aws.amazon.com/keyspaces/)
2. Nel pannello di navigazione, scegli Tabelle, quindi seleziona Crea tabella.
3. Nella pagina Crea tabella, nella sezione Dettagli della tabella, seleziona un keyspace e fornisci un nome per la nuova tabella.
4. Nella sezione Schema, crea lo schema per la tua tabella.
5. Nella sezione Impostazioni tabella, scegli Personalizza impostazioni.
6. Continua con i timestamp sul lato client.

Scegli Attiva i timestamp lato client per attivare i timestamp lato client per la tabella.

7. Scegliere Create table (Crea tabella). La tabella viene creata con i timestamp lato client attivati.

## Cassandra Query Language (CQL)

### Crea una nuova tabella usando CQL

1. Per creare una nuova tabella con timestamp lato client abilitati utilizzando CQL, puoi utilizzare il seguente esempio.

```
CREATE TABLE my_keyspace.my_table (
 userid uuid,
 time timeuuid,
 subject text,
 body text,
 user inet,
 PRIMARY KEY (userid, time)
) WITH CUSTOM_PROPERTIES = {'client_side_timestamps': {'status': 'enabled'}};
```

2. Per confermare le impostazioni dei timestamp sul lato client per la nuova tabella, utilizzate un'SELECTistruzione per rivedere quanto mostrato nell'esempio seguente.  
`custom_properties`

```
SELECT custom_properties from system_schema_mcs.tables where keyspace_name =
'my_keyspace' and table_name = 'my_table';
```

L'output di questa istruzione mostra lo stato dei timestamp lato client.

```
'client_side_timestamps': {'status': 'enabled'}
```

## AWS CLI

### Crea una nuova tabella utilizzando il AWS CLI

1. Per creare una nuova tabella con i timestamp sul lato client abilitati, puoi utilizzare il seguente esempio.

```
./aws keyspaces create-table \
--keyspace-name my_keyspace \

```

```
--table-name my_table \
--client-side-timestamps 'status=ENABLED' \
--schema-definition 'allColumns=[{name=id,type=int},{name=date,type=timestamp},
{name=name,type=text}],partitionKeys=[{name=id}]'
```

2. Per confermare che i timestamp lato client sono attivati per la nuova tabella, esegui il codice seguente.

```
./aws keyspaces get-table \
--keyspace-name my_keyspace \
--table-name my_table
```

L'output dovrebbe essere simile a quello di questo esempio.

```
{
 "keyspaceName": "my_keyspace",
 "tableName": "my_table",
 "resourceArn": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/
my_keyspace/table/my_table",
 "creationTimestamp": 1662681206.032,
 "status": "ACTIVE",
 "schemaDefinition": {
 "allColumns": [
 {
 "name": "id",
 "type": "int"
 },
 {
 "name": "date",
 "type": "timestamp"
 },
 {
 "name": "name",
 "type": "text"
 }
],
 "partitionKeys": [
 {
 "name": "id"
 }
],
 "clusteringKeys": [],
 "staticColumns": []
 }
}
```



```
 },
 "capacitySpecification": {
 "throughputMode": "PAY_PER_REQUEST",
 "lastUpdateToPayPerRequestTimestamp": 1662681206.032
 },
 "encryptionSpecification": {
 "type": "AWS_OWNED_KMS_KEY"
 },
 "pointInTimeRecovery": {
 "status": "DISABLED"
 },
 "clientSideTimestamps": {
 "status": "ENABLED"
 },
 "ttl": {
 "status": "ENABLED"
 },
 "defaultTimeToLive": 0,
 "comment": {
 "message": ""
 }
 }
}
```

## Configura i timestamp lato client per una tabella in Amazon Keyspaces

Segui questi esempi per attivare i timestamp lato client per le tabelle esistenti utilizzando Amazon Keyspaces Console di gestione AWS, Cassandra Query Language (CQL) o. AWS Command Line Interface

### Console

Per attivare i timestamp lato client per una tabella esistente (console)

1. [Accedi a e apri Console di gestione AWS la console Amazon Keyspaces a casahttps://console.aws.amazon.com/keyspaces/.](https://console.aws.amazon.com/keyspaces/)
2. Scegli la tabella che desideri aggiornare, quindi scegli la scheda Impostazioni aggiuntive.
3. Nella scheda Impostazioni aggiuntive, vai a Modifica timestamp lato client e seleziona Attiva timestamp lato client
4. Scegli Salva modifiche per modificare le impostazioni della tabella.

## Cassandra Query Language (CQL)

### Utilizzo di un'istruzione CQL

1. Attiva i timestamp lato client per una tabella esistente con l'istruzione CQL. ALTER TABLE

```
ALTER TABLE my_table WITH custom_properties = {'client_side_timestamps':
{'status': 'enabled'}};;
```

2. Per confermare le impostazioni dei timestamp sul lato client per la nuova tabella, utilizzate un'SELECTistruzione per rivedere quanto mostrato nell'esempio seguente.  
custom\_properties

```
SELECT custom_properties from system_schema_mcs.tables where keyspace_name =
'my_keyspace' and table_name = 'my_table';
```

L'output di questa istruzione mostra lo stato dei timestamp lato client.

```
'client_side_timestamps': {'status': 'enabled'}
```

## AWS CLI

### Usando il AWS CLI

1. È possibile attivare i timestamp sul lato client per una tabella esistente AWS CLI utilizzando l'esempio seguente.

```
./aws keyspaces update-table \
--keyspace-name my_keyspace \
--table-name my_table \
--client-side-timestamps 'status=ENABLED'
```

2. Per confermare che i timestamp lato client sono attivati per la tabella, esegui il codice seguente.

```
./aws keyspaces get-table \
--keyspace-name my_keyspace \
--table-name my_table
```

L'output dovrebbe essere simile a questo esempio e indicare lo stato dei timestamp lato client come. ENABLED

```
{
 "keyspaceName": "my_keyspace",
 "tableName": "my_table",
 "resourceArn": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/
my_keyspace/table/my_table",
 "creationTimestamp": 1662681312.906,
 "status": "ACTIVE",
 "schemaDefinition": {
 "allColumns": [
 {
 "name": "id",
 "type": "int"
 },
 {
 "name": "date",
 "type": "timestamp"
 },
 {
 "name": "name",
 "type": "text"
 }
],
 "partitionKeys": [
 {
 "name": "id"
 }
],
 "clusteringKeys": [],
 "staticColumns": []
 },
 "capacitySpecification": {
 "throughputMode": "PAY_PER_REQUEST",
 "lastUpdateToPayPerRequestTimestamp": 1662681312.906
 },
 "encryptionSpecification": {
 "type": "AWS_OWNED_KMS_KEY"
 },
 "pointInTimeRecovery": {
 "status": "DISABLED"
 }
}
```

```

 },
 "clientSideTimestamps": {
 "status": "ENABLED"
 },
 "ttl": {
 "status": "ENABLED"
 },
 "defaultTimeToLive": 0,
 "comment": {
 "message": ""
 }
}

```

## Usa i timestamp lato client nelle query in Amazon Keyspaces

Dopo aver attivato i timestamp lato client, è possibile passare il timestamp nelle istruzioni INSERT, UPDATE e con la clausola DELETE USING TIMESTAMP

Il valore del timestamp bigint rappresenta un numero di microsecondi a partire dall'ora base standard nota come: 1° gennaio 1970 alle 00:00:00 GMT. epoch Un timestamp fornito dal cliente deve rientrare nell'intervallo di 2 giorni passati e 5 minuti futuri rispetto all'ora corrente dell'orologio da parete.

Amazon Keyspaces conserva i metadati con data e ora per tutta la durata dei dati. Puoi utilizzare la WRITETIME funzione per cercare i timestamp che si sono verificati anni fa. Per ulteriori informazioni sulla sintassi CQL, vedere. [the section called “Istruzioni DML”](#)

La seguente istruzione CQL è un esempio di come utilizzare un timestamp come.  
update\_parameter

```

INSERT INTO catalog.book_awards (year, award, rank, category, book_title, author,
publisher)
VALUES (2022, 'Wolf', 4, 'Non-Fiction', 'Science Update', 'Ana Carolina Silva',
'SomePublisher')
USING TIMESTAMP 1669069624;

```

Se non specifichi un timestamp nella tua query CQL, Amazon Keyspaces utilizza il timestamp passato dal driver del client. Se il driver client non fornisce alcun timestamp, Amazon Keyspaces assegna un timestamp lato server per l'operazione di scrittura.

Per visualizzare il valore del timestamp memorizzato per una colonna specifica, puoi utilizzare la `WRITETIME` funzione in un'istruzione, come mostrato nell'esempio seguente. `SELECT`

```
SELECT year, award, rank, category, book_title, author, publisher, WRITETIME(year),
 WRITETIME(award), WRITETIME(rank),
 WRITETIME(category), WRITETIME(book_title), WRITETIME(author), WRITETIME(publisher)
from catalog.book_awards;
```

## Replica multiregionale per Amazon Keyspaces (per Apache Cassandra)

Puoi utilizzare la replica multiregionale di Amazon Keyspaces per replicare i tuoi dati con una replica automatizzata, completamente gestita e attiva-attiva su qualsiasi dispositivo di tua scelta. Regioni AWS Con la replica attiva-attiva, ogni regione è in grado di eseguire letture e scritture in modo isolato. È possibile migliorare sia la disponibilità che la resilienza in caso di degrado regionale, sfruttando al contempo le operazioni di lettura e scrittura locali a bassa latenza per applicazioni globali.

Con la replica multiregionale, Amazon Keyspaces replica in modo asincrono i dati tra le regioni e i dati vengono in genere propagati tra le regioni entro un secondo. Inoltre, con la replica in più regioni, non avrai più il difficile compito di risolvere i conflitti e correggere i problemi di divergenza dei dati, così puoi concentrarti sulla tua applicazione.

Per impostazione predefinita, Amazon Keyspaces replica i dati su tre [zone di disponibilità](#) all'interno della stessa Regione AWS per garantire durabilità e disponibilità elevata. Con la replica multiregionale, puoi creare keyspaces multiregionali che replicano le tabelle in aree geografiche diverse a tua scelta. Regioni AWS

### Argomenti

- [Vantaggi dell'utilizzo della replica in più regioni](#)
- [Capacità, modalità e prezzi](#)
- [Come funziona la replica multiregionale in Amazon Keyspaces](#)
- [Note sull'utilizzo della replica multiregionale di Amazon Keyspaces](#)
- [Configurazione della replica multiregionale per Amazon Keyspaces \(per Apache Cassandra\)](#)

## Vantaggi dell'utilizzo della replica in più regioni

La replica in più regioni offre i seguenti vantaggi.

- **Lecture e scritture globali con latenza di un millisecondo:** in Amazon Keyspaces, la replica è attiva-attiva. Puoi eseguire operazioni di lettura e scrittura localmente dalle regioni più vicine ai tuoi clienti con una latenza di un millisecondo su qualsiasi scala. Puoi utilizzare le tabelle multiregionali di Amazon Keyspaces per applicazioni globali che richiedono tempi di risposta rapidi in qualsiasi parte del mondo.
- **Continuità aziendale e protezione migliorate dal degrado in un'unica regione:** con la replica multiregionale, puoi ripristinare il sistema dal degrado in un'unica area reindirizzando l'applicazione in un' Regione AWS altra regione del tuo keyspace multiregionale. Poiché Amazon Keyspaces offre una replica attiva-attiva, non vi è alcun impatto sulle operazioni di lettura e scrittura.

Amazon Keyspaces tiene traccia di tutte le scritture che sono state eseguite sul tuo keyspace multiregionale ma non sono state propagate a tutte le regioni di replica. Una volta che la regione è tornata online, Amazon Keyspaces sincronizza automaticamente tutte le modifiche mancanti in modo da consentirti il ripristino senza alcun impatto sulle applicazioni.

- **Replica ad alta velocità tra regioni:** la replica multiregionale utilizza la replica fisica rapida e basata sullo storage dei dati tra le regioni, con un ritardo di replica che in genere è inferiore a 1 secondo.

La replica in Amazon Keyspaces ha un impatto minimo o nullo sulle query del database perché non condivide le risorse di elaborazione con l'applicazione. Ciò significa che puoi affrontare casi d'uso con throughput di scrittura elevato o casi d'uso con picchi o picchi improvvisi di throughput senza alcun impatto sull'applicazione.

- **Coerenza e risoluzione dei conflitti:** tutte le modifiche apportate ai dati in qualsiasi regione vengono replicate nelle altre regioni in uno spazio chiave multiregionale. Se le applicazioni aggiornano gli stessi dati in regioni diverse contemporaneamente, possono verificarsi conflitti.

Per garantire la coerenza finale, Amazon Keyspaces utilizza timestamp a livello di cella e un last writer consente la riconciliazione tra gli aggiornamenti simultanei. La risoluzione dei conflitti è completamente gestita e avviene in background senza alcun impatto sulle applicazioni.

Per ulteriori informazioni sulle configurazioni e sulle funzionalità supportate, vedere [the section called “Note per l'utilizzo”](#).

## Capacità, modalità e prezzi

Per uno spazio di chiavi multiregionale, è possibile utilizzare la modalità di capacità su richiesta o la modalità di capacità fornita. Per ulteriori informazioni, consulta [the section called “Configura le modalità di capacità di lettura/scrittura”](#).

Per la modalità on demand, ti viene fatturata 1 unità di richiesta di scrittura (WRU) per scrivere fino a 1 KB di dati per riga, allo stesso modo delle tabelle a regione singola. Ma ti vengono fatturate le scritture in ogni regione del tuo spazio chiave multiregionale. Ad esempio, la scrittura di una riga di 3 KB di dati in uno spazio chiave multiregionale con due regioni richiede 6 WRUs:  $3 * 2 = 6$ . WRUs Inoltre, le scritture che includono dati statici e non statici richiedono operazioni di scrittura aggiuntive.

Per la modalità provisioning, ti viene fatturata 1 unità di capacità di scrittura (WCU) per scrivere fino a 1 KB di dati per riga, allo stesso modo delle tabelle a regione singola. Ma ti vengono fatturate le scritture in ogni regione del tuo spazio chiave multiregionale. Ad esempio, la scrittura di una riga di 3 KB di dati al secondo in uno spazio chiave multiregionale con due regioni richiede 6 WCUs:  $3 * 2 = 6$ . WCUs Inoltre, le scritture che includono dati statici e non statici richiedono operazioni di scrittura aggiuntive.

Per ulteriori informazioni sui prezzi, consulta i prezzi di [Amazon Keyspaces \(per Apache Cassandra\)](#).

## Come funziona la replica multiregionale in Amazon Keyspaces

Questa sezione fornisce una panoramica di come funziona la replica multiregionale di Amazon Keyspaces. Per ulteriori informazioni sui prezzi, consulta i prezzi di [Amazon Keyspaces \(per Apache Cassandra\)](#).

### Argomenti

- [Come funziona la replica multiregionale in Amazon Keyspaces](#)
- [Risoluzione dei conflitti di replica in più regioni](#)
- [Replica in più regioni e disaster recovery](#)
- [La replica multiregionale è disattivata per impostazione predefinita Regioni AWS](#)
- [Replica e integrazione in più regioni con ripristino \(PITR\) point-in-time](#)
- [Replica in più regioni e integrazione con i servizi AWS](#)

## Come funziona la replica multiregionale in Amazon Keyspaces

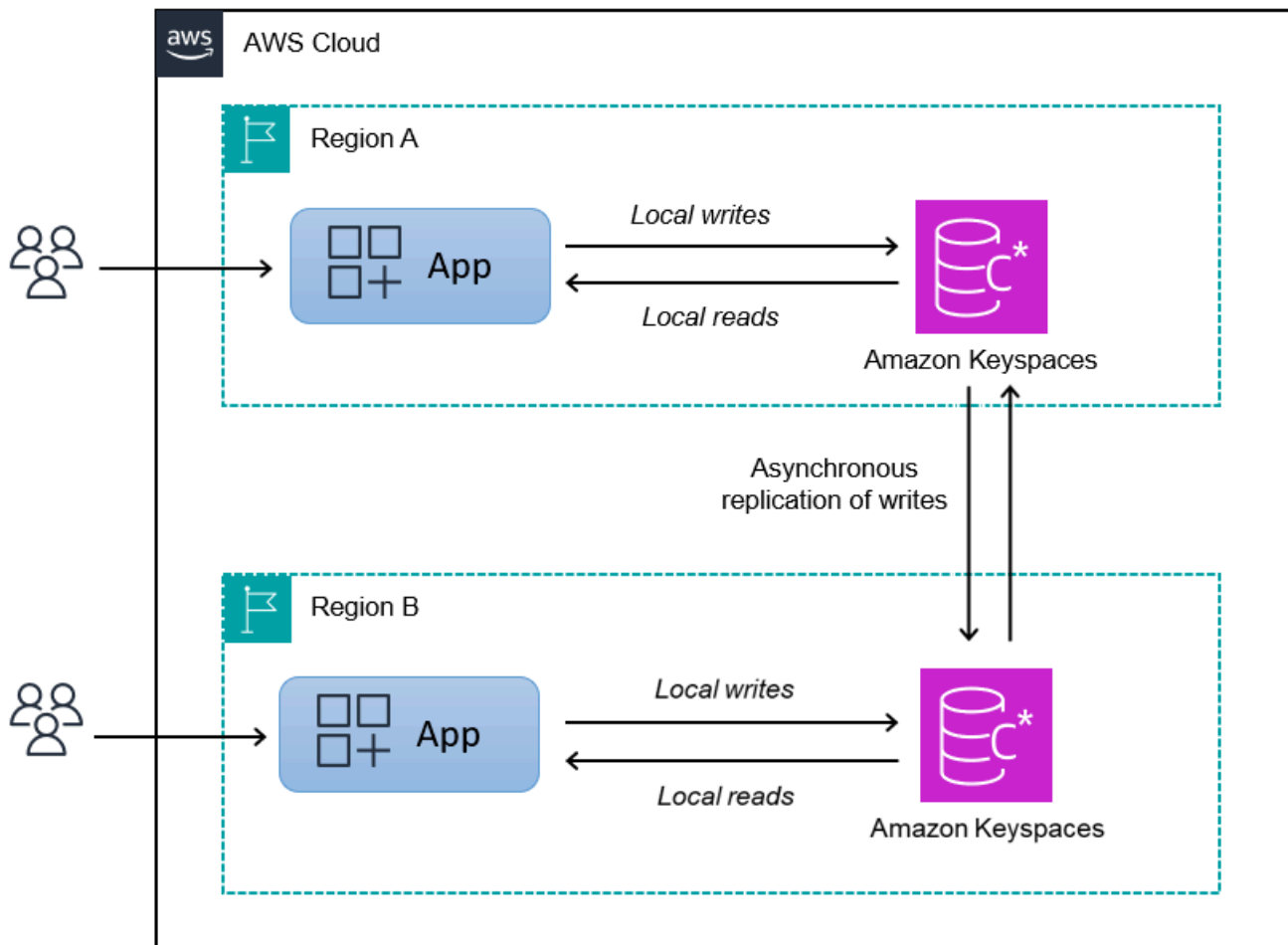
La replica multiregionale di Amazon Keyspaces implementa un'architettura di resilienza dei dati che distribuisce i dati in modo indipendente e distribuito geograficamente. Regioni AWS Utilizza la replica attiva-attiva, che fornisce una bassa latenza locale e ciascuna regione è in grado di eseguire letture e scritture in modo isolato.

Quando crei uno spazio chiave multiregionale Amazon Keyspaces, puoi selezionare regioni aggiuntive in cui verranno replicati i dati. Ogni tabella creata in uno spazio di chiavi multiregione è composta da più tabelle di replica (una per regione) che Amazon Keyspaces considera come una singola unità.

Ogni replica ha lo stesso nome di tabella e lo stesso schema di chiave primaria. Quando un'applicazione scrive dati su una tabella locale in una regione, i dati vengono scritti in modo duraturo utilizzando il livello di coerenza. LOCAL\_QUORUM Amazon Keyspaces replica automaticamente i dati in modo asincrono nelle altre regioni di replica. Il ritardo di replica tra le regioni è in genere inferiore a un secondo e non influisce sulle prestazioni o sulla velocità effettiva dell'applicazione.

Dopo aver scritto i dati, è possibile leggerli dalla tabella multiregionale in un'altra regione di replica con i livelli di coerenza. LOCAL\_ONE/LOCAL\_QUORUM Per ulteriori informazioni sulle configurazioni e sulle funzionalità supportate, consulta. [the section called “Note per l'utilizzo”](#)





## Risoluzione dei conflitti di replica in più regioni

La replica multiregionale di Amazon Keyspaces è completamente gestita, il che significa che non è necessario eseguire attività di replica come eseguire regolarmente operazioni di riparazione per risolvere i problemi di sincronizzazione dei dati. Amazon Keyspaces monitora la coerenza dei dati tra tabelle diverse Regioni AWS rilevando e riparando i conflitti e sincronizza automaticamente le repliche.

Amazon Keyspaces utilizza il metodo last writer wins per la riconciliazione dei dati. Con questo meccanismo di risoluzione dei conflitti, tutte le regioni di uno spazio chiave multiregionale concordano sull'ultimo aggiornamento e convergono verso uno stato in cui dispongono tutte di dati identici. Il processo di riconciliazione non ha alcun impatto sulle prestazioni delle applicazioni. Per supportare la risoluzione dei conflitti, i timestamp lato client vengono attivati automaticamente per le tabelle multiregionali e non possono essere disattivati. Per ulteriori informazioni, consulta [the section called “Timestamp lato client”](#).

## Replica in più regioni e disaster recovery

Con la replica multiregionale di Amazon Keyspaces, le scritture vengono replicate in modo asincrono in ogni regione. Nel raro caso di deterioramento o guasto di una singola regione, la replica multiregionale consente di ripristinare il sistema in caso di emergenza con un impatto minimo o nullo sull'applicazione. Il ripristino in caso di emergenza viene in genere misurato utilizzando i valori di Recovery time objective (RTO) e Recovery point objective (RPO).

Obiettivo relativo al tempo di ripristino: il tempo impiegato da un sistema per tornare allo stato operativo dopo un disastro. L'RTO misura la quantità di downtime tollerabile dal carico di lavoro, misurata in termini di tempo. Per i piani di disaster recovery che utilizzano la replica multiregionale per eseguire il failover in una regione non interessata, l'RTO può essere quasi pari a zero. L'RTO è limitato dalla rapidità con cui l'applicazione è in grado di rilevare la condizione di errore e reindirizzare il traffico verso un'altra regione.

Obiettivo del punto di ripristino: la quantità di dati che è possibile perdere (misurata in termini di tempo). Per i piani di disaster recovery che utilizzano la replica multiregionale per eseguire il failover in una regione non interessata, l'RPO è in genere di secondi a una cifra. L'RPO è limitato dalla latenza di replica alla replica di destinazione del failover.

In caso di guasto o deterioramento regionale, non è necessario promuovere una regione secondaria o eseguire procedure di failover del database perché la replica in Amazon Keyspaces è attiva-attiva. Puoi invece utilizzare Amazon Route 53 per indirizzare la tua applicazione verso la regione integra più vicina. Per ulteriori informazioni su Route 53, consulta [Cos'è Amazon Route 53?](#).

Se una singola unità Regione AWS viene isolata o danneggiata, l'applicazione può reindirizzare il traffico verso un'altra regione utilizzando Route 53 per eseguire letture e scritture su una tabella di replica diversa. È inoltre possibile applicare una logica aziendale personalizzata per determinare quando reindirizzare le richieste verso altre regioni. Un esempio di ciò è rendere l'applicazione consapevole dei molteplici endpoint disponibili.

Quando la regione torna online, Amazon Keyspaces riprende a propagare le scritture in sospeso da quella regione alle tabelle di replica in altre regioni. Riprende inoltre la propagazione delle scritture da altre tabelle di replica alla regione che è ora di nuovo online.

## La replica multiregionale è disattivata per impostazione predefinita Regioni AWS

La replica multiregionale di Amazon Keyspaces è supportata nei seguenti casi, Regioni AWS che sono disabilitati per impostazione predefinita:

- Regione Africa (Città del Capo)
- Regione Medio Oriente (EAU)
- Regione Asia Pacifico (Hong Kong)
- Regione Medio Oriente (Bahrein)

Prima di poter utilizzare una regione disabilitata per impostazione predefinita con la replica multiregionale di Amazon Keyspaces, devi prima abilitare la regione. [Per ulteriori informazioni, consulta \*Abilita o disabilita Regioni AWS nel tuo account nella Guida per l'AWS Organizations\* utente.](#)

Dopo aver abilitato una regione, puoi creare nuove risorse Amazon Keyspaces nella regione e aggiungere la regione a uno spazio di chiavi multiregionale.

Quando disabiliti una regione utilizzata dalla replica multiregionale di Amazon Keyspaces, Amazon Keyspaces avvia un periodo di prova di 24 ore. Durante questa finestra temporale, puoi aspettarti il seguente comportamento:

- Amazon Keyspaces continua a eseguire operazioni DML (Data Manipulation Language) nelle regioni abilitate.
- Amazon Keyspaces sospende la replica degli aggiornamenti dei dati dalle regioni abilitate alla regione disabilitata.
- Amazon Keyspaces blocca tutte le richieste DDL (Data Definition Language) nella regione disattivata.

Se hai disabilitato la regione per errore, puoi riattivarla entro 24 ore. Se riattivi la regione durante il periodo di prova di 24 ore, Amazon Keyspaces intraprenderà le seguenti azioni:

- Riprendi automaticamente tutte le repliche nella regione riattivata.
- Replica tutti gli aggiornamenti dei dati che hanno avuto luogo nelle regioni abilitate mentre la regione era disattivata per garantire la coerenza dei dati.
- Continua automaticamente tutte le operazioni di replica in più aree geografiche.

Nel caso in cui la regione rimanga disattivata dopo la chiusura della finestra di 24 ore, Amazon Keyspaces intraprende le seguenti azioni per rimuovere definitivamente la regione dalla replica multiregionale:

- Rimuovi la regione disabilitata da tutti gli spazi chiave di replica multiregione.

- Converti le repliche delle tabelle di replica multiregione nella regione disattivata in spazi chiave e tabelle a regione singola.
- Amazon Keyspaces non elimina alcuna risorsa dalla regione disattivata.

Dopo che Amazon Keyspaces ha rimosso definitivamente la regione disabilitata dallo spazio di chiavi multiregione, non puoi aggiungere nuovamente la regione disabilitata.

## Replica e integrazione in più regioni con ripristino (PITR) point-in-time

Point-in-time il ripristino è supportato per le tabelle multiregionali. Per ripristinare correttamente una tabella multiregionale con PITR, devono essere soddisfatte le seguenti condizioni.

- La tabella di origine e quella di destinazione devono essere configurate come tabelle multiregionali.
- Le regioni di replica per lo spazio chiave della tabella di origine e per lo spazio chiave della tabella di destinazione devono essere le stesse.
- PITR deve essere abilitato su tutte le repliche della tabella di origine.

È possibile eseguire l'istruzione di ripristino da qualsiasi regione in cui è disponibile la tabella di origine. Amazon Keyspaces ripristina automaticamente la tabella di destinazione in ogni regione. Per ulteriori informazioni su PITR, consulta [the section called “Come funziona”](#).

Quando crei una tabella multiregionale, le impostazioni PITR che definisci durante il processo di creazione vengono applicate automaticamente a tutte le tabelle in tutte le regioni. Quando modifichi le impostazioni PITR utilizzando `ALTER TABLE`, Amazon Keyspaces applica l'aggiornamento solo alla tabella locale e non alle repliche in altre regioni. Per abilitare PITR per una tabella multiregionale esistente, devi ripetere l'istruzione per tutte le `ALTER TABLE` repliche.

## Replica in più regioni e integrazione con i servizi AWS

È possibile monitorare le prestazioni di replica tra tabelle diverse Regioni AWS utilizzando i CloudWatch parametri di Amazon. La seguente metrica fornisce il monitoraggio continuo degli spazi chiave multiregionali.

- **ReplicationLatency**— Questa metrica misura il tempo impiegato per la replica o `deletes` dal passaggio da una tabella di `updates` replica a un'altra tabella di replica in uno spazio chiave multiregionale. `inserts`

Per ulteriori informazioni su come monitorare le metriche, vedere [CloudWatch the section called “Monitoraggio con CloudWatch”](#)

## Note sull'utilizzo della replica multiregionale di Amazon Keyspaces

Considera quanto segue quando utilizzi la replica multiregionale con Amazon Keyspaces.

- [Puoi selezionare qualsiasi tipo di pubblico disponibile.](#) Regioni AWS Per ulteriori informazioni su Regioni AWS [ciò che è disabilitato per impostazione predefinita](#), consulta [the section called “Regioni disattivate per impostazione predefinita”](#).
- AWS GovCloud (US) Regions e le regioni della Cina non sono supportate.
- Prendi in considerazione le seguenti soluzioni alternative fino a quando le funzionalità non saranno disponibili:

Configura Time to Live (TTL) durante la creazione della tabella multiregionale. Non potrai abilitare e disabilitare il TTL o modificare il valore TTL in un secondo momento. Per ulteriori informazioni, consulta [the section called “Fai scadere i dati con Time to Live”](#).

- Per la crittografia a riposo, usa una chiave AWS proprietaria. Le chiavi gestite dal cliente non sono attualmente supportate per le tabelle multiregionali. Per ulteriori informazioni, consulta la pagina

[the section called “Come funziona”](#).

- È possibile utilizzare ALTER KEYSPACE per aggiungere una regione a uno spazio di chiavi a regione singola o multiarea. Per ulteriori informazioni, consulta [the section called “Aggiungi una regione a un keyspace”](#).
  - Prima di aggiungere una regione a uno spazio di chiavi a regione singola, assicuratevi che nessuna tabella sotto lo spazio chiave sia configurata con chiavi gestite dal cliente.
  - Tutti i tag esistenti configurati per gli spazi chiave o le tabelle non vengono replicati nella nuova regione.
- Quando utilizzi la gestione della capacità assegnata con Amazon Keyspaces auto Scaling, assicurati di utilizzare le operazioni dell'API Amazon Keyspaces per creare e configurare le tabelle multiregionali. Le operazioni API Application Auto Scaling sottostanti che Amazon Keyspaces chiama per tuo conto non hanno funzionalità multiregionali.

Per ulteriori informazioni, consulta [the section called “Aggiornamento della capacità assegnata e delle impostazioni di ridimensionamento automatico per una tabella multiregionale”](#). Per ulteriori informazioni su come stimare la velocità effettiva della capacità di scrittura delle tabelle

multiregionali predisposte, consulta. [the section called “Stima della capacità per una tabella multiregionale”](#)

- Sebbene i dati vengano replicati automaticamente nelle regioni selezionate di una tabella multiregionale, quando un client si connette a un endpoint in una regione e interroga la `system.peers` tabella, la query restituisce solo informazioni locali. Il risultato della query appare al client come un singolo cluster di data center.
- La replica multiregionale di Amazon Keyspaces è asincrona e supporta la coerenza per le scritture. `LOCAL_QUORUM LOCAL_QUORUM` la coerenza richiede che un aggiornamento di una riga venga mantenuto in modo duraturo su due repliche nella regione locale prima di restituire l'esito positivo al client. La propagazione delle scritture nella regione (o nelle regioni) replicate viene quindi eseguita in modo asincrono.

La replica multiregionale di Amazon Keyspaces non supporta la replica o la coerenza sincrona. `QUORUM`

- Quando crei uno spazio chiave o una tabella multiregionale, tutti i tag definiti durante il processo di creazione vengono applicati automaticamente a tutti gli spazi chiave e le tabelle in tutte le regioni. Quando modificate i tag esistenti utilizzando `ALTER KEYSPACE` o `ALTER TABLE`, l'aggiornamento viene applicato solo allo spazio chiave o alla tabella nella regione in cui state apportando la modifica.
- Amazon CloudWatch fornisce una `ReplicationLatency` metrica per ogni regione replicata. Calcola questa metrica tracciando le righe in arrivo, confrontando l'ora di arrivo con il tempo di scrittura iniziale e calcolando una media. Le tempistiche vengono memorizzate CloudWatch nella regione di origine. Per ulteriori informazioni, consulta [the section called “Monitoraggio con CloudWatch”](#).

Può essere utile visualizzare i tempi medi e massimi per determinare il ritardo di replica medio e peggiore dei casi. Non esiste alcun Accordo sul livello di servizio (SLA) per questa latenza.

- Quando si utilizza una tabella multiregionale in modalità on demand, è possibile osservare un aumento della latenza per la replica asincrona delle scritture se una replica di tabella registra un nuovo picco di traffico. Analogamente a come Amazon Keyspaces adatta automaticamente la capacità di una tabella on-demand a regione singola al traffico dell'applicazione che riceve, Amazon Keyspaces adatta automaticamente la capacità di una replica di tabella on-demand multiregionale al traffico che riceve. L'aumento della latenza di replica è transitorio perché Amazon Keyspaces alloca automaticamente più capacità all'aumentare del volume di traffico. Una volta che tutte le repliche si saranno adattate al volume di traffico, la latenza di replica dovrebbe tornare alla

normalità. Per ulteriori informazioni, consulta [the section called “Proprietà di dimensionamento e picco di traffico”](#).

- Quando si utilizza una tabella multiregionale in modalità provisioning, se l'applicazione supera la capacità di throughput assegnata, è possibile che si verifichino errori di capacità insufficienti e un aumento della latenza di replica. Per garantire che ci sia sempre una capacità di lettura e scrittura sufficiente per tutte le repliche Regioni AWS di tabella in una tabella multiregionale, ti consigliamo di configurare la scalabilità automatica di Amazon Keyspaces. La scalabilità automatica di Amazon Keyspaces ti aiuta a fornire la capacità di throughput in modo efficiente per carichi di lavoro variabili regolando automaticamente la capacità di throughput in risposta al traffico effettivo delle applicazioni. Per ulteriori informazioni, consulta [the section called “Come funziona la scalabilità automatica per le tabelle multiregionali”](#).

## Configurazione della replica multiregionale per Amazon Keyspaces (per Apache Cassandra)

Puoi utilizzare la console, Cassandra Query Language (CQL) o AWS Command Line Interface per creare e gestire spazi chiave e tabelle multiregionali in Amazon Keyspaces.

Questa sezione fornisce esempi di come creare e gestire spazi chiave e tabelle multiregionali. Tutte le tabelle create in uno spazio chiave multiarea ereditano automaticamente le impostazioni multiregione dallo spazio chiave.

Per ulteriori informazioni sulle configurazioni e sulle funzionalità supportate, consulta [the section called “Note per l'utilizzo”](#)

### Argomenti

- [Configura le autorizzazioni IAM necessarie per creare spazi chiave e tabelle multiregionali](#)
- [Configura le autorizzazioni IAM necessarie per aggiungere un a un keyspace Regione AWS](#)
- [Crea uno spazio chiave multiregionale in Amazon Keyspaces](#)
- [Aggiungere un Regione AWS a un keyspace in Amazon Keyspaces](#)
- [Controllate l'avanzamento della replica quando aggiungete una nuova regione a un keyspace](#)
- [Crea una tabella multiregionale con impostazioni predefinite in Amazon Keyspaces](#)
- [Crea una tabella multiregionale in modalità provisioning con scalabilità automatica in Amazon Keyspaces](#)

- [Aggiorna la capacità fornita e le impostazioni di ridimensionamento automatico per una tabella multiregionale in Amazon Keyspaces](#)
- [Visualizza la capacità fornita e le impostazioni di ridimensionamento automatico per una tabella multiregionale in Amazon Keyspaces](#)
- [Disattiva la scalabilità automatica per una tabella in Amazon Keyspaces](#)
- [Imposta manualmente la capacità fornita di una tabella multiregionale in Amazon Keyspaces](#)

## Configura le autorizzazioni IAM necessarie per creare spazi chiave e tabelle multiregionali

Per creare correttamente spazi chiave e tabelle multiregionali, il responsabile IAM deve essere in grado di creare un ruolo collegato ai servizi. Questo ruolo collegato ai servizi è un tipo unico di ruolo IAM predefinito da Amazon Keyspaces. Include tutte le autorizzazioni richieste da Amazon Keyspaces per eseguire azioni per tuo conto. Per ulteriori informazioni sul ruolo collegato al servizio, consulta [the section called “Replica in più regioni”](#).

Per creare il ruolo collegato ai servizi richiesto dalla replica multiregionale, la policy per il principale IAM richiede i seguenti elementi:

- `iam:CreateServiceLinkedRole`— L'azione che il principale può eseguire.
- `arn:aws:iam::*:role/aws-service-role/replication.cassandra.amazonaws.com/AWSServiceRoleForKeyspacesReplication`— La risorsa su cui è possibile eseguire l'azione.
- `iam:AWSServiceName": "replication.cassandra.amazonaws.com`— L'unico AWS servizio a cui è possibile collegare questo ruolo è Amazon Keyspaces.

Di seguito è riportato un esempio della politica che concede le autorizzazioni minime richieste a un principale per creare spazi chiave e tabelle multiregionali.

```
{
 "Effect": "Allow",
 "Action": "iam:CreateServiceLinkedRole",
 "Resource": "arn:aws:iam::*:role/aws-service-role/
replication.cassandra.amazonaws.com/AWSServiceRoleForKeyspacesReplication",
 "Condition": {"StringLike": {"iam:AWSServiceName":
"replication.cassandra.amazonaws.com"}}
}
```



Per ulteriori autorizzazioni IAM per keyspace e tabelle multiregionali, consulta le [Azioni, risorse e chiavi di condizione per Amazon Keyspaces \(per Apache Cassandra\)](#) nel Service Authorization Reference.

## Configura le autorizzazioni IAM necessarie per aggiungere un a un keyspace Regione AWS

Per aggiungere una regione a un keyspace, il principale IAM necessita delle seguenti autorizzazioni:

- `cassandra:Alter`
- `cassandra:AlterMultiRegionResource`
- `cassandra:Create`
- `cassandra:CreateMultiRegionResource`
- `cassandra:Select`
- `cassandra:SelectMultiRegionResource`
- `cassandra:Modify`
- `cassandra:ModifyMultiRegionResource`

Se la tabella è configurata in modalità provisioning con la scalabilità automatica abilitata, sono necessarie le seguenti autorizzazioni aggiuntive.

- `application-autoscaling:RegisterScalableTarget`
- `application-autoscaling:DeregisterScalableTarget`
- `application-autoscaling:DescribeScalableTargets`
- `application-autoscaling:PutScalingPolicy`
- `application-autoscaling:DescribeScalingPolicies`

Per aggiungere correttamente una regione a uno spazio chiave a regione singola, il principale IAM deve anche essere in grado di creare un ruolo collegato al servizio. Questo ruolo collegato ai servizi è un tipo unico di ruolo IAM predefinito da Amazon Keyspaces. Include tutte le autorizzazioni richieste da Amazon Keyspaces per eseguire azioni per tuo conto. Per ulteriori informazioni sul ruolo collegato al servizio, consulta [the section called “Replica in più regioni”](#).

Per creare il ruolo collegato ai servizi richiesto dalla replica multiregionale, la policy per il principale IAM richiede i seguenti elementi:

- `iam:CreateServiceLinkedRole`— L'azione che il principale può eseguire.
- `arn:aws:iam::*:role/aws-service-role/replication.cassandra.amazonaws.com/AWSServiceRoleForKeyspacesReplication`— La risorsa su cui è possibile eseguire l'azione.
- `iam:AWSServiceName": "replication.cassandra.amazonaws.com`— L'unico AWS servizio a cui è possibile collegare questo ruolo è Amazon Keyspaces.

Di seguito è riportato un esempio della politica che concede le autorizzazioni minime richieste a un principale per aggiungere una regione a un keyspace.

```
{
 "Effect": "Allow",
 "Action": "iam:CreateServiceLinkedRole",
 "Resource": "arn:aws:iam::*:role/aws-service-role/
replication.cassandra.amazonaws.com/AWSServiceRoleForKeyspacesReplication",
 "Condition": {"StringLike": {"iam:AWSServiceName":
"replication.cassandra.amazonaws.com"}}
}
```

## Crea uno spazio chiave multiregionale in Amazon Keyspaces

Questa sezione fornisce esempi su come creare uno spazio chiave multiregionale. Puoi farlo sulla console Amazon Keyspaces, utilizzando CQL o il AWS CLI. Tutte le tabelle che crei in uno spazio chiave multiregionale ereditano automaticamente le impostazioni multiregionali dallo spazio chiave.

### Note

Quando crei uno spazio di chiavi multiregionale, Amazon Keyspaces crea un ruolo collegato al servizio con il nome nel tuo account.

`AWSServiceRoleForAmazonKeyspacesReplication` Questo ruolo consente ad Amazon Keyspaces di replicare le scritture su tutte le repliche di una tabella multiregionale per tuo conto. Per ulteriori informazioni, consulta [the section called “Replica in più regioni”](#).

## Console

### Crea uno spazio di chiavi multiregionale (console)

1. [Accedi a e apri Console di gestione AWS la console Amazon Keyspaces a casahttps://console.aws.amazon.com/keyspaces/.](https://console.aws.amazon.com/keyspaces/)

2. Nel riquadro di navigazione, scegli Keyspaces, quindi scegli Crea keyspace.
3. Per il nome dello spazio chiave, inserisci il nome per lo spazio chiave.
4. Nella sezione Replica multiregione, è possibile aggiungere le regioni aggiuntive disponibili nell'elenco.
5. Per finire, scegli Crea spazio chiave.

## Cassandra Query Language (CQL)

### Crea uno spazio chiave multiregionale usando CQL

1. Per creare uno spazio chiave multiregione, utilizzare `NetworkTopologyStrategy` per specificare in Regioni AWS che modo verrà replicato lo spazio chiave. È necessario includere la regione corrente e almeno una regione aggiuntiva.

Tutte le tabelle nello spazio delle chiavi ereditano la strategia di replica dallo spazio delle chiavi. Non è possibile modificare la strategia di replica a livello di tabella.

`NetworkTopologyStrategy`— Il fattore di replica per ogni regione è tre perché Amazon Keyspaces replica i dati su [tre zone di disponibilità](#) all'interno della Regione AWS stessa, per impostazione predefinita.

La seguente istruzione CQL ne è un esempio.

```
CREATE KEYSPACE mykeyspace
WITH REPLICATION = {'class':'NetworkTopologyStrategy', 'us-east-1':'3', 'ap-southeast-1':'3', 'eu-west-1':'3' };
```

2. È possibile utilizzare un'istruzione CQL per interrogare la `tables` tabella nello `system_multiregion_info` spazio delle chiavi per elencare a livello di codice le regioni e lo stato della tabella multiregione specificata. Il codice seguente ne è un esempio.

```
SELECT * from system_multiregion_info.tables WHERE keyspace_name = 'mykeyspace'
AND table_name = 'mytable';
```

L'output dell'istruzione è simile al seguente:

| keyspace_name | table_name | region    | status |
|---------------|------------|-----------|--------|
| mykeyspace    | mytable    | us-east-1 | ACTIVE |

|            |         |                |        |
|------------|---------|----------------|--------|
| mykeyspace | mytable | ap-southeast-1 | ACTIVE |
| mykeyspace | mytable | eu-west-1      | ACTIVE |

## CLI

Crea un nuovo spazio chiave multiregionale utilizzando il AWS CLI

- Per creare uno spazio di chiavi multiregione, puoi utilizzare la seguente istruzione CLI. Specificate la regione corrente e almeno una regione aggiuntiva in `regionList`

```
aws keyspaces create-keyspace --keyspace-name mykeyspace \
--replication-specification replicationStrategy=MULTI_REGION,regionList=us-
east-1,eu-west-1
```

Per creare una tabella multiregionale, consulta [the section called “Crea una tabella multiregionale con impostazioni predefinite”](#) e [the section called “Crea una tabella multiregionale in modalità provisioning”](#).

## Aggiungere un Regione AWS a un keyspace in Amazon Keyspaces

È possibile aggiungerne uno nuovo Regione AWS a uno spazio chiave che sia uno spazio chiave singolo o multiarea. La nuova regione di replica viene applicata a tutte le tabelle nel keyspace.

Per modificare una regione singola in uno spazio chiave multiarea, è necessario abilitare i timestamp sul lato client per tutte le tabelle nello spazio chiave. Per ulteriori informazioni, consulta [the section called “Timestamp lato client”](#).

Se stai aggiungendo una regione aggiuntiva a uno spazio chiave multiregionale, Amazon Keyspaces deve replicare le tabelle esistenti nella nuova regione utilizzando un ripristino interregionale una tantum per ogni tabella esistente. I costi di ripristino per ogni tabella vengono fatturati per GB, per ulteriori informazioni consulta la pagina dei prezzi di [Backup e ripristino](#) su Amazon Keyspaces (per Apache Cassandra). Non è previsto alcun costo per il trasferimento dei dati tra le regioni per questa operazione di ripristino. Oltre ai dati, tutte le proprietà della tabella ad eccezione dei tag verranno replicate nella nuova regione.

Puoi utilizzare l'`ALTER KEYSPACE` istruzione in CQL, il `update-keyspace` comando con o la AWS CLI console per aggiungere una nuova regione a uno spazio di chiavi singolo o multiregionale in Amazon Keyspaces. Per eseguire correttamente l'istruzione, l'account che stai utilizzando deve trovarsi in una delle regioni in cui lo spazio delle chiavi è già disponibile. Durante l'aggiunta della

replica, non è possibile eseguire altre operazioni DDL (Data Definition Language) sulle risorse che vengono aggiornate e replicate.

Per ulteriori informazioni sulle autorizzazioni necessarie per aggiungere una regione, vedere. [the section called “Configura le autorizzazioni IAM per aggiungere una regione”](#)

#### Note

Quando aggiungi una regione aggiuntiva a uno spazio chiave per regione singola, Amazon Keyspaces crea un ruolo collegato al servizio con il nome nel tuo account. `AWSServiceRoleForAmazonKeyspacesReplication` Questo ruolo consente ad Amazon Keyspaces di replicare le tabelle in nuove regioni e di replicare le scritture da una tabella su tutte le repliche di una tabella multiregionale per tuo conto. Per ulteriori informazioni, consulta [the section called “Replica in più regioni”](#).

## Console

Segui questi passaggi per aggiungere una regione a un keyspace utilizzando la console Amazon Keyspaces.

Aggiungi una regione a un keyspace (console)

1. [Accedi a e apri Console di gestione AWS la console Amazon Keyspaces a casahttps://console.aws.amazon.com/keyspaces/](https://console.aws.amazon.com/keyspaces/).
2. Nel riquadro di navigazione, scegli Keyspaces, quindi scegli uno spazio chiave dall'elenco.
3. Scegliere la scheda Regioni AWS.
4. Nella Regioni AWSscheda, scegli Aggiungi regione.
5. Nella finestra di dialogo Aggiungi regione, scegli la regione aggiuntiva che desideri aggiungere allo spazio delle chiavi.
6. Per finire, scegliete Aggiungi.

## Cassandra Query Language (CQL)

Aggiungi una regione a uno spazio chiave usando CQL

- Per aggiungere una nuova regione a uno spazio chiave, è possibile utilizzare la seguente istruzione. In questo esempio, lo spazio delle chiavi è già disponibile nelle regioni Stati Uniti

orientali (Virginia settentrionale) e Stati Uniti occidentali (Oregon) e l'istruzione CQL sta aggiungendo la regione Stati Uniti occidentali (California settentrionale).

```
ALTER KEYSPACE my_keyspace
WITH REPLICATION = {
 'class': 'NetworkTopologyStrategy',
 'us-east-1': '3',
 'us-west-2': '3',
 'us-west-1': '3'
} AND CLIENT_SIDE_TIMESTAMPS = {'status': 'ENABLED'};
```

## CLI

Aggiungere una regione a uno spazio chiave utilizzando il AWS CLI

- Per aggiungere una nuova regione a un keyspace utilizzando la CLI, puoi usare il seguente esempio. Si noti che il valore predefinito per `client-side-timestamps` è `DISABLED`. Con il `update-keyspace` comando, è necessario modificare il valore in `ENABLED`

```
aws keyspaces update-keyspace \
--keyspace-name my_keyspace \
--replication-specification '{"replicationStrategy": "MULTI_REGION",
"regionList": ["us-east-1", "eu-west-1", "eu-west-3"] }' \
--client-side-timestamps '{"status": "ENABLED"}'
```

## Controllate l'avanzamento della replica quando aggiungete una nuova regione a un keyspace

L'aggiunta di una nuova regione a uno spazio di chiavi Amazon Keyspaces è un'operazione che richiede molto tempo. Per tenere traccia dei progressi puoi utilizzare le query mostrate in questa sezione.

### Cassandra Query Language (CQL)

Utilizzo di CQL per verificare l'avanzamento dell'aggiunta della regione

- Per verificare lo stato di avanzamento della creazione delle nuove repliche di tabella in un determinato spazio di chiavi, è possibile interrogare la tabella. `system_multiregion_info.keyspaces` La seguente istruzione CQL ne è un esempio.

```
SELECT keyspace_name, region, status, tables_replication_progress
FROM system_multiregion_info.keyspaces
WHERE keyspace_name = 'my_keyspace';
```

Mentre è in corso un'operazione di replica, lo stato mostra l'avanzamento della creazione della tabella nella nuova regione. Questo è un esempio in cui 5 tabelle su 10 sono state replicate nella nuova regione.

| keyspace_name | region    | status   | tables_replication_progress |
|---------------|-----------|----------|-----------------------------|
| my_keyspace   | us-east-1 | Updating |                             |
| my_keyspace   | us-west-2 | Updating |                             |
| my_keyspace   | eu-west-1 | Creating | 50%                         |

Una volta completato correttamente il processo di replica, l'output dovrebbe essere simile a questo esempio.

| keyspace_name | region    | status |
|---------------|-----------|--------|
| my_keyspace   | us-east-1 | Active |
| my_keyspace   | us-west-2 | Active |
| my_keyspace   | eu-west-1 | Active |

## CLI

Utilizzo di AWS CLI per verificare lo stato di avanzamento dell'aggiunta della regione

- Per confermare lo stato della creazione della replica della tabella per un determinato keyspace, è possibile utilizzare il seguente esempio.

```
aws keyspaces get-keyspace \
--keyspace-name my_keyspace
```

L'output dovrebbe essere simile a quello di questo esempio.

```
{
 "keyspaceName": "my_keyspace",
```

```
{
 "resourceArn": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/
my_keyspace/",
 "replicationStrategy": "MULTI_REGION",
 "replicationRegions": [
 "us-east-1",
 "eu-west-1"
]
 "replicationGroupStatus": [
 {
 "RegionName": "us-east-1",
 "KeyspaceStatus": "Active"
 },
 {
 "RegionName": "eu-west-1",
 "KeyspaceStatus": "Creating",
 "TablesReplicationProgress": "50.0%"
 }
]
}
```

## Crea una tabella multiregionale con impostazioni predefinite in Amazon Keyspaces

Questa sezione fornisce esempi di come creare una tabella multiregionale in modalità on-demand con tutte le impostazioni predefinite. Puoi farlo sulla console Amazon Keyspaces, utilizzando CQL o il AWS CLI. Tutte le tabelle che crei in uno spazio chiave multiregionale ereditano automaticamente le impostazioni multiregionali dallo spazio chiave.

Per creare uno spazio chiave multiregionale, vedere. [the section called “Crea uno spazio chiave multiregionale”](#)

### Console

Creare una tabella multiregionale con impostazioni predefinite (console)

1. [Accedi a e apri Console di gestione AWS la console Amazon Keyspaces a casahttps://console.aws.amazon.com/keyspaces/.](https://console.aws.amazon.com/keyspaces/)
2. Scegli uno spazio di chiavi multiregionale.
3. Nella scheda Tabelle, scegli Crea tabella.
4. In Nome tabella, inserisci il nome della tabella. Le informazioni in Regioni AWS cui questa tabella viene replicata sono mostrate nella casella delle informazioni.



5. Continuare con lo schema della tabella.
6. In Impostazioni tabella, continua con l'opzione Impostazioni predefinite. Nota le seguenti impostazioni predefinite per le tabelle con più regioni.
  - Modalità di capacità: la modalità di capacità predefinita è On-demand. Per ulteriori informazioni sulla configurazione della modalità provisioned, vedere [the section called “Crea una tabella multiregionale in modalità provisioning”](#)
  - Gestione delle chiavi di crittografia: è supportata solo l'Chiave di proprietà di AWS
  - Timestamp lato client: questa funzionalità è richiesta per le tabelle multiregionali.
- Scegli Personalizza impostazioni se devi attivare Time to Live (TTL) per la tabella e tutte le sue repliche.

 Note

Non sarai in grado di modificare le impostazioni TTL su una tabella multiregionale esistente.

7. Per finire, scegli Crea tabella.

## Cassandra Query Language (CQL)

Crea una tabella multiregionale in modalità on-demand con impostazioni predefinite

- Per creare una tabella multiregionale con impostazioni predefinite, è possibile utilizzare la seguente istruzione CQL.

```
CREATE TABLE mykeyspace.mytable(pk int, ck int, PRIMARY KEY (pk, ck))
 WITH CUSTOM_PROPERTIES = {
 'capacity_mode':{
 'throughput_mode':'PAY_PER_REQUEST'
 },
 'point_in_time_recovery':{
 'status':'enabled'
 },
 'encryption_specification':{
 'encryption_type':'AWS_OWNED_KMS_KEY'
 },
 'client_side_timestamps':{
```

```
'status': 'enabled'
}
```

## CLI

### Usando il AWS CLI

1. Per creare una tabella multiregionale con impostazioni predefinite, è sufficiente specificare lo schema. È possibile utilizzare l'esempio seguente.

```
aws keyspaces create-table --keyspace-name mykeyspace --table-name mytable \
--schema-definition 'allColumns=[{name=pk,type=int}],partitionKeys={name= pk}'
```

L'output del comando è:

```
{
 "resourceArn": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/
mykeyspace/table/mytable"
}
```

2. Per confermare le impostazioni della tabella, puoi usare la seguente dichiarazione.

```
aws keyspaces get-table --keyspace-name mykeyspace --table-name mytable
```

L'output mostra tutte le impostazioni predefinite di una tabella multiregionale.

```
{
 "keyspaceName": "mykeyspace",
 "tableName": "mytable",
 "resourceArn": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/
mykeyspace/table/mytable",
 "creationTimestamp": "2023-12-19T16:50:37.639000+00:00",
 "status": "ACTIVE",
 "schemaDefinition": {
 "allColumns": [
 {
 "name": "pk",
 "type": "int"
 }
]
 }
}
```

```

],
 "partitionKeys": [
 {
 "name": "pk"
 }
],
 "clusteringKeys": [],
 "staticColumns": []
 },
 "capacitySpecification": {
 "throughputMode": "PAY_PER_REQUEST",
 "lastUpdateToPayPerRequestTimestamp": "2023-12-19T16:50:37.639000+00:00"
 },
 "encryptionSpecification": {
 "type": "AWS_OWNED_KMS_KEY"
 },
 "pointInTimeRecovery": {
 "status": "DISABLED"
 },
 "defaultTimeToLive": 0,
 "comment": {
 "message": ""
 },
 "clientSideTimestamps": {
 "status": "ENABLED"
 },
 "replicaSpecifications": [
 {
 "region": "us-east-1",
 "status": "ACTIVE",
 "capacitySpecification": {
 "throughputMode": "PAY_PER_REQUEST",
 "lastUpdateToPayPerRequestTimestamp": 1702895811.469
 }
 },
 {
 "region": "eu-north-1",
 "status": "ACTIVE",
 "capacitySpecification": {
 "throughputMode": "PAY_PER_REQUEST",
 "lastUpdateToPayPerRequestTimestamp": 1702895811.121
 }
 }
]
}

```

```
}
```

## Crea una tabella multiregionale in modalità provisioning con scalabilità automatica in Amazon Keyspaces

Questa sezione fornisce esempi su come creare una tabella multiregione in modalità provisioning con ridimensionamento automatico. Puoi farlo sulla console Amazon Keyspaces, utilizzando CQL o il AWS CLI

Per ulteriori informazioni sulle configurazioni supportate e sulle funzionalità di replica in più regioni, consulta. [the section called “Note per l'utilizzo”](#)

Per creare uno spazio chiave multiregionale, vedere. [the section called “Crea uno spazio chiave multiregionale”](#)

Quando si crea una nuova tabella multiarea in modalità provisioning con impostazioni di ridimensionamento automatico, è possibile specificare le impostazioni generali per la tabella valide per tutto Regioni AWS ciò in cui la tabella viene replicata. È quindi possibile sovrascrivere le impostazioni della capacità di lettura e leggere le impostazioni di ridimensionamento automatico per ogni replica. La capacità di scrittura, tuttavia, rimane sincronizzata tra tutte le repliche per garantire una capacità sufficiente per replicare le scritture in tutte le regioni.

### Note

La scalabilità automatica di Amazon Keyspaces richiede la presenza di un ruolo collegato al servizio (AWSServiceRoleForApplicationAutoScaling\_CassandraTable) che esegue azioni di ridimensionamento automatico per tuo conto. Questo ruolo viene creato automaticamente per te. Per ulteriori informazioni, consulta [the section called “Uso di ruoli collegati ai servizi”](#).

## Console

Crea una nuova tabella multiregionale con il ridimensionamento automatico abilitato

1. [Accedi a e apri Console di gestione AWS la console Amazon Keyspaces a casahttps://console.aws.amazon.com/keyspaces/.](https://console.aws.amazon.com/keyspaces/)
2. Scegli uno spazio di chiavi multiregionale.

3. Nella scheda Tabelle, scegli Crea tabella.
4. Nella pagina Crea tabella nella sezione Dettagli tabella, seleziona uno spazio chiave e fornisci un nome per la nuova tabella.
5. Nella sezione Colonne, crea lo schema per la tua tabella.
6. Nella sezione Chiave primaria, definisci la chiave primaria della tabella e seleziona le colonne di clustering opzionali.
7. Nella sezione Impostazioni tabella, scegli Personalizza impostazioni.
8. Continua con le impostazioni della capacità di lettura/scrittura.
9. Per la modalità Capacity, scegli Provisioned.
10. Nella sezione Capacità di lettura, conferma che l'opzione Scale automaticamente sia selezionata.

È possibile scegliere di configurare le stesse unità di capacità di lettura per tutte le unità in Regioni AWS cui viene replicata la tabella. In alternativa, è possibile deselezionare la casella di controllo e configurare la capacità di lettura per ciascuna regione in modo diverso.

Se si sceglie di configurare ogni regione in modo diverso, si selezionano le unità di capacità di lettura minima e massima per ogni replica della tabella, nonché l'utilizzo previsto.

- Unità di capacità minima: immettete il valore per il livello minimo di velocità effettiva che la tabella deve essere sempre pronta a supportare. Il valore deve essere compreso tra 1 e la quota massima di throughput al secondo prevista per l'account (40.000 per impostazione predefinita).
- Unità di capacità massima: inserisci la quantità massima di velocità effettiva che desideri fornire per la tabella. Il valore deve essere compreso tra 1 e la quota massima di throughput al secondo prevista per l'account (40.000 per impostazione predefinita).
- Utilizzo obiettivo: inserisci un tasso di utilizzo obiettivo compreso tra il 20% e il 90%. Quando il traffico supera il tasso di utilizzo target definito, la capacità viene automaticamente aumentata. Quando il traffico scende al di sotto dell'obiettivo definito, viene nuovamente ridimensionato automaticamente.
- Deselezionate la casella di controllo Ridimensiona automaticamente se desiderate assegnare manualmente la capacità di lettura della tabella. Questa impostazione si applica a tutte le repliche della tabella.

 Note

Per garantire una capacità di lettura sufficiente per tutte le repliche, consigliamo la scalabilità automatica di Amazon Keyspaces per tabelle multiregionali fornite.

 Note

Per ulteriori informazioni sulle quote predefinite per il tuo account e su come aumentarle, consulta. [Quote](#)

11. Nella sezione Capacità di scrittura, conferma che sia selezionata l'opzione Scala automaticamente. Quindi configura le unità di capacità per la tabella. Le unità di capacità di scrittura rimangono sincronizzate in tutte le Regioni AWS regioni per garantire una capacità sufficiente per replicare gli eventi di scrittura tra le regioni.
  - Clear Scale automaticamente se si desidera fornire manualmente la capacità di scrittura della tabella. Questa impostazione si applica a tutte le repliche della tabella.

 Note

Per garantire una capacità di scrittura sufficiente per tutte le repliche, consigliamo la scalabilità automatica di Amazon Keyspaces per tabelle multiregionali fornite.

12. Scegliere Create table (Crea tabella). La tabella viene creata con i parametri di ridimensionamento automatico specificati.

## Cassandra Query Language (CQL)

Crea una tabella multiregionale con modalità di capacità assegnata e scalabilità automatica tramite CQL

- Per creare una tabella multiarea in modalità provisioning con ridimensionamento automatico, è necessario innanzitutto specificare la modalità di capacità definendo `CUSTOM_PROPERTIES` per la tabella. Dopo aver specificato la modalità di capacità assegnata, è possibile configurare le impostazioni di ridimensionamento automatico per la tabella utilizzando `AUTOSCALING_SETTINGS`

Per informazioni dettagliate sulle impostazioni di ridimensionamento automatico, sulla politica di tracciamento degli obiettivi, sul valore target e sulle impostazioni opzionali, vedere [the section called “Crea una nuova tabella con ridimensionamento automatico”](#).

Per definire la capacità di lettura di una replica di tabella in una regione specifica, è possibile configurare i seguenti parametri come parte della tabella: `replica_updates`

- La regione
- Le unità di capacità di lettura fornite (opzionali)
- Impostazioni di ridimensionamento automatico per la capacità di lettura (opzionale)

L'esempio seguente mostra un'CREATE TABLEistruzione per una tabella multiregionale in modalità provisioning. Le impostazioni generali di autoscaling della capacità di scrittura e lettura sono le stesse. Tuttavia, le impostazioni di lettura automatica specificano periodi di recupero aggiuntivi di 60 secondi prima di aumentare o ridurre la capacità di lettura della tabella. Inoltre, le impostazioni di ridimensionamento automatico della capacità di lettura per la regione Stati Uniti orientali (Virginia settentrionale) sono superiori a quelle di altre repliche. Inoltre, il valore target è impostato al 70% anziché al 50%.

```
CREATE TABLE mykeyspace.mytable(pk int, ck int, PRIMARY KEY (pk, ck))
WITH CUSTOM_PROPERTIES = {
 'capacity_mode': {
 'throughput_mode': 'PROVISIONED',
 'read_capacity_units': 5,
 'write_capacity_units': 5
 }
} AND AUTOSCALING_SETTINGS = {
 'provisioned_write_capacity_autoscaling_update': {
 'maximum_units': 10,
 'minimum_units': 5,
 'scaling_policy': {
 'target_tracking_scaling_policy_configuration': {
 'target_value': 50
 }
 }
 },
 'provisioned_read_capacity_autoscaling_update': {
 'maximum_units': 10,
 'minimum_units': 5,
```

```

 'scaling_policy': {
 'target_tracking_scaling_policy_configuration': {
 'target_value': 50,
 'scale_in_cooldown': 60,
 'scale_out_cooldown': 60
 }
 },
 'replica_updates': {
 'us-east-1': {
 'provisioned_read_capacity_autoscaling_update': {
 'maximum_units': 20,
 'minimum_units': 5,
 'scaling_policy': {
 'target_tracking_scaling_policy_configuration': {
 'target_value': 70
 }
 }
 }
 }
 }
 }
};

```

## CLI

Crea una nuova tabella multiregione in modalità provisioning con ridimensionamento automatico utilizzando il AWS CLI

- Per creare una tabella multiregione in modalità provisioning con configurazione di ridimensionamento automatico, è possibile utilizzare il. AWS CLI Tieni presente che è necessario utilizzare il `create-table` comando CLI di Amazon Keyspaces per configurare le impostazioni di autoscaling multiregionale. Questo perché Application Auto Scaling, il servizio che Amazon Keyspaces utilizza per eseguire l'auto scaling per tuo conto, non supporta più regioni.

Per ulteriori informazioni sulle impostazioni di ridimensionamento automatico, sulla politica di tracciamento del target, sul valore target e sulle impostazioni opzionali, vedere [the section called “Crea una nuova tabella con ridimensionamento automatico”](#).

Per definire la capacità di lettura di una replica di tabella in una regione specifica, è possibile configurare i seguenti parametri come parte della tabella: `replicaSpecifications`



- La regione
- Le unità di capacità di lettura fornite (opzionali)
- Impostazioni di ridimensionamento automatico per la capacità di lettura (opzionale)

Quando crei tabelle multiregione predisposte con impostazioni di ridimensionamento automatico complesse e configurazioni diverse per le repliche di tabelle, è utile caricare le impostazioni di ridimensionamento automatico della tabella e le configurazioni di replica dai file JSON.

Per utilizzare il seguente esempio di codice, è possibile scaricare i file JSON di esempio da [auto-scaling.zip](#) ed estrarre `auto-scaling.json` e `replication.json`. Prendi nota del percorso dei file.

In questo esempio, i file JSON si trovano nella directory corrente. Per le diverse opzioni relative al percorso dei file, consultate [Come caricare i parametri da un file](#).

```
aws keyspaces create-table --keyspace-name mykeyspace --table-name mytable \
--schema-definition 'allColumns=[{name=pk,type=int},
{name=ck,type=int}],partitionKeys=[{name=pk},{name=ck}]' \
--capacity-specification
 throughputMode=PROVISIONED,readCapacityUnits=1,writeCapacityUnits=1 \
--auto-scaling-specification file://auto-scaling.json \
--replica-specifications file://replication.json
```

## Aggiorna la capacità fornita e le impostazioni di ridimensionamento automatico per una tabella multiregionale in Amazon Keyspaces

Questa sezione include esempi su come utilizzare la console, CQL e gestire le impostazioni di scalabilità automatica AWS CLI di Amazon Keyspaces delle tabelle multiregionali predisposte. Per ulteriori informazioni sulle opzioni generali di configurazione della scalabilità automatica e sul loro funzionamento, vedere [the section called “Gestisci la capacità di throughput con la scalabilità automatica”](#).

Tieni presente che se utilizzi la modalità di capacità fornita per tabelle multiregionali, devi sempre utilizzare le chiamate API Amazon Keyspaces per configurare la scalabilità automatica. Questo perché le operazioni sottostanti dell'API Application Auto Scaling non riconoscono la regione.

Per ulteriori informazioni su come stimare la velocità effettiva della capacità di scrittura delle tabelle multiregionali predisposte, vedere [the section called “Stima della capacità per una tabella multiregionale”](#)

Per ulteriori informazioni sull'API Amazon Keyspaces, consulta Amazon [Keyspaces](#) API Reference.

Quando si aggiornano le impostazioni della modalità di provisioning o della scalabilità automatica di una tabella multiregione, è possibile aggiornare le impostazioni della capacità di lettura e la configurazione di scalabilità automatica in lettura per ogni replica della tabella.

La capacità di scrittura, tuttavia, rimane sincronizzata tra tutte le repliche per garantire una capacità sufficiente per replicare le scritture in tutte le regioni.

### Cassandra Query Language (CQL)

Aggiorna la capacità fornita e le impostazioni di ridimensionamento automatico di una tabella multiregionale utilizzando CQL

- È possibile utilizzare `ALTER TABLE` per aggiornare la modalità di capacità e le impostazioni di ridimensionamento automatico di una tabella esistente. Se stai aggiornando una tabella che è attualmente in modalità di capacità su richiesta, `capacity_mode` è obbligatorio. Se la tabella è già in modalità di capacità assegnata, questo campo può essere omesso.

Per informazioni dettagliate sulle impostazioni di ridimensionamento automatico, sulla politica di tracciamento degli obiettivi, sul valore target e sulle impostazioni opzionali, vedere [the section called “Crea una nuova tabella con ridimensionamento automatico”](#).

Nella stessa istruzione, puoi anche aggiornare la capacità di lettura e le impostazioni di ridimensionamento automatico delle repliche di tabella in regioni specifiche aggiornando la proprietà della `replica_updates` tabella. La seguente dichiarazione ne è un esempio.

```
ALTER TABLE mykeyspace.mytable
WITH CUSTOM_PROPERTIES = {
 'capacity_mode': {
 'throughput_mode': 'PROVISIONED',
 'read_capacity_units': 1,
 'write_capacity_units': 1
 }
} AND AUTOSCALING_SETTINGS = {
 'provisioned_write_capacity_autoscaling_update': {
 'maximum_units': 10,
```

```

 'minimum_units': 5,
 'scaling_policy': {
 'target_tracking_scaling_policy_configuration': {
 'target_value': 50
 }
 },
 'provisioned_read_capacity_autoscaling_update': {
 'maximum_units': 10,
 'minimum_units': 5,
 'scaling_policy': {
 'target_tracking_scaling_policy_configuration': {
 'target_value': 50,
 'scale_in_cooldown': 60,
 'scale_out_cooldown': 60
 }
 }
 },
 'replica_updates': {
 'us-east-1': {
 'provisioned_read_capacity_autoscaling_update': {
 'maximum_units': 20,
 'minimum_units': 5,
 'scaling_policy': {
 'target_tracking_scaling_policy_configuration': {
 'target_value': 70
 }
 }
 }
 }
 }
 };

```

## CLI

Aggiornare la capacità fornita e le impostazioni di ridimensionamento automatico di una tabella multiregionale utilizzando il AWS CLI

- Per aggiornare la modalità di provisioning e la configurazione di ridimensionamento automatico di una tabella esistente, è possibile utilizzare il AWS CLI `update-table` comando.

Tieni presente che devi utilizzare i comandi CLI di Amazon Keyspaces per creare o modificare le impostazioni di auto scaling multiregionale. Questo perché Application Auto Scaling, il servizio che Amazon Keyspaces utilizza per eseguire la scalabilità automatica della capacità delle tabelle per tuo conto, non supporta più di una tabella. Regioni AWS

Per aggiornare la capacità di lettura di una replica di tabella in una regione specifica, puoi modificare uno dei seguenti parametri opzionali della tabella: `replicaSpecifications`

- Le unità di capacità di lettura fornite (facoltative)
- Impostazioni di ridimensionamento automatico per la capacità di lettura (opzionale)

Quando aggiorni tabelle multiregionali con impostazioni di ridimensionamento automatico complesse e configurazioni diverse per le repliche delle tabelle, è utile caricare le impostazioni di ridimensionamento automatico della tabella e le configurazioni di replica dai file JSON.

Per utilizzare il seguente esempio di codice, è possibile scaricare i file JSON di esempio da [auto-scaling.zip](#) ed estrarre `auto-scaling.json` e `replication.json`. Prendi nota del percorso dei file.

In questo esempio, i file JSON si trovano nella directory corrente. Per le diverse opzioni relative al percorso dei file, consultate [Come caricare i parametri da un file](#).

```
aws keyspaces update-table --keyspace-name mykeyspace --table-name mytable \
--capacity-specification
 throughputMode=PROVISIONED,readCapacityUnits=1,writeCapacityUnits=1 \
--auto-scaling-specification file://auto-scaling.json \
--replica-specifications file://replication.json
```

## Visualizza la capacità fornita e le impostazioni di ridimensionamento automatico per una tabella multiregionale in Amazon Keyspaces

Puoi visualizzare la capacità fornita di una tabella multiregionale e le impostazioni di scalabilità automatica sulla console Amazon Keyspaces, utilizzando CQL o il. AWS CLI Questa sezione fornisce esempi di come eseguire questa operazione utilizzando CQL e. AWS CLI

## Cassandra Query Language (CQL)

Visualizza la capacità fornita e le impostazioni di ridimensionamento automatico di una tabella multiregionale utilizzando CQL

- Per visualizzare la configurazione di ridimensionamento automatico di una tabella multiregione, utilizzare il comando seguente.

```
SELECT * FROM system_multiregion_info.autoscaling WHERE keyspace_name =
'mykeyspace' AND table_name = 'mytable';
```

L'output di questo comando è simile al seguente:

```
keyspace_name | table_name | region |
provisioned_read_capacity_autoscaling_update

provisioned_write_capacity_autoscaling_update
-----+-----+-----
+-----+-----+-----
+-----+-----+-----
mykeyspace | mytable | ap-southeast-1 | {'minimum_units': 5,
'maximum_units': 10, 'scaling_policy':
{'target_tracking_scaling_policy_configuration': {'scale_out_cooldown':
60, 'disable_scale_in': false, 'target_value': 50, 'scale_in_cooldown':
60}}} | {'minimum_units': 5, 'maximum_units': 10, 'scaling_policy':
{'target_tracking_scaling_policy_configuration': {'scale_out_cooldown': 0,
'disable_scale_in': false, 'target_value': 50, 'scale_in_cooldown': 0}}}
mykeyspace | mytable | us-east-1 | {'minimum_units': 5,
'maximum_units': 20, 'scaling_policy':
{'target_tracking_scaling_policy_configuration': {'scale_out_cooldown':
60, 'disable_scale_in': false, 'target_value': 70, 'scale_in_cooldown':
60}}} | {'minimum_units': 5, 'maximum_units': 10, 'scaling_policy':
{'target_tracking_scaling_policy_configuration': {'scale_out_cooldown': 0,
'disable_scale_in': false, 'target_value': 50, 'scale_in_cooldown': 0}}}
mykeyspace | mytable | eu-west-1 | {'minimum_units': 5,
'maximum_units': 10, 'scaling_policy':
{'target_tracking_scaling_policy_configuration': {'scale_out_cooldown':
60, 'disable_scale_in': false, 'target_value': 50, 'scale_in_cooldown':
60}}} | {'minimum_units': 5, 'maximum_units': 10, 'scaling_policy':
{'target_tracking_scaling_policy_configuration': {'scale_out_cooldown': 0,
'disable_scale_in': false, 'target_value': 50, 'scale_in_cooldown': 0}}}
```

## CLI

Visualizza la capacità fornita e le impostazioni di ridimensionamento automatico di una tabella multiregionale utilizzando il AWS CLI

- Per visualizzare la configurazione di ridimensionamento automatico di una tabella multiregione, è possibile utilizzare l'get-table-auto-scaling-settingsoperazione. Il seguente comando CLI ne è un esempio.

```
aws keyspaces get-table-auto-scaling-settings --keyspace-name mykeyspace --table-name mytable
```

Vedrai il seguente output.

```
{
 "keyspaceName": "mykeyspace",
 "tableName": "mytable",
 "resourceArn": "arn:aws:cassandra:us-east-1:777788889999:/keyspace/mykeyspace/table/mytable",
 "autoScalingSpecification": {
 "writeCapacityAutoScaling": {
 "autoScalingDisabled": false,
 "minimumUnits": 5,
 "maximumUnits": 10,
 "scalingPolicy": {
 "targetTrackingScalingPolicyConfiguration": {
 "disableScaleIn": false,
 "scaleInCooldown": 0,
 "scaleOutCooldown": 0,
 "targetValue": 50.0
 }
 }
 },
 "readCapacityAutoScaling": {
 "autoScalingDisabled": false,
 "minimumUnits": 5,
 "maximumUnits": 20,
 "scalingPolicy": {
 "targetTrackingScalingPolicyConfiguration": {
 "disableScaleIn": false,
 "scaleInCooldown": 60,
 "scaleOutCooldown": 60,

```

```

 "targetValue": 70.0
 }
 }
 },
 "replicaSpecifications": [
 {
 "region": "us-east-1",
 "autoScalingSpecification": {
 "writeCapacityAutoScaling": {
 "autoScalingDisabled": false,
 "minimumUnits": 5,
 "maximumUnits": 10,
 "scalingPolicy": {
 "targetTrackingScalingPolicyConfiguration": {
 "disableScaleIn": false,
 "scaleInCooldown": 0,
 "scaleOutCooldown": 0,
 "targetValue": 50.0
 }
 }
 },
 "readCapacityAutoScaling": {
 "autoScalingDisabled": false,
 "minimumUnits": 5,
 "maximumUnits": 20,
 "scalingPolicy": {
 "targetTrackingScalingPolicyConfiguration": {
 "disableScaleIn": false,
 "scaleInCooldown": 60,
 "scaleOutCooldown": 60,
 "targetValue": 70.0
 }
 }
 }
 }
 },
 {
 "region": "eu-north-1",
 "autoScalingSpecification": {
 "writeCapacityAutoScaling": {
 "autoScalingDisabled": false,
 "minimumUnits": 5,
 "maximumUnits": 10,

```

```

 "scalingPolicy": {
 "targetTrackingScalingPolicyConfiguration": {
 "disableScaleIn": false,
 "scaleInCooldown": 0,
 "scaleOutCooldown": 0,
 "targetValue": 50.0
 }
 },
 "readCapacityAutoScaling": {
 "autoScalingDisabled": false,
 "minimumUnits": 5,
 "maximumUnits": 10,
 "scalingPolicy": {
 "targetTrackingScalingPolicyConfiguration": {
 "disableScaleIn": false,
 "scaleInCooldown": 60,
 "scaleOutCooldown": 60,
 "targetValue": 50.0
 }
 }
 }
 }
}

```

## Disattiva la scalabilità automatica per una tabella in Amazon Keyspaces

Questa sezione fornisce esempi su come disattivare la scalabilità automatica per una tabella multiregione in modalità di capacità fornita. Puoi farlo sulla console Amazon Keyspaces, utilizzando CQL o il AWS CLI

### Important

Si consiglia di utilizzare la scalabilità automatica per le tabelle multiregionali che utilizzano la modalità di capacità assegnata. Per ulteriori informazioni, consulta [the section called “Stima della capacità per una tabella multiregionale”](#).



 Note

Per eliminare il ruolo collegato al servizio utilizzato da Application Auto Scaling, è necessario disabilitare il ridimensionamento automatico su tutte le tabelle dell'account in tutto. Regioni AWS

## Console

Disattiva il ridimensionamento automatico di Amazon Keyspaces per una tabella multiregionale esistente sulla console

1. [Accedi a e apri Console di gestione AWS la console Amazon Keyspaces a casahttps://console.aws.amazon.com/keyspaces/.](https://console.aws.amazon.com/keyspaces/)
2. Scegli la tabella su cui vuoi lavorare e scegli la scheda Capacità.
3. Nella sezione Impostazioni di capacità, scegli Modifica.
4. Per disabilitare il ridimensionamento automatico di Amazon Keyspaces, deseleziona la casella di controllo Scala automaticamente. La disabilitazione del ridimensionamento automatico annulla la registrazione della tabella come destinazione scalabile con Application Auto Scaling. Per eliminare il ruolo collegato al servizio utilizzato da Application Auto Scaling per accedere alla tabella Amazon Keyspaces, segui la procedura riportata di seguito. [the section called “Eliminazione di un ruolo collegato al servizio per Amazon Keyspaces”](#)
5. Una volta definite le impostazioni di ridimensionamento automatico, scegli Salva.

## Cassandra Query Language (CQL)

Disattiva il ridimensionamento automatico per una tabella multiregione utilizzando CQL

- È possibile utilizzare `ALTER TABLE` per disattivare il ridimensionamento automatico per una tabella esistente. Tieni presente che non puoi disattivare il ridimensionamento automatico per una singola replica di tabella.

Nell'esempio seguente, la scalabilità automatica è disattivata per la capacità di lettura della tabella.

```
ALTER TABLE mykeyspace.mytable
WITH AUTOSCALING_SETTINGS = {
 'provisioned_read_capacity_autoscaling_update': {
```

```
 'autoscaling_disabled': true
 }
};
```

## CLI

Disattiva il ridimensionamento automatico per una tabella multiregionale utilizzando il AWS CLI

- È possibile utilizzare il AWS CLI `update-table` comando per disattivare il ridimensionamento automatico per una tabella esistente. Tieni presente che non puoi disattivare il ridimensionamento automatico per una singola replica di tabella.

Nell'esempio seguente, la scalabilità automatica è disattivata per la capacità di lettura della tabella.

```
aws keyspaces update-table --keyspace-name mykeyspace --table-name mytable
 \ --auto-scaling-specification
 readCapacityAutoScaling={autoScalingDisabled=true}
```

## Imposta manualmente la capacità fornita di una tabella multiregionale in Amazon Keyspaces

Se è necessario disattivare la scalabilità automatica per una tabella multiregionale, è possibile assegnare manualmente la capacità di lettura della tabella per una tabella di replica utilizzando CQL o il AWS CLI

### Note

Si consiglia di utilizzare la scalabilità automatica per le tabelle multiregionali che utilizzano la modalità di capacità assegnata. Per ulteriori informazioni, consulta [the section called “Stima della capacità per una tabella multiregionale”](#).

## Cassandra Query Language (CQL)

Impostazione manuale della capacità fornita di una tabella multiregionale utilizzando CQL

- È possibile utilizzare `ALTER TABLE` per fornire manualmente la capacità di lettura della tabella per una tabella di replica.

```
ALTER TABLE mykeyspace.mytable
WITH CUSTOM_PROPERTIES = {
 'capacity_mode': {
 'throughput_mode': 'PROVISIONED',
 'read_capacity_units': 1,
 'write_capacity_units': 1
 },
 'replica_updates': {
 'us-east-1': {
 'read_capacity_units': 2
 }
 }
};
```

## CLI

Imposta manualmente la capacità fornita di una tabella multiregionale utilizzando il AWS CLI

- Se è necessario disattivare la scalabilità automatica per una tabella multiregionale, è possibile utilizzare `update-table` per fornire manualmente la capacità di lettura della tabella per una tabella di replica.

```
aws keyspaces update-table --keyspace-name mykeyspace --table-name mytable \
--capacity-specification
 throughputMode=PROVISIONED,readCapacityUnits=1,writeCapacityUnits=1 \
--replica-specifications region="us-east-1",readCapacityUnits=5
```

# Backup e ripristino dei dati con point-in-time ripristino per Amazon Keyspaces

Point-in-time recovery (PITR) aiuta a proteggere le tabelle Amazon Keyspaces da operazioni di scrittura o cancellazione accidentali fornendo backup continui dei dati delle tabelle.

Ad esempio, supponiamo che uno script di test scriva accidentalmente su una tabella Amazon Keyspaces di produzione. Con point-in-time il ripristino, puoi ripristinare i dati di quella tabella in qualsiasi momento dall'attivazione di PITR negli ultimi 35 giorni. Se elimini una tabella con point-in-time il ripristino abilitato, puoi interrogare i dati della tabella eliminata per 35 giorni (senza costi aggiuntivi) e ripristinarla allo stato in cui si trovava appena prima del punto di eliminazione.

Puoi ripristinare una tabella Amazon Keyspaces in un determinato momento utilizzando la console, l' AWS SDK e () o Cassandra Query Language AWS Command Line Interface (CQL AWS CLI). Per ulteriori informazioni, consulta [Usa point-in-time il ripristino in Amazon Keyspaces](#).

Point-in-time le operazioni non hanno alcun impatto sulle prestazioni o sulla disponibilità sulla tabella di base e il ripristino di una tabella non richiede un throughput aggiuntivo.

Per informazioni sulle quote PITR, vedere. [Quote](#)

Per informazioni sui prezzi, consulta i prezzi di [Amazon Keyspaces \(per Apache Cassandra\)](#).

## Argomenti

- [Come funziona point-in-time il ripristino in Amazon Keyspaces](#)
- [Usa point-in-time il ripristino in Amazon Keyspaces](#)

## Come funziona point-in-time il ripristino in Amazon Keyspaces

Questa sezione fornisce una panoramica di come funziona il point-in-time ripristino di Amazon Keyspaces (PITR). Per ulteriori informazioni sui prezzi, consulta i prezzi di [Amazon Keyspaces \(per Apache Cassandra\)](#).

## Argomenti

- [Finestra temporale per i backup continui PITR](#)
- [Impostazioni di ripristino PITR](#)
- [Ripristino PITR di tabelle crittografate](#)

- [Ripristino PITR di tabelle multiregionali](#)
- [Ripristino PITR di tabelle con tipi definiti dall'utente \(\) UDTs](#)
- [Tempo di ripristino della tabella con PITR](#)
- [Amazon Keyspaces PITR e integrazione con i servizi AWS](#)

## Finestra temporale per i backup continui PITR

Amazon Keyspaces PITR utilizza due timestamp per mantenere l'intervallo di tempo per il quale sono disponibili backup ripristinabili per una tabella.

- Data del primo ripristino: indica l'ora del primo backup ripristinabile. Il primo backup ripristinabile risale a 35 giorni o quando PITR è stato abilitato, a seconda di quale delle due date sia più recente. La finestra di backup massima di 35 giorni non può essere modificata.
- Ora corrente: il timestamp dell'ultimo backup ripristinabile è l'ora corrente. Se durante il ripristino non viene fornito alcun timestamp, viene utilizzata l'ora corrente.

Quando PITR è abilitato, è possibile eseguire il ripristino in qualsiasi momento compreso tra `EarliestRestorableDateTime` e `CurrentTime`. È possibile ripristinare i dati della tabella solo in un momento in cui PITR era abilitato.

Se si disattiva PITR e successivamente lo si riattiva nuovamente, si reimposta l'ora di inizio del primo backup disponibile a quella in cui PITR è stato riattivato. Ciò significa che la disattivazione di PITR cancella la cronologia dei backup.

### Note

Le operazioni DDL (Data Definition Language) sulle tabelle, come le modifiche allo schema, vengono eseguite in modo asincrono. È possibile visualizzare solo le operazioni completate nei dati della tabella ripristinata, ma è possibile visualizzare azioni aggiuntive nella tabella di origine se erano in corso al momento del ripristino. Per un elenco delle istruzioni DDL, vedere [the section called “Istruzioni DDL”](#).

Non è necessario che una tabella sia attiva per essere ripristinata. Puoi anche ripristinare le tabelle eliminate se PITR era abilitato sulla tabella eliminata e l'eliminazione è avvenuta nella finestra di backup (o negli ultimi 35 giorni).

 Note

Se viene creata una nuova tabella con lo stesso nome qualificato (ad esempio, `mykeyspace.mytable`) di una tabella eliminata in precedenza, la tabella eliminata non sarà più ripristinabile. Se si tenta di eseguire questa operazione dalla console, viene visualizzato un avviso.

## Impostazioni di ripristino PITR

Quando ripristini una tabella utilizzando PITR, Amazon Keyspaces ripristina lo schema e i dati della tabella di origine allo stato in base al timestamp selezionato `day:hour:minute:second ()` su una nuova tabella. PITR non sovrascrive le tabelle esistenti.

Oltre allo schema e ai dati della tabella, PITR ripristina i dati `custom_properties` dalla tabella di origine. A differenza dei dati della tabella, che vengono ripristinati in base al timestamp selezionato tra la prima ora di ripristino e l'ora corrente, le proprietà personalizzate vengono sempre ripristinate in base alle impostazioni della tabella all'ora corrente.

Le impostazioni della tabella ripristinata corrispondono alle impostazioni della tabella di origine con il timestamp di quando è stato avviato il ripristino. Se desideri sovrascrivere queste impostazioni durante il ripristino, puoi farlo utilizzando `WITH custom_properties`. Le proprietà personalizzate includono le seguenti impostazioni.

- Modalità di capacità in lettura/scrittura
- Impostazioni della capacità di throughput fornite
- Impostazioni PITR

Se la tabella è in modalità di capacità fornita con ridimensionamento automatico abilitato, l'operazione di ripristino ripristina anche le impostazioni di ridimensionamento automatico della tabella. È possibile sovrascriverli utilizzando il `autoscaling_settings` parametro in CQL o `autoScalingSpecification` con la CLI. Per ulteriori informazioni sulle impostazioni di ridimensionamento automatico, vedere [the section called “Gestisci la capacità di throughput con la scalabilità automatica”](#).

Quando esegui un ripristino completo della tabella, tutte le impostazioni per la tabella ripristinata provengono dalle impostazioni correnti della tabella di origine al momento del ripristino.

Ad esempio, supponi che il throughput assegnato di una tabella sia stato recentemente abbassato a 50 unità di capacità in lettura e 50 unità di capacità in scrittura. Quindi ripristini lo stato della tabella a tre settimane fa. In quel momento, la velocità effettiva assegnata era impostata su 100 unità di capacità di lettura e 100 unità di capacità di scrittura. In questo caso, Amazon Keyspaces ripristina i dati della tabella a quel momento, ma utilizza le impostazioni di throughput attualmente fornite (50 unità di capacità di lettura e 50 unità di capacità di scrittura).

Le seguenti impostazioni non vengono ripristinate ed è necessario configurarle manualmente per la nuova tabella.

- Amazon Keyspaces modifica i flussi di acquisizione dati (CDC)
- AWS Identity and Access Management politiche (IAM)
- CloudWatch Parametri e allarmi Amazon
- Tag (possono essere aggiunti all'istruzione CQL utilizzando) `RESTORE WITH TAGS`

## Ripristino PITR di tabelle crittografate

Quando ripristini una tabella utilizzando PITR, Amazon Keyspaces ripristina le impostazioni di crittografia della tabella di origine. Se la tabella è stata crittografata con un Chiave di proprietà di AWS (impostazione predefinita), la tabella viene ripristinata automaticamente con la stessa impostazione. Se la tabella che desideri ripristinare è stata crittografata utilizzando una chiave gestita dal cliente, la stessa chiave gestita dal cliente deve essere accessibile ad Amazon Keyspaces per ripristinare i dati della tabella.

Puoi modificare le impostazioni di crittografia della tabella al momento del ripristino. Per passare da una chiave gestita dal cliente Chiave di proprietà di AWS a una chiave gestita dal cliente, è necessario fornire una chiave gestita dal cliente valida e accessibile al momento del ripristino.

Se desideri passare da una chiave gestita dal cliente a una Chiave di proprietà di AWS, verifica che Amazon Keyspaces abbia accesso alla chiave gestita dal cliente della tabella di origine per ripristinare la tabella con un. Chiave di proprietà di AWS Per ulteriori informazioni sulle impostazioni di crittografia a riposo per le tabelle, consulta [the section called “Come funziona”](#).

### Note

Se la tabella è stata eliminata perché Amazon Keyspaces ha perso l'accesso alla tua chiave gestita dal cliente, devi assicurarti che la chiave gestita dal cliente sia accessibile ad Amazon Keyspaces prima di provare a ripristinare la tabella. Una tabella crittografata con una chiave

gestita dal cliente non può essere ripristinata se Amazon Keyspaces non ha accesso a tale chiave. Per ulteriori informazioni, consulta [Risoluzione dei problemi di accesso tramite chiave](#) nella Guida per gli AWS Key Management Service sviluppatori.

## Ripristino PITR di tabelle multiregionali

È possibile ripristinare una tabella multiregionale utilizzando PITR. Affinché l'operazione di ripristino abbia esito positivo, è necessario abilitare PITR su tutte le repliche della tabella di origine e sia la tabella di origine che quella di destinazione devono essere replicate nella stessa. Regioni AWS

Amazon Keyspaces ripristina le impostazioni della tabella di origine in ciascuna delle regioni replicate che fanno parte del keyspace. Puoi anche sovrascrivere le impostazioni durante l'operazione di ripristino. Per ulteriori informazioni sulle impostazioni che possono essere modificate durante il ripristino, vedere [the section called “Ripristina le impostazioni”](#).

Per ulteriori informazioni sulla replica in più regioni, vedere. [the section called “Come funziona”](#)

## Ripristino PITR di tabelle con tipi definiti dall'utente () UDTs

È possibile ripristinare una tabella che utilizza UDTs. Affinché l'operazione di ripristino abbia esito positivo, i riferimenti UDTs devono esistere ed essere validi nel keyspace.

Se manca un UDT richiesto quando tenti di ripristinare una tabella, Amazon Keyspaces tenta di ripristinare automaticamente lo schema UDT e poi continua a ripristinare la tabella.

Se hai rimosso e ricreato l'UDT, Amazon Keyspaces ripristina l'UDT con il nuovo schema dell'UDT e rifiuta la richiesta di ripristino della tabella utilizzando lo schema UDT originale. In questo caso, se desideri ripristinare la tabella con il vecchio schema UDT, puoi ripristinare la tabella in un nuovo keyspace. Quando si elimina e si ricrea un UDT, anche se lo schema dell'UDT ricreato è lo stesso dell'UDT eliminato, l'UDT ricreato viene considerato un nuovo UDT. In questo caso, Amazon Keyspaces rifiuta la richiesta di ripristino della tabella con il vecchio schema UDT.

Se l'UDT è mancante e Amazon Keyspaces tenta di ripristinarlo, il tentativo fallisce se è stato raggiunto il numero massimo UDTs di account nella regione.

Per ulteriori informazioni sulle quote UDT e sui valori predefiniti, consulta. [the section called “Quote e valori predefiniti per i tipi definiti dall'utente \(UDTs\) in Amazon Keyspaces”](#) Per ulteriori informazioni sull'utilizzo di UDTs, vedere. [the section called “Tipi definiti dall'utente \(\) UDTs”](#)



## Tempo di ripristino della tabella con PITR

Il tempo necessario per ripristinare una tabella si basa su molteplici fattori e non è sempre direttamente correlato alle dimensioni della tabella.

Di seguito sono riportate alcune considerazioni relative ai tempi di ripristino.

- I backup vengono ripristinati in una nuova tabella. Le operazioni necessarie per la creazione di una nuova tabella e l'avvio del processo di ripristino possono richiedere fino a 20 minuti (anche se la tabella è vuota).
- I tempi di ripristino per tabelle di grandi dimensioni con modelli di dati ben distribuiti possono essere di diverse ore o più.
- Se la tabella di origine contiene dati notevolmente distorti, il tempo di ripristino potrebbe aumentare. Ad esempio, se la chiave primaria della tabella utilizza il mese dell'anno come chiave di partizione e tutti i dati provengono dal mese di dicembre, i dati sono distorti.

Una best practice quando si pianifica un ripristino di emergenza consiste nel documentare regolarmente i tempi medi di completamento del ripristino e stabilire in che modo questi tempi influiscono sull'obiettivo del tempo di ripristino complessivo.

## Amazon Keyspaces PITR e integrazione con i servizi AWS

Le seguenti operazioni PITR vengono registrate utilizzando AWS CloudTrail per consentire il monitoraggio e il controllo continui.

- Crea una nuova tabella con PITR abilitato o disabilitato.
- Abilita o disabilita PITR su una tabella esistente.
- Ripristina una tabella attiva o eliminata.

Per ulteriori informazioni, consulta [Registrazione delle chiamate API Amazon Keyspaces con AWS CloudTrail](#).

È possibile eseguire le seguenti azioni PITR utilizzando CloudFormation.

- Crea una nuova tabella con PITR abilitato o disabilitato.
- Abilita o disabilita PITR su una tabella esistente.

Per ulteriori informazioni, consulta [Cassandra Resource Type Reference nella Guida](#) per l'[CloudFormation utente](#).

## Usa point-in-time il ripristino in Amazon Keyspaces

Con Amazon Keyspaces (per Apache Cassandra), puoi ripristinare le tabelle in un momento specifico utilizzando Point-in-Time Restore (PITR). PITR consente di ripristinare una tabella a uno stato precedente negli ultimi 35 giorni, fornendo funzionalità di protezione e ripristino dei dati. Questa funzionalità è utile in casi quali l'eliminazione accidentale dei dati, gli errori delle applicazioni o per scopi di test. È possibile ripristinare i dati in modo rapido ed efficiente, riducendo al minimo i tempi di inattività e la perdita di dati. Le seguenti sezioni ti guidano nel processo di ripristino delle tabelle utilizzando PITR in Amazon Keyspaces, garantendo l'integrità dei dati e la continuità aziendale.

### Argomenti

- [Configura le autorizzazioni IAM della tabella di ripristino per Amazon Keyspaces PITR](#)
- [Configurare PITR per una tabella in Amazon Keyspaces](#)
- [Disattiva PITR per una tabella Amazon Keyspaces](#)
- [Ripristina una tabella dal backup a un momento temporale specificato in Amazon Keyspaces](#)
- [Ripristina una tabella eliminata utilizzando Amazon Keyspaces PITR](#)

## Configura le autorizzazioni IAM della tabella di ripristino per Amazon Keyspaces PITR

Questa sezione riassume come configurare le autorizzazioni per un principale AWS Identity and Access Management (IAM) per ripristinare le tabelle Amazon Keyspaces. In IAM, la policy AWS gestita AmazonKeyspacesFullAccess include le autorizzazioni per ripristinare le tabelle Amazon Keyspaces. Per implementare una policy personalizzata con autorizzazioni minime richieste, considera i requisiti descritti nella sezione successiva.

Per ripristinare correttamente una tabella, il principale IAM necessita delle seguenti autorizzazioni minime:

- `cassandra:Restore`— L'azione di ripristino è necessaria per ripristinare la tabella di destinazione.
- `cassandra:Select`— L'azione di selezione è necessaria per leggere dalla tabella di origine.
- `cassandra:TagResource`— L'azione tag è facoltativa e richiesta solo se l'operazione di ripristino aggiunge tag.

Questo è un esempio di policy che concede le autorizzazioni minime richieste a un utente per ripristinare le tabelle nel keyspace. `mykeyspace`

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "cassandra:Restore",
 "cassandra:Select"
],
 "Resource": [
 "arn:aws:cassandra:us-east-1:111122223333:/keyspace/mykeyspace/*",
 "arn:aws:cassandra:us-east-1:111122223333:/keyspace/system*"
]
 }
]
}
```

Potrebbero essere necessarie autorizzazioni aggiuntive per ripristinare una tabella in base ad altre funzionalità selezionate. Ad esempio, se la tabella di origine è crittografata a riposo con una chiave gestita dal cliente, Amazon Keyspaces deve disporre delle autorizzazioni per accedere alla chiave gestita dal cliente della tabella di origine per ripristinare correttamente la tabella. Per ulteriori informazioni, consulta [the section called “Tabelle PITR e crittografate”](#).

Se utilizzi policy IAM con [chiavi di condizione](#) per limitare il traffico in entrata a fonti specifiche, devi assicurarti che Amazon Keyspaces sia autorizzato a eseguire un'operazione di ripristino per conto del tuo principale. Devi aggiungere una chiave di `aws:ViaAWSService` condizione alla tua policy IAM se la policy limita il traffico in entrata a uno dei seguenti fattori:

- Endpoint VPC con `aws:SourceVpce`
- Intervalli IP con `aws:SourceIp`
- VPCs con `aws:SourceVpc`

La chiave di `aws:ViaAWSService` condizione consente l'accesso quando un AWS servizio effettua una richiesta utilizzando le credenziali del principale. Per ulteriori informazioni, consulta [IAM JSON Policy elements: Condition key](#) nella IAM User Guide.

Di seguito è riportato un esempio di policy che limita il traffico di origine a un indirizzo IP specifico e consente ad Amazon Keyspaces di ripristinare una tabella per conto del principale.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "CassandraAccessForCustomIp",
 "Effect": "Allow",
 "Action": "cassandra:*",
 "Resource": "*",
 "Condition": {
 "Bool": {
 "aws:ViaAWSService": "false"
 },
 "ForAnyValue:IpAddress": {
 "aws:SourceIp": [
 "123.45.167.89"
]
 }
 }
 },
 {
 "Sid": "CassandraAccessForAwsService",
 "Effect": "Allow",
 "Action": "cassandra:*",
 "Resource": "*",
 "Condition": {
 "Bool": {
 "aws:ViaAWSService": "true"
 }
 }
 }
]
}
```

Per un esempio di politica che utilizza la chiave `aws:ViaAWSService` global condition, vedi. [the section called “Policy degli endpoint VPC e ripristino di Amazon point-in-time Keyspaces \(PITR\)”](#)

## Configurare PITR per una tabella in Amazon Keyspaces

Puoi configurare una tabella in Amazon Keyspaces per le operazioni di backup e ripristino utilizzando PITR con la console, CQL e AWS CLI

Quando si crea una nuova tabella utilizzando CQL o il AWS CLI, è necessario abilitare esplicitamente Pitr nell'istruzione create table. Quando si crea una nuova tabella utilizzando la console, Pitr sarà abilitato per impostazione predefinita.

Per informazioni su come ripristinare una tabella, consulta [the section called “Ripristina una tabella in un determinato momento”](#).

## Console

Configurare Pitr per una tabella utilizzando la console

1. [Accedi a e apri Console di gestione AWS la console Amazon Keyspaces a casahttps://console.aws.amazon.com/keyspaces/](https://console.aws.amazon.com/keyspaces/).
2. Nel pannello di navigazione, scegli Tabelle e seleziona la tabella che desideri modificare.
3. Nella scheda Backup, scegli Modifica.
4. Nella sezione Modifica impostazioni point-in-time di ripristino, seleziona Abilita Point-in-time ripristino.
5. Scegli Save changes (Salva modifiche).

## Cassandra Query Language (CQL)

Configura Pitr per una tabella utilizzando CQL

1. È possibile gestire le impostazioni Pitr per le tabelle utilizzando la `point_in_time_recovery` proprietà personalizzata.

Per abilitare Pitr durante la creazione di una nuova tabella, è necessario impostare lo stato su `point_in_time_recovery enabled` È possibile utilizzare il seguente comando CQL come esempio.

```
CREATE TABLE "my_keyspace1"."my_table1"(
 "id" int,
 "name" ascii,
 "date" timestamp,
 PRIMARY KEY("id"))
WITH CUSTOM_PROPERTIES = {
 'capacity_mode':{'throughput_mode':'PAY_PER_REQUEST'},
 'point_in_time_recovery':{'status':'enabled'}
}
```

 Note

Se non viene specificata alcuna point-in-time proprietà personalizzata di point-in-time ripristino, il ripristino è disabilitato per impostazione predefinita.

2. Per abilitare Pitr per una tabella esistente utilizzando CQL, esegui il seguente comando CQL.

```
ALTER TABLE mykeyspace.mytable
WITH custom_properties = {'point_in_time_recovery': {'status': 'enabled'}}
```

## CLI

Configurare Pitr per una tabella utilizzando il AWS CLI

1. È possibile gestire le impostazioni Pitr per le tabelle utilizzando l'UpdateTableAPI.

Per abilitare Pitr durante la creazione di una nuova tabella, è necessario includere `point-in-time-recovery 'status=ENABLED'` nel comando di creazione della tabella. È possibile utilizzare il AWS CLI comando seguente come esempio. Il comando è stato suddiviso in righe separate per migliorare la leggibilità.

```
aws keyspaces create-table --keyspace-name 'myKeyspace' --table-name 'myTable'
 --schema-definition 'allColumns=[{name=id,type=int},
{name=name,type=text},{name=date,type=timestamp}],partitionKeys=[{name=id}]'
 --point-in-time-recovery 'status=ENABLED'
```

 Note

Se non viene specificato alcun valore di point-in-time ripristino, point-in-time il ripristino è disabilitato per impostazione predefinita.

2. Per confermare l'impostazione di point-in-time ripristino per una tabella, è possibile utilizzare il seguente AWS CLI comando.

```
aws keyspaces get-table --keyspace-name 'myKeyspace' --table-name 'myTable'
```

3. Per abilitare PITR per una tabella esistente utilizzando il AWS CLI, esegui il comando seguente.

```
aws keyspaces update-table --keyspace-name 'myKeyspace' --table-name 'myTable'
--point-in-time-recovery 'status=ENABLED'
```

## Disattiva PITR per una tabella Amazon Keyspaces

Puoi disattivare PITR per una tabella Amazon Keyspaces in qualsiasi momento utilizzando la console, CQL o il AWS CLI

### Important

La disabilitazione di PITR elimina immediatamente la cronologia di backup, anche se riattivi PITR sul tavolo entro 35 giorni.

Per informazioni su come ripristinare una tabella, consulta [the section called “Ripristina una tabella in un determinato momento”](#)

### Console

Disabilitare PITR per una tabella utilizzando la console

1. [Accedi a e apri Console di gestione AWS la console Amazon Keyspaces a casahttps://console.aws.amazon.com/keyspaces/.](https://console.aws.amazon.com/keyspaces/)
2. Nel pannello di navigazione, scegli Tabelle e seleziona la tabella che desideri modificare.
3. Nella scheda Backup, scegli Modifica.
4. Nella sezione Modifica impostazioni point-in-time di ripristino, deseleziona la casella di controllo Abilita Point-in-time ripristino.
5. Scegli Save changes (Salva modifiche).

### Cassandra Query Language (CQL)

Disabilita PITR per una tabella utilizzando CQL

- Per disabilitare PITR per una tabella esistente, esegui il seguente comando CQL.

```
ALTER TABLE mykeyspace.mytable
WITH custom_properties = {'point_in_time_recovery': {'status': 'disabled'}}
```

## CLI

Disattivate Pitr per una tabella utilizzando il AWS CLI

- Per disabilitare Pitr per una tabella esistente, eseguite il comando seguente AWS CLI .

```
aws keyspaces update-table --keyspace-name 'myKeyspace' --table-name 'myTable'
--point-in-time-recovery 'status=DISABLED'
```

## Ripristina una tabella dal backup a un momento temporale specificato in Amazon Keyspaces

La sezione seguente mostra come ripristinare una tabella Amazon Keyspaces esistente in un momento specifico.

### Note

Questa procedura presuppone che la tabella che stai utilizzando sia stata configurata con il ripristino. point-in-time Per abilitare Pitr per una tabella, vedere. [the section called “Configura Pitr”](#)

### Important

Mentre è in corso un ripristino, non modificare o eliminare le policy AWS Identity and Access Management (IAM) che concedono al principale IAM (ad esempio, utente, gruppo o ruolo) l'autorizzazione a eseguire il ripristino. Diversamente, si potrebbero verificare comportamenti imprevisti. Ad esempio, se rimuovi i permessi di scrittura per una tabella durante il ripristino della tabella, l'`RestoreTableToPointInTime` operazione sottostante non può scrivere nessuno dei dati ripristinati nella tabella.

È possibile modificare o eliminare le autorizzazioni solo dopo il completamento dell'operazione di ripristino.



## Console

Ripristina una tabella in un momento specifico utilizzando la console

1. [Accedi a e apri Console di gestione AWS la console Amazon Keyspaces a casa](https://console.aws.amazon.com/keyspaces/)<https://console.aws.amazon.com/keyspaces/>.
2. Nel riquadro di navigazione sul lato sinistro della console scegli Tables (Tabelle).
3. Nell'elenco delle tabelle, scegli la tabella che desideri ripristinare.
4. Nella scheda Backup della tabella, nella sezione Point-in-time Ripristino, scegli Ripristina.
5. Per il nuovo nome della tabella, inserisci un nuovo nome per la tabella ripristinata, ad esempio **mytable\_restored**.
6. Per definire il momento dell'operazione di ripristino, puoi scegliere tra due opzioni:
  - Seleziona l'ora più vicina preconfigurata.
  - Seleziona Specificare data e ora e inserisci la data e l'ora in cui desideri ripristinare la nuova tabella.

### Note

È possibile eseguire il ripristino in qualsiasi momento compreso tra la prima ora e l'ora corrente. Amazon Keyspaces ripristina i dati della tabella allo stato in base alla data e all'ora selezionate (giorno:ora:minuto:secondo).

7. Scegli Ripristina per avviare il processo di ripristino.

La tabella in fase di ripristino è visualizzata con lo stato Restoring (Ripristino). Al termine del processo di ripristino, lo stato della tabella ripristinata diventa Attivo.

## Cassandra Query Language (CQL)

Ripristina una tabella in un determinato momento utilizzando CQL

1. È possibile ripristinare una tabella attiva in un intervallo point-in-time compreso tra l'ora corrente `earliest_restorable_timestamp` e quella corrente. L'ora corrente è l'impostazione predefinita.

Per confermare che point-in-time il ripristino è abilitato per la tabella, interroga il `system_schema_mcs.tables` file come mostrato in questo esempio.

```
SELECT custom_properties
FROM system_schema_mcs.tables
WHERE keyspace_name = 'mykeyspace' AND table_name = 'mytable';
```

Point-in-time il ripristino è abilitato come illustrato nell'output di esempio seguente.

```
custom_properties

{
 ...,
 "point_in_time_recovery": {
 "earliest_restorable_timestamp": "2020-06-30T19:19:21.175Z"
 "status": "enabled"
 }
}
```

2.
  - Ripristina la tabella all'ora corrente. Quando si omette la `WITH restore_timestamp = ...` clausola, viene utilizzato il timestamp corrente.

```
RESTORE TABLE mykeyspace.mytable_restored
FROM TABLE mykeyspace.mytable;
```

- È inoltre possibile ripristinare un momento specifico, definito da un `restore_timestamp` formato ISO 8601. Puoi specificare un momento qualsiasi negli ultimi 35 giorni. Ad esempio, il comando seguente ripristina la tabella a `EarliestRestorableDateTime`.

```
RESTORE TABLE mykeyspace.mytable_restored
FROM TABLE mykeyspace.mytable
WITH restore_timestamp = '2020-06-30T19:19:21.175Z';
```

Per una descrizione completa della sintassi, vedere [the section called “RIPRISTINARE LA TABELLA”](#) nella guida di riferimento del linguaggio.

3. Per verificare che il ripristino della tabella sia andato a buon fine, interrogate il `system_schema_mcs.tables` per confermare lo stato della tabella.

```
SELECT status
FROM system_schema_mcs.tables
```

```
WHERE keyspace_name = 'mykeyspace' AND table_name = 'mytable_restored'
```

L'interrogazione mostra il seguente risultato.

```
status

RESTORING
```

La tabella in fase di ripristino è visualizzata con lo stato Restoring (Ripristino). Al termine del processo di ripristino, lo stato della tabella diventa Attivo.

## CLI

Ripristina una tabella in un determinato momento utilizzando il AWS CLI

1. Crea una tabella semplice denominata `myTable` con PITR abilitato. Il comando è stato suddiviso in righe separate per motivi di leggibilità.

```
aws keyspaces create-table --keyspace-name 'myKeyspace' --table-name 'myTable'
 --schema-definition 'allColumns=[{name=id,type=int},
{name=name,type=text},{name=date,type=timestamp}],partitionKeys=[{name=id}]'
 --point-in-time-recovery 'status=ENABLED'
```

2. Confermate le proprietà della nuova tabella ed esaminatela `earliestRestorableTimestamp` per PITR.

```
aws keyspaces get-table --keyspace-name 'myKeyspace' --table-name 'myTable'
```

L'output di questo comando restituisce quanto segue.

```
{
 "keyspaceName": "myKeyspace",
 "tableName": "myTable",
 "resourceArn": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/
myKeyspace/table/myTable",
 "creationTimestamp": "2022-06-20T14:34:57.049000-07:00",
 "status": "ACTIVE",
 "schemaDefinition": {
 "allColumns": [
 {
```

```

 "name": "id",
 "type": "int"
 },
 {
 "name": "date",
 "type": "timestamp"
 },
 {
 "name": "name",
 "type": "text"
 }
],
 "partitionKeys": [
 {
 "name": "id"
 }
],
 "clusteringKeys": [],
 "staticColumns": []
 },
 "capacitySpecification": {
 "throughputMode": "PAY_PER_REQUEST",
 "lastUpdateToPayPerRequestTimestamp": "2022-06-20T14:34:57.049000-07:00"
 },
 "encryptionSpecification": {
 "type": "AWS_OWNED_KMS_KEY"
 },
 "pointInTimeRecovery": {
 "status": "ENABLED",
 "earliestRestorableTimestamp": "2022-06-20T14:35:13.693000-07:00"
 },
 "defaultTimeToLive": 0,
 "comment": {
 "message": ""
 }
}

```

3. • Per ripristinare una tabella in un determinato momento, specificate a `restore_timestamp` in formato ISO 8601. Puoi scegliere qualsiasi momento negli ultimi 35 giorni a intervalli di un secondo. Ad esempio, il comando seguente ripristina la tabella a `EarliestRestorableDateTime`.

```
aws keyspaces restore-table --source-keyspace-name 'myKeyspace' --source-table-name 'myTable' --target-keyspace-name 'myKeyspace' --target-table-name 'myTable_restored' --restore-timestamp "2022-06-20 21:35:14.693"
```

L'output di questo comando restituisce l'ARN della tabella ripristinata.

```
{
 "restoredTableARN": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/myKeyspace/table/myTable_restored"
}
```

- Per ripristinare la tabella all'ora corrente, è possibile omettere il `restore-timestamp` parametro.

```
aws keyspaces restore-table --source-keyspace-name 'myKeyspace' --source-table-name 'myTable' --target-keyspace-name 'myKeyspace' --target-table-name 'myTable_restored1'"
```

## Ripristina una tabella eliminata utilizzando Amazon Keyspaces PITR

La procedura seguente mostra come ripristinare una tabella eliminata dal backup al momento dell'eliminazione. È possibile eseguire questa operazione utilizzando CQL o il AWS CLI.

### Note

Questa procedura presuppone che PITR sia stato abilitato nella tabella eliminata.

## Cassandra Query Language (CQL)

### Ripristina una tabella eliminata utilizzando CQL

1. Per confermare che point-in-time il ripristino è abilitato per una tabella eliminata, interroga la tabella di sistema. Vengono mostrate solo le tabelle con point-in-time il ripristino abilitato.

```
SELECT custom_properties
FROM system_schema_mcs.tables_history
```

```
WHERE keyspace_name = 'mykeyspace' AND table_name = 'my_table';
```

L'interrogazione mostra il seguente risultato.

```
custom_properties

{
 ...,
 "point_in_time_recovery":{
 "restorable_until_time":"2020-08-04T00:48:58.381Z",
 "status":"enabled"
 }
}
```

2. Ripristina la tabella al momento dell'eliminazione con la seguente istruzione di esempio.

```
RESTORE TABLE mykeyspace.mytable_restored
FROM TABLE mykeyspace.mytable;
```

## CLI

Ripristina una tabella eliminata utilizzando il AWS CLI

1. Eliminare una tabella creata in precedenza con PITR abilitato. Il comando seguente è un esempio.

```
aws keyspaces delete-table --keyspace-name 'myKeyspace' --table-name 'myTable'
```

2. Ripristina la tabella eliminata al momento dell'eliminazione con il seguente comando.

```
aws keyspaces restore-table --source-keyspace-name 'myKeyspace' --source-
table-name 'myTable' --target-keyspace-name 'myKeyspace' --target-table-name
'myTable_restored2'
```

L'output di questo comando restituisce l'ARN della tabella ripristinata.

```
{
 "restoredTableARN": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/
myKeyspace/table/myTable_restored2"
}
```

# Fai scadere i dati con Time to Live (TTL) per Amazon Keyspaces (per Apache Cassandra)

Amazon Keyspaces (per Apache Cassandra) Time to Live (TTL) ti aiuta a semplificare la logica delle applicazioni e ottimizzare il prezzo dello storage facendo scadere automaticamente i dati dalle tabelle. I dati che non ti servono più vengono eliminati automaticamente dalla tabella in base al valore Time to Live che hai impostato.

Ciò semplifica la conformità alle politiche di conservazione dei dati basate su requisiti aziendali, di settore o normativi che definiscono per quanto tempo devono essere conservati i dati o specificano quando i dati devono essere eliminati.

Ad esempio, puoi utilizzare il TTL in un' AdTech applicazione per pianificare quando i dati di annunci specifici scadono e non sono più visibili ai clienti. Puoi anche utilizzare TTL per ritirare automaticamente i dati più vecchi e risparmiare sui costi di archiviazione.

È possibile impostare un valore TTL predefinito per l'intera tabella e sovrascrivere tale valore per singole righe e colonne. Le operazioni TTL non influiscono sulle prestazioni dell'applicazione. Inoltre, il numero di righe e colonne contrassegnate per scadere con TTL non influisce sulla disponibilità della tabella.

Amazon Keyspaces filtra automaticamente i dati scaduti in modo che i dati scaduti non vengano restituiti nei risultati delle query o non siano disponibili per l'uso nelle istruzioni DML (Data Manipulation Language). Amazon Keyspaces in genere elimina i dati scaduti dallo storage entro 10 giorni dalla data di scadenza.

In rari casi, Amazon Keyspaces potrebbe non essere in grado di eliminare i dati entro 10 giorni se c'è un'attività sostenuta sulla partizione di storage sottostante per proteggere la disponibilità. In questi casi, Amazon Keyspaces continua a tentare di eliminare i dati scaduti una volta che il traffico sulla partizione diminuisce.

Una volta eliminati definitivamente i dati dallo storage, smetti di incorrere in costi di archiviazione.

È possibile impostare, modificare o disabilitare le impostazioni TTL predefinite per tabelle nuove ed esistenti utilizzando la console, Cassandra Query Language (CQL) o. AWS CLI

Nelle tabelle con TTL predefinito configurato, è possibile utilizzare le istruzioni CQL per sovrascrivere le impostazioni TTL predefinite della tabella e applicare valori TTL personalizzati a righe e colonne. Per ulteriori informazioni, consultare [the section called “Si usa INSERT per impostare un TTL”](#)

[personalizzato per le nuove righe](#) e [the section called “Si usa UPDATE per impostare un TTL personalizzato per righe e colonne”](#).

I prezzi TTL si basano sulla dimensione delle righe che vengono eliminate o aggiornate utilizzando Time to Live. Le operazioni TTL vengono misurate in unità di. TTL `deletes` Viene utilizzata un'eliminazione TTL per KB di dati per riga eliminata o aggiornata.

Ad esempio, per aggiornare una riga che memorizza 2,5 KB di dati ed eliminare contemporaneamente una o più colonne all'interno della riga sono necessarie tre eliminazioni TTL. In alternativa, per eliminare un'intera riga che contiene 3,5 KB di dati sono necessarie quattro eliminazioni TTL.

Viene utilizzata un'eliminazione TTL per KB di dati eliminati per riga. Per ulteriori informazioni sui prezzi, consulta i prezzi di [Amazon Keyspaces \(per Apache Cassandra\)](#).

#### Argomenti

- [Amazon Keyspaces Time to Live e integrazione con i servizi AWS](#)
- [Crea una nuova tabella con le impostazioni Time to Live \(TTL\) predefinite](#)
- [Aggiorna il valore Time to Live \(TTL\) predefinito di una tabella](#)
- [Crea una tabella con le impostazioni Time to Live \(TTL\) personalizzate abilitate](#)
- [Aggiorna la tabella con Time to Live \(TTL\) personalizzato](#)
- [Utilizzate l'INSERTistruzione per impostare valori Time to Live \(TTL\) personalizzati per le nuove righe](#)
- [Utilizzate l'UPDATEistruzione per modificare le impostazioni Time to Live \(TTL\) personalizzate per righe e colonne](#)

## Amazon Keyspaces Time to Live e integrazione con i servizi AWS

La seguente metrica TTL è disponibile in Amazon CloudWatch per consentire il monitoraggio continuo.

- `TTLDeletes`— Le unità utilizzate per eliminare o aggiornare i dati di seguito utilizzando Time to Live (TTL).

Per ulteriori informazioni su come monitorare le CloudWatch metriche, consulta. [the section called “Monitoraggio con CloudWatch”](#)



Quando lo usi CloudFormation, puoi attivare il TTL durante la creazione di una tabella Amazon Keyspaces. Per ulteriori informazioni, consulta la [AWS CloudFormation Guida per l'utente di](#) .

## Crea una nuova tabella con le impostazioni Time to Live (TTL) predefinite

In Amazon Keyspaces, puoi impostare un valore TTL predefinito per tutte le righe di una tabella al momento della creazione della tabella.

Il valore TTL predefinito per una tabella è zero, il che significa che i dati non scadono automaticamente. Se il valore TTL predefinito per una tabella è maggiore di zero, viene aggiunto un timestamp di scadenza a ogni riga.

I valori TTL sono impostati in secondi e il valore massimo configurabile è 630.720.000 secondi, che equivale a 20 anni.

Dopo la creazione della tabella, è possibile sovrascrivere l'impostazione TTL predefinita della tabella per righe o colonne specifiche con istruzioni CQL DML. Per ulteriori informazioni, consultare [the section called “Si usa INSERT per impostare un TTL personalizzato per le nuove righe”](#) e [the section called “Si usa UPDATE per impostare un TTL personalizzato per righe e colonne”](#).

Quando abiliti il TTL su una tabella, Amazon Keyspaces inizia a memorizzare metadati aggiuntivi relativi a TTL per ogni riga. Inoltre, TTL utilizza i timestamp di scadenza per tenere traccia della scadenza di righe o colonne. I timestamp vengono archiviati come metadati di riga e contribuiscono al costo di archiviazione della riga.

Dopo aver abilitato la funzionalità TTL, non è possibile disabilitarla per una tabella. L'impostazione della tabella `default_time_to_live` su 0 disabilita i tempi di scadenza predefiniti per i nuovi dati, ma non disattiva la funzionalità TTL né ripristina la tabella ai metadati di archiviazione o al comportamento di scrittura originali di Amazon Keyspaces.

Gli esempi seguenti mostrano come creare una nuova tabella con un valore TTL predefinito.

### Console

Crea una nuova tabella con un valore predefinito Time to Live utilizzando la console.

1. [Accedi a e apri Console di gestione AWS la console Amazon Keyspaces a casahttps://console.aws.amazon.com/keyspaces/](https://console.aws.amazon.com/keyspaces/).
2. Nel pannello di navigazione, scegli Tabelle, quindi seleziona Crea tabella.

3. Nella pagina Crea tabella, nella sezione Dettagli della tabella, seleziona un keyspace e fornisci un nome per la nuova tabella.
4. Nella sezione Schema, crea lo schema per la tua tabella.
5. Nella sezione Impostazioni tabella, scegli Personalizza impostazioni.
6. Continua su Time to Live (TTL).

In questo passaggio, si selezionano le impostazioni TTL predefinite per la tabella.

Per il periodo TTL predefinito, inserisci l'ora di scadenza e scegli l'unità di tempo che hai inserito, ad esempio secondi, giorni o anni. Amazon Keyspaces memorizzerà il valore in pochi secondi.

7. Scegliere Create table (Crea tabella). La tabella viene creata con il valore TTL predefinito specificato.

## Cassandra Query Language (CQL)

Crea una nuova tabella con un valore TTL predefinito utilizzando CQL

1. L'istruzione seguente crea una nuova tabella con il valore TTL predefinito impostato su 3.024.000 secondi, che rappresenta 35 giorni.

```
CREATE TABLE my_table (
 userid uuid,
 time timeuuid,
 subject text,
 body text,
 user inet,
 PRIMARY KEY (userid, time)
) WITH default_time_to_live = 3024000;
```

2. Per confermare le impostazioni TTL per la nuova tabella, utilizzate l'`cqlshDESCRIBE`istruzione come illustrato nell'esempio seguente. L'output mostra l'impostazione TTL predefinita per la tabella come `default_time_to_live`

```
DESC TABLE my_table;
```

```
CREATE TABLE my_keyspace.my_table (
 userid uuid,
```

```

 time timeuuid,
 body text,
 subject text,
 user inet,
 PRIMARY KEY (userid, time)
) WITH CLUSTERING ORDER BY (time ASC)
 AND bloom_filter_fp_chance = 0.01
 AND caching = {'class': 'com.amazonaws.cassandra.DefaultCaching'}
 AND comment = ''
 AND compaction = {'class': 'com.amazonaws.cassandra.DefaultCompaction'}
 AND compression = {'class': 'com.amazonaws.cassandra.DefaultCompression'}
 AND crc_check_chance = 1.0
 AND dclocal_read_repair_chance = 0.0
 AND default_time_to_live = 3024000
 AND gc_grace_seconds = 7776000
 AND max_index_interval = 2048
 AND memtable_flush_period_in_ms = 3600000
 AND min_index_interval = 128
 AND read_repair_chance = 0.0
 AND speculative_retry = '99PERCENTILE';

```

## CLI

Creare una nuova tabella con un valore TTL predefinito utilizzando il AWS CLI

1. È possibile utilizzare il comando seguente per creare una nuova tabella con il valore TTL predefinito impostato su un anno.

```

aws keyspaces create-table --keyspace-name 'myKeyspace' --table-name 'myTable' \
 --schema-definition 'allColumns=[{name=id,type=int},
 {name=name,type=text},{name=date,type=timestamp}],partitionKeys=[{name=id}]' \
 --default-time-to-live '31536000'

```

2. Per confermare lo stato TTL della tabella, è possibile utilizzare il seguente comando.

```

aws keyspaces get-table --keyspace-name 'myKeyspace' --table-name 'myTable'

```

L'output del comando è simile a quello dell'esempio seguente

```

{
 "keyspaceName": "myKeyspace",

```

```

 "tableName": "myTable",
 "resourceArn": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/
myKeyspace/table/myTable",
 "creationTimestamp": "2024-09-02T10:52:22.190000+00:00",
 "status": "ACTIVE",
 "schemaDefinition": {
 "allColumns": [
 {
 "name": "id",
 "type": "int"
 },
 {
 "name": "date",
 "type": "timestamp"
 },
 {
 "name": "name",
 "type": "text"
 }
],
 "partitionKeys": [
 {
 "name": "id"
 }
],
 "clusteringKeys": [],
 "staticColumns": []
 },
 "capacitySpecification": {
 "throughputMode": "PAY_PER_REQUEST",
 "lastUpdateToPayPerRequestTimestamp": "2024-09-02T10:52:22.190000+00:00"
 },
 "encryptionSpecification": {
 "type": "AWS_OWNED_KMS_KEY"
 },
 "pointInTimeRecovery": {
 "status": "DISABLED"
 },
 "ttl": {
 "status": "ENABLED"
 },
 "defaultTimeToLive": 31536000,
 "comment": {
 "message": ""
 }
 }

```

```
 },
 "replicaSpecifications": []
}
```

## Aggiorna il valore Time to Live (TTL) predefinito di una tabella

È possibile aggiornare una tabella esistente con un nuovo valore TTL predefinito. I valori TTL sono impostati in secondi e il valore massimo configurabile è 630.720.000 secondi, che equivale a 20 anni.

Quando abiliti il TTL su una tabella, Amazon Keyspaces inizia a memorizzare metadati aggiuntivi relativi a TTL per ogni riga. Inoltre, TTL utilizza i timestamp di scadenza per tenere traccia della scadenza di righe o colonne. I timestamp vengono archiviati come metadati di riga e contribuiscono al costo di archiviazione della riga.

Dopo aver abilitato il TTL per una tabella, è possibile sovrascrivere l'impostazione TTL predefinita della tabella per righe o colonne specifiche con istruzioni CQL DML. Per ulteriori informazioni, consultare [the section called “Si usa INSERT per impostare un TTL personalizzato per le nuove righe”](#) e [the section called “Si usa UPDATE per impostare un TTL personalizzato per righe e colonne”](#).

Dopo aver abilitato la funzionalità TTL, non è possibile disabilitarla per una tabella. L'impostazione della tabella `default_time_to_live` su 0 disabilita i tempi di scadenza predefiniti per i nuovi dati, ma non disattiva la funzionalità TTL né ripristina la tabella ai metadati di archiviazione o al comportamento di scrittura originali di Amazon Keyspaces.

Segui questi passaggi per aggiornare le impostazioni Time to Live predefinite per le tabelle esistenti utilizzando la console, CQL o. AWS CLI

### Console

Aggiorna il valore TTL predefinito di una tabella utilizzando la console

1. [Accedi a e apri Console di gestione AWS la console Amazon Keyspaces a casa](https://console.aws.amazon.com/keyspaces/)<https://console.aws.amazon.com/keyspaces/>.
2. Scegli la tabella che desideri aggiornare, quindi scegli la scheda Impostazioni aggiuntive.
3. Continua su Time to Live (TTL) e scegli Modifica.
4. Per il periodo TTL predefinito, inserisci l'ora di scadenza e scegli l'unità di tempo, ad esempio secondi, giorni o anni. Amazon Keyspaces memorizzerà il valore in pochi secondi. Ciò non modifica il valore TTL delle righe esistenti.

5. Una volta definite le impostazioni TTL, scegli Salva modifiche.

## Cassandra Query Language (CQL)

Aggiorna il valore TTL predefinito di una tabella utilizzando CQL

1. È possibile utilizzare ALTER TABLE per modificare le impostazioni Time to Live (TTL) predefinite di una tabella. Per aggiornare le impostazioni TTL predefinite della tabella a 2.592.000 secondi, che rappresentano 30 giorni, puoi utilizzare la seguente istruzione.

```
ALTER TABLE my_table WITH default_time_to_live = 2592000;
```

2. Per confermare le impostazioni TTL per la tabella aggiornata, utilizzate l'`cqlshDESCRIBE`istruzione come illustrato nell'esempio seguente. L'output mostra l'impostazione TTL predefinita per la tabella come `default_time_to_live`

```
DESC TABLE my_table;
```

L'output dell'istruzione dovrebbe essere simile a quello di questo esempio.

```
CREATE TABLE my_keyspace.my_table (
 id int PRIMARY KEY,
 date timestamp,
 name text
) WITH bloom_filter_fp_chance = 0.01
 AND caching = {'class': 'com.amazonaws.cassandra.DefaultCaching'}
 AND comment = ''
 AND compaction = {'class': 'com.amazonaws.cassandra.DefaultCompaction'}
 AND compression = {'class': 'com.amazonaws.cassandra.DefaultCompression'}
 AND crc_check_chance = 1.0
 AND dclocal_read_repair_chance = 0.0
 AND default_time_to_live = 2592000
 AND gc_grace_seconds = 7776000
 AND max_index_interval = 2048
 AND memtable_flush_period_in_ms = 3600000
 AND min_index_interval = 128
 AND read_repair_chance = 0.0
 AND speculative_retry = '99PERCENTILE';
```

## CLI

Aggiornate il valore TTL predefinito di una tabella utilizzando il AWS CLI

1. È possibile utilizzare `update-table` una tabella per modificare il valore TTL predefinito. Per aggiornare le impostazioni TTL predefinite della tabella a 2.592.000 secondi, che rappresentano 30 giorni, puoi utilizzare la seguente istruzione.

```
aws keyspaces update-table --keyspace-name 'myKeyspace' --table-name 'myTable'
--default-time-to-live '2592000'
```

2. Per confermare il valore TTL predefinito aggiornato, puoi utilizzare la seguente istruzione.

```
aws keyspaces get-table --keyspace-name 'myKeyspace' --table-name 'myTable'
```

L'output dell'istruzione dovrebbe essere simile a quello dell'esempio seguente.

```
{
 "keyspaceName": "myKeyspace",
 "tableName": "myTable",
 "resourceArn": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/
myKeyspace/table/myTable",
 "creationTimestamp": "2024-09-02T10:52:22.190000+00:00",
 "status": "ACTIVE",
 "schemaDefinition": {
 "allColumns": [
 {
 "name": "id",
 "type": "int"
 },
 {
 "name": "date",
 "type": "timestamp"
 },
 {
 "name": "name",
 "type": "text"
 }
],
 "partitionKeys": [
 {
 "name": "id"
 }
]
 }
}
```

```
 }
],
 "clusteringKeys": [],
 "staticColumns": []
},
"capacitySpecification": {
 "throughputMode": "PAY_PER_REQUEST",
 "lastUpdateToPayPerRequestTimestamp": "2024-09-02T10:52:22.190000+00:00"
},
"encryptionSpecification": {
 "type": "AWS_OWNED_KMS_KEY"
},
"pointInTimeRecovery": {
 "status": "DISABLED"
},
"ttl": {
 "status": "ENABLED"
},
"defaultTimeToLive": 2592000,
"comment": {
 "message": ""
},
"replicaSpecifications": []
}
```

## Crea una tabella con le impostazioni Time to Live (TTL) personalizzate abilitate

Per creare una nuova tabella con impostazioni personalizzate di Time to Live che possono essere applicate a righe e colonne senza abilitare le impostazioni TTL predefinite per l'intera tabella, puoi utilizzare i seguenti comandi.

### Note

Se viene creata una tabella con le impostazioni `ttl` personalizzate abilitate, non è possibile disabilitare l'impostazione in un secondo momento.



## Cassandra Query Language (CQL)

Crea una nuova tabella con impostazioni TTL personalizzate utilizzando CQL

- ```
CREATE TABLE my_keyspace.my_table (id int primary key) WITH  
CUSTOM_PROPERTIES={'ttl':{'status': 'enabled'}};
```

CLI

Crea una nuova tabella con impostazioni TTL personalizzate utilizzando il AWS CLI

1. È possibile utilizzare il seguente comando per creare una nuova tabella con TTL abilitato.

```
aws keyspaces create-table --keyspace-name 'myKeyspace' --table-name 'myTable' \  
                           --schema-definition  
'allColumns=[{name=id,type=int},{name=name,type=text},  
{name=date,type=timestamp}],partitionKeys=[{name=id}]]' \  
                           --ttl 'status=ENABLED'
```

2. Per confermare che il TTL è abilitato per la tabella, puoi usare la seguente istruzione.

```
aws keyspaces get-table --keyspace-name 'myKeyspace' --table-name 'myTable'
```

L'output dell'istruzione dovrebbe essere simile a quello dell'esempio seguente.

```
{  
  "keyspaceName": "myKeyspace",  
  "tableName": "myTable",  
  "resourceArn": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/  
myKeyspace/table/myTable",  
  "creationTimestamp": "2024-09-02T10:52:22.190000+00:00",  
  "status": "ACTIVE",  
  "schemaDefinition": {  
    "allColumns": [  
      {  
        "name": "id",  
        "type": "int"  
      },  
      {  
        "name": "date",  
        "type": "timestamp"  
      }  
    ]  
  }  
}
```

```

        },
        {
            "name": "name",
            "type": "text"
        }
    ],
    "partitionKeys": [
        {
            "name": "id"
        }
    ],
    "clusteringKeys": [],
    "staticColumns": []
},
"capacitySpecification": {
    "throughputMode": "PAY_PER_REQUEST",
    "lastUpdateToPayPerRequestTimestamp": "2024-09-02T11:18:55.796000+00:00"
},
"encryptionSpecification": {
    "type": "AWS_OWNED_KMS_KEY"
},
"pointInTimeRecovery": {
    "status": "DISABLED"
},
"ttl": {
    "status": "ENABLED"
},
"defaultTimeToLive": 0,
"comment": {
    "message": ""
},
"replicaSpecifications": []
}

```

Aggiorna la tabella con Time to Live (TTL) personalizzato

Per abilitare le impostazioni personalizzate Time to Live per una tabella in modo che i valori TTL possano essere applicati a singole righe e colonne senza impostare un valore TTL predefinito per l'intera tabella, puoi utilizzare i seguenti comandi.

 Note

Dopo `ttl` è abilitato, non è possibile disabilitarlo per la tabella.

Cassandra Query Language (CQL)

Abilita le impostazioni TTL personalizzate per una tabella utilizzando CQL

- ```
ALTER TABLE my_table WITH CUSTOM_PROPERTIES={'ttl':{'status': 'enabled'}};
```

## CLI

Abilita le impostazioni TTL personalizzate per una tabella utilizzando il AWS CLI

- È possibile utilizzare il comando seguente per aggiornare l'impostazione TTL personalizzata di una tabella.

```
aws keyspaces update-table --keyspace-name 'myKeyspace' --table-name 'myTable'
--ttl 'status=ENABLED'
```

- Per confermare che il TTL è ora abilitato per la tabella, puoi usare la seguente istruzione.

```
aws keyspaces get-table --keyspace-name 'myKeyspace' --table-name 'myTable'
```

L'output dell'istruzione dovrebbe essere simile a quello dell'esempio seguente.

```
{
 "keyspaceName": "myKeyspace",
 "tableName": "myTable",
 "resourceArn": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/
myKeyspace/table/myTable",
 "creationTimestamp": "2024-09-02T11:32:27.349000+00:00",
 "status": "ACTIVE",
 "schemaDefinition": {
 "allColumns": [
 {
 "name": "id",
 "type": "int"
 }
]
 }
}
```

```
 {
 "name": "date",
 "type": "timestamp"
 },
 {
 "name": "name",
 "type": "text"
 }
],
 "partitionKeys": [
 {
 "name": "id"
 }
],
 "clusteringKeys": [],
 "staticColumns": []
},
"capacitySpecification": {
 "throughputMode": "PAY_PER_REQUEST",
 "lastUpdateToPayPerRequestTimestamp": "2024-09-02T11:32:27.349000+00:00"
},
"encryptionSpecification": {
 "type": "AWS_OWNED_KMS_KEY"
},
"pointInTimeRecovery": {
 "status": "DISABLED"
},
"ttl": {
 "status": "ENABLED"
},
"defaultTimeToLive": 0,
"comment": {
 "message": ""
},
"replicaSpecifications": []
}
```

## Utilizzate l'**INSERT**istruzione per impostare valori Time to Live (TTL) personalizzati per le nuove righe

### Note

Prima di poter impostare valori TTL personalizzati per le righe utilizzando l'INSERTistruzione, è necessario abilitare il TTL personalizzato nella tabella. Per ulteriori informazioni, consulta [the section called “Aggiorna il TTL personalizzato della tabella”](#).

Per sovrascrivere il valore TTL predefinito di una tabella impostando le date di scadenza per le singole righe, puoi utilizzare l'istruzione: INSERT

- INSERT— Inserisce una nuova riga di dati con un set di valori TTL.

L'impostazione dei valori TTL per le nuove righe utilizzando l'INSERTistruzione ha la precedenza sull'impostazione TTL predefinita della tabella.

La seguente istruzione CQL inserisce una riga di dati nella tabella e modifica l'impostazione TTL predefinita su 259.200 secondi (che equivalgono a 3 giorni).

```
INSERT INTO my_table (userid, time, subject, body, user)
 VALUES (B79CB3BA-745E-5D9A-8903-4A02327A7E09, 96a29100-5e25-11ec-90d7-
b5d91eceda0a, 'Message', 'Hello', '205.212.123.123')
 USING TTL 259200;
```

Per confermare le impostazioni TTL per la riga inserita, utilizzare la seguente istruzione.

```
SELECT TTL (subject) from my_table;
```

## Utilizzate l'**UPDATE**istruzione per modificare le impostazioni Time to Live (TTL) personalizzate per righe e colonne

### Note

Prima di poter impostare valori TTL personalizzati per righe e colonne, è necessario abilitare il TTL nella tabella. Per ulteriori informazioni, consulta [the section called “Aggiorna il TTL personalizzato della tabella”](#).

Puoi utilizzare l'UPDATEistruzione per sovrascrivere il valore TTL predefinito di una tabella impostando la data di scadenza per singole righe e colonne:

- **Righe:** puoi aggiornare una riga di dati esistente con un valore TTL personalizzato.
- **Colonne:** puoi aggiornare un sottoinsieme di colonne all'interno delle righe esistenti con un valore TTL personalizzato.

L'impostazione dei valori TTL per righe e colonne ha la precedenza sull'impostazione TTL predefinita per la tabella.

Per modificare le impostazioni TTL della colonna «oggetto» inserita in precedenza da 259.200 secondi (3 giorni) a 86.400 secondi (un giorno), utilizzare la seguente istruzione.

```
UPDATE my_table USING TTL 86400 set subject = 'Updated Message' WHERE userid =
B79CB3BA-745E-5D9A-8903-4A02327A7E09 and time = 96a29100-5e25-11ec-90d7-b5d91eceda0a;
```

È possibile eseguire una semplice query di selezione per visualizzare il record aggiornato prima della scadenza.

```
SELECT * from my_table;
```

La query mostra il seguente risultato.

| userid | subject | user  | time  | body  |
|--------|---------|-------|-------|-------|
| -----  | -----   | ----- | ----- | ----- |
| +      | +       | +     | +     | +     |

```
b79cb3ba-745e-5d9a-8903-4a02327a7e09 | 96a29100-5e25-11ec-90d7-b5d91eceda0a | Hello |
Updated Message | 205.212.123.123
50554d6e-29bb-11e5-b345-feff819cdc9f | cf03fb21-59b5-11ec-b371-dff626ab9620 | Hello |
Message | 205.212.123.123
```

Per confermare che la scadenza è avvenuta correttamente, esegui nuovamente la stessa query dopo l'ora di scadenza configurata.

```
SELECT * from my_table;
```

La query mostra il seguente risultato dopo la scadenza della colonna «oggetto».

```
userid | time | body |
subject | user
-----+-----+-----
+-----+-----+-----
b79cb3ba-745e-5d9a-8903-4a02327a7e09 | 96a29100-5e25-11ec-90d7-b5d91eceda0a | Hello |
null | 205.212.123.123
50554d6e-29bb-11e5-b345-feff819cdc9f | cf03fb21-59b5-11ec-b371-dff626ab9620 | Hello |
Message | 205.212.123.123
```

## Utilizzo di questo servizio con un SDK AWS

AWS i kit di sviluppo software (SDKs) 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                                        |
|----------------------------------------|---------------------------------------------------------|
| <a href="#">AWS SDK per C++</a>        | <a href="#">AWS SDK per C++ esempi di codice</a>        |
| <a href="#">AWS CLI</a>                | <a href="#">AWS CLI esempi di codice</a>                |
| <a href="#">AWS SDK per Go</a>         | <a href="#">AWS SDK per Go esempi di codice</a>         |
| <a href="#">AWS SDK per Java</a>       | <a href="#">AWS SDK per Java esempi di codice</a>       |
| <a href="#">AWS SDK per JavaScript</a> | <a href="#">AWS SDK per JavaScript esempi di codice</a> |
| <a href="#">AWS SDK per Kotlin</a>     | <a href="#">AWS SDK per Kotlin esempi di codice</a>     |

| Documentazione sugli SDK                     | Esempi di codice                                              |
|----------------------------------------------|---------------------------------------------------------------|
| <a href="#">AWS SDK per .NET</a>             | <a href="#">AWS SDK per .NET esempi di codice</a>             |
| <a href="#">AWS SDK per PHP</a>              | <a href="#">AWS SDK per PHP esempi di codice</a>              |
| <a href="#">AWS Strumenti per PowerShell</a> | <a href="#">AWS Strumenti per PowerShell esempi di codice</a> |
| <a href="#">AWS SDK per Python (Boto3)</a>   | <a href="#">AWS SDK per Python (Boto3) esempi di codice</a>   |
| <a href="#">AWS SDK per Ruby</a>             | <a href="#">AWS SDK per Ruby esempi di codice</a>             |
| <a href="#">AWS SDK for Rust</a>             | <a href="#">AWS SDK for Rust esempi di codice</a>             |
| <a href="#">SDK AWS per SAP ABAP</a>         | <a href="#">SDK AWS per SAP ABAP esempi di codice</a>         |
| <a href="#">SDK AWS per Swift</a>            | <a href="#">SDK AWS per Swift esempi di codice</a>            |

#### Esempio di disponibilità

Non riesci a trovare quello che ti serve? Richiedi un esempio di codice utilizzando il link [Provide feedback \(Fornisci un feedback\)](#) nella parte inferiore di questa pagina.

## Utilizzo di tag ed etichette per le risorse Amazon Keyspaces

Puoi etichettare le risorse di Amazon Keyspaces (per Apache Cassandra) utilizzando i tag. I tag consentono di classificare le risorse in diversi modi, ad esempio per scopo, proprietario, ambiente o altri criteri. I tag consentono di eseguire le seguenti operazioni:

- identificare rapidamente una risorsa in base ai tag a questa assegnati.
- Visualizza le AWS fatture suddivise per tag.
- Controlla l'accesso alle risorse Amazon Keyspaces in base ai tag. Per esempi di policy IAM che utilizzano i tag, consulta [the section called “Autorizzazione basata sui tag Amazon Keyspaces”](#).

Il tagging è supportato da AWS servizi come Amazon Elastic Compute Cloud (Amazon EC2), Amazon Simple Storage Service (Amazon S3), Amazon Keyspaces e molti altri. Un tagging efficiente



può fornire informazioni dettagliate sui costi abilitando la creazione di report su servizi che recano tag specifici.

Per iniziare a utilizzare il tagging, procedi come segue:

1. comprendere [Restrizioni all'uso dei tag per etichettare le risorse in Amazon Keyspaces](#);
2. creare tag utilizzando [Aggiungi tag a keyspace, tabelle e stream in Amazon Keyspaces](#);
3. Utilizzalo [Crea report di allocazione dei costi utilizzando tag per Amazon Keyspaces](#) per tenere traccia dei AWS costi per tag attivo.

Infine, è buona norma seguire strategie di tagging ottimali. Per informazioni, consulta [Strategie di assegnazione di tag di AWS](#).

## Restrizioni all'uso dei tag per etichettare le risorse in Amazon Keyspaces

Ogni tag consiste di una chiave e di un valore, entrambi personalizzabili. Le restrizioni si applicano come segue:

- Ogni keyspace, tabella o stream di Amazon Keyspaces può avere un solo tag con la stessa chiave. Se provi ad aggiungere un tag esistente (con la stessa chiave), il valore del tag esistente viene caricato al nuovo valore;
- I tag applicati a uno spazio chiave non si applicano automaticamente alle tabelle all'interno di tale spazio chiave. Per applicare lo stesso tag a uno spazio chiave e a tutte le relative tabelle, ogni risorsa deve essere etichettata individualmente.
- I tag applicati a una tabella non si applicano automaticamente al flusso di quella tabella. Per applicare gli stessi tag a una tabella e allo stream durante la creazione della tabella, puoi usare il `PropagateTagsOnEnable` flag quando crei la tabella. Utilizzando questo flag, Amazon Keyspaces applica i tag della tabella allo stream durante la creazione dello stream. Quando lo stream è attivo, le modifiche ai tag della tabella non si applicano allo stream.
- Quando create uno spazio chiave o una tabella multiregione, tutti i tag definiti durante il processo di creazione vengono applicati automaticamente a tutti gli spazi chiave e le tabelle in tutte le regioni. Quando modificate i tag esistenti utilizzando `ALTER KEYSPACE` o `ALTER TABLE`, l'aggiornamento viene applicato solo allo spazio chiave o alla tabella nella regione in cui state apportando la modifica.
- Un valore agisce come descrittore all'interno di una categoria di tag (chiave). In Amazon Keyspaces il valore non può essere vuoto o nullo.

- I valori e le chiavi dei tag rispettano la distinzione tra maiuscole e minuscole.
- La lunghezza massima della chiave è di 128 caratteri Unicode.
- La lunghezza massima del valore è di 256 caratteri Unicode.
- i caratteri consentiti sono lettere, spazi vuoti, numeri e i seguenti caratteri speciali: + - = . \_ : /
- Il numero massimo di tag per risorsa è 50.
- AWS ai nomi e ai valori dei tag assegnati viene assegnato automaticamente il `aws :` prefisso, che non è possibile assegnare. AWS-i nomi dei tag assegnati non vengono conteggiati ai fini del limite di 50 tag. I nomi dei tag assegnati dall'utente hanno il prefisso `user :` nel report di allocazione dei costi;
- Non puoi retrodatare l'applicazione di un tag.

## Aggiungi tag a keyspace, tabelle e stream in Amazon Keyspaces

Puoi aggiungere, elencare, modificare o eliminare tag per keyspace, tabelle e stream utilizzando la console Amazon Keyspaces, il o Cassandra Query Language AWS CLI(CQL). Puoi quindi attivare questi tag definiti dall'utente in modo che vengano visualizzati sulla console per il monitoraggio dell'allocazione dei Gestione dei costi e fatturazione AWS costi. Per ulteriori informazioni, consulta [Crea report di allocazione dei costi utilizzando tag per Amazon Keyspaces](#).

Per la modifica in blocco, puoi anche utilizzare Tag Editor sulla console. Per ulteriori informazioni, consulta [Lavorare con il Tag Editor](#) nella Guida per l'utente di AWS Resource Groups .

Per informazioni sulla struttura dei tag, consulta [Restrizioni all'uso dei tag per etichettare le risorse in Amazon Keyspaces](#).

### Argomenti

- [Aggiungi tag quando crei un nuovo keyspace](#)
- [Aggiungere tag a un keyspace](#)
- [Eliminare i tag da un keyspace](#)
- [Visualizza i tag di un keyspace](#)
- [Aggiungi tag quando crei una nuova tabella](#)
- [Aggiungere tag a una tabella](#)
- [Eliminare tag da una tabella](#)

- [Visualizza i tag di una tabella](#)
- [Aggiungi tag a un nuovo stream durante la creazione di una tabella](#)
- [Aggiungi tag a un nuovo stream per una tabella esistente](#)
- [Aggiungere nuovi tag a uno stream](#)
- [Eliminare i tag da uno stream](#)
- [Visualizza i tag di uno stream](#)

## Aggiungi tag quando crei un nuovo keyspace

Puoi utilizzare la console Amazon Keyspaces, CQL o AWS CLI aggiungere tag quando crei un nuovo keyspace.

### Console

Imposta un tag quando crei un nuovo keyspace utilizzando la console

1. [Accedi a e apri Console di gestione AWS la console Amazon Keyspaces a casahttps://console.aws.amazon.com/keyspaces/.](https://console.aws.amazon.com/keyspaces/)
2. Nel riquadro di navigazione, scegli Keyspaces, quindi scegli Crea keyspace.
3. Nella pagina Crea spazio chiave, fornisci un nome per lo spazio chiave.
4. In Tag scegli Aggiungi nuovo tag e inserisci una chiave e un valore.
5. Scegli Crea spazio chiave.

### Cassandra Query Language (CQL)

Imposta un tag quando crei un nuovo keyspace usando CQL

- L'esempio seguente crea un nuovo keyspace con tag.

```
CREATE KEYSPACE mykeyspace WITH TAGS = {'key1':'val1', 'key2':'val2'};
```

### CLI

Imposta un tag quando crei un nuovo keyspace usando il AWS CLI

- L'istruzione seguente crea un nuovo keyspace con tag.

```
aws keyspaces create-keyspace --keyspace-name 'myKeyspace' --tags
'key=key1,value=val1' 'key=key2,value=val2'
```

## Aggiungere tag a un keyspace

Gli esempi seguenti mostrano come aggiungere tag a un keyspace in Amazon Keyspaces.

### Console

Aggiungi un tag a un keyspace esistente utilizzando la console

1. [Accedi a e apri Console di gestione AWS la console Amazon Keyspaces a casa](https://console.aws.amazon.com/keyspaces/)<https://console.aws.amazon.com/keyspaces/>.
2. Nel pannello di navigazione, scegli Keyspaces.
3. Scegli uno spazio chiave dall'elenco. Quindi scegli la scheda Tag in cui puoi visualizzare i tag del keyspace.
4. Scegli Gestisci tag per aggiungere, modificare o eliminare tag.
5. Scegli Save changes (Salva modifiche).

### Cassandra Query Language (CQL)

Aggiungi un tag a uno spazio chiave esistente usando CQL

- ```
ALTER KEYSPACE mykeyspace ADD TAGS {'key1':'val1', 'key2':'val2'};
```

CLI

Aggiungi un tag a uno spazio chiave esistente usando il AWS CLI

- L'esempio seguente mostra come aggiungere nuovi tag a un keyspace esistente.

```
aws keyspaces tag-resource --resource-arn 'arn:aws:cassandra:us-east-1:111122223333:/keyspace/myKeyspace/' --tags 'key=key3,value=val3'  
'key=key4,value=val4'
```

Eliminare i tag da un keyspace

Console

Elimina un tag da un keyspace esistente utilizzando la console

1. [Accedi a e apri Console di gestione AWS la console Amazon Keyspaces a casa](https://console.aws.amazon.com/keyspaces/)<https://console.aws.amazon.com/keyspaces/>.
2. Nel pannello di navigazione, scegli Keyspaces.
3. Scegli uno spazio chiave dall'elenco. Quindi scegli la scheda Tag in cui puoi visualizzare i tag del keyspace.
4. Scegli Gestisci tag ed elimina i tag che non ti servono più.
5. Scegli Save changes (Salva modifiche).

Cassandra Query Language (CQL)

Elimina un tag da uno spazio chiave esistente utilizzando CQL

- ```
ALTER KEYSPACE mykeyspace DROP TAGS {'key1':'val1', 'key2':'val2'};
```

### CLI

Eliminare un tag da uno spazio chiave esistente utilizzando il AWS CLI

- L'istruzione seguente rimuove i tag specificati da un keyspace.

```
aws keyspaces untag-resource --resource-arn 'arn:aws:cassandra:us-east-1:111122223333:/keyspace/myKeyspace/' --tags 'key=key3,value=val3'
'key=key4,value=val4'
```

## Visualizza i tag di un keyspace

Gli esempi seguenti mostrano come leggere i tag utilizzando la console, CQL o il AWS CLI

## Console

Visualizza i tag di un keyspace utilizzando la console Amazon Keyspaces

1. [Accedi a e apri Console di gestione AWS la console Amazon Keyspaces a casahttps://console.aws.amazon.com/keyspaces/.](https://console.aws.amazon.com/keyspaces/)
2. Nel pannello di navigazione, scegli Keyspaces.
3. Scegli uno spazio chiave dall'elenco. Quindi scegli la scheda Tag in cui puoi visualizzare i tag del keyspace.

## Cassandra Query Language (CQL)

Visualizza i tag di un keyspace usando CQL

Per leggere i tag allegati a un keyspace, utilizzate la seguente istruzione CQL.

```
SELECT * FROM system_schema_mcs.tags WHERE valid_where_clause;
```

La WHERE clausola è obbligatoria e deve utilizzare uno dei seguenti formati:

- `keyspace_name = 'mykeyspace' AND resource_type = 'keyspace'`
- `resource_id = arn`
- L'istruzione seguente mostra se un keyspace ha dei tag.

```
SELECT * FROM system_schema_mcs.tags WHERE keyspace_name = 'mykeyspace' AND
resource_type = 'keyspace';
```

L'output della query è simile al seguente.

```
resource_id | keyspace_name
| resource_name | resource_type | tags

+-----+-----+-----+-----
arn:aws:cassandra:us-east-1:111122223333:/keyspace/mykeyspace/ | mykeyspace
| mykeyspace | keyspace | {'key1': 'val1', 'key2': 'val2'}
```

## CLI

Visualizza i tag di un keyspace usando il AWS CLI

- Questo esempio mostra come elencare i tag della risorsa specificata.

```
aws keyspaces list-tags-for-resource --resource-arn 'arn:aws:cassandra:us-east-1:111122223333:/keyspace/myKeyspace/'
```

L'output dell'ultimo comando è simile al seguente.

```
{
 "tags": [
 {
 "key": "key1",
 "value": "val1"
 },
 {
 "key": "key2",
 "value": "val2"
 },
 {
 "key": "key3",
 "value": "val3"
 },
 {
 "key": "key4",
 "value": "val4"
 }
]
}
```

## Aggiungi tag quando crei una nuova tabella

Puoi utilizzare la console Amazon Keyspaces, CQL o il AWS CLI per aggiungere tag a nuove tabelle quando le crei.

## Console

Aggiungi un tag quando crei una nuova tabella utilizzando (console)

1. [Accedi a e apri Console di gestione AWS la console Amazon Keyspaces a casahttps://console.aws.amazon.com/keyspaces/.](https://console.aws.amazon.com/keyspaces/)
2. Nel pannello di navigazione, scegli Tabelle, quindi seleziona Crea tabella.
3. Nella pagina Crea tabella, nella sezione Dettagli della tabella, seleziona un keyspace e fornisci un nome per la tabella.
4. Nella sezione Schema, crea lo schema per la tua tabella.
5. Nella sezione Impostazioni tabella, scegli Personalizza impostazioni.
6. Vai alla sezione Tag della tabella, opzionale, e scegli Aggiungi nuovo tag per creare nuovi tag.
7. Scegliere Create table (Crea tabella).

## Cassandra Query Language (CQL)

Aggiungi tag quando crei una nuova tabella usando CQL

- L'esempio seguente crea una nuova tabella con tag.

```
CREATE TABLE mytable(...) WITH TAGS = {'key1':'val1', 'key2':'val2'};
```

## CLI

Aggiungi tag quando crei una nuova tabella utilizzando il AWS CLI

- L'esempio seguente mostra come creare una nuova tabella con tag. Il comando crea una tabella MyTable in uno spazio di chiavi MyKeySpace già esistente. Nota che il comando è stato suddiviso in diverse righe per facilitarne la leggibilità.

```
aws keyspaces create-table --keyspace-name 'myKeyspace' --table-name 'myTable'
 --schema-definition 'allColumns=[{name=id,type=int},
{name=name,type=text},{name=date,type=timestamp}],partitionKeys=[{name=id}]'
 --tags 'key=key1,value=val1' 'key=key2,value=val2'
```



## Aggiungere tag a una tabella

Puoi aggiungere tag a una tabella esistente in Amazon Keyspaces utilizzando la console, CQL o il AWS CLI

### Console

Aggiungi tag a una tabella utilizzando la console Amazon Keyspaces

1. [Accedi a e apri Console di gestione AWS la console Amazon Keyspaces a casahttps://console.aws.amazon.com/keyspaces/.](https://console.aws.amazon.com/keyspaces/)
2. Nel pannello di navigazione, seleziona Tabelle.
3. Scegli una tabella dall'elenco e scegli la scheda Tag.
4. Scegli Gestisci tag per aggiungere tag alla tabella.
5. Scegli Save changes (Salva modifiche).

### Cassandra Query Language (CQL)

Aggiungi tag a una tabella usando CQL

- La seguente dichiarazione mostra come aggiungere tag a una tabella esistente.

```
ALTER TABLE mykeyspace.mytable ADD TAGS {'key1':'val1', 'key2':'val2'};
```

### CLI

Aggiungere tag a una tabella utilizzando il AWS CLI

- L'esempio seguente mostra come aggiungere nuovi tag a una tabella esistente.

```
aws keyspaces tag-resource --resource-arn 'arn:aws:cassandra:us-east-1:111122223333:/keyspace/myKeyspace/table/myTable' --tags 'key=key3,value=val3' 'key=key4,value=val4'
```

## Eliminare tag da una tabella

### Console

Eliminare i tag da una tabella utilizzando la console Amazon Keyspaces

1. [Accedi a e apri Console di gestione AWS la console Amazon Keyspaces a casa](https://console.aws.amazon.com/keyspaces/)<https://console.aws.amazon.com/keyspaces/>.
2. Nel pannello di navigazione, seleziona Tabelle.
3. Scegli una tabella dall'elenco e scegli la scheda Tag.
4. Scegli Gestisci tag per eliminare i tag dalla tabella.
5. Scegli Save changes (Salva modifiche).

### Cassandra Query Language (CQL)

Elimina i tag da una tabella usando CQL

- L'istruzione seguente mostra come eliminare i tag da una tabella esistente.

```
ALTER TABLE mytable DROP TAGS {'key3':'val3', 'key4':'val4'};
```

### CLI

Aggiungere tag a una tabella utilizzando il AWS CLI

- L'istruzione seguente rimuove i tag specificati da uno spazio di chiavi.

```
aws keyspaces untag-resource --resource-arn 'arn:aws:cassandra:us-east-1:111122223333:keyspace/myKeyspace/table/myTable' --tags
'key=key3,value=val3' 'key=key4,value=val4'
```

## Visualizza i tag di una tabella

Gli esempi seguenti mostrano come visualizzare i tag di una tabella in Amazon Keyspaces utilizzando la console, CQL o il AWS CLI

## Console

Visualizza i tag di una tabella utilizzando la console

1. [Accedi a e apri Console di gestione AWS la console Amazon Keyspaces a casahttps://console.aws.amazon.com/keyspaces/.](https://console.aws.amazon.com/keyspaces/)
2. Nel pannello di navigazione, seleziona Tabelle.
3. Scegli una tabella dall'elenco e scegli la scheda Tag.

## Cassandra Query Language (CQL)

Visualizza i tag di una tabella usando CQL

Per leggere i tag allegati a una tabella, utilizzate la seguente istruzione CQL.

```
SELECT * FROM system_schema_mcs.tags WHERE valid_where_clause;
```

La WHERE clausola è obbligatoria e deve utilizzare uno dei seguenti formati:

- `keyspace_name = 'mykeyspace' AND resource_name = 'mytable'`
- `resource_id = arn`
- La seguente query restituisce i tag della tabella specificata.

```
SELECT * FROM system_schema_mcs.tags WHERE keyspace_name = 'mykeyspace' AND
resource_name = 'mytable';
```

L'output di tale query è simile al seguente.

```
resource_id |
keyspace_name | resource_name | resource_type | tags

+-----+-----+-----+-----+
arn:aws:cassandra:us-east-1:111122223333:/keyspace/mykeyspace/table/mytable |
mykeyspace | mytable | table | {'key1': 'val1', 'key2':
'val2'}
```

## CLI

Visualizza i tag di una tabella utilizzando il AWS CLI

- Questo esempio mostra come elencare i tag della risorsa specificata.

```
aws keyspaces list-tags-for-resource --resource-arn
'arn:aws:arn:aws:cassandra:us-east-1:111122223333:/keyspace/my_keyspace/
table/my_table/stream/2025-05-11T21:21:33.291cassandra:us-east-1:111122223333:/
keyspace/myKeyspace/table/myTable'
```

L'output dell'ultimo comando è simile al seguente.

```
{
 "tags": [
 {
 "key": "key1",
 "value": "val1"
 },
 {
 "key": "key2",
 "value": "val2"
 },
 {
 "key": "key3",
 "value": "val3"
 },
 {
 "key": "key4",
 "value": "val4"
 }
]
}
```

## Aggiungi tag a un nuovo stream durante la creazione di una tabella

Puoi aggiungere tag agli stream quando crei una nuova tabella con uno stream usando CQL o per taggare uno stream. AWS CLI

 Note

Amazon Keyspaces CDC richiede la presenza di un ruolo collegato al servizio (AWSServiceRoleForAmazonKeyspacesCDC) che pubblica i dati metrici provenienti dai flussi di Amazon Keyspaces CDC nel tuo account per tuo conto. "cloudwatch:namespace": "AWS/Cassandra" CloudWatch Questo ruolo viene creato automaticamente per te. Per ulteriori informazioni, consulta [the section called "Stream CDC"](#).

## Cassandra Query Language (CQL)

Aggiungi tag a uno stream quando crei una nuova tabella utilizzando CQL

1. Per creare una nuova tabella con uno stream e applicare automaticamente i tag della tabella allo stream, puoi usare il 'propagate\_tags': 'TABLE' flag. La seguente dichiarazione ne è un esempio.

```
CREATE TABLE mytable (pk int, ck text, PRIMARY KEY(pk))
WITH TAGS={'key1':'val1', 'key2':'val2'}
AND cdc = TRUE
AND CUSTOM_PROPERTIES={
 'cdc_specification': {
 'view_type': 'NEW_IMAGE',
 'propagate_tags': 'TABLE'
 }
};
```

2. Per applicare nuovi tag allo stream, puoi usare il seguente esempio.

```
CREATE TABLE mytable (pk int, ck text, PRIMARY KEY(pk))
WITH TAGS={'key1':'val1', 'key2':'val2'}
AND cdc = TRUE
AND CUSTOM_PROPERTIES={
 'cdc_specification': {
 'view_type': 'NEW_IMAGE',
 'tags': { 'key': 'string', 'value': 'string' },
 }
};
```

## CLI

Aggiungi tag a uno stream quando crei una nuova tabella utilizzando il AWS CLI

1. Per creare una tabella con uno stream e applicare automaticamente i tag della tabella allo stream, puoi usare il `propagateTags=Table` flag. Il codice seguente ne è un esempio.

```
aws keyspaces create-table \
--keyspace-name 'my_keyspace' \
--table-name 'my_table' \
--schema-definition 'allColumns=[{name=pk,type=int},
{name=ck,type=text}],clusteringKeys=[{name=ck,orderBy=ASC}],partitionKeys=[{name=pk}]' \
--tags key=tag_key,value=tag_value \
--cdc-specification propagateTags=TABLE,status=ENABLED,viewType=NEW_IMAGE
```

2. Per applicare tag diversi allo stream, puoi usare il seguente esempio.

```
aws keyspaces create-table \
--keyspace-name 'my_keyspace' \
--table-name 'my_table' \
--schema-definition 'allColumns=[{name=pk,type=int},
{name=ck,type=text}],clusteringKeys=[{name=ck,orderBy=ASC}],partitionKeys=[{name=pk}]' \
--tags key=tag_key,value=tag_value \
--cdc-specification 'status=ENABLED,viewType=NEW_IMAGE,tags=[{key=tag_key,
value=tag_value}]'
```

## Aggiungi tag a un nuovo stream per una tabella esistente

Puoi aggiungere tag quando crei un nuovo stream per una tabella esistente. Puoi usare il `PropagateTags` flag per applicare i tag della tabella allo stream o specificare nuovi tag per lo stream. Puoi usare CQL o the AWS CLI per taggare un nuovo stream.

### Note

Amazon Keyspaces CDC richiede la presenza di un ruolo collegato al servizio (`AWSServiceRoleForAmazonKeyspacesCDC`) che pubblica i dati metrici provenienti dai flussi di Amazon Keyspaces CDC nel tuo account per tuo conto.

"cloudwatch:namespace": "AWS/Cassandra" CloudWatch Questo ruolo viene creato automaticamente per te. Per ulteriori informazioni, consulta [the section called “Stream CDC”](#).

## Console

Aggiungi tag quando crei un nuovo stream utilizzando (console)

1. [Accedi a e apri Console di gestione AWS la console Amazon Keyspaces a casa](https://console.aws.amazon.com/keyspaces/)<https://console.aws.amazon.com/keyspaces/>.
2. Nel pannello di navigazione, scegli Tabelle, quindi scegli la tabella per cui desideri aggiungere uno stream.
3. Scegli la scheda Streams.
4. Nella sezione Dettagli dello stream, scegli Modifica.
5. Seleziona Attiva gli stream.
6. Seleziona il tipo di visualizzazione e continua con Tag per creare tag per lo stream.
7. È possibile selezionare una delle seguenti opzioni:
  - Nessun tag: utilizza questa opzione se non desideri creare alcun tag per lo stream.
  - Copia i tag dalla tabella: utilizza questa opzione se desideri copiare i tag dalla tabella allo stream. Dopo aver copiato i tag, puoi modificarli per lo stream. Nota che questa opzione è disponibile solo se la tabella contiene tag.
  - Aggiungi nuovi tag: puoi aggiungere fino a 50 tag per lo stream scegliendo Aggiungi nuovo tag.
8. Scegli Save changes (Salva modifiche).

## Cassandra Query Language (CQL)

Aggiungi tag quando crei un nuovo stream

1. Per creare un nuovo flusso per una tabella esistente e applicare i tag della tabella allo stream, puoi usare il 'propagate\_tags': 'TABLE' flag. Le seguenti affermazioni ne sono un esempio.

```
ALTER TABLE mytable WITH cdc = TRUE AND CUSTOM_PROPERTIES={ 'cdc_specification':
 { 'view_type': 'NEW_IMAGE', 'propagate_tags': 'TABLE' } };
```

2. Per creare un nuovo flusso per una tabella esistente e specificare nuovi tag, puoi usare l'esempio seguente.

```
ALTER TABLE mytable WITH cdc = TRUE AND CUSTOM_PROPERTIES={ 'cdc_specification':
 { 'view_type': 'NEW_IMAGE', 'tags': { 'key': 'string', 'value': 'string' }} };
```

## CLI

Aggiungi tag quando crei un nuovo stream usando il AWS CLI

1. Per creare un nuovo stream con tag, puoi usare il `propagateTags=TABLE` flag per applicare automaticamente i tag della tabella allo stream. Il codice seguente ne è un esempio.

```
aws keyspaces update-table \
--keyspace-name 'my_keyspace' \
--table-name 'my_table' \
--cdc-specification propagateTags=TABLE,status=ENABLED,viewType=NEW_IMAGE
```

2. Per creare un nuovo flusso per una tabella esistente e specificare nuovi tag, puoi usare il seguente esempio.

```
aws keyspaces update-table \
--keyspace-name 'my_keyspace' \
--table-name 'my_table' \
--cdc-specification 'status=ENABLED,viewType=NEW_IMAGE,tags=[{key=tag_key,
value=tag_value}]'
```

## Aggiungere nuovi tag a uno stream

Puoi aggiungere nuovi tag a uno stream esistente in Amazon Keyspaces utilizzando CQL o il AWS CLI. Puoi aggiungere tag solo allo stream più recente.

## Console

Aggiungi tag a uno stream esistente (console)

1. [Accedi a e apri Console di gestione AWS la console Amazon Keyspaces a casahttps://console.aws.amazon.com/keyspaces/.](https://console.aws.amazon.com/keyspaces/)



2. Nel pannello di navigazione, scegli Tabelle, quindi scegli la tabella con lo stream a cui desideri taggare.
3. Scegli la scheda Streams.
4. Nella sezione Tag scegli Gestisci tag.
5. Scegli Aggiungi nuovo tag per aggiungere un nuovo tag. Puoi creare fino a 50 tag ripetendo questo passaggio.
6. Scegli Save changes (Salva modifiche).

## Cassandra Query Language (CQL)

Aggiungi tag a uno stream usando CQL

- La seguente dichiarazione mostra come aggiungere tag a uno stream esistente.

```
ALTER TABLE mykeyspace.mytable ADD TAGS_FOR_CDC {'key1':'val1', 'key2':'val2'};
```

## CLI

Aggiungere tag a uno stream utilizzando il AWS CLI

- L'esempio seguente mostra come aggiungere nuovi tag a uno stream esistente.

```
aws keyspaces tag-resource --resource-arn 'arn:aws:cassandra:us-east-1:111122223333:/keyspace/my_keyspace/table/my_table/stream/2025-05-11T21:21:33.291' --tags 'key=key3,value=val3' 'key=key4,value=val4'
```

## Eliminare i tag da uno stream

Per eliminare i tag da uno stream, puoi utilizzare CQL o. AWS CLI Puoi eliminare solo i tag per lo stream più recente.

## Console

Eliminare i tag da una tabella utilizzando la console Amazon Keyspaces

1. [Accedi a e apri Console di gestione AWS la console Amazon Keyspaces a casahttps://console.aws.amazon.com/keyspaces/.](https://console.aws.amazon.com/keyspaces/)
2. Nel pannello di navigazione, seleziona Tabelle.
3. Scegli una tabella dall'elenco e scegli la scheda Streams.
4. Nella sezione Tag, scegli Gestisci tag per eliminare i tag dalla tabella.
5. Dopo il tag che desideri eliminare, scegli Rimuovi.
6. Scegli Save changes (Salva modifiche).

## Cassandra Query Language (CQL)

Elimina i tag da uno stream usando CQL

- La seguente dichiarazione mostra come eliminare i tag da uno stream esistente.

```
ALTER TABLE mytable DROP TAGS_FOR_CDC {'key3':'val3', 'key4':'val4'};
```

## CLI

Eliminare i tag da uno stream utilizzando il AWS CLI

- L'istruzione seguente rimuove i tag specificati da uno stream.

```
aws keyspaces untag-resource --resource-arn 'arn:aws:cassandra:us-east-1:111122223333:/keyspace/my_keyspace/table/my_table/stream/2025-05-11T21:21:33.291' --tags 'key=key3,value=val3'
'key=key4,value=val4'
```

## Visualizza i tag di uno stream

Gli esempi seguenti mostrano come visualizzare i tag di uno stream in Amazon Keyspaces utilizzando CQL o il AWS CLI

## Console

Visualizza i tag di uno stream utilizzando la console

1. [Accedi a e apri Console di gestione AWS la console Amazon Keyspaces a casa](https://console.aws.amazon.com/keyspaces/)<https://console.aws.amazon.com/keyspaces/>.
2. Nel pannello di navigazione, seleziona Tabelle.
3. Scegli una tabella dall'elenco e scegli la scheda Streams.
4. Puoi visualizzare i tag dello stream nella sezione Tag.

## Cassandra Query Language (CQL)

Visualizza i tag di uno stream usando CQL

Per leggere i tag allegati a uno stream, è necessario specificare l'ARN della risorsa dello stream nella WHERE clausola. La seguente sintassi CQL ne è un esempio.

```
SELECT * FROM system_schema_mcs.tags WHERE resource_id = stream_arn;
```

- La seguente query restituisce i tag per lo stream specificato.

```
SELECT tags FROM system_schema_mcs.tags WHERE resource_id =
'arn:aws:cassandra:us-east-1:111122223333:/keyspace/my_keyspace/table/my_table/
stream/2025-05-06T17:17:39.800';
```

L'output di tale query è simile al seguente.

```
resource_id
 | keyspace_name | resource_name
resource_type | tags
-----+-----+-----
+-----+-----+-----
arn:aws:cassandra:us-east-1:111122223333:/keyspace/my_keyspace/table/my_table/
stream/2025-04-02T23:00:07.052 | singleks | 2025-04-02T23:00:07.052 |
stream | {'tagkey': 'tagval'}
```

## CLI

Visualizza i tag di uno stream utilizzando il AWS CLI

- Questo esempio mostra come elencare i tag per tutti gli stream nel keyspace specificato.

```
aws keyspaces list-tags-for-resource --resource-arn 'arn:aws:cassandra:us-east-1:111122223333:/keyspace/my_keyspace/table/my_table/stream/2025-05-11T21:21:33.291'
```

L'output dell'ultimo comando è simile al seguente.

```
{
 "tags": [
 {
 "key": "key1",
 "value": "val1"
 },
 {
 "key": "key2",
 "value": "val2"
 },
 {
 "key": "key3",
 "value": "val3"
 },
 {
 "key": "key4",
 "value": "val4"
 }
]
}
```

## Crea report di allocazione dei costi utilizzando tag per Amazon Keyspaces

AWS utilizza i tag per organizzare i costi delle risorse nel report di allocazione dei costi. AWS fornisce due tipi di tag di allocazione dei costi:

- Un AWS tag generato. AWS definisce, crea e applica questo tag per te.
- Tag definiti dall'utente. L'utente definisce, crea e applica questi tag.

È necessario attivare entrambi i tipi di tag separatamente per poterli visualizzare in Cost Explorer o in un report di allocazione dei costi.

Per attivare i tag AWS generati:

1. Accedi Console di gestione AWS e apri la console Billing and Cost Management <https://console.aws.amazon.com/billing/a/casa#/>.
2. Nel riquadro di navigazione scegli Cost Allocation Tags (Tag per l'allocazione dei costi).
3. In Tag per l'allocazione dei costi generati da AWS scegli Attiva.

Per attivare i tag definiti dall'utente:

1. Accedi Console di gestione AWS e apri la console Billing and Cost Management <https://console.aws.amazon.com/billing/a/casa#/>.
2. Nel riquadro di navigazione scegli Cost Allocation Tags (Tag per l'allocazione dei costi).
3. In Tag per l'allocazione dei costi definiti dall'utente scegli Attiva.

Dopo aver creato e attivato i tag, AWS genera un rapporto di allocazione dei costi con l'utilizzo e i costi raggruppati in base ai tag attivi. Il rapporto sull'allocazione dei costi include tutti i AWS costi per ogni periodo di fatturazione. Il report include sia le risorse taggate, sia quelle non taggate, per permetterti di organizzare in modo chiaro le spese per le risorse.

#### Note

Attualmente, i dati trasferiti da Amazon Keyspaces non verranno suddivisi per tag nei report di allocazione dei costi.

Per ulteriori informazioni, consulta [Utilizzo dei tag per l'allocazione dei costi](#).

## Crea risorse Amazon Keyspaces con AWS CloudFormation

Amazon Keyspaces è integrato con AWS CloudFormation, un servizio che ti aiuta a modellare e configurare AWS keyspaces e tabelle in modo da dedicare meno tempo alla creazione e alla gestione delle risorse e dell'infrastruttura. Crei un modello che descrive gli spazi chiave e le tabelle che desideri e si CloudFormation occupa del provisioning e della configurazione di tali risorse per te.

Quando lo utilizzi CloudFormation, puoi riutilizzare il modello per configurare le risorse Amazon Keyspaces in modo coerente e ripetuto. Descrivi le tue risorse una sola volta e poi fornisci le stesse risorse più e più volte in più regioni Account AWS .

## Amazon Keyspaces e modelli CloudFormation

[Per effettuare il provisioning e configurare le risorse per Amazon Keyspaces, devi conoscere CloudFormation i modelli.](#) I modelli sono file di testo formattati in JSON o YAML. Questi modelli descrivono le risorse che desideri fornire nei tuoi CloudFormation stack. Se non conosci JSON o YAML, puoi usare CloudFormation Designer per iniziare a usare i modelli. CloudFormation [Per ulteriori informazioni, consulta Cos'è il designer? CloudFormation](#) nella Guida AWS CloudFormation per l'utente.

Amazon Keyspaces supporta la creazione di keyspaces e tabelle in. CloudFormation Per le tabelle create utilizzando i CloudFormation modelli, puoi specificare lo schema, la read/write modalità, le impostazioni di throughput assegnate e altre funzionalità supportate. Per ulteriori informazioni, inclusi esempi di modelli JSON e YAML per keyspace e tabelle, consulta il riferimento al tipo di risorsa Amazon [Keyspaces \(for Apache Cassandra\)](#) nel Template Reference.AWS CloudFormation

## Scopri di più su CloudFormation

Per ulteriori informazioni CloudFormation, consulta le seguenti risorse:

- [AWS CloudFormation](#)
- [AWS CloudFormation Guida per l'utente](#)
- [AWS CloudFormation Guida per l'utente dell'interfaccia a riga di comando](#)

## Utilizzo di NoSQL Workbench con Amazon Keyspaces (per Apache Cassandra)

NoSQL Workbench è un'applicazione lato client che ti aiuta a progettare e visualizzare più facilmente modelli di dati non relazionali per Amazon Keyspaces. I client NoSQL Workbench sono disponibili per Windows, macOS e Linux.

### Progettazione di modelli di dati e creazione automatica di risorse

NoSQL Workbench fornisce un'interfaccia per progettare e creare point-and-click modelli di dati Amazon Keyspaces. Puoi creare facilmente nuovi modelli di dati partendo da zero definendo

spazi chiave, tabelle e colonne. Puoi anche importare modelli di dati esistenti e apportare modifiche (come aggiungere, modificare o rimuovere colonne) per adattare i modelli di dati alle nuove applicazioni. NoSQL Workbench consente quindi di eseguire il commit dei modelli di dati su Amazon Keyspaces o Apache Cassandra e di creare gli spazi chiave e le tabelle automaticamente. Per imparare a creare modelli di dati, consulta e. [the section called “Crea un modello di dati”](#) [the section called “Modifica un modello di dati”](#)

## Visualizzazione dei modelli di dati

Utilizzando NoSQL Workbench, puoi visualizzare i tuoi modelli di dati per garantire che siano in grado di supportare le query e i modelli di accesso dell'applicazione. Puoi anche salvare ed esportare i tuoi modelli di dati in una varietà di formati per la collaborazione, la documentazione e le presentazioni. Per ulteriori informazioni, consulta [the section called “Visualizza un modello di dati”](#).

## Argomenti

- [Scarica NoSQL Workbench](#)
- [Guida introduttiva a NoSQL Workbench](#)
- [Visualizza modelli di dati con NoSQL Workbench](#)
- [Crea un nuovo modello di dati con NoSQL Workbench](#)
- [Modifica i modelli di dati esistenti con NoSQL Workbench](#)
- [Come eseguire il commit di modelli di dati in Amazon Keyspaces e Apache Cassandra](#)
- [Modelli di dati di esempio in NoSQL Workbench](#)
- [Cronologia delle versioni di NoSQL Workbench](#)

## Scarica NoSQL Workbench

Segui queste istruzioni per scaricare e installare NoSQL Workbench.

Per scaricare e installare NoSQL Workbench

1. Utilizzare uno dei collegamenti seguenti per scaricare gratuitamente NoSQL Workbench.

| Sistema operativo | Collegamento per il download       |
|-------------------|------------------------------------|
| macOS             | <a href="#">Download per macOS</a> |

| Sistema operativo | Collegamento per il download         |
|-------------------|--------------------------------------|
| Linux*            | <a href="#">Scarica per Linux</a>    |
| Windows           | <a href="#">Download per Windows</a> |

\* NoSQL Workbench supporta Ubuntu 12.04, Fedora 21 e Debian 8 o qualsiasi versione più recente di queste distribuzioni Linux.

- Una volta completato il download, avvia l'applicazione e segui le istruzioni sullo schermo per completare l'installazione.

## Guida introduttiva a NoSQL Workbench

Per iniziare a usare NoSQL Workbench, nella pagina Database Catalog di NoSQL Workbench, scegli Amazon Keyspaces, quindi scegli Launch.

**aws NoSQL Workbench**  
A client-side application for designing, creating, querying, and managing NoSQL databases.

How it works

- Data modeler**
  - Build new data models
  - Add tables and indexes
  - Import and export models
- Visualizer**
  - Add sample data
  - Visualize data layout and structure
  - Commit model the cloud
- Operation builder**
  - Build operations and queries
  - Use a guided form
  - Generate code for data-plane operations

Get started by choosing a database service

Amazon DynamoDB
 Launch

Amazon Keyspaces (for Apache Cassandra)
 Launch

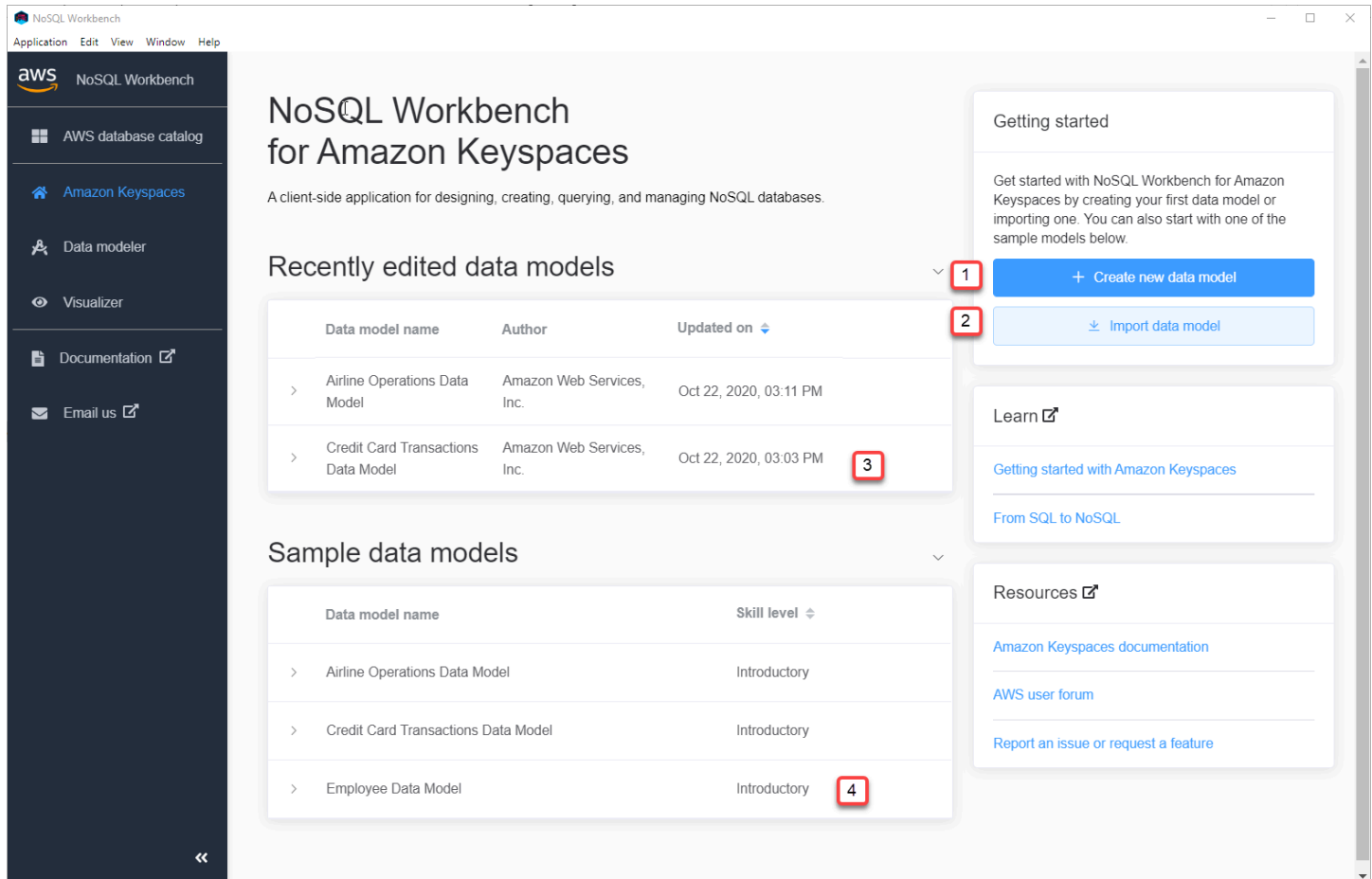
| Type        | Description                                                                                                                                                                                                                                                                                                            | Use cases                                                                                      |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------|
| Key-value   | Amazon DynamoDB is a key-value and document database that delivers single-digit-millisecond performance at any scale. It's a fully managed, multi-region, multi-master, durable database with built-in security, backup and restore, and in-memory caching for internet-scale applications. <a href="#">Learn more</a> | High-traffic web apps, e-commerce systems, and gaming applications                             |
| Wide-column | Amazon Keyspaces is a scalable, highly available, and managed Apache Cassandra-compatible database service. You can run your Cassandra workloads on AWS using the same Cassandra application code and developer tools you use today. <a href="#">Learn more</a>                                                        | High-scale industrial apps for equipment maintenance, fleet management, and route optimization |

By using NoSQL Workbench you agree to the [License Agreement](#), [AWS Customer Agreement](#), [AWS Service Terms](#), [AWS Privacy Notice](#), and [Source Code Notice](#). If you already have an AWS Customer Agreement, you agree that the terms of that agreement govern your installation and use of this product. See also [third party notices](#).



Si apre la home page di NoSQL Workbench per Amazon Keyspaces, dove sono disponibili le seguenti opzioni per iniziare:

1. Crea un nuovo modello di dati.
2. Importa un modello di dati esistente in formato JSON.
3. Apri un modello di dati modificato di recente.
4. Apri uno dei modelli di esempio disponibili.



Ciascuna delle opzioni apre il modellatore di dati NoSQL Workbench. Per continuare a creare un nuovo modello di dati, consulta [the section called “Crea un modello di dati”](#). Per modificare un modello di dati esistente, vedere [the section called “Modifica un modello di dati”](#).

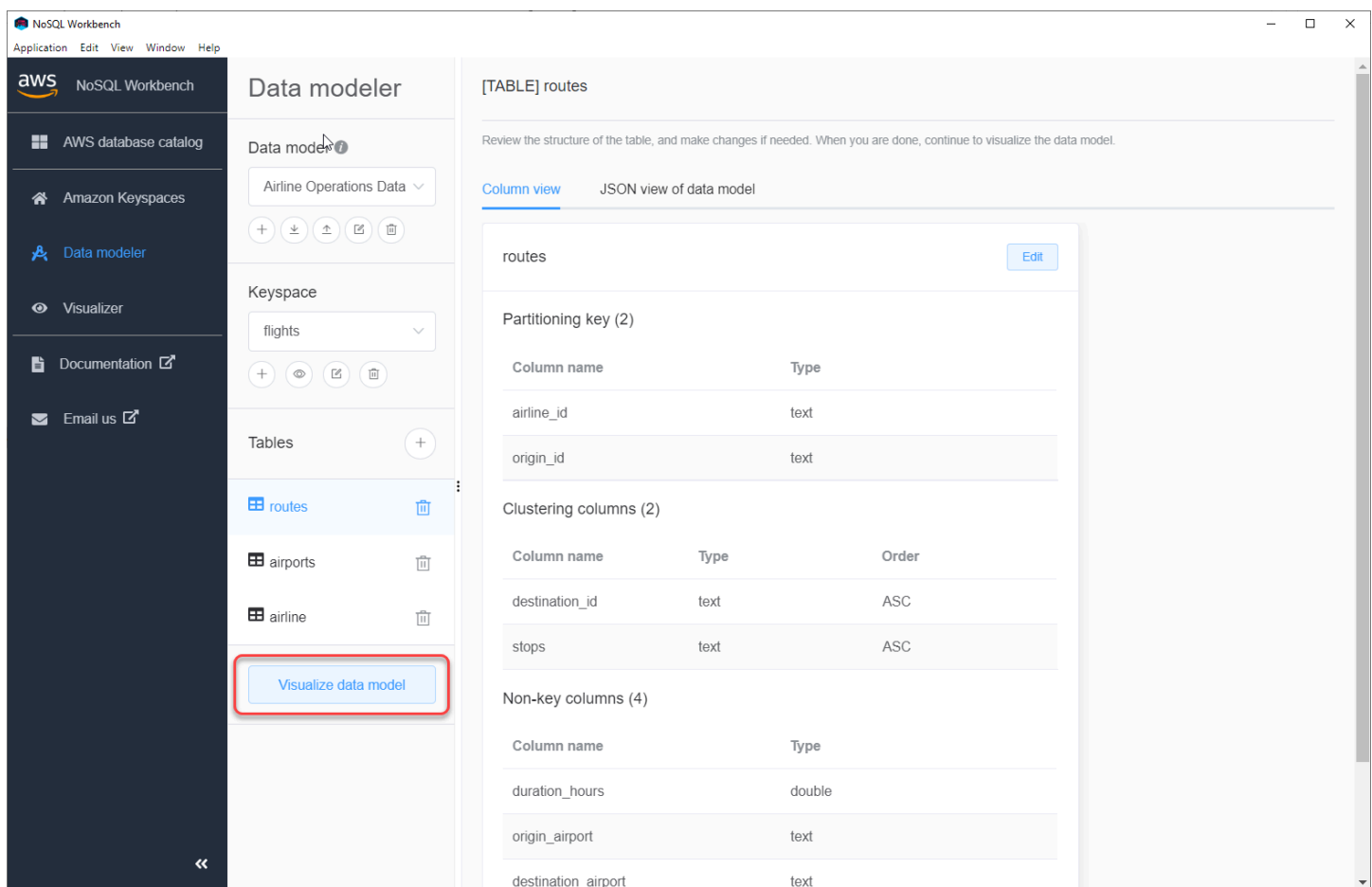
## Visualizza modelli di dati con NoSQL Workbench

Utilizzando NoSQL Workbench, puoi visualizzare i tuoi modelli di dati per garantire che siano in grado di supportare le query e i modelli di accesso dell'applicazione. Puoi anche salvare ed esportare i tuoi modelli di dati in una varietà di formati per la collaborazione, la documentazione e le presentazioni.

Dopo aver creato un nuovo modello di dati o modificato un modello di dati esistente, è possibile visualizzare il modello.

### Visualizzazione di modelli di dati con NoSQL Workbench

Dopo aver completato il modello di dati nel modellatore di dati, scegli **Visualizza modello di dati**.



Si accede al visualizzatore di dati in NoSQL Workbench. Il visualizzatore di dati fornisce una rappresentazione visiva dello schema della tabella e consente di aggiungere dati di esempio. Per aggiungere dati di esempio a una tabella, scegliete una tabella dal modello, quindi scegliete **Modifica**. Per aggiungere una nuova riga di dati, scegli **Aggiungi nuova riga** nella parte inferiore dello schermo. Al termine, scegliere **Save (Salva)**.

The screenshot shows the AWS NoSQL Workbench Visualizer interface. On the left is a sidebar with navigation options: AWS database catalog, Amazon Keyspaces, Data modeler, Visualizer (selected), Documentation, and Email us. The main area is titled 'Visualizer' and shows the 'Data model' for 'Airline Operations'. The 'Keyspace' is set to 'flights'. Under 'Tables', a list includes 'routes', 'airports', and 'airline'. The 'routes' table is selected, and its schema is displayed in a table view. The schema has a primary key 'airline\_id (text)' and clustering columns 'destination\_id (text)' and 'stops (text)'. Non-key columns are 'duration\_hours (double)' and 'origin\_airport (text)'. Below the schema, there are three rows of sample data. At the bottom of the table view, there is a button labeled '+ Add new row' which is highlighted with a red rectangle. At the top right of the table view, there are 'Cancel' and 'Save' buttons. Below the table view, there are three buttons: 'Aggregate view', 'Commit to Amazon Keyspaces', and 'Commit to Apache Cassandra'.

| Primary key       |                       |              | Non-key columns         |                       |
|-------------------|-----------------------|--------------|-------------------------|-----------------------|
| Partition key     | Clustering columns    |              |                         |                       |
| airline_id (text) | destination_id (text) | stops (text) | duration_hours (double) | origin_airport (text) |
| acme_airlines     | MCI                   | 1            | 0.33333333              | Newark Liberty        |
| trusted_airlines  | MIC                   | 2            | 0.33333333              | Phoenix Sky Ha        |
| freedom_airline   | EWR                   | 1            | 0.33333333              | San Francisco         |

## Visualizzazione aggregata

Dopo aver confermato lo schema della tabella, puoi aggregare le visualizzazioni del modello di dati.

**Visualizer**

Data model ⓘ

Airline Operations Data ▾

↓ ↑

Keyspace

flights ▾

Tables

- routes
- airports
- airline

**Aggregate view**

Commit to Amazon Keyspaces

Commit to Apache Cassandra

**[TABLE] routes** Edit

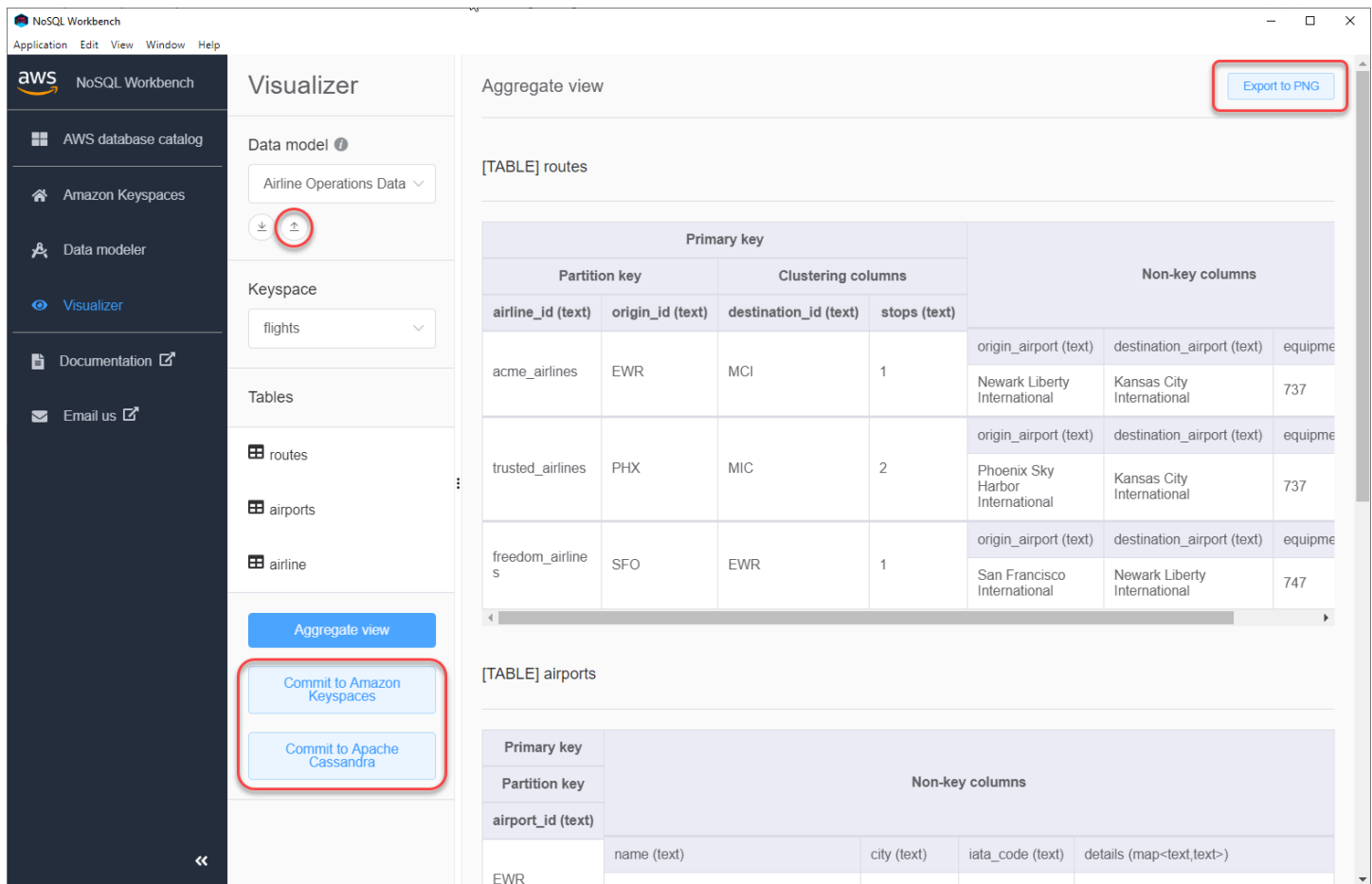
Use this view to get a visual representation of your tables schema, and add sample data. When you are done, commit your model to Amazon Keyspaces.

| Primary key       |                  |                       |              | Non-key columns                  |                              |           |
|-------------------|------------------|-----------------------|--------------|----------------------------------|------------------------------|-----------|
| Partition key     |                  | Clustering columns    |              |                                  |                              |           |
| airline_id (text) | origin_id (text) | destination_id (text) | stops (text) | origin_airport (text)            | destination_airport (text)   | equipment |
| acme_airlines     | EWR              | MCI                   | 1            | Newark Liberty International     | Kansas City International    | 737       |
| trusted_airlines  | PHX              | MIC                   | 2            | Phoenix Sky Harbor International | Kansas City International    | 737       |
| freedom_airlines  | SFO              | EWR                   | 1            | San Francisco International      | Newark Liberty International | 747       |

Dopo aver aggregato la vista del modello di dati, è possibile esportare la vista in un file PNG. Per esportare il modello di dati in un file JSON, scegliete il segno di caricamento sotto il nome del modello di dati.

#### Note

Puoi esportare il modello di dati in formato JSON in qualsiasi momento del processo di progettazione.



Sono disponibili le seguenti opzioni per confermare le modifiche:

- Impegnati a utilizzare Amazon Keyspaces
- Effettua il commit in un cluster Apache Cassandra

Per ulteriori informazioni su come eseguire il commit delle modifiche, consulta [the section called “Esegui il commit di un modello di dati”](#)

## Crea un nuovo modello di dati con NoSQL Workbench

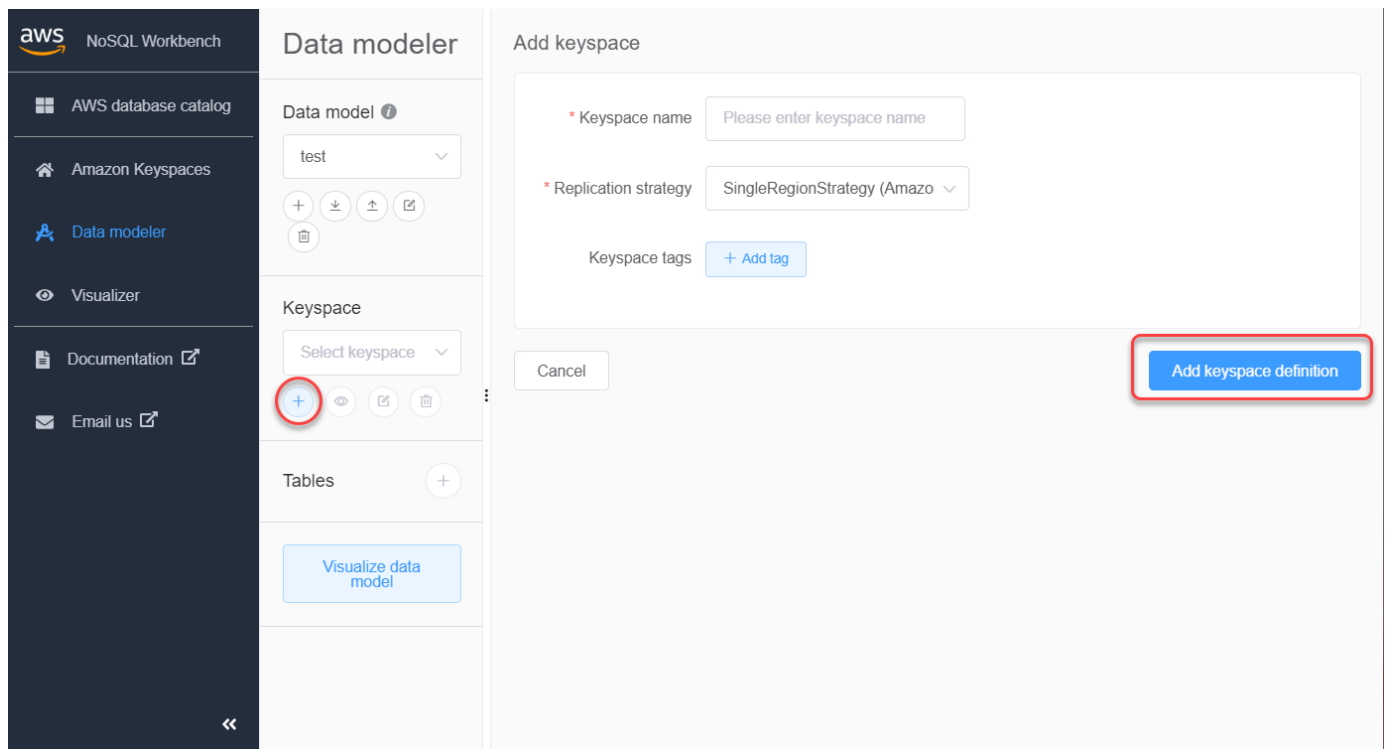
Puoi utilizzare il modellatore di dati NoSQL Workbench per progettare nuovi modelli di dati basati sui modelli di accesso ai dati dell'applicazione. Per creare un nuovo modello di dati per Amazon Keyspaces, puoi utilizzare il modellatore di dati NoSQL Workbench per creare keyspace, tabelle e colonne. Segui questi passaggi per creare un nuovo modello di dati.

1. Per creare un nuovo keyspace, scegli il segno più in Keyspace.

In questo passaggio, scegli le seguenti proprietà e impostazioni.

- Nome dello spazio chiave: immettete il nome del nuovo spazio chiave.
- Strategia di replica: scegli la strategia di replica per il keyspace. Amazon Keyspaces lo utilizza SingleRegionStrategy per replicare i dati tre volte automaticamente in più AWS zone di disponibilità. Se hai intenzione di affidare il modello di dati a un cluster Apache Cassandra, puoi scegliere o. SimpleStrategyNetworkTopologyStrategy
- Tag Keyspaces: i tag delle risorse sono opzionali e consentono di classificare le risorse in diversi modi, ad esempio per scopo, proprietario, ambiente o altri criteri. Per ulteriori informazioni sui tag per le risorse di Amazon Keyspaces, consulta. [the section called “Lavorare con i tag”](#)

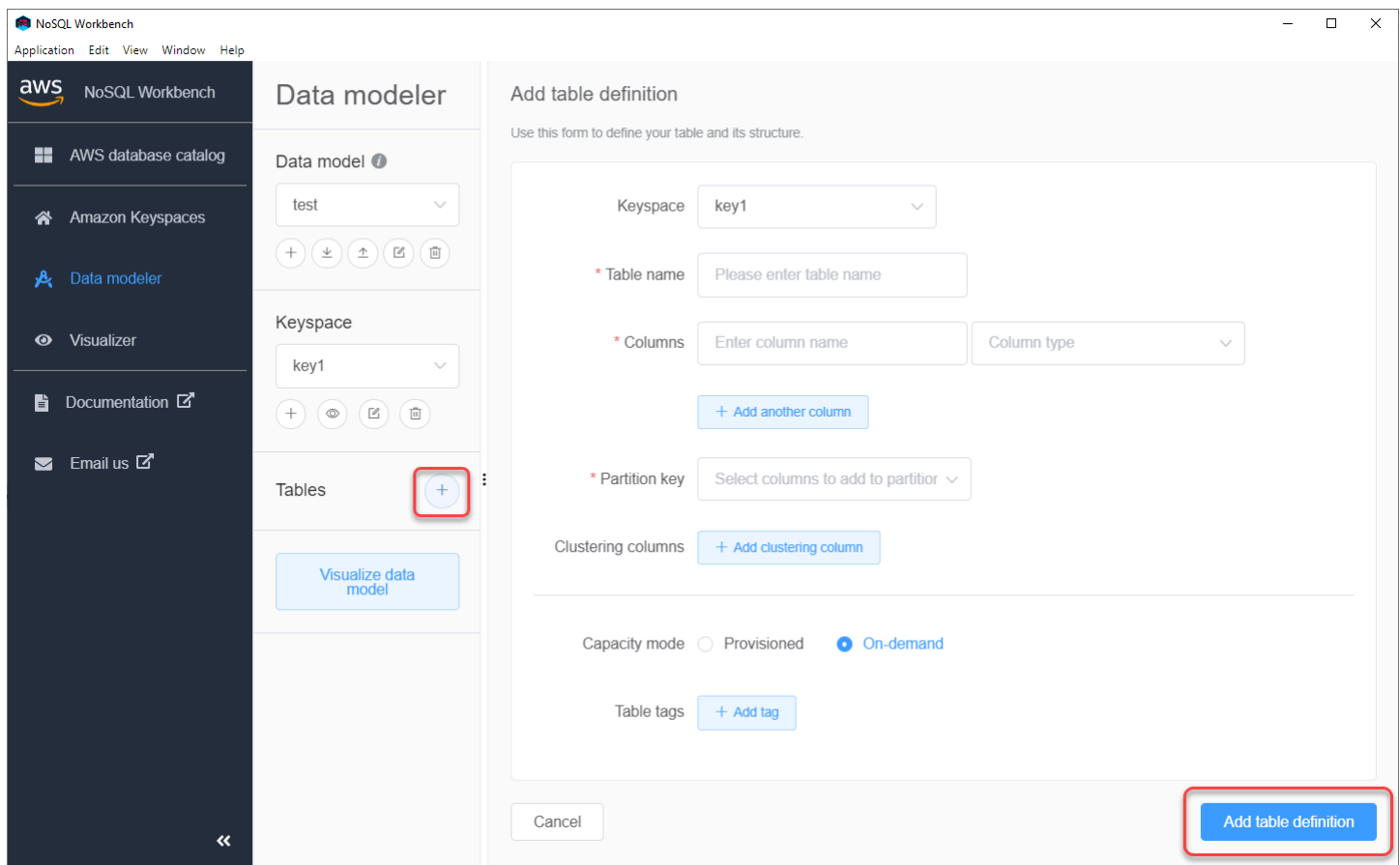
2. Scegli Aggiungi definizione dello spazio chiave per creare lo spazio chiave.



3. Per creare una nuova tabella, scegli il segno più accanto a Tabelle. In questo passaggio, definisci le seguenti proprietà e impostazioni.

- Nome tabella: il nome della nuova tabella.
- Colonne: aggiungi un nome di colonna e scegli il tipo di dati. Ripeti questi passaggi per ogni colonna dello schema.
- Chiave di partizione: scegli le colonne per la chiave di partizione.

- Colonne di raggruppamento: scegli le colonne di raggruppamento (opzionale).
  - Modalità capacità: scegli la modalità di capacità di lettura/scrittura per la tabella. È possibile scegliere la capacità fornita o su richiesta. Per ulteriori informazioni sulle modalità di capacità, consulta [the section called “Configura le modalità di capacità di lettura/scrittura”](#)
  - Tag di tabella: i tag delle risorse sono opzionali e consentono di classificare le risorse in diversi modi, ad esempio per scopo, proprietario, ambiente o altri criteri. Per ulteriori informazioni sui tag per le risorse di Amazon Keyspaces, consulta [the section called “Lavorare con i tag”](#)
4. Scegli Aggiungi definizione di tabella per creare la nuova tabella.
  5. Ripeti questi passaggi per creare tabelle aggiuntive.
  6. Continua [the section called “Visualizzazione di un modello di dati”](#) a visualizzare il modello di dati che hai creato.



## Modifica i modelli di dati esistenti con NoSQL Workbench

È possibile utilizzare il modellatore di dati per importare e modificare modelli di dati esistenti creati utilizzando NoSQL Workbench. Il modellatore di dati include anche alcuni modelli di dati di esempio per aiutarti a iniziare con la modellazione dei dati. I modelli di dati che è possibile modificare con NoSQL Workbench possono essere modelli di dati importati da un file, modelli di dati di esempio forniti o modelli di dati creati in precedenza.

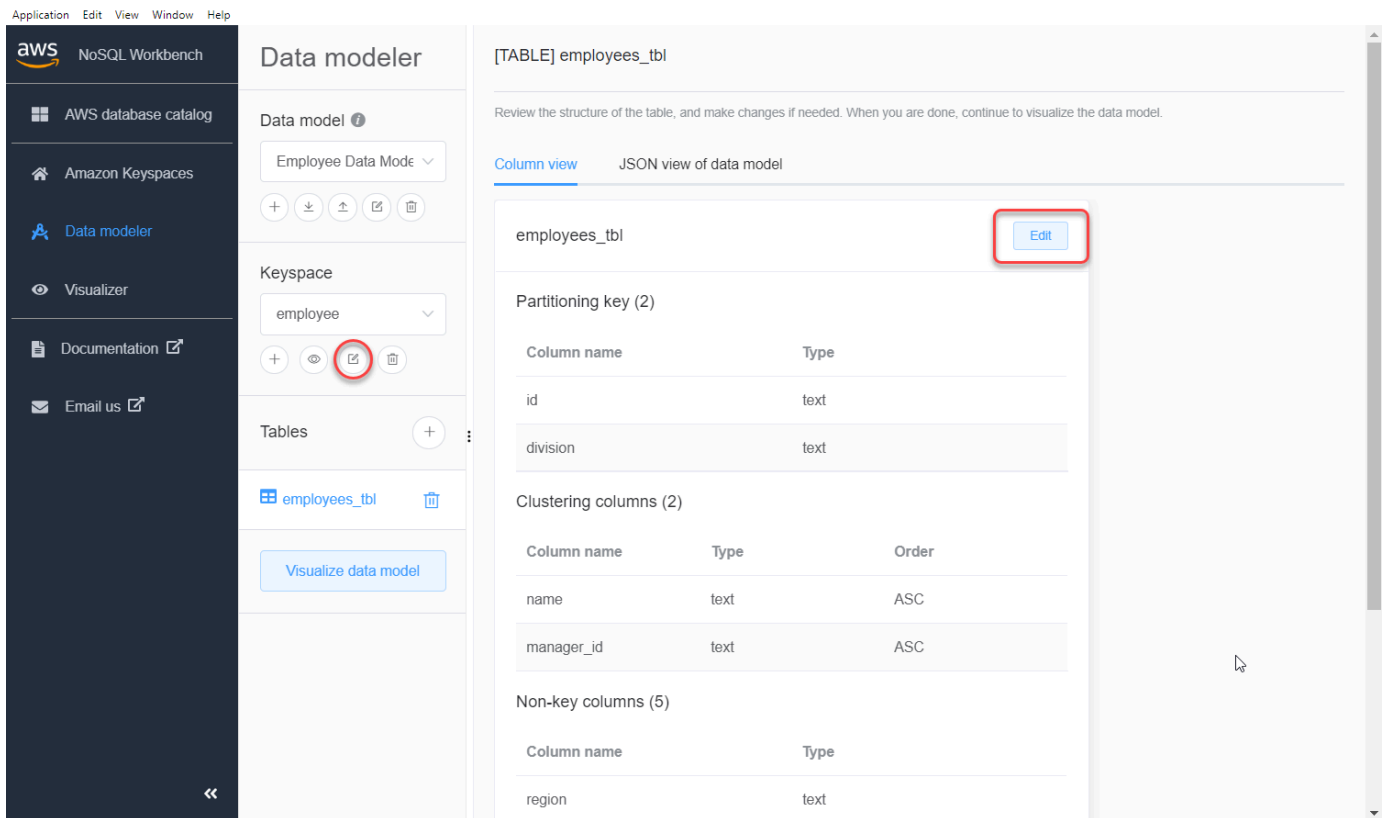
1. Per modificare uno spazio chiave, scegli il simbolo di modifica in Keyspace.

In questo passaggio, puoi modificare le seguenti proprietà e impostazioni.

- Nome dello spazio chiave: immettere il nome del nuovo spazio chiave.
- Strategia di replica: scegli la strategia di replica per il keyspace. Amazon Keyspaces lo utilizza `SingleRegionStrategy` per replicare i dati tre volte automaticamente in più AWS zone di disponibilità. Se hai intenzione di affidare il modello di dati a un cluster Apache Cassandra, puoi scegliere o. `SimpleStrategyNetworkTopologyStrategy`
- Tag Keyspaces: i tag delle risorse sono opzionali e consentono di classificare le risorse in diversi modi, ad esempio per scopo, proprietario, ambiente o altri criteri. Per ulteriori informazioni sui tag per le risorse di Amazon Keyspaces, consulta. [the section called “Lavorare con i tag”](#)

2. Scegli Salva modifiche per aggiornare lo spazio delle chiavi.





3. Per modificare una tabella, scegli Modifica accanto al nome della tabella. In questo passaggio, puoi aggiornare le seguenti proprietà e impostazioni.
  - Nome tabella: il nome della nuova tabella.
  - Colonne: aggiungi un nome di colonna e scegli il tipo di dati. Ripeti questi passaggi per ogni colonna dello schema.
  - Chiave di partizione: scegli le colonne per la chiave di partizione.
  - Colonne di raggruppamento: scegli le colonne di raggruppamento (opzionale).
  - Modalità capacità: scegli la modalità di capacità di lettura/scrittura per la tabella. È possibile scegliere la capacità fornita o su richiesta. Per ulteriori informazioni sulle modalità di capacità, consulta. [the section called “Configura le modalità di capacità di lettura/scrittura”](#)
  - Tag di tabella: i tag delle risorse sono opzionali e consentono di classificare le risorse in diversi modi, ad esempio per scopo, proprietario, ambiente o altri criteri. Per ulteriori informazioni sui tag per le risorse di Amazon Keyspaces, consulta. [the section called “Lavorare con i tag”](#)
4. Scegli Salva modifiche per aggiornare la tabella.
5. Continua [the section called “Visualizzazione di un modello di dati”](#) a visualizzare il modello di dati che hai aggiornato.

# Come eseguire il commit di modelli di dati in Amazon Keyspaces e Apache Cassandra

Questa sezione mostra come eseguire il commit di modelli di dati completati nei cluster Amazon Keyspaces e Apache Cassandra. Questo processo crea automaticamente le risorse lato server per gli spazi chiave e le tabelle in base alle impostazioni definite nel modello di dati.

The screenshot shows the NoSQL Workbench Visualizer interface. On the left, a sidebar contains navigation links: AWS database catalog, Amazon Keyspaces, Data modeler, Visualizer (selected), Documentation, and Email us. The main area is titled 'Visualizer' and shows a 'Data model' for 'Airline Operations Data'. Below this, the 'Keyspace' is set to 'flights' and the 'Tables' list includes 'routes', 'airports', and 'airline'. At the bottom of the sidebar, three buttons are visible: 'Aggregate view', 'Commit to Amazon Keyspaces' (highlighted with a red box), and 'Commit to Apache Cassandra'. The main panel displays a visual representation of the 'routes' table schema. It includes a table with columns: 'airline\_id (text)', 'origin\_id (text)', 'destination\_id (text)', 'stops (text)', 'origin\_airport (text)', 'destination\_airport (text)', and 'equipment'. The table is divided into 'Primary key' (Partition key and Clustering columns) and 'Non-key columns'. The data rows show flight routes for 'acme\_airlines', 'trusted\_airlines', and 'freedom\_airlines'.

| Primary key       |                  |                       |              | Non-key columns                  |                              |           |
|-------------------|------------------|-----------------------|--------------|----------------------------------|------------------------------|-----------|
| Partition key     |                  | Clustering columns    |              |                                  |                              |           |
| airline_id (text) | origin_id (text) | destination_id (text) | stops (text) | origin_airport (text)            | destination_airport (text)   | equipment |
| acme_airlines     | EWR              | MCI                   | 1            | Newark Liberty International     | Kansas City International    | 737       |
| trusted_airlines  | PHX              | MIC                   | 2            | Phoenix Sky Harbor International | Kansas City International    | 737       |
| freedom_airlines  | SFO              | EWR                   | 1            | San Francisco International      | Newark Liberty International | 747       |

## Argomenti

- [Prima di iniziare](#)
- [Connect ad Amazon Keyspaces con credenziali specifiche del servizio](#)
- [Connect ad Amazon Keyspaces con credenziali AWS Identity and Access Management \(IAM\)](#)
- [Usa una connessione salvata](#)
- [Impegnati con Apache Cassandra](#)

## Prima di iniziare

Amazon Keyspaces richiede l'uso di Transport Layer Security (TLS) per proteggere le connessioni con i client. Per connetterti ad Amazon Keyspaces tramite TLS, devi completare la seguente attività prima di iniziare.

- Scarica i seguenti certificati digitali e salva i file localmente o nella tua home directory.
  1. AmazonRootCA1
  2. AmazonRootCA2
  3. AmazonRootCA3
  4. AmazonRootCA4
  5. Starfield Class 2 Root (opzionale, per compatibilità con le versioni precedenti)

Per scaricare i certificati, puoi usare i seguenti comandi.

```
curl -O https://www.amazontrust.com/repository/AmazonRootCA1.pem
curl -O https://www.amazontrust.com/repository/AmazonRootCA2.pem
curl -O https://www.amazontrust.com/repository/AmazonRootCA3.pem
curl -O https://www.amazontrust.com/repository/AmazonRootCA4.pem
curl -O https://certs.secureserver.net/repository/sf-class2-root.crt
```

### Note

Amazon Keyspaces utilizzava in precedenza certificati TLS ancorati alla CA Starfield Class 2. AWS sta migrando tutte le Regioni AWS verso certificati emessi nell'ambito di Amazon Trust Services (Amazon Root CAs 1—4). Durante questa transizione, configura i client in modo che si fidino sia di Amazon Root CAs 1-4 che di Starfield root per garantire la compatibilità in tutte le regioni.

Combina tutti i certificati scaricati in un unico pem file con il nome indicato *keyspaces-bundle.pem* nei nostri esempi. Per aggiungerlo, eseguire il comando seguente: Prendi nota del percorso del file, ne avrai bisogno in seguito.

```
cat AmazonRootCA1.pem \
 AmazonRootCA2.pem \
 AmazonRootCA3.pem \
 AmazonRootCA4.pem \
```

```
sf-class2-root.crt \
> keyspaces-bundle.pem
```

Dopo aver salvato il file del certificato, puoi connetterti ad Amazon Keyspaces. Un'opzione è quella di connettersi utilizzando credenziali specifiche del servizio. Le credenziali specifiche del servizio sono un nome utente e una password associati a un utente IAM specifico e possono essere utilizzate solo con il servizio specificato. La seconda opzione consiste nel connettersi con le credenziali IAM che utilizzano il [processo AWS Signature Version 4 \(SigV4\)](#). Per ulteriori informazioni su queste due opzioni, consulta [the section called “Crea credenziali di accesso programmatico”](#)

Per connetterti con credenziali specifiche del servizio, consulta [the section called “Connect con credenziali specifiche del servizio”](#)

Per connetterti con le credenziali IAM, consulta [the section called “Connessione con le credenziali IAM”](#)

## Connect ad Amazon Keyspaces con credenziali specifiche del servizio

Questa sezione mostra come utilizzare le credenziali specifiche del servizio per eseguire il commit del modello di dati creato o modificato con NoSQL Workbench.

1. Per creare una nuova connessione utilizzando credenziali specifiche del servizio, scegli la scheda **Connetti** utilizzando nome utente e password.
  - Prima di iniziare, è necessario creare credenziali specifiche del servizio utilizzando il processo documentato in [the section called “Crea credenziali specifiche per il servizio”](#)

Dopo aver ottenuto le credenziali specifiche del servizio, è possibile continuare a configurare la connessione. Continua con una delle seguenti operazioni:

- Nome utente: immettere il nome utente.
- Password: immettere la password.
- Regione AWS— Per le regioni disponibili, vedere [the section called “Endpoint di servizio”](#).
- Porta: Amazon Keyspaces utilizza la porta 9142.

In alternativa, puoi importare le credenziali salvate da un file.

2. Scegli **Commit** per aggiornare Amazon Keyspaces con il modello di dati.

## Commit to Amazon Keyspaces

 On this page, you can create server-side resources such as keyspaces and tables for the chosen data model.



< Use saved connections    Connect by using IAM credentials    Connect by using user name >



You can generate service-specific credentials to allow your users to access Amazon Keyspaces using AWS Management Console or AWS CLI.

[How to generate Amazon Keyspaces credentials](#)

\* User Name

anika

\* Password

.....



\* AWS Region

us-east-1



\* Port

9142

OR

 Import from credential file

Cancel

Reset

Commit

## Connect ad Amazon Keyspaces con credenziali AWS Identity and Access Management (IAM)

Questa sezione mostra come utilizzare le credenziali IAM per eseguire il commit del modello di dati creato o modificato con NoSQL Workbench.

1. Per creare una nuova connessione utilizzando le credenziali IAM, scegli la scheda Connect by using IAM credentials.
  - Prima di iniziare, devi creare le credenziali IAM utilizzando uno dei seguenti metodi.
    - Per l'accesso alla console, utilizza il nome utente IAM e la relativa password per accedere alla pagina [Console di gestione AWS](#) di accesso IAM. Per informazioni sulle credenziali AWS di sicurezza, incluso l'accesso programmatico e le alternative alle credenziali a lungo termine, consulta le credenziali di [AWS sicurezza nella Guida per l'utente IAM](#). Per i dettagli sull'accesso al tuo Account AWS, consulta [Come accedere a nella Guida per l'AWS](#)utente.Accedi ad AWS
    - Per l'accesso alla CLI, sono necessari un ID chiave di accesso e una chiave di accesso segreta. Utilizza credenziali temporanee al posto delle chiavi di accesso a lungo termine quando possibile. Le credenziali temporanee includono un ID della chiave di accesso, una chiave di accesso segreta e un token di sicurezza che ne indica la scadenza. Per ulteriori informazioni, consulta [Using temporary credentials with AWS resources](#) nella IAM User Guide.
    - Per l'accesso alle API sono necessari un ID chiave di accesso e una chiave di accesso segreta. Utilizza le chiavi di accesso utente IAM invece delle chiavi di accesso Utente root dell'account AWS . Per ulteriori informazioni sulla creazione delle chiavi di accesso, consulta [Manage access keys for IAM users](#) nella IAM User Guide.

Per ulteriori informazioni, consulta [Gestione delle chiavi di accesso per gli utenti IAM](#).

Dopo aver ottenuto le credenziali IAM, puoi continuare a configurare la connessione.

- Nome della connessione: il nome della connessione.
- Regione AWS— Per le regioni disponibili, vedere [the section called “Endpoint di servizio”](#).
- ID della chiave di accesso: immettere l'ID della chiave di accesso.
- Chiave di accesso segreta: immettere la chiave di accesso segreta.

- Porta: Amazon Keyspaces utilizza la porta 9142.
  - AWS certificato pubblico: indica il AWS certificato che è stato scaricato nel primo passaggio.
  - Connessione persistente: seleziona questa casella di controllo se desideri salvare i segreti di AWS connessione localmente.
2. Scegli Commit per aggiornare Amazon Keyspaces con il modello di dati.

**i** On this page, you can create server-side resources such as keyspaces and tables for the chosen data model.

< Use saved connections Connect by using IAM credentials Connect by using user name >

\* Connection name

Connection 2

\* AWS Region

us-east-2

\* Access key ID

AKIAIOSFODNN7EXAMPLE

\* Secret access key

.....



\* Port

9142

\* AWS public certificate

⬇ Choose AWS public certificate

AmazonRootCA1.pem

Choose an AWS public certificate for a Signature Version 4–based connection to Amazon Keyspaces. **i**

Persist connection



If you select this check box, AWS connection secrets will be persisted in

C:\Users\awsuser\aws\credentials

Cancel

Reset

Commit



## Usa una connessione salvata

Se in precedenza hai configurato una connessione ad Amazon Keyspaces, puoi utilizzarla come connessione predefinita per eseguire modifiche al modello di dati. Scegli la scheda Usa connessioni salvate e continua a eseguire il commit degli aggiornamenti.

### Commit to Amazon Keyspaces



On this page, you can create server-side resources such as keyspaces and tables for the chosen data model.

< **Use saved connections**

Connect by using IAM credentials

Connect by using user name : >

\* Saved connections

default



\* Port

9142

\* AWS public certificate

[Choose AWS public certificate](#)

AmazonRootCA1.pem

Choose an AWS public certificate for a Signature Version 4-based connection to Amazon Keyspaces.

Cancel

Reset

Commit

## Impegnati con Apache Cassandra

Questa sezione illustra come effettuare le connessioni a un cluster Apache Cassandra per eseguire il commit del modello di dati creato o modificato con NoSQL Workbench.

 Note

Solo i modelli di dati che sono stati creati `SimpleStrategy` o che `NetworkTopologyStrategy` possono essere salvati nei cluster Apache Cassandra. Per modificare la strategia di replica, modifica lo spazio delle chiavi nel data modeler.

1.
  - Nome utente: immettere il nome utente se l'autenticazione è abilitata nel cluster.
  - Password: immettere la password se l'autenticazione è abilitata nel cluster.
  - Punti di contatto: inserire i punti di contatto.
  - Centro dati locale: immettere il nome del data center locale.
  - Porta: la connessione utilizza la porta 9042.
2. Scegli Commit per aggiornare il cluster Apache Cassandra con il modello di dati.

## Commit to Apache Cassandra



Configure the connection to the Apache Cassandra cluster so that you can commit your data model to the database, including keyspaces, tables, and sample data. The user name and password are not required, and are necessary only if authentication is enabled on the cluster

User name

Password

\* Contact points

[+ Add another contact point](#)

\* Local data center

\* Port

Cancel

Reset

Commit

## Modelli di dati di esempio in NoSQL Workbench

La home page del modellatore e del visualizzatore mostra una serie di modelli di esempio forniti con NoSQL Workbench. In questa sezione vengono descritti questi modelli e i loro potenziali utilizzi.

### Argomenti

- [Modello di dati del dipendente](#)
- [Modello di dati sulle transazioni con carta di credito](#)
- [Modello di dati operativi delle compagnie aeree](#)

### Modello di dati del dipendente

Questo modello di dati rappresenta uno schema Amazon Keyspaces per un'applicazione di database per dipendenti.

Le applicazioni che accedono alle informazioni sui dipendenti di una determinata azienda possono utilizzare questo modello di dati.

I modelli di accesso supportati da questo modello di dati sono:

- Recupero del record di un dipendente con un determinato ID.
- Recupero del record di un dipendente con un determinato ID e divisione.
- Recupero del record di un dipendente con un determinato ID e nome.

### Modello di dati sulle transazioni con carta di credito

Questo modello di dati rappresenta uno schema Amazon Keyspaces per le transazioni con carta di credito nei negozi al dettaglio.

L'archiviazione delle transazioni con carta di credito non solo aiuta i negozi a tenere la contabilità, ma aiuta anche i gestori dei negozi ad analizzare le tendenze di acquisto, il che può aiutarli a prevedere e pianificare.

I modelli di accesso supportati da questo modello di dati sono:

- Recupero delle transazioni per numero di carta di credito, mese, anno e data.
- Recupero delle transazioni per numero di carta di credito, categoria e data.
- Recupero delle transazioni per categoria, ubicazione e numero di carta di credito.

- Recupero delle transazioni in base al numero di carta di credito e allo stato della controversia.

## Modello di dati operativi delle compagnie aeree

Questo modello di dati mostra i dati sui voli aerei, inclusi aeroporti, compagnie aeree e rotte aeree.

I componenti chiave della modellazione Amazon Keyspaces dimostrati sono coppie chiave-valore, archivi dati a colonna larga, chiavi composite e tipi di dati complessi come mappe per dimostrare modelli di accesso ai dati NoSQL comuni.

I modelli di accesso supportati da questo modello di dati sono:

- Recupero di rotte provenienti da una determinata compagnia aerea in un determinato aeroporto.
- Recupero di rotte con un determinato aeroporto di destinazione.
- Recupero di aeroporti con voli diretti.
- Recupero dei dati dell'aeroporto e della compagnia aerea.

## Cronologia delle versioni di NoSQL Workbench

La tabella seguente descrive le modifiche importanti in ogni versione dell'applicazione lato client NoSQL Workbench.

| Modifica                                                     | Descrizione                                                                                                                                                                                                                                                   | Data            |
|--------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| Workbench NoSQL per Amazon Keyspaces — GA.                   | NoSQL Workbench per Amazon Keyspaces è generalmente disponibile.                                                                                                                                                                                              | 28 ottobre 2020 |
| Non è stata rilasciata nessuna anteprima di NoSQL Workbench. | NoSQL Workbench è un'applicazione lato client che ti aiuta a progettare e visualizzare più facilmente modelli di dati non relazionali per Amazon Keyspaces. I client NoSQL Workbench sono disponibili per Windows, macOS e Linux. Per ulteriori informazioni, | 5 ottobre 2020  |

| Modifica | Descrizione                                                     | Data |
|----------|-----------------------------------------------------------------|------|
|          | consulta <a href="#">NoSQL Workbench for Amazon Keyspaces</a> . |      |

# Esempi di codice per l'utilizzo di Amazon Keyspaces AWS SDKs

I seguenti esempi di codice mostrano come usare Amazon Keyspaces con un kit di sviluppo AWS software (SDK).

Le nozioni di base sono esempi di codice che mostrano come eseguire le operazioni essenziali all'interno di un servizio.

Le operazioni sono estratti di codice da programmi più grandi e devono essere eseguite nel contesto. Sebbene le operazioni mostrino come richiamare le singole funzioni del servizio, è possibile visualizzarle contestualizzate negli scenari correlati.

Per un elenco completo di guide ed esempi di codice per sviluppatori AWS SDK, consulta. [Utilizzo di questo servizio con un SDK AWS](#) Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

Nozioni di base

## Salve Amazon Keyspaces

I seguenti esempi di codice mostrano come iniziare a utilizzare Amazon Keyspaces.

.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 KeyspacesActions;

public class HelloKeyspaces
{
 private static ILogger logger = null!;
```

```

static async Task Main(string[] args)
{
 // Set up dependency injection for Amazon Keyspaces (for Apache
 Cassandra).
 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<IAmazonKeyspaces>()
 .AddTransient<KeyspacesWrapper>()
)
 .Build();

 logger = LoggerFactory.Create(builder => { builder.AddConsole(); })
 .CreateLogger<HelloKeyspaces>();

 var keyspacesClient =
 host.Services.GetRequiredService<IAmazonKeyspaces>();
 var keyspacesWrapper = new KeyspacesWrapper(keyspacesClient);

 Console.WriteLine("Hello, Amazon Keyspaces! Let's list your keyspaces:");
 await keyspacesWrapper.ListKeyspaces();
}
}

```

- Per i dettagli sull'API, consulta la [ListKeyspaces](#) sezione AWS SDK per .NET 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](#).



```
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.keyspaces.KeyspacesClient;
import software.amazon.awssdk.services.keyspaces.model.KeyspaceSummary;
import software.amazon.awssdk.services.keyspaces.model.KeyspacesException;
import software.amazon.awssdk.services.keyspaces.model.ListKeyspacesRequest;
import software.amazon.awssdk.services.keyspaces.model.ListKeyspacesResponse;
import java.util.List;

/**
 * Before running this Java (v2) code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 */
public class HelloKeyspaces {
 public static void main(String[] args) {
 Region region = Region.US_EAST_1;
 KeyspacesClient keyClient = KeyspacesClient.builder()
 .region(region)
 .build();

 listKeyspaces(keyClient);
 }

 public static void listKeyspaces(KeyspacesClient keyClient) {
 try {
 ListKeyspacesRequest keyspacesRequest =
 ListKeyspacesRequest.builder()
 .maxResults(10)
 .build();

 ListKeyspacesResponse response =
 keyClient.listKeyspaces(keyspacesRequest);
 List<KeyspaceSummary> keyspaces = response.keyspaces();
 for (KeyspaceSummary keyspace : keyspaces) {
 System.out.println("The name of the keyspace is " +
 keyspace.keyspaceName());
 }
 } catch (KeyspacesException e) {
```

```
 System.err.println(e.awsErrorDetails().errorMessage());
 System.exit(1);
 }
}
```

- Per i dettagli sull'API, consulta la [ListKeyspaces](#) sezione AWS SDK for Java 2.x 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](#).

```
/**
Before running this Kotlin code example, set up your development environment,
including your credentials.

For more information, see the following documentation topic:

https://docs.aws.amazon.com/sdk-for-kotlin/latest/developer-guide/setup.html
*/

suspend fun main() {
 listKeyspaces()
}

suspend fun listKeyspaces() {
 val keyspacesRequest =
 ListKeyspacesRequest {
 maxResults = 10
 }

 KeyspacesClient.fromEnvironment { region = "us-east-1" }.use { keyClient ->
```

```

 val response = keyClient.listKeyspaces(keyspacesRequest)
 response.keyspaces?.forEach { keyspace ->
 println("The name of the keyspace is ${keyspace.keyspaceName}")
 }
 }
}

```

- Per i dettagli sull'API, [ListKeyspaces](#) consulta AWS SDK for Kotlin 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

def hello_keyspaces(keyspaces_client):
 """
 Use the AWS SDK for Python (Boto3) to create an Amazon Keyspaces (for Apache
 Cassandra)
 client and list the keyspaces in your account.
 This example uses the default settings specified in your shared credentials
 and config files.

 :param keyspaces_client: A Boto3 Amazon Keyspaces Client object. This object
 wraps
 the low-level Amazon Keyspaces service API.
 """
 print("Hello, Amazon Keyspaces! Let's list some of your keyspaces:\n")
 for ks in keyspaces_client.list_keyspaces(maxResults=5).get("keyspaces", []):
 print(ks["keyspaceName"])
 print(f"\t{ks['resourceArn']}")

if __name__ == "__main__":

```

```
hello_keyspaces(boto3.client("keyspaces"))
```

- Per i dettagli sull'API, consulta [ListKeyspaces AWS SDK for Python \(Boto3\) API Reference](#).

## Esempi di codice

- [Esempi di base per l'utilizzo di Amazon Keyspaces AWS SDKs](#)
  - [Salve Amazon Keyspaces](#)
  - [Scopri le nozioni di base di Amazon Keyspaces con un SDK AWS](#)
  - [Azioni per l'utilizzo di Amazon Keyspaces AWS SDKs](#)
    - [Utilizzalo CreateKeyspace con un SDK AWS](#)
    - [Utilizzare CreateTable con un SDK AWS](#)
    - [Utilizzare DeleteKeyspace con un SDK AWS](#)
    - [Utilizzare DeleteTable con un SDK AWS](#)
    - [Utilizzare GetKeyspace con un SDK AWS](#)
    - [Utilizzare GetTable con un SDK AWS](#)
    - [Utilizzare ListKeyspaces con un SDK AWS](#)
    - [Utilizzare ListTables con un SDK AWS](#)
    - [Utilizzare RestoreTable con un SDK AWS](#)
    - [Utilizzare UpdateTable con un SDK AWS](#)

## Esempi di base per l'utilizzo di Amazon Keyspaces AWS SDKs

I seguenti esempi di codice mostrano come utilizzare le basi di Amazon Keyspaces (per Apache Cassandra) con AWS SDKs

### Esempi

- [Salve Amazon Keyspaces](#)
- [Scopri le nozioni di base di Amazon Keyspaces con un SDK AWS](#)
- [Azioni per l'utilizzo di Amazon Keyspaces AWS SDKs](#)
  - [Utilizzalo CreateKeyspace con un SDK AWS](#)
  - [Utilizzare CreateTable con un SDK AWS](#)
  - [Utilizzare DeleteKeyspace con un SDK AWS](#)

- [Utilizzare DeleteTable con un SDK AWS](#)
- [Utilizzare GetKeyspace con un SDK AWS](#)
- [Utilizzare GetTable con un SDK AWS](#)
- [Utilizzare ListKeyspaces con un SDK AWS](#)
- [Utilizzare ListTables con un SDK AWS](#)
- [Utilizzare RestoreTable con un SDK AWS](#)
- [Utilizzare UpdateTable con un SDK AWS](#)

## Salve Amazon Keyspaces

I seguenti esempi di codice mostrano come iniziare a utilizzare Amazon Keyspaces.

.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 KeyspacesActions;

public class HelloKeyspaces
{
 private static ILogger logger = null!;

 static async Task Main(string[] args)
 {
 // Set up dependency injection for Amazon Keyspaces (for Apache
 // Cassandra).
 using var host = Host.CreateDefaultBuilder(args)
 .ConfigureLogging(logging =>
 logging.AddFilter("System", LogLevel.Debug)
 .AddFilter<DebugLoggerProvider>("Microsoft",
 LogLevel.Information)
 .AddFilter<ConsoleLoggerProvider>("Microsoft",
 LogLevel.Trace))
 .Build();
 }
}
```

```

 .ConfigureServices((_, services) =>
 services.AddAWSService<IAmazonKeyspaces>()
 .AddTransient<KeyspacesWrapper>()
)
 .Build();

logger = LoggerFactory.Create(builder => { builder.AddConsole(); })
 .CreateLogger<HelloKeyspaces>();

var keyspacesClient =
host.Services.GetRequiredService<IAmazonKeyspaces>();
var keyspacesWrapper = new KeyspacesWrapper(keyspacesClient);

Console.WriteLine("Hello, Amazon Keyspaces! Let's list your keyspaces:");
await keyspacesWrapper.ListKeyspaces();
 }
}

```

- Per i dettagli sull'API, consulta la [ListKeyspaces](#) sezione AWS SDK per .NET 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](#).

```

import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.keyspaces.KeyspacesClient;
import software.amazon.awssdk.services.keyspaces.model.KeyspaceSummary;
import software.amazon.awssdk.services.keyspaces.model.KeyspacesException;
import software.amazon.awssdk.services.keyspaces.model.ListKeyspacesRequest;
import software.amazon.awssdk.services.keyspaces.model.ListKeyspacesResponse;
import java.util.List;

/**
 * Before running this Java (v2) code example, set up your development

```

```
* environment, including your credentials.
*
* For more information, see the following documentation topic:
*
* https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
*/
public class HelloKeyspaces {
 public static void main(String[] args) {
 Region region = Region.US_EAST_1;
 KeyspacesClient keyClient = KeyspacesClient.builder()
 .region(region)
 .build();

 listKeyspaces(keyClient);
 }

 public static void listKeyspaces(KeyspacesClient keyClient) {
 try {
 ListKeyspacesRequest keyspacesRequest =
ListKeyspacesRequest.builder()
 .maxResults(10)
 .build();

 ListKeyspacesResponse response =
keyClient.listKeyspaces(keyspacesRequest);
 List<KeyspaceSummary> keyspaces = response.keyspaces();
 for (KeyspaceSummary keyspace : keyspaces) {
 System.out.println("The name of the keyspace is " +
keyspace.keyspaceName());
 }

 } catch (KeyspacesException e) {
 System.err.println(e.awsErrorDetails().errorMessage());
 System.exit(1);
 }
 }
}
```

- Per i dettagli sull'API, consulta la [ListKeyspaces](#) sezione AWS SDK for Java 2.x 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](#).

```
/**
Before running this Kotlin code example, set up your development environment,
including your credentials.

For more information, see the following documentation topic:

https://docs.aws.amazon.com/sdk-for-kotlin/latest/developer-guide/setup.html
*/

suspend fun main() {
 listKeyspaces()
}

suspend fun listKeyspaces() {
 val keyspacesRequest =
 ListKeyspacesRequest {
 maxResults = 10
 }

 KeyspacesClient.fromEnvironment { region = "us-east-1" }.use { keyClient ->
 val response = keyClient.listKeyspaces(keyspacesRequest)
 response.keyspaces?.forEach { keyspace ->
 println("The name of the keyspace is ${keyspace.keyspaceName}")
 }
 }
}
```

- Per i dettagli sull'API, [ListKeyspaces](#) consulta AWS SDK for Kotlin 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

def hello_keyspaces(keyspaces_client):
 """
 Use the AWS SDK for Python (Boto3) to create an Amazon Keyspaces (for Apache
 Cassandra)
 client and list the keyspaces in your account.
 This example uses the default settings specified in your shared credentials
 and config files.

 :param keyspaces_client: A Boto3 Amazon Keyspaces Client object. This object
 wraps
 the low-level Amazon Keyspaces service API.
 """
 print("Hello, Amazon Keyspaces! Let's list some of your keyspaces:\n")
 for ks in keyspaces_client.list_keyspaces(maxResults=5).get("keyspaces", []):
 print(ks["keyspaceName"])
 print(f"\t{ks['resourceArn']}")

if __name__ == "__main__":
 hello_keyspaces(boto3.client("keyspaces"))
```

- Per i dettagli sull'API, consulta [ListKeyspaces AWS](#) SDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. [Utilizzo di questo servizio con un SDK AWS](#) Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

# Scopri le nozioni di base di Amazon Keyspaces con un SDK AWS

Gli esempi di codice seguenti mostrano come:

- Crea uno spazio di chiavi e una tabella. Lo schema della tabella contiene i dati dei film e il point-in-time ripristino è abilitato.
- Connect al keyspace utilizzando una connessione TLS sicura con autenticazione SigV4.
- Esegui una query sulla tabella. Aggiungi, recupera e aggiorna i dati dei film.
- Aggiorna la tabella. Aggiungi una colonna per tenere traccia dei film guardati.
- Ripristina lo stato precedente della tabella e ripulisci le risorse.

.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](#).

```
global using System.Security.Cryptography.X509Certificates;
global using Amazon.Keyspaces;
global using Amazon.Keyspaces.Model;
global using KeyspacesActions;
global using KeyspacesScenario;
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;
global using Newtonsoft.Json;

namespace KeyspacesBasics;

/// <summary>
/// Amazon Keyspaces (for Apache Cassandra) scenario. Shows some of the basic
```

```
/// actions performed with Amazon Keyspaces.
/// </summary>
public class KeyspacesBasics
{
 private static ILogger logger = null!;

 static async Task Main(string[] args)
 {
 // Set up dependency injection for the Amazon service.
 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<IAmazonKeyspaces>()
 .AddTransient<KeyspacesWrapper>()
 .AddTransient<CassandraWrapper>()
)
 .Build();

 logger = LoggerFactory.Create(builder => { builder.AddConsole(); })
 .CreateLogger<KeyspacesBasics>();

 var configuration = new ConfigurationBuilder()
 .SetBasePath(Directory.GetCurrentDirectory())
 .AddJsonFile("settings.json") // Load test settings from .json file.
 .AddJsonFile("settings.local.json",
 true) // Optionally load local settings.
 .Build();

 var keyspacesWrapper =
 host.Services.GetRequiredService<KeyspacesWrapper>();
 var uiMethods = new UiMethods();

 var keyspaceName = configuration["KeyspaceName"];
 var tableName = configuration["TableName"];

 bool success; // Used to track the results of some operations.

 uiMethods.DisplayOverview();
 uiMethods.PressEnter();
 }
}
```

```
// Create the keyspace.
var keyspaceArn = await keyspacesWrapper.CreateKeyspace(keyspaceName);

// Wait for the keyspace to be available. GetKeyspace results in a
// resource not found error until it is ready for use.
try
{
 var getKeyspaceArn = "";
 Console.WriteLine($"Created {keyspaceName}. Waiting for it to become
available. ");
 do
 {
 getKeyspaceArn = await
keyspacesWrapper.GetKeyspace(keyspaceName);
 Console.WriteLine(". ");
 } while (getKeyspaceArn != keyspaceArn);
}
catch (ResourceNotFoundException)
{
 Console.WriteLine("Waiting for keyspace to be created.");
}

Console.WriteLine($"\\n\\nThe keyspace {keyspaceName} is ready for use.");

uiMethods.PressEnter();

// Create the table.
// First define the schema.
var allColumns = new List<ColumnDefinition>
{
 new ColumnDefinition { Name = "title", Type = "text" },
 new ColumnDefinition { Name = "year", Type = "int" },
 new ColumnDefinition { Name = "release_date", Type = "timestamp" },
 new ColumnDefinition { Name = "plot", Type = "text" },
};

var partitionKeys = new List<PartitionKey>
{
 new PartitionKey { Name = "year", },
 new PartitionKey { Name = "title" },
};

var tableSchema = new SchemaDefinition
```

```

 {
 AllColumns = allColumns,
 PartitionKeys = partitionKeys,
 };

 var tableArn = await keyspacesWrapper.CreateTable(keyspaceName,
tableSchema, tableName);

 // Wait for the table to be active.
 try
 {
 var resp = new GetTableResponse();
 Console.WriteLine("Waiting for the new table to be active. ");
 do
 {
 try
 {
 resp = await keyspacesWrapper.GetTable(keyspaceName,
tableName);

 Console.WriteLine(".");
 }
 catch (ResourceNotFoundException)
 {
 Console.WriteLine(".");
 }
 } while (resp.Status != TableStatus.ACTIVE);

 // Display the table's schema.
 Console.WriteLine($"\\nTable {tableName} has been created in
{keyspaceName}");
 Console.WriteLine("Let's take a look at the schema.");
 uiMethods.DisplayTitle("All columns");
 resp.SchemaDefinition.AllColumns.ForEach(column =>
 {
 Console.WriteLine($"{column.Name, -40}\\t{column.Type, -20}");
 });

 uiMethods.DisplayTitle("Cluster keys");
 resp.SchemaDefinition.ClusteringKeys.ForEach(clusterKey =>
 {
 Console.WriteLine($"{clusterKey.Name, -40}\\t{clusterKey.OrderBy, -20}");
 });
 }

```

```
 uiMethods.DisplayTitle("Partition keys");
 resp.SchemaDefinition.PartitionKeys.ForEach(partitionKey =>
 {
 Console.WriteLine($"{partitionKey.Name}");
 });

 uiMethods.PressEnter();
 }
 catch (ResourceNotFoundException ex)
 {
 Console.WriteLine($"Error: {ex.Message}");
 }

 // Access Apache Cassandra using the Cassandra drive for C#.
 var cassandraWrapper =
host.Services.GetRequiredService<CassandraWrapper>();
 var movieFilePath = configuration["MovieFile"];

 Console.WriteLine("Let's add some movies to the table we created.");
 var inserted = await cassandraWrapper.InsertIntoMovieTable(keyspaceName,
tableName, movieFilePath);

 uiMethods.PressEnter();

 Console.WriteLine("Added the following movies to the table:");
 var rows = await cassandraWrapper.GetMovies(keyspaceName, tableName);
 uiMethods.DisplayTitle("All Movies");

 foreach (var row in rows)
 {
 var title = row.GetValue<string>("title");
 var year = row.GetValue<int>("year");
 var plot = row.GetValue<string>("plot");
 var release_date = row.GetValue<DateTime>("release_date");
 Console.WriteLine($"{release_date}\t{title}\t{year}\n{plot}");
 Console.WriteLine(uiMethods.SepBar);
 }

 // Update the table schema
 uiMethods.DisplayTitle("Update table schema");
 Console.WriteLine("Now we will update the table to add a boolean field
called watched.");

 // First save the current time as a UTC Date so the original
```

```
// table can be restored later.
var timeChanged = DateTime.UtcNow;

// Now update the schema.
var resourceArn = await keyspacesWrapper.UpdateTable(keyspaceName,
tableName);
uiMethods.PressEnter();

Console.WriteLine("Now let's mark some of the movies as watched.");

// Pick some files to mark as watched.
var movieToWatch = rows[2].GetValue<string>("title");
var watchedMovieYear = rows[2].GetValue<int>("year");
var changedRows = await cassandraWrapper.MarkMovieAsWatched(keyspaceName,
tableName, movieToWatch, watchedMovieYear);

movieToWatch = rows[6].GetValue<string>("title");
watchedMovieYear = rows[6].GetValue<int>("year");
changedRows = await cassandraWrapper.MarkMovieAsWatched(keyspaceName,
tableName, movieToWatch, watchedMovieYear);

movieToWatch = rows[9].GetValue<string>("title");
watchedMovieYear = rows[9].GetValue<int>("year");
changedRows = await cassandraWrapper.MarkMovieAsWatched(keyspaceName,
tableName, movieToWatch, watchedMovieYear);

movieToWatch = rows[10].GetValue<string>("title");
watchedMovieYear = rows[10].GetValue<int>("year");
changedRows = await cassandraWrapper.MarkMovieAsWatched(keyspaceName,
tableName, movieToWatch, watchedMovieYear);

movieToWatch = rows[13].GetValue<string>("title");
watchedMovieYear = rows[13].GetValue<int>("year");
changedRows = await cassandraWrapper.MarkMovieAsWatched(keyspaceName,
tableName, movieToWatch, watchedMovieYear);

uiMethods.DisplayTitle("Watched movies");
Console.WriteLine("These movies have been marked as watched:");
rows = await cassandraWrapper.GetWatchedMovies(keyspaceName, tableName);
foreach (var row in rows)
{
 var title = row.GetValue<string>("title");
 var year = row.GetValue<int>("year");
 Console.WriteLine($"{title,-40}\t{year,8}");
}
```

```

 }
 uiMethods.PressEnter();

 Console.WriteLine("We can restore the table to its previous state but
that can take up to 20 minutes to complete.");
 string answer;
 do
 {
 Console.WriteLine("Do you want to restore the table? (y/n)");
 answer = Console.ReadLine();
 } while (answer.ToLower() != "y" && answer.ToLower() != "n");

 if (answer == "y")
 {
 var restoredTableName = $"{tableName}_restored";
 var restoredTableArn = await keyspacesWrapper.RestoreTable(
 keyspaceName,
 tableName,
 restoredTableName,
 timeChanged);
 // Loop and call GetTable until the table is gone. Once it has been
 // deleted completely, GetTable will raise a
ResourceNotFoundException.
 bool wasRestored = false;

 try
 {
 do
 {
 var resp = await keyspacesWrapper.GetTable(keyspaceName,
restoredTableName);
 wasRestored = (resp.Status == TableStatus.ACTIVE);
 } while (!wasRestored);
 }
 catch (ResourceNotFoundException)
 {
 // If the restored table raised an error, it isn't
 // ready yet.
 Console.Write(".");
 }
 }

 uiMethods.DisplayTitle("Clean up resources.");

```



```

 // Delete the table.
 success = await keyspacesWrapper.DeleteTable(keyspaceName, tableName);

 Console.WriteLine($"Table {tableName} successfully deleted from
{keyspaceName}.");
 Console.WriteLine("Waiting for the table to be removed completely. ");

 // Loop and call GetTable until the table is gone. Once it has been
 // deleted completely, GetTable will raise a ResourceNotFoundException.
 bool wasDeleted = false;

 try
 {
 do
 {
 var resp = await keyspacesWrapper.GetTable(keyspaceName,
tableName);
 } while (!wasDeleted);
 }
 catch (ResourceNotFoundException ex)
 {
 wasDeleted = true;
 Console.WriteLine($"{ex.Message} indicates that the table has been
deleted.");
 }

 // Delete the keyspace.
 success = await keyspacesWrapper.DeleteKeyspace(keyspaceName);
 Console.WriteLine("The keyspace has been deleted and the demo is now
complete.");
 }
}

```

```

namespace KeyspacesActions;

/// <summary>
/// Performs Amazon Keyspaces (for Apache Cassandra) actions.
/// </summary>
public class KeyspacesWrapper
{
 private readonly IAmazonKeyspaces _amazonKeyspaces;

```

```
/// <summary>
/// Constructor for the KeyspaceWrapper.
/// </summary>
/// <param name="amazonKeyspaces">An Amazon Keyspaces client object.</param>
public KeyspacesWrapper(IAmazonKeyspaces amazonKeyspaces)
{
 _amazonKeyspaces = amazonKeyspaces;
}

/// <summary>
/// Create a new keyspace.
/// </summary>
/// <param name="keyspaceName">The name for the new keyspace.</param>
/// <returns>The Amazon Resource Name (ARN) of the new keyspace.</returns>
public async Task<string> CreateKeyspace(string keyspaceName)
{
 var response =
 await _amazonKeyspaces.CreateKeyspaceAsync(
 new CreateKeyspaceRequest { KeyspaceName = keyspaceName });
 return response.ResourceArn;
}

/// <summary>
/// Create a new Amazon Keyspaces table.
/// </summary>
/// <param name="keyspaceName">The keyspace where the table will be
created.</param>
/// <param name="schema">The schema for the new table.</param>
/// <param name="tableName">The name of the new table.</param>
/// <returns>The Amazon Resource Name (ARN) of the new table.</returns>
public async Task<string> CreateTable(string keyspaceName, SchemaDefinition
schema, string tableName)
{
 var request = new CreateTableRequest
 {
 KeyspaceName = keyspaceName,
 SchemaDefinition = schema,
 TableName = tableName,
 PointInTimeRecovery = new PointInTimeRecovery { Status =
PointInTimeRecoveryStatus.ENABLED }
 };
};
```

```

 var response = await _amazonKeyspaces.CreateTableAsync(request);
 return response.ResourceArn;
 }

 /// <summary>
 /// Delete an existing keyspace.
 /// </summary>
 /// <param name="keyspaceName"></param>
 /// <returns>A Boolean value indicating the success of the action.</returns>
 public async Task<bool> DeleteKeyspace(string keyspaceName)
 {
 var response = await _amazonKeyspaces.DeleteKeyspaceAsync(
 new DeleteKeyspaceRequest { KeyspaceName = keyspaceName });
 return response.HttpStatusCode == HttpStatusCode.OK;
 }

 /// <summary>
 /// Delete an Amazon Keyspaces table.
 /// </summary>
 /// <param name="keyspaceName">The keyspace containing the table.</param>
 /// <param name="tableName">The name of the table to delete.</param>
 /// <returns>A Boolean value indicating the success of the action.</returns>
 public async Task<bool> DeleteTable(string keyspaceName, string tableName)
 {
 var response = await _amazonKeyspaces.DeleteTableAsync(
 new DeleteTableRequest { KeyspaceName = keyspaceName, TableName =
tableName });
 return response.HttpStatusCode == HttpStatusCode.OK;
 }

 /// <summary>
 /// Get data about a keyspace.
 /// </summary>
 /// <param name="keyspaceName">The name of the keyspace.</param>
 /// <returns>The Amazon Resource Name (ARN) of the keyspace.</returns>
 public async Task<string> GetKeyspace(string keyspaceName)
 {
 var response = await _amazonKeyspaces.GetKeyspaceAsync(
 new GetKeyspaceRequest { KeyspaceName = keyspaceName });
 return response.ResourceArn;
 }

```

```

 /// <summary>
 /// Get information about an Amazon Keyspaces table.
 /// </summary>
 /// <param name="keyspaceName">The keyspace containing the table.</param>
 /// <param name="tableName">The name of the Amazon Keyspaces table.</param>
 /// <returns>The response containing data about the table.</returns>
 public async Task<GetTableResponse> GetTable(string keyspaceName, string
tableName)
 {
 var response = await _amazonKeyspaces.GetTableAsync(
 new GetTableRequest { KeyspaceName = keyspaceName, TableName =
tableName });
 return response;
 }

 /// <summary>
 /// Lists all keyspaces for the account.
 /// </summary>
 /// <returns>Async task.</returns>
 public async Task ListKeyspaces()
 {
 var paginator = _amazonKeyspaces.Paginators.ListKeyspaces(new
ListKeyspacesRequest());

 Console.WriteLine("{0, -30}\t{1}", "Keyspace name", "Keyspace ARN");
 Console.WriteLine(new string('-', Console.WindowWidth));
 await foreach (var keyspace in paginator.Keyspaces)
 {
 Console.WriteLine($"{keyspace.KeyspaceName, -30}\t{keyspace.ResourceArn}");
 }
 }

 /// <summary>
 /// Lists the Amazon Keyspaces tables in a keyspace.
 /// </summary>
 /// <param name="keyspaceName">The name of the keyspace.</param>
 /// <returns>A list of TableSummary objects.</returns>
 public async Task<List<TableSummary>> ListTables(string keyspaceName)
 {

```

```

 var response = await _amazonKeyspaces.ListTablesAsync(new
ListTablesRequest { KeyspaceName = keyspaceName });
 response.Tables.ForEach(table =>
 {

Console.WriteLine($"{table.KeyspaceName}\t{table.TableName}\t{table.ResourceArn}");
 });

 return response.Tables;
 }

 /// <summary>
 /// Restores the specified table to the specified point in time.
 /// </summary>
 /// <param name="keyspaceName">The keyspace containing the table.</param>
 /// <param name="tableName">The name of the table to restore.</param>
 /// <param name="timestamp">The time to which the table will be restored.</
param>
 /// <returns>The Amazon Resource Name (ARN) of the restored table.</returns>
 public async Task<string> RestoreTable(string keyspaceName, string tableName,
string restoredTableName, DateTime timestamp)
 {
 var request = new RestoreTableRequest
 {
 RestoreTimestamp = timestamp,
 SourceKeyspaceName = keyspaceName,
 SourceTableName = tableName,
 TargetKeyspaceName = keyspaceName,
 TargetTableName = restoredTableName
 };

 var response = await _amazonKeyspaces.RestoreTableAsync(request);
 return response.RestoredTableARN;
 }

 /// <summary>
 /// Updates the movie table to add a boolean column named watched.
 /// </summary>
 /// <param name="keyspaceName">The keyspace containing the table.</param>
 /// <param name="tableName">The name of the table to change.</param>
 /// <returns>The Amazon Resource Name (ARN) of the updated table.</returns>
 public async Task<string> UpdateTable(string keyspaceName, string tableName)

```

```

 {
 var newColumn = new ColumnDefinition { Name = "watched", Type =
"boolean" };
 var request = new UpdateTableRequest
 {
 KeyspaceName = keyspaceName,
 TableName = tableName,
 AddColumns = new List<ColumnDefinition> { newColumn }
 };
 var response = await _amazonKeyspaces.UpdateTableAsync(request);
 return response.ResourceArn;
 }
}

```

```

using System.Net;
using Cassandra;

namespace KeyspacesScenario;

/// <summary>
/// Class to perform CRUD methods on an Amazon Keyspaces (for Apache Cassandra)
/// database.
///
/// NOTE: This sample uses a plain text authenticator for example purposes only.
/// Recommended best practice is to use a SigV4 authentication plugin, if
/// available.
/// </summary>
public class CassandraWrapper
{
 private readonly IConfiguration _configuration;
 private readonly string _localPathToFile;
 private const string _certLocation = "https://certs.secureserver.net/
repository/sf-class2-root.crt";
 private const string _certFileName = "sf-class2-root.crt";
 private readonly X509Certificate2Collection _certCollection;
 private X509Certificate2 _amazoncert;
 private Cluster _cluster;

 // User name and password for the service.
 private string _userName = null!;

```

```
private string _pwd = null!;

public CassandraWrapper()
{
 _configuration = new ConfigurationBuilder()
 .SetBasePath(Directory.GetCurrentDirectory())
 .AddJsonFile("settings.json") // Load test settings from .json file.
 .AddJsonFile("settings.local.json",
 true) // Optionally load local settings.
 .Build();

 _localPathToFile = Path.GetTempPath();

 // Get the Starfield digital certificate and save it locally.
 var client = new WebClient();
 client.DownloadFile(_certLocation, $"{_localPathToFile}/
{_certFileName}");

 //var httpClient = new HttpClient();
 //var httpResult = httpClient.Get(fileUrl);
 //using var resultStream = await httpResult.Content.ReadAsStreamAsync();
 //using var fileStream = File.Create(pathToSave);
 //resultStream.CopyTo(fileStream);

 _certCollection = new X509Certificate2Collection();
 _amazoncert = new X509Certificate2($"{_localPathToFile}/
{_certFileName}");

 // Get the user name and password stored in the configuration file.
 _userName = _configuration["UserName"]!;
 _pwd = _configuration["Password"]!;

 // For a list of Service Endpoints for Amazon Keyspaces, see:
 // https://docs.aws.amazon.com/keyspaces/latest/devguide/
 programmatic.endpoints.html
 var awsEndpoint = _configuration["ServiceEndpoint"];

 _cluster = Cluster.Builder()
 .AddContactPoints(awsEndpoint)
 .WithPort(9142)
 .WithAuthProvider(new PlainTextAuthProvider(_userName, _pwd))
 .WithSSL(new SSLOptions().SetCertificateCollection(_certCollection))
 .WithQueryOptions(
 new QueryOptions()
```

```

 .SetConsistencyLevel(ConsistencyLevel.LocalQuorum)
 .SetSerialConsistencyLevel(ConsistencyLevel.LocalSerial))
 .Build();
}

/// <summary>
/// Loads the contents of a JSON file into a list of movies to be
/// added to the Apache Cassandra table.
/// </summary>
/// <param name="movieFileName">The full path to the JSON file.</param>
/// <returns>A list of movie objects.</returns>
public List<Movie> ImportMoviesFromJson(string movieFileName, int numToImport
= 0)
{
 if (!File.Exists(movieFileName))
 {
 return null!;
 }

 using var sr = new StreamReader(movieFileName);
 string json = sr.ReadToEnd();

 var allMovies = JsonConvert.DeserializeObject<List<Movie>>(json);

 // If numToImport = 0, return all movies in the collection.
 if (numToImport == 0)
 {
 // Now return the entire list of movies.
 return allMovies;
 }
 else
 {
 // Now return the first numToImport entries.
 return allMovies.GetRange(0, numToImport);
 }
}

/// <summary>
/// Insert movies into the movie table.
/// </summary>
/// <param name="keyspaceName">The keyspace containing the table.</param>
/// <param name="movieTableName">The Amazon Keyspaces table.</param>
/// <param name="movieFilePath">The path to the resource file containing
/// movie data to insert into the table.</param>

```



```

 /// <returns>A Boolean value indicating the success of the action.</returns>
 public async Task<bool> InsertIntoMovieTable(string keyspaceName, string
movieTableName, string movieFilePath, int numToImport = 20)
 {
 // Get some movie data from the movies.json file
 var movies = ImportMoviesFromJson(movieFilePath, numToImport);

 var session = _cluster.Connect(keyspaceName);

 string insertCql;

 RowSet rs;

 // Now we insert the numToImport movies into the table.
 foreach (var movie in movies)
 {
 // Escape single quote characters in the plot.
 insertCql = $"INSERT INTO {keyspaceName}.{movieTableName}
(title, year, release_date, plot) values($${movie.Title}$$, {movie.Year},
'{movie.Info.Release_Date.ToString("yyyy-MM-dd")} ', $${movie.Info.Plot}$$)";
 rs = await session.ExecuteAsync(new SimpleStatement(insertCql));
 }

 return true;
 }

 /// <summary>
 /// Gets all of the movies in the movies table.
 /// </summary>
 /// <param name="keyspaceName">The keyspace containing the table.</param>
 /// <param name="tableName">The name of the table.</param>
 /// <returns>A list of row objects containing movie data.</returns>
 public async Task<List<Row>> GetMovies(string keyspaceName, string tableName)
 {
 var session = _cluster.Connect();
 RowSet rs;
 try
 {
 rs = await session.ExecuteAsync(new SimpleStatement($"SELECT * FROM
{keyspaceName}.{tableName}"));

 // Extract the row data from the returned RowSet.
 var rows = rs.GetRows().ToList();
 return rows;
 }
 }

```

```

 }
 catch (Exception ex)
 {
 Console.WriteLine(ex.Message);
 return null!;
 }
}

/// <summary>
/// Mark a movie in the movie table as watched.
/// </summary>
/// <param name="keyspaceName">The keyspace containing the table.</param>
/// <param name="tableName">The name of the table.</param>
/// <param name="title">The title of the movie to mark as watched.</param>
/// <param name="year">The year the movie was released.</param>
/// <returns>A set of rows containing the changed data.</returns>
public async Task<List<Row>> MarkMovieAsWatched(string keyspaceName, string
tableName, string title, int year)
{
 var session = _cluster.Connect();
 string updateCql = $"UPDATE {keyspaceName}.{tableName} SET watched=true
WHERE title = ${title} AND year = {year}";
 var rs = await session.ExecuteAsync(new SimpleStatement(updateCql));
 var rows = rs.GetRows().ToList();
 return rows;
}

/// <summary>
/// Retrieve the movies in the movies table where watched is true.
/// </summary>
/// <param name="keyspaceName">The keyspace containing the table.</param>
/// <param name="tableName">The name of the table.</param>
/// <returns>A list of row objects containing information about movies
/// where watched is true.</returns>
public async Task<List<Row>> GetWatchedMovies(string keyspaceName, string
tableName)
{
 var session = _cluster.Connect();
 RowSet rs;
 try
 {
 rs = await session.ExecuteAsync(new SimpleStatement($"SELECT
title, year, plot FROM {keyspaceName}.{tableName} WHERE watched = true ALLOW
FILTERING"));
 }
}

```

```
 // Extract the row data from the returned RowSet.
 var rows = rs.GetRows().ToList();
 return rows;
 }
 catch (Exception ex)
 {
 Console.WriteLine(ex.Message);
 return null!;
 }
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK per .NET .
  - [CreateKeyspace](#)
  - [CreateTable](#)
  - [DeleteKeyspace](#)
  - [DeleteTable](#)
  - [GetKeyspace](#)
  - [GetTable](#)
  - [ListKeyspaces](#)
  - [ListTables](#)
  - [RestoreTable](#)
  - [UpdateTable](#)

## 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.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
 *
 * Before running this Java code example, you must create a
 * Java keystore (JKS) file and place it in your project's resources folder.
 *
 * This file is a secure file format used to hold certificate information for
 * Java applications. This is required to make a connection to Amazon Keyspaces.
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/keyspaces/latest/devguide/using_java_driver.html
 *
 * This Java example performs the following tasks:
 *
 * 1. Create a keyspace.
 * 2. Check for keyspace existence.
 * 3. List keyspaces using a paginator.
 * 4. Create a table with a simple movie data schema and enable point-in-time
 * recovery.
 * 5. Check for the table to be in an Active state.
 * 6. List all tables in the keyspace.
 * 7. Use a Cassandra driver to insert some records into the Movie table.
 * 8. Get all records from the Movie table.
 * 9. Get a specific Movie.
 * 10. Get a UTC timestamp for the current time.
 * 11. Update the table schema to add a 'watched' Boolean column.
 * 12. Update an item as watched.
 * 13. Query for items with watched = True.
 * 14. Restore the table back to the previous state using the timestamp.
 * 15. Check for completion of the restore action.
 * 16. Delete the table.
 * 17. Confirm that both tables are deleted.
 * 18. Delete the keyspace.
 */

public class ScenarioKeyspaces {
```

```
public static final String DASHES = new String(new char[80]).replace("\0",
"-");

/*
 * Usage:
 * fileName - The name of the JSON file that contains movie data. (Get this
file
 * from the GitHub repo at resources/sample_file.)
 * keyspaceName - The name of the keyspace to create.
 */
public static void main(String[] args) throws InterruptedException,
IOException {
 String fileName = "<Replace with the JSON file that contains movie
data>";
 String keyspaceName = "<Replace with the name of the keyspace to
create>";
 String titleUpdate = "The Family";
 int yearUpdate = 2013;
 String tableName = "Movie";
 String tableNameRestore = "MovieRestore";
 Region region = Region.US_EAST_1;
 KeyspacesClient keyClient = KeyspacesClient.builder()
 .region(region)
 .build();

 DriverConfigLoader loader =
DriverConfigLoader.fromClasspath("application.conf");
 CqlSession session = CqlSession.builder()
 .withConfigLoader(loader)
 .build();

 System.out.println(DASHES);
 System.out.println("Welcome to the Amazon Keyspaces example scenario.");
 System.out.println(DASHES);

 System.out.println(DASHES);
 System.out.println("1. Create a keyspace.");
 createKeySpace(keyClient, keyspaceName);
 System.out.println(DASHES);

 System.out.println(DASHES);
 Thread.sleep(5000);
 System.out.println("2. Check for keyspace existence.");
 checkKeyspaceExistence(keyClient, keyspaceName);
}
```

```
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("3. List keyspaces using a paginator.");
listKeyspacesPaginator(keyClient);
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("4. Create a table with a simple movie data schema and
enable point-in-time recovery.");
createTable(keyClient, keyspaceName, tableName);
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("5. Check for the table to be in an Active state.");
Thread.sleep(6000);
checkTable(keyClient, keyspaceName, tableName);
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("6. List all tables in the keyspace.");
listTables(keyClient, keyspaceName);
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("7. Use a Cassandra driver to insert some records into
the Movie table.");
Thread.sleep(6000);
loadData(session, fileName, keyspaceName);
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("8. Get all records from the Movie table.");
getMovieData(session, keyspaceName);
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("9. Get a specific Movie.");
getSpecificMovie(session, keyspaceName);
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("10. Get a UTC timestamp for the current time.");
ZonedDateTime utc = ZonedDateTime.now(ZoneOffset.UTC);
```

```
System.out.println("DATETIME = " + Date.from(utc.toInstant()));
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("11. Update the table schema to add a watched Boolean
column.");
updateTable(keyClient, keyspaceName, tableName);
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("12. Update an item as watched.");
Thread.sleep(10000); // Wait 10 secs for the update.
updateRecord(session, keyspaceName, titleUpdate, yearUpdate);
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("13. Query for items with watched = True.");
getWatchedData(session, keyspaceName);
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("14. Restore the table back to the previous state
using the timestamp.");
System.out.println("Note that the restore operation can take up to 20
minutes.");
restoreTable(keyClient, keyspaceName, utc);
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("15. Check for completion of the restore action.");
Thread.sleep(5000);
checkRestoredTable(keyClient, keyspaceName, "MovieRestore");
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("16. Delete both tables.");
deleteTable(keyClient, keyspaceName, tableName);
deleteTable(keyClient, keyspaceName, tableNameRestore);
System.out.println(DASHES);

System.out.println(DASHES);
System.out.println("17. Confirm that both tables are deleted.");
checkTableDelete(keyClient, keyspaceName, tableName);
checkTableDelete(keyClient, keyspaceName, tableNameRestore);
```

```
 System.out.println(DASHES);

 System.out.println(DASHES);
 System.out.println("18. Delete the keyspace.");
 deleteKeyspace(keyClient, keyspaceName);
 System.out.println(DASHES);

 System.out.println(DASHES);
 System.out.println("The scenario has completed successfully.");
 System.out.println(DASHES);
 }

 public static void deleteKeyspace(KeyspacesClient keyClient, String
keyspaceName) {
 try {
 DeleteKeyspaceRequest deleteKeyspaceRequest =
DeleteKeyspaceRequest.builder()
 .keyspaceName(keyspaceName)
 .build();

 keyClient.deleteKeyspace(deleteKeyspaceRequest);

 } catch (KeyspacesException e) {
 System.err.println(e.awsErrorDetails().errorMessage());
 System.exit(1);
 }
 }

 public static void checkTableDelete(KeyspacesClient keyClient, String
keyspaceName, String tableName)
 throws InterruptedException {
 try {
 String status;
 GetTableResponse response;
 GetTableRequest tableRequest = GetTableRequest.builder()
 .keyspaceName(keyspaceName)
 .tableName(tableName)
 .build();

 // Keep looping until table cannot be found and a
ResourceNotFoundException is
 // thrown.
 while (true) {
 response = keyClient.getTable(tableRequest);
```



```
 status = response.statusAsString();
 System.out.println(". The table status is " + status);
 Thread.sleep(500);
 }

 } catch (ResourceNotFoundException e) {
 System.err.println(e.awsErrorDetails().errorMessage());
 }
 System.out.println("The table is deleted");
}

public static void deleteTable(KeyspacesClient keyClient, String
keyspaceName, String tableName) {
 try {
 DeleteTableRequest tableRequest = DeleteTableRequest.builder()
 .keyspaceName(keyspaceName)
 .tableName(tableName)
 .build();

 keyClient.deleteTable(tableRequest);

 } catch (KeyspacesException e) {
 System.err.println(e.awsErrorDetails().errorMessage());
 System.exit(1);
 }
}

public static void checkRestoredTable(KeyspacesClient keyClient, String
keyspaceName, String tableName)
 throws InterruptedException {
 try {
 boolean tableStatus = false;
 String status;
 GetTableResponse response = null;
 GetTableRequest tableRequest = GetTableRequest.builder()
 .keyspaceName(keyspaceName)
 .tableName(tableName)
 .build();

 while (!tableStatus) {
 response = keyClient.getTable(tableRequest);
 status = response.statusAsString();
 System.out.println("The table status is " + status);
 }
 }
}
```

```

 if (status.compareTo("ACTIVE") == 0) {
 tableStatus = true;
 }
 Thread.sleep(500);
 }

 List<ColumnDefinition> cols =
response.schemaDefinition().allColumns();
 for (ColumnDefinition def : cols) {
 System.out.println("The column name is " + def.name());
 System.out.println("The column type is " + def.type());
 }

 } catch (KeyspacesException e) {
 System.err.println(e.awsErrorDetails().errorMessage());
 System.exit(1);
 }
}

public static void restoreTable(KeyspacesClient keyClient, String
keyspaceName, ZonedDateTime utc) {
 try {
 Instant myTime = utc.toInstant();
 RestoreTableRequest restoreTableRequest =
RestoreTableRequest.builder()
 .restoreTimestamp(myTime)
 .sourceTableName("Movie")
 .targetKeyspaceName(keyspaceName)
 .targetTableName("MovieRestore")
 .sourceKeyspaceName(keyspaceName)
 .build();

 RestoreTableResponse response =
keyClient.restoreTable(restoreTableRequest);
 System.out.println("The ARN of the restored table is " +
response.restoredTableARN());

 } catch (KeyspacesException e) {
 System.err.println(e.awsErrorDetails().errorMessage());
 System.exit(1);
 }
}

public static void getWatchedData(CqlSession session, String keyspaceName) {

```

```

 ResultSet resultSet = session
 .execute("SELECT * FROM \"" + keyspaceName + "\".\"Movie\" WHERE
watched = true ALLOW FILTERING;");
 resultSet.forEach(item -> {
 System.out.println("The Movie title is " + item.getString("title"));
 System.out.println("The Movie year is " + item.getInt("year"));
 System.out.println("The plot is " + item.getString("plot"));
 });
 }

 public static void updateRecord(CqlSession session, String keySpace, String
titleUpdate, int yearUpdate) {
 String sqlStatement = "UPDATE \"" + keySpace
 + "\".\"Movie\" SET watched=true WHERE title = :k0 AND year
= :k1;";
 BatchStatementBuilder builder =
BatchStatement.builder(DefaultBatchType.UNLOGGED);
 builder.setConsistencyLevel(ConsistencyLevel.LOCAL_QUORUM);
 PreparedStatement preparedStatement = session.prepare(sqlStatement);
 builder.addStatement(preparedStatement.boundStatementBuilder()
 .setString("k0", titleUpdate)
 .setInt("k1", yearUpdate)
 .build());

 BatchStatement batchStatement = builder.build();
 session.execute(batchStatement);
 }

 public static void updateTable(KeyspacesClient keyClient, String keySpace,
String tableName) {
 try {
 ColumnDefinition def = ColumnDefinition.builder()
 .name("watched")
 .type("boolean")
 .build();

 UpdateTableRequest tableRequest = UpdateTableRequest.builder()
 .keyspaceName(keySpace)
 .tableName(tableName)
 .addColumnns(def)
 .build();

 keyClient.updateTable(tableRequest);
 }
 }

```

```

 } catch (KeyspacesException e) {
 System.err.println(e.awsErrorDetails().errorMessage());
 System.exit(1);
 }
 }

 public static void getSpecificMovie(CqlSession session, String keyspaceName)
 {
 ResultSet resultSet = session.execute(
 "SELECT * FROM \"" + keyspaceName + "\".\"Movie\" WHERE title =
'The Family' ALLOW FILTERING ;");
 resultSet.forEach(item -> {
 System.out.println("The Movie title is " + item.getString("title"));
 System.out.println("The Movie year is " + item.getInt("year"));
 System.out.println("The plot is " + item.getString("plot"));
 });
 }

 // Get records from the Movie table.
 public static void getMovieData(CqlSession session, String keyspaceName) {
 ResultSet resultSet = session.execute("SELECT * FROM \"" + keyspaceName +
 "\".\"Movie\";");
 resultSet.forEach(item -> {
 System.out.println("The Movie title is " + item.getString("title"));
 System.out.println("The Movie year is " + item.getInt("year"));
 System.out.println("The plot is " + item.getString("plot"));
 });
 }

 // Load data into the table.
 public static void loadData(CqlSession session, String fileName, String
keySpace) throws IOException {
 String sqlStatement = "INSERT INTO \"" + keySpace + "\".\"Movie\" (title,
year, plot) values (:k0, :k1, :k2)";
 JsonParser parser = new JsonFactory().createParser(new File(fileName));
 com.fasterxml.jackson.databind.JsonNode rootNode = new
ObjectMapper().readTree(parser);
 Iterator<JsonNode> iter = rootNode.iterator();
 ObjectNode currentNode;
 int t = 0;
 while (iter.hasNext()) {

 // Add 20 movies to the table.
 if (t == 20)

```

```

 break;
 currentNode = (ObjectNode) iter.next();

 int year = currentNode.path("year").asInt();
 String title = currentNode.path("title").asText();
 String plot = currentNode.path("info").path("plot").toString();

 // Insert the data into the Amazon Keyspaces table.
 BatchStatementBuilder builder =
BatchStatement.builder(DefaultBatchType.UNLOGGED);
 builder.setConsistencyLevel(ConsistencyLevel.LOCAL_QUORUM);
 PreparedStatement preparedStatement = session.prepare(sqlStatement);
 builder.addStatement(preparedStatement.boundStatementBuilder()
 .setString("k0", title)
 .setInt("k1", year)
 .setString("k2", plot)
 .build());

 BatchStatement batchStatement = builder.build();
 session.execute(batchStatement);
 t++;
 }

 System.out.println("You have added " + t + " records successfully!");
}

public static void listTables(KeyspacesClient keyClient, String keyspaceName)
{
 try {
 ListTablesRequest tablesRequest = ListTablesRequest.builder()
 .keyspaceName(keyspaceName)
 .build();

 ListTablesIterable listRes =
keyClient.listTablesPaginator(tablesRequest);
 listRes.stream()
 .flatMap(r -> r.tables().stream())
 .forEach(content -> System.out.println(" ARN: " +
content.resourceArn() +
 " Table name: " + content.tableName()));

 } catch (KeyspacesException e) {
 System.err.println(e.awsErrorDetails().errorMessage());
 System.exit(1);
 }
}

```

```

 }
}

public static void checkTable(KeyspacesClient keyClient, String keyspaceName,
String tableName)
 throws InterruptedException {
 try {
 boolean tableStatus = false;
 String status;
 GetTableResponse response = null;
 GetTableRequest tableRequest = GetTableRequest.builder()
 .keyspaceName(keyspaceName)
 .tableName(tableName)
 .build();

 while (!tableStatus) {
 response = keyClient.getTable(tableRequest);
 status = response.statusAsString();
 System.out.println(". The table status is " + status);

 if (status.compareTo("ACTIVE") == 0) {
 tableStatus = true;
 }
 Thread.sleep(500);
 }

 List<ColumnDefinition> cols =
response.schemaDefinition().allColumns();
 for (ColumnDefinition def : cols) {
 System.out.println("The column name is " + def.name());
 System.out.println("The column type is " + def.type());
 }

 } catch (KeyspacesException e) {
 System.err.println(e.awsErrorDetails().errorMessage());
 System.exit(1);
 }
}

public static void createTable(KeyspacesClient keyClient, String keySpace,
String tableName) {
 try {
 // Set the columns.
 ColumnDefinition defTitle = ColumnDefinition.builder()

```

```
 .name("title")
 .type("text")
 .build();

ColumnDefinition defYear = ColumnDefinition.builder()
 .name("year")
 .type("int")
 .build();

ColumnDefinition defReleaseDate = ColumnDefinition.builder()
 .name("release_date")
 .type("timestamp")
 .build();

ColumnDefinition defPlot = ColumnDefinition.builder()
 .name("plot")
 .type("text")
 .build();

List<ColumnDefinition> collist = new ArrayList<>();
collist.add(defTitle);
collist.add(defYear);
collist.add(defReleaseDate);
collist.add(defPlot);

// Set the keys.
PartitionKey yearKey = PartitionKey.builder()
 .name("year")
 .build();

PartitionKey titleKey = PartitionKey.builder()
 .name("title")
 .build();

List<PartitionKey> keyList = new ArrayList<>();
keyList.add(yearKey);
keyList.add(titleKey);

SchemaDefinition schemaDefinition = SchemaDefinition.builder()
 .partitionKeys(keyList)
 .allColumns(collist)
 .build();

PointInTimeRecovery timeRecovery = PointInTimeRecovery.builder()
```

```
 .status(PointInTimeRecoveryStatus.ENABLED)
 .build();

 CreateTableRequest tableRequest = CreateTableRequest.builder()
 .keyspaceName(keySpace)
 .tableName(tableName)
 .schemaDefinition(schemaDefinition)
 .pointInTimeRecovery(timeRecovery)
 .build();

 CreateTableResponse response = keyClient.createTable(tableRequest);
 System.out.println("The table ARN is " + response.resourceArn());

 } catch (KeyspacesException e) {
 System.err.println(e.awsErrorDetails().errorMessage());
 System.exit(1);
 }
}

public static void listKeyspacesPaginator(KeyspacesClient keyClient) {
 try {
 ListKeyspacesRequest keyspacesRequest =
ListKeyspacesRequest.builder()
 .maxResults(10)
 .build();

 ListKeyspacesIterable listRes =
keyClient.listKeyspacesPaginator(keyspacesRequest);
 listRes.stream()
 .flatMap(r -> r.keyspaces().stream())
 .forEach(content -> System.out.println(" Name: " +
content.keyspaceName()));

 } catch (KeyspacesException e) {
 System.err.println(e.awsErrorDetails().errorMessage());
 System.exit(1);
 }
}

public static void checkKeyspaceExistence(KeyspacesClient keyClient, String
keyspaceName) {
 try {
 GetKeyspaceRequest keyspaceRequest = GetKeyspaceRequest.builder()
 .keyspaceName(keyspaceName)
```



```

 .build();

 GetKeyspaceResponse response =
keyClient.getKeyspace(keyspaceRequest);
 String name = response.keyspaceName();
 System.out.println("The " + name + " KeySpace is ready");

 } catch (KeyspacesException e) {
 System.err.println(e.awsErrorDetails().errorMessage());
 System.exit(1);
 }
}

public static void createKeySpace(KeyspacesClient keyClient, String
keyspaceName) {
 try {
 CreateKeyspaceRequest keyspaceRequest =
CreateKeyspaceRequest.builder()
 .keyspaceName(keyspaceName)
 .build();

 CreateKeyspaceResponse response =
keyClient.createKeyspace(keyspaceRequest);
 System.out.println("The ARN of the KeySpace is " +
response.resourceArn());

 } catch (KeyspacesException e) {
 System.err.println(e.awsErrorDetails().errorMessage());
 System.exit(1);
 }
}
}

```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API AWS SDK for Java 2.x .

- [CreateKeyspace](#)
- [CreateTable](#)
- [DeleteKeyspace](#)
- [DeleteTable](#)
- [GetKeyspace](#)

- [GetTable](#)
- [ListKeyspaces](#)
- [ListTables](#)
- [RestoreTable](#)
- [UpdateTable](#)

## Kotlin

### SDK per Kotlin

#### 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 Kotlin code example, set up your development environment, including your credentials.

For more information, see the following documentation topic:

<https://docs.aws.amazon.com/sdk-for-kotlin/latest/developer-guide/setup.html>

This example uses a secure file format to hold certificate information for Kotlin applications. This is required to make a connection to Amazon Keyspaces. For more information, see the following documentation topic:

[https://docs.aws.amazon.com/keyspaces/latest/devguide/using\\_java\\_driver.html](https://docs.aws.amazon.com/keyspaces/latest/devguide/using_java_driver.html)

This Kotlin example performs the following tasks:

1. Create a keyspace.
2. Check for keyspace existence.
3. List keyspaces using a paginator.
4. Create a table with a simple movie data schema and enable point-in-time recovery.
5. Check for the table to be in an Active state.
6. List all tables in the keyspace.

```

7. Use a Cassandra driver to insert some records into the Movie table.
8. Get all records from the Movie table.
9. Get a specific Movie.
10. Get a UTC timestamp for the current time.
11. Update the table schema to add a 'watched' Boolean column.
12. Update an item as watched.
13. Query for items with watched = True.
14. Restore the table back to the previous state using the timestamp.
15. Check for completion of the restore action.
16. Delete the table.
17. Confirm that both tables are deleted.
18. Delete the keyspace.
*/

/*
Usage:
 fileName - The name of the JSON file that contains movie data. (Get this
file from the GitHub repo at resources/sample_file.)
 keyspaceName - The name of the keyspace to create.
*/
val DASHES: String = String(CharArray(80)).replace("\u0000", "-")

suspend fun main() {
 val fileName = "<Replace with the JSON file that contains movie data>"
 val keyspaceName = "<Replace with the name of the keyspace to create>"
 val titleUpdate = "The Family"
 val yearUpdate = 2013
 val tableName = "MovieKotlin"
 val tableNameRestore = "MovieRestore"

 val loader = DriverConfigLoader.fromClasspath("application.conf")
 val session =
 CqlSession
 .builder()
 .withConfigLoader(loader)
 .build()

 println(DASHES)
 println("Welcome to the Amazon Keyspaces example scenario.")
 println(DASHES)

 println(DASHES)
 println("1. Create a keyspace.")
 createKeySpace(keyspaceName)

```

```
println(DASHES)

println(DASHES)
delay(5000)
println("2. Check for keyspace existence.")
checkKeyspaceExistence(keyspaceName)
println(DASHES)

println(DASHES)
println("3. List keyspace using a paginator.")
listKeyspacesPaginator()
println(DASHES)

println(DASHES)
println("4. Create a table with a simple movie data schema and enable point-
in-time recovery.")
createTable(keyspaceName, tableName)
println(DASHES)

println(DASHES)
println("5. Check for the table to be in an Active state.")
delay(6000)
checkTable(keyspaceName, tableName)
println(DASHES)

println(DASHES)
println("6. List all tables in the keyspace.")
listTables(keyspaceName)
println(DASHES)

println(DASHES)
println("7. Use a Cassandra driver to insert some records into the Movie
table.")
delay(6000)
loadData(session, fileName, keyspaceName)
println(DASHES)

println(DASHES)
println("8. Get all records from the Movie table.")
getMovieData(session, keyspaceName)
println(DASHES)

println(DASHES)
println("9. Get a specific Movie.")
```

```
getSpecificMovie(session, keyspaceName)
println(DASHES)

println(DASHES)
println("10. Get a UTC timestamp for the current time.")
val utc = ZonedDateTime.now(ZoneOffset.UTC)
println("DATETIME = ${Date.from(utc.toInstant())}")
println(DASHES)

println(DASHES)
println("11. Update the table schema to add a watched Boolean column.")
updateTable(keyspaceName, tableName)
println(DASHES)

println(DASHES)
println("12. Update an item as watched.")
delay(10000) // Wait 10 seconds for the update.
updateRecord(session, keyspaceName, titleUpdate, yearUpdate)
println(DASHES)

println(DASHES)
println("13. Query for items with watched = True.")
getWatchedData(session, keyspaceName)
println(DASHES)

println(DASHES)
println("14. Restore the table back to the previous state using the
timestamp.")
println("Note that the restore operation can take up to 20 minutes.")
restoreTable(keyspaceName, utc)
println(DASHES)

println(DASHES)
println("15. Check for completion of the restore action.")
delay(5000)
checkRestoredTable(keyspaceName, "MovieRestore")
println(DASHES)

println(DASHES)
println("16. Delete both tables.")
deleteTable(keyspaceName, tableName)
deleteTable(keyspaceName, tableNameRestore)
println(DASHES)
```

```

println(DASHES)
println("17. Confirm that both tables are deleted.")
checkTableDelete(keyspaceName, tableName)
checkTableDelete(keyspaceName, tableNameRestore)
println(DASHES)

println(DASHES)
println("18. Delete the keyspace.")
deleteKeyspace(keyspaceName)
println(DASHES)

println(DASHES)
println("The scenario has completed successfully.")
println(DASHES)
}

suspend fun deleteKeyspace(keyspaceNameVal: String?) {
 val deleteKeyspaceRequest =
 DeleteKeyspaceRequest {
 keyspaceName = keyspaceNameVal
 }

 KeyspacesClient.fromEnvironment { region = "us-east-1" }.use { keyClient ->
 keyClient.deleteKeyspace(deleteKeyspaceRequest)
 }
}

suspend fun checkTableDelete(
 keyspaceNameVal: String?,
 tableNameVal: String?,
) {
 var status: String
 var response: GetTableResponse
 val tableRequest =
 GetTableRequest {
 keyspaceName = keyspaceNameVal
 tableName = tableNameVal
 }

 try {
 KeyspacesClient.fromEnvironment { region = "us-east-1" }.use { keyClient
->
 // Keep looping until the table cannot be found and a
ResourceNotFoundException is thrown.

```

```

 while (true) {
 response = keyClient.getTable(tableRequest)
 status = response.status.toString()
 println(". The table status is $status")
 delay(500)
 }
 }
} catch (e: ResourceNotFoundException) {
 println(e.message)
}
println("The table is deleted")
}

suspend fun deleteTable(
 keyspaceNameVal: String?,
 tableNameVal: String?,
) {
 val tableRequest =
 DeleteTableRequest {
 keyspaceName = keyspaceNameVal
 tableName = tableNameVal
 }

 KeyspacesClient.fromEnvironment { region = "us-east-1" }.use { keyClient ->
 keyClient.deleteTable(tableRequest)
 }
}

suspend fun checkRestoredTable(
 keyspaceNameVal: String?,
 tableNameVal: String?,
) {
 var tableStatus = false
 var status: String
 var response: GetTableResponse? = null

 val tableRequest =
 GetTableRequest {
 keyspaceName = keyspaceNameVal
 tableName = tableNameVal
 }

 KeyspacesClient.fromEnvironment { region = "us-east-1" }.use { keyClient ->
 while (!tableStatus) {

```

```

 response = keyClient.getTable(tableRequest)
 status = response!!.status.toString()
 println("The table status is $status")

 if (status.compareTo("ACTIVE") == 0) {
 tableStatus = true
 }
 delay(500)
 }

 val cols = response!!.schemaDefinition?.allColumns
 if (cols != null) {
 for (def in cols) {
 println("The column name is ${def.name}")
 println("The column type is ${def.type}")
 }
 }
}

suspend fun restoreTable(
 keyspaceName: String?,
 utc: ZonedDateTime,
) {
 // Create an aws.smithy.kotlin.runtime.time.Instant value.
 val timeStamp =
 aws.smithy.kotlin.runtime.time
 .Instant(utc.toInstant())
 val restoreTableRequest =
 RestoreTableRequest {
 restoreTimestamp = timeStamp
 sourceTableName = "MovieKotlin"
 targetKeyspaceName = keyspaceName
 targetTableName = "MovieRestore"
 sourceKeyspaceName = keyspaceName
 }

 KeyspacesClient.fromEnvironment { region = "us-east-1" }.use { keyClient ->
 val response = keyClient.restoreTable(restoreTableRequest)
 println("The ARN of the restored table is ${response.restoredTableArn}")
 }
}

fun getWatchedData(

```



```

 session: CqlSession,
 keyspaceName: String,
) {
 val resultSet = session.execute("SELECT * FROM \"\$keyspaceName\".
 \"MovieKotlin\" WHERE watched = true ALLOW FILTERING;")
 resultSet.forEach { item: Row ->
 println("The Movie title is ${item.getString("title")}")
 println("The Movie year is ${item.getInt("year")}")
 println("The plot is ${item.getString("plot")}")
 }
 }

fun updateRecord(
 session: CqlSession,
 keySpace: String,
 titleUpdate: String?,
 yearUpdate: Int,
) {
 val sqlStatement =
 "UPDATE \"\$keySpace\".\"MovieKotlin\" SET watched=true WHERE title = :k0
 AND year = :k1;"
 val builder = BatchStatement.builder(DefaultBatchType.UNLOGGED)
 builder.setConsistencyLevel(ConsistencyLevel.LOCAL_QUORUM)
 val preparedStatement = session.prepare(sqlStatement)
 builder.addStatement(
 preparedStatement
 .boundStatementBuilder()
 .setString("k0", titleUpdate)
 .setInt("k1", yearUpdate)
 .build(),
)
 val batchStatement = builder.build()
 session.execute(batchStatement)
}

suspend fun updateTable(
 keySpace: String?,
 tableNameVal: String?,
) {
 val def =
 ColumnDefinition {
 name = "watched"
 type = "boolean"
 }
}

```

```

 val tableRequest =
 UpdateTableRequest {
 keyspaceName = keySpace
 tableName = tableNameVal
 addColumns = listOf(def)
 }

 KeyspacesClient.fromEnvironment { region = "us-east-1" }.use { keyClient ->
 keyClient.updateTable(tableRequest)
 }
 }

fun getSpecificMovie(
 session: CqlSession,
 keyspaceName: String,
) {
 val resultSet =
 session.execute("SELECT * FROM \"\$keyspaceName\".\"MovieKotlin\" WHERE
title = 'The Family' ALLOW FILTERING ;")

 resultSet.forEach { item: Row ->
 println("The Movie title is ${item.getString("title")}")
 println("The Movie year is ${item.getInt("year")}")
 println("The plot is ${item.getString("plot")}")
 }
}

// Get records from the Movie table.
fun getMovieData(
 session: CqlSession,
 keyspaceName: String,
) {
 val resultSet = session.execute("SELECT * FROM \"\$keyspaceName\".
\"MovieKotlin\";")
 resultSet.forEach { item: Row ->
 println("The Movie title is ${item.getString("title")}")
 println("The Movie year is ${item.getInt("year")}")
 println("The plot is ${item.getString("plot")}")
 }
}

// Load data into the table.
fun loadData(

```

```

 session: CqlSession,
 fileName: String,
 keySpace: String,
) {
 val sqlStatement =
 "INSERT INTO \"\$keySpace\".\"MovieKotlin\" (title, year, plot) values
(:k0, :k1, :k2)"
 val parser = JsonFactory().createParser(File(fileName))
 val rootNode = ObjectMapper().readTree<JsonNode>(parser)
 val iter: Iterator<JsonNode> = rootNode.iterator()
 var currentNode: JsonNode

 var t = 0
 while (iter.hasNext()) {
 if (t == 50) {
 break
 }

 currentNode = iter.next() as JsonNode
 val year = currentNode.path("year").asInt()
 val title = currentNode.path("title").asText()
 val info = currentNode.path("info").toString()

 // Insert the data into the Amazon Keyspaces table.
 val builder = BatchStatement.builder(DefaultBatchType.UNLOGGED)
 builder.setConsistencyLevel(ConsistencyLevel.LOCAL_QUORUM)
 val preparedStatement: PreparedStatement = session.prepare(sqlStatement)
 builder.addStatement(
 preparedStatement
 .boundStatementBuilder()
 .setString("k0", title)
 .setInt("k1", year)
 .setString("k2", info)
 .build(),
)

 val batchStatement = builder.build()
 session.execute(batchStatement)
 t++
 }
 }

suspend fun listTables(keyspaceNameVal: String?) {
 val tablesRequest =

```

```

 ListTablesRequest {
 keyspaceName = keyspaceNameVal
 }

KeyspacesClient.fromEnvironment { region = "us-east-1" }.use { keyClient ->
 keyClient
 .listTablesPaginated(tablesRequest)
 .transform { it.tables?.forEach { obj -> emit(obj) } }
 .collect { obj ->
 println(" ARN: ${obj.resourceArn} Table name: ${obj.tableName}")
 }
 }
}

suspend fun checkTable(
 keyspaceNameVal: String?,
 tableNameVal: String?,
) {
 var tableStatus = false
 var status: String
 var response: GetTableResponse? = null

 val tableRequest =
 GetTableRequest {
 keyspaceName = keyspaceNameVal
 tableName = tableNameVal
 }

 KeyspacesClient.fromEnvironment { region = "us-east-1" }.use { keyClient ->
 while (!tableStatus) {
 response = keyClient.getTable(tableRequest)
 status = response!!.status.toString()
 println(". The table status is $status")
 if (status.compareTo("ACTIVE") == 0) {
 tableStatus = true
 }
 delay(500)
 }
 val cols: List<ColumnDefinition>? =
response!!.schemaDefinition?.allColumns
 if (cols != null) {
 for (def in cols) {
 println("The column name is ${def.name}")
 println("The column type is ${def.type}")
 }
 }
 }
}

```

```
 }
 }
}

suspend fun createTable(
 keySpaceVal: String?,
 tableNameVal: String?,
) {
 // Set the columns.
 val defTitle =
 ColumnDefinition {
 name = "title"
 type = "text"
 }

 val defYear =
 ColumnDefinition {
 name = "year"
 type = "int"
 }

 val defReleaseDate =
 ColumnDefinition {
 name = "release_date"
 type = "timestamp"
 }

 val defPlot =
 ColumnDefinition {
 name = "plot"
 type = "text"
 }

 val collist = ArrayList<ColumnDefinition>()
 collist.add(defTitle)
 collist.add(defYear)
 collist.add(defReleaseDate)
 collist.add(defPlot)

 // Set the keys.
 val yearKey =
 PartitionKey {
 name = "year"
 }
}
```

```
val titleKey =
 PartitionKey {
 name = "title"
 }

val keyList = ArrayList<PartitionKey>()
keyList.add(yearKey)
keyList.add(titleKey)

val schemaDefinition0b =
 SchemaDefinition {
 partitionKeys = keyList
 allColumns = colList
 }

val timeRecovery =
 PointInTimeRecovery {
 status = PointInTimeRecoveryStatus.Enabled
 }

val tableRequest =
 CreateTableRequest {
 keyspaceName = keySpaceVal
 tableName = tableNameVal
 schemaDefinition = schemaDefinition0b
 pointInTimeRecovery = timeRecovery
 }

KeyspacesClient.fromEnvironment { region = "us-east-1" }.use { keyClient ->
 val response = keyClient.createTable(tableRequest)
 println("The table ARN is ${response.resourceArn}")
}

suspend fun listKeyspacesPaginator() {
 KeyspacesClient.fromEnvironment { region = "us-east-1" }.use { keyClient ->
 keyClient
 .listKeyspacesPaginated(ListKeyspacesRequest {})
 .transform { it.keyspaces?.forEach { obj -> emit(obj) } }
 .collect { obj ->
 println("Name: ${obj.keyspaceName}")
 }
 }
}
```

```
}

suspend fun checkKeyspaceExistence(keyspaceNameVal: String?) {
 val keyspaceRequest =
 GetKeyspaceRequest {
 keyspaceName = keyspaceNameVal
 }

 KeyspacesClient.fromEnvironment { region = "us-east-1" }.use { keyClient ->
 val response: GetKeyspaceResponse =
 keyClient.getKeyspace(keyspaceRequest)
 val name = response.keyspaceName
 println("The $name KeySpace is ready")
 }
}

suspend fun createKeySpace(keyspaceNameVal: String) {
 val keyspaceRequest =
 CreateKeyspaceRequest {
 keyspaceName = keyspaceNameVal
 }

 KeyspacesClient.fromEnvironment { region = "us-east-1" }.use { keyClient ->
 val response = keyClient.createKeyspace(keyspaceRequest)
 println("The ARN of the KeySpace is ${response.resourceArn}")
 }
}
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API SDK AWS per Kotlin.
  - [CreateKeyspace](#)
  - [CreateTable](#)
  - [DeleteKeyspace](#)
  - [DeleteTable](#)
  - [GetKeyspace](#)
  - [GetTable](#)
  - [ListKeyspaces](#)
  - [ListTables](#)
  - [RestoreTable](#)

- [UpdateTable](#)

## 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](#).

Esegui uno scenario interattivo al prompt dei comandi.

```
class KeyspaceScenario:
 """Runs an interactive scenario that shows how to get started using Amazon
 Keyspaces."""

 def __init__(self, ks_wrapper):
 """
 :param ks_wrapper: An object that wraps Amazon Keyspace actions.
 """
 self.ks_wrapper = ks_wrapper

 @demo_func
 def create_keyspace(self):
 """
 1. Creates a keyspace.
 2. Lists up to 10 keyspace in your account.
 """
 print("Let's create a keyspace.")
 ks_name = q.ask(
 "Enter a name for your new keyspace.\nThe name can contain only
letters, "
 "numbers and underscores: ",
 q.non_empty,
)
 if self.ks_wrapper.exists_keyspace(ks_name):
 print(f"A keyspace named {ks_name} exists.")
 else:
 ks_arn = self.ks_wrapper.create_keyspace(ks_name)
 ks_exists = False
```



```

 while not ks_exists:
 wait(3)
 ks_exists = self.ks_wrapper.exists_keyspace(ks_name)
 print(f"Created a new keyspace.\n\t{ks_arn}.")
 print("The first 10 keyspaces in your account are:\n")
 self.ks_wrapper.list_keyspaces(10)

@demo_func
def create_table(self):
 """
 1. Creates a table in the keyspace. The table is configured with a schema
 to hold
 movie data and has point-in-time recovery enabled.
 2. Waits for the table to be in an active state.
 3. Displays schema information for the table.
 4. Lists tables in the keyspace.
 """
 print("Let's create a table for movies in your keyspace.")
 table_name = q.ask("Enter a name for your table: ", q.non_empty)
 table = self.ks_wrapper.get_table(table_name)
 if table is not None:
 print(
 f"A table named {table_name} already exists in keyspace "
 f"{self.ks_wrapper.ks_name}."
)
 else:
 table_arn = self.ks_wrapper.create_table(table_name)
 print(f"Created table {table_name}:\n\t{table_arn}")
 table = {"status": None}
 print("Waiting for your table to be ready...")
 while table["status"] != "ACTIVE":
 wait(5)
 table = self.ks_wrapper.get_table(table_name)
 print(f"Your table is {table['status']}. Its schema is:")
 pp(table["schemaDefinition"])
 print("\nThe tables in your keyspace are:\n")
 self.ks_wrapper.list_tables()

@demo_func
def ensure_tls_cert(self):
 """
 Ensures you have a TLS certificate available to use to secure the
 connection

```

```

 to the keyspace. This function downloads a default certificate or lets
you
 specify your own.
 """
 print("To connect to your keyspace, you must have a TLS certificate.")
 print("Checking for TLS certificate...")
 cert_path = os.path.join(
 os.path.dirname(__file__), QueryManager.DEFAULT_CERT_FILE
)
 if not os.path.exists(cert_path):
 cert_choice = q.ask(
 f"Press enter to download a certificate from
{QueryManager.CERT_URL} "
 f"or enter the full path to the certificate you want to use: "
)
 if cert_choice:
 cert_path = cert_choice
 else:
 cert = requests.get(QueryManager.CERT_URL).text
 with open(cert_path, "w") as cert_file:
 cert_file.write(cert)
 else:
 q.ask(f"Certificate {cert_path} found. Press Enter to continue.")
 print(
 f"Certificate {cert_path} will be used to secure the connection to
your keyspace."
)
 return cert_path

@demo_func
def query_table(self, qm, movie_file):
 """
 1. Adds movies to the table from a sample movie data file.
 2. Gets a list of movies from the table and lets you select one.
 3. Displays more information about the selected movie.
 """
 qm.add_movies(self.ks_wrapper.table_name, movie_file)
 movies = qm.get_movies(self.ks_wrapper.table_name)
 print(f"Added {len(movies)} movies to the table:")
 sel = q.choose("Pick one to learn more about it: ", [m.title for m in
movies])
 movie_choice = qm.get_movie(
 self.ks_wrapper.table_name, movies[sel].title, movies[sel].year
)

```

```

 print(movie_choice.title)
 print(f"\tReleased: {movie_choice.release_date}")
 print(f"\tPlot: {movie_choice.plot}")

@demo_func
def update_and_restore_table(self, qm):
 """
 1. Updates the table by adding a column to track watched movies.
 2. Marks some of the movies as watched.
 3. Gets the list of watched movies from the table.
 4. Restores to a movies_restored table at a previous point in time.
 5. Gets the list of movies from the restored table.
 """
 print("Let's add a column to record which movies you've watched.")
 pre_update_timestamp = datetime.utcnow()
 print(
 f"Recorded the current UTC time of {pre_update_timestamp} so we can
 restore the table later."
)
 self.ks_wrapper.update_table()
 print("Waiting for your table to update...")
 table = {"status": "UPDATING"}
 while table["status"] != "ACTIVE":
 wait(5)
 table = self.ks_wrapper.get_table(self.ks_wrapper.table_name)
 print("Column 'watched' added to table.")
 q.ask(
 "Let's mark some of the movies as watched. Press Enter when you're
 ready.\n"
)
 movies = qm.get_movies(self.ks_wrapper.table_name)
 for movie in movies[:10]:
 qm.watched_movie(self.ks_wrapper.table_name, movie.title, movie.year)
 print(f"Marked {movie.title} as watched.")
 movies = qm.get_movies(self.ks_wrapper.table_name, watched=True)
 print("-" * 88)
 print("The watched movies in our table are:\n")
 for movie in movies:
 print(movie.title)
 print("-" * 88)
 if q.ask(
 "Do you want to restore the table to the way it was before all of
 these\n"
 "updates? Keep in mind, this can take up to 20 minutes. (y/n) ",
):

```

```

 q.is_ynsno,
):
 starting_table_name = self.ks_wrapper.table_name
 table_name_restored =
self.ks_wrapper.restore_table(pre_update_timestamp)
 table = {"status": "RESTORING"}
 while table["status"] != "ACTIVE":
 wait(10)
 table = self.ks_wrapper.get_table(table_name_restored)
 print(
 f"Restored {starting_table_name} to {table_name_restored} "
 f"at a point in time of {pre_update_timestamp}."
)
 movies = qm.get_movies(table_name_restored)
 print("Now the movies in our table are:")
 for movie in movies:
 print(movie.title)

def cleanup(self, cert_path):
 """
 1. Deletes the table and waits for it to be removed.
 2. Deletes the keyspace.

 :param cert_path: The path of the TLS certificate used in the demo. If
the
 certificate was downloaded during the demo, it is
removed.
 """
 if q.ask(
 f"Do you want to delete your {self.ks_wrapper.table_name} table and "
 f"{self.ks_wrapper.ks_name} keyspace? (y/n) ",
 q.is_ynsno,
):
 table_name = self.ks_wrapper.table_name
 self.ks_wrapper.delete_table()
 table = self.ks_wrapper.get_table(table_name)
 print("Waiting for the table to be deleted.")
 while table is not None:
 wait(5)
 table = self.ks_wrapper.get_table(table_name)
 print("Table deleted.")
 self.ks_wrapper.delete_keyspace()
 print(

```

```

 "Keyspace deleted. If you chose to restore your table during the
"
 "demo, the original table is also deleted."
)
 if cert_path == os.path.join(
 os.path.dirname(__file__), QueryManager.DEFAULT_CERT_FILE
) and os.path.exists(cert_path):
 os.remove(cert_path)
 print("Removed certificate that was downloaded for this demo.")

 def run_scenario(self):
 logging.basicConfig(level=logging.INFO, format="%(levelname)s:
%(message)s")

 print("-" * 88)
 print("Welcome to the Amazon Keyspaces (for Apache Cassandra) demo.")
 print("-" * 88)

 self.create_keyspace()
 self.create_table()
 cert_file_path = self.ensure_tls_cert()
 # Use a context manager to ensure the connection to the keyspace is
 closed.
 with QueryManager(
 cert_file_path, boto3.DEFAULT_SESSION, self.ks_wrapper.ks_name
) as qm:
 self.query_table(qm, "../..../resources/sample_files/movies.json")
 self.update_and_restore_table(qm)
 self.cleanup(cert_file_path)

 print("\nThanks for watching!")
 print("-" * 88)

 if __name__ == "__main__":
 try:
 scenario = KeyspaceScenario(KeyspaceWrapper.from_client())
 scenario.run_scenario()
 except Exception:
 logging.exception("Something went wrong with the demo.")

```

Definisci una classe che racchiude le azioni dello spazio delle chiavi e delle tabelle.

```
class KeyspaceWrapper:
 """Encapsulates Amazon Keyspaces (for Apache Cassandra) keyspace and table
 actions."""

 def __init__(self, keyspaces_client):
 """
 :param keyspaces_client: A Boto3 Amazon Keyspaces client.
 """
 self.keyspaces_client = keyspaces_client
 self.ks_name = None
 self.ks_arn = None
 self.table_name = None

 @classmethod
 def from_client(cls):
 keyspaces_client = boto3.client("keyspaces")
 return cls(keyspaces_client)

 def create_keyspace(self, name):
 """
 Creates a keyspace.

 :param name: The name to give the keyspace.
 :return: The Amazon Resource Name (ARN) of the new keyspace.
 """
 try:
 response = self.keyspaces_client.create_keyspace(keyspaceName=name)
 self.ks_name = name
 self.ks_arn = response["resourceArn"]
 except ClientError as err:
 logger.error(
 "Couldn't create %s. Here's why: %s: %s",
 name,
 err.response["Error"]["Code"],
 err.response["Error"]["Message"],
)
 raise
 else:
 return self.ks_arn

 def exists_keyspace(self, name):
```

```

 """
 Checks whether a keyspace exists.

 :param name: The name of the keyspace to look up.
 :return: True when the keyspace exists. Otherwise, False.
 """
 try:
 response = self.keyspaces_client.get_keyspace(keyspaceName=name)
 self.ks_name = response["keyspaceName"]
 self.ks_arn = response["resourceArn"]
 exists = True
 except ClientError as err:
 if err.response["Error"]["Code"] == "ResourceNotFoundException":
 logger.info("Keyspace %s does not exist.", name)
 exists = False
 else:
 logger.error(
 "Couldn't verify %s exists. Here's why: %s: %s",
 name,
 err.response["Error"]["Code"],
 err.response["Error"]["Message"],
)
 raise
 return exists

def list_keyspaces(self, limit):
 """
 Lists the keyspaces in your account.

 :param limit: The maximum number of keyspaces to list.
 """
 try:
 ks_paginator = self.keyspaces_client.get_paginator("list_keyspaces")
 for page in ks_paginator.paginate(PaginationConfig={"MaxItems":
limit}):
 for ks in page["keyspaces"]:
 print(ks["keyspaceName"])
 print(f"\t{ks['resourceArn']}")
 except ClientError as err:
 logger.error(
 "Couldn't list keyspaces. Here's why: %s: %s",
 err.response["Error"]["Code"],
 err.response["Error"]["Message"],

```

```

)
 raise

def create_table(self, table_name):
 """
 Creates a table in the keyspace.
 The table is created with a schema for storing movie data
 and has point-in-time recovery enabled.

 :param table_name: The name to give the table.
 :return: The ARN of the new table.
 """
 try:
 response = self.keyspaces_client.create_table(
 keyspaceName=self.ks_name,
 tableName=table_name,
 schemaDefinition={
 "allColumns": [
 {"name": "title", "type": "text"},
 {"name": "year", "type": "int"},
 {"name": "release_date", "type": "timestamp"},
 {"name": "plot", "type": "text"},
],
 "partitionKeys": [{"name": "year"}, {"name": "title"}],
 },
 pointInTimeRecovery={"status": "ENABLED"},
)
 except ClientError as err:
 logger.error(
 "Couldn't create table %s. Here's why: %s: %s",
 table_name,
 err.response["Error"]["Code"],
 err.response["Error"]["Message"],
)
 raise
 else:
 return response["resourceArn"]

def get_table(self, table_name):
 """
 Gets data about a table in the keyspace.

```



```

:param table_name: The name of the table to look up.
:return: Data about the table.
"""
try:
 response = self.keyspaces_client.get_table(
 keyspaceName=self.ks_name, tableName=table_name
)
 self.table_name = table_name
except ClientError as err:
 if err.response["Error"]["Code"] == "ResourceNotFoundException":
 logger.info("Table %s does not exist.", table_name)
 self.table_name = None
 response = None
 else:
 logger.error(
 "Couldn't verify %s exists. Here's why: %s: %s",
 table_name,
 err.response["Error"]["Code"],
 err.response["Error"]["Message"],
)
 raise
return response

def list_tables(self):
 """
 Lists the tables in the keyspace.
 """
 try:
 table_paginator = self.keyspaces_client.get_paginator("list_tables")
 for page in table_paginator.paginate(keyspaceName=self.ks_name):
 for table in page["tables"]:
 print(table["tableName"])
 print(f"\t{table['resourceArn']}")
 except ClientError as err:
 logger.error(
 "Couldn't list tables in keyspace %s. Here's why: %s: %s",
 self.ks_name,
 err.response["Error"]["Code"],
 err.response["Error"]["Message"],
)
 raise

```

```

def update_table(self):
 """
 Updates the schema of the table.

 This example updates a table of movie data by adding a new column
 that tracks whether the movie has been watched.
 """
 try:
 self.keyspaces_client.update_table(
 keyspaceName=self.ks_name,
 tableName=self.table_name,
 addColumns=[{"name": "watched", "type": "boolean"}],
)
 except ClientError as err:
 logger.error(
 "Couldn't update table %s. Here's why: %s: %s",
 self.table_name,
 err.response["Error"]["Code"],
 err.response["Error"]["Message"],
)
 raise

def restore_table(self, restore_timestamp):
 """
 Restores the table to a previous point in time. The table is restored
 to a new table in the same keyspace.

 :param restore_timestamp: The point in time to restore the table. This
 time must be in UTC format.

 :return: The name of the restored table.
 """
 try:
 restored_table_name = f"{self.table_name}_restored"
 self.keyspaces_client.restore_table(
 sourceKeyspaceName=self.ks_name,
 sourceTableName=self.table_name,
 targetKeyspaceName=self.ks_name,
 targetTableName=restored_table_name,
 restoreTimestamp=restore_timestamp,
)
 except ClientError as err:
 logger.error(

```

```
 "Couldn't restore table %s. Here's why: %s: %s",
 restore_timestamp,
 err.response["Error"]["Code"],
 err.response["Error"]["Message"],
)
 raise
else:
 return restored_table_name

def delete_table(self):
 """
 Deletes the table from the keyspace.
 """
 try:
 self.keyspaces_client.delete_table(
 keyspaceName=self.ks_name, tableName=self.table_name
)
 self.table_name = None
 except ClientError as err:
 logger.error(
 "Couldn't delete table %s. Here's why: %s: %s",
 self.table_name,
 err.response["Error"]["Code"],
 err.response["Error"]["Message"],
)
 raise

def delete_keyspace(self):
 """
 Deletes the keyspace.
 """
 try:
 self.keyspaces_client.delete_keyspace(keyspaceName=self.ks_name)
 self.ks_name = None
 except ClientError as err:
 logger.error(
 "Couldn't delete keyspace %s. Here's why: %s: %s",
 self.ks_name,
 err.response["Error"]["Code"],
 err.response["Error"]["Message"],
)
 raise
```

Definisci una classe che crea una connessione TLS a un keyspace, si autentica con SigV4 e invia query CQL a una tabella nello spazio delle chiavi.

```
class QueryManager:
 """
 Manages queries to an Amazon Keyspaces (for Apache Cassandra) keyspace.
 Queries are secured by TLS and authenticated by using the Signature V4
 (SigV4)
 AWS signing protocol. This is more secure than sending username and password
 with a plain-text authentication provider.

 This example downloads a default certificate to secure TLS, or lets you
 specify
 your own.

 This example uses a table of movie data to demonstrate basic queries.
 """

 DEFAULT_CERT_FILE = "sf-class2-root.crt"
 CERT_URL = f"https://certs.secureserver.net/repository/sf-class2-root.crt"

 def __init__(self, cert_file_path, boto_session, keyspace_name):
 """
 :param cert_file_path: The path and file name of the certificate used for
 TLS.
 :param boto_session: A Boto3 session. This is used to acquire your AWS
 credentials.
 :param keyspace_name: The name of the keyspace to connect.
 """
 self.cert_file_path = cert_file_path
 self.boto_session = boto_session
 self.ks_name = keyspace_name
 self.cluster = None
 self.session = None

 def __enter__(self):
 """
 Creates a session connection to the keyspace that is secured by TLS and

```

```

 authenticated by SigV4.
 """
 ssl_context = SSLContext(PROTOCOL_TLSv1_2)
 ssl_context.load_verify_locations(self.cert_file_path)
 ssl_context.verify_mode = CERT_REQUIRED
 auth_provider = SigV4AuthProvider(self.boto_session)
 contact_point = f"cassandra.
{self.boto_session.region_name}.amazonaws.com"
 exec_profile = ExecutionProfile(
 consistency_level=ConsistencyLevel.LOCAL_QUORUM,
 load_balancing_policy=DCAwareRoundRobinPolicy(),
)
 self.cluster = Cluster(
 [contact_point],
 ssl_context=ssl_context,
 auth_provider=auth_provider,
 port=9142,
 execution_profiles={EXEC_PROFILE_DEFAULT: exec_profile},
 protocol_version=4,
)
 self.cluster.__enter__()
 self.session = self.cluster.connect(self.ks_name)
 return self

 def __exit__(self, *args):
 """
 Exits the cluster. This shuts down all existing session connections.
 """
 self.cluster.__exit__(*args)

 def add_movies(self, table_name, movie_file_path):
 """
 Gets movies from a JSON file and adds them to a table in the keyspace.

 :param table_name: The name of the table.
 :param movie_file_path: The path and file name of a JSON file that
contains movie data.
 """
 with open(movie_file_path, "r") as movie_file:
 movies = json.loads(movie_file.read())
 stmt = self.session.prepare(
 f"INSERT INTO {table_name} (year, title, release_date, plot) VALUES
(?, ?, ?, ?);"
)

```

```

 for movie in movies[:20]:
 self.session.execute(
 stmt,
 parameters=[
 movie["year"],
 movie["title"],
 date.fromisoformat(movie["info"]
["release_date"].partition("T")[0]),
 movie["info"]["plot"],
],
)

 def get_movies(self, table_name, watched=None):
 """
 Gets the title and year of the full list of movies from the table.

 :param table_name: The name of the movie table.
 :param watched: When specified, the returned list of movies is filtered
to
 either movies that have been watched or movies that have
not
 been watched. Otherwise, all movies are returned.
 :return: A list of movies in the table.
 """
 if watched is None:
 stmt = SimpleStatement(f"SELECT title, year from {table_name}")
 params = None
 else:
 stmt = SimpleStatement(
 f"SELECT title, year from {table_name} WHERE watched = %s ALLOW
FILTERING"
)
 params = [watched]
 return self.session.execute(stmt, parameters=params).all()

 def get_movie(self, table_name, title, year):
 """
 Gets a single movie from the table, by title and year.

 :param table_name: The name of the movie table.
 :param title: The title of the movie.
 :param year: The year of the movie's release.
 :return: The requested movie.
 """

```

```
 return self.session.execute(
 SimpleStatement(
 f"SELECT * from {table_name} WHERE title = %s AND year = %s"
),
 parameters=[title, year],
).one()

 def watched_movie(self, table_name, title, year):
 """
 Updates a movie as having been watched.

 :param table_name: The name of the movie table.
 :param title: The title of the movie.
 :param year: The year of the movie's release.
 """
 self.session.execute(
 SimpleStatement(
 f"UPDATE {table_name} SET watched=true WHERE title = %s AND year
= %s"
),
 parameters=[title, year],
)
```

- Per informazioni dettagliate sull'API, consulta i seguenti argomenti nella Documentazione di riferimento delle API SDK AWS per Python (Boto3).
  - [CreateKeyspace](#)
  - [CreateTable](#)
  - [DeleteKeyspace](#)
  - [DeleteTable](#)
  - [GetKeyspace](#)
  - [GetTable](#)
  - [ListKeyspaces](#)
  - [ListTables](#)
  - [RestoreTable](#)
  - [UpdateTable](#)

Per un elenco completo delle guide per sviluppatori SDK e degli esempi di AWS codice, consulta. [Utilizzo di questo servizio con un SDK AWS](#) Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

## Azioni per l'utilizzo di Amazon Keyspaces AWS SDKs

I seguenti esempi di codice mostrano come eseguire singole azioni di Amazon Keyspaces con. AWS SDKs Ogni esempio include un collegamento a GitHub, dove puoi trovare le istruzioni per la configurazione e l'esecuzione del codice.

Gli esempi seguenti includono solo le operazioni più comunemente utilizzate. Per un elenco completo, consulta il riferimento all'API [Amazon Keyspaces \(for Apache Cassandra\)](#).

### Esempi

- [Utilizzalo CreateKeyspace con un SDK AWS](#)
- [Utilizzare CreateTable con un SDK AWS](#)
- [Utilizzare DeleteKeyspace con un SDK AWS](#)
- [Utilizzare DeleteTable con un SDK AWS](#)
- [Utilizzare GetKeyspace con un SDK AWS](#)
- [Utilizzare GetTable con un SDK AWS](#)
- [Utilizzare ListKeyspaces con un SDK AWS](#)
- [Utilizzare ListTables con un SDK AWS](#)
- [Utilizzare RestoreTable con un SDK AWS](#)
- [Utilizzare UpdateTable con un SDK AWS](#)

### Utilizzalo **CreateKeyspace** con un SDK AWS

Gli esempi di codice seguenti mostrano come utilizzare CreateKeyspace.

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 a new keyspace.
/// </summary>
/// <param name="keyspaceName">The name for the new keyspace.</param>
/// <returns>The Amazon Resource Name (ARN) of the new keyspace.</returns>
public async Task<string> CreateKeyspace(string keyspaceName)
{
 var response =
 await _amazonKeyspaces.CreateKeyspaceAsync(
 new CreateKeyspaceRequest { KeyspaceName = keyspaceName });
 return response.ResourceArn;
}
```

- Per i dettagli sull'API, consulta la [CreateKeyspace](#) sezione AWS SDK per .NET 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](#).

```
public static void createKeySpace(KeyspacesClient keyClient, String
keyspaceName) {
```

```
try {
 CreateKeyspaceRequest keyspaceRequest =
CreateKeyspaceRequest.builder()
 .keyspaceName(keyspaceName)
 .build();

 CreateKeyspaceResponse response =
keyClient.createKeyspace(keyspaceRequest);
 System.out.println("The ARN of the KeySpace is " +
response.resourceArn());

} catch (KeyspacesException e) {
 System.err.println(e.awsErrorDetails().errorMessage());
 System.exit(1);
}
```

- Per i dettagli sull'API, consulta la [CreateKeyspaces](#) sezione AWS SDK for Java 2.x 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 createKeySpace(keyspaceNameVal: String) {
 val keyspaceRequest =
 CreateKeyspaceRequest {
 keyspaceName = keyspaceNameVal
 }

 KeyspacesClient.fromEnvironment { region = "us-east-1" }.use { keyClient ->
 val response = keyClient.createKeyspace(keyspaceRequest)
 println("The ARN of the KeySpace is ${response.resourceArn}")
 }
}
```

```
}
```

- Per i dettagli sull'API, [CreateKeyspace](#) consulta AWS SDK for Kotlin 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 KeyspaceWrapper:
 """Encapsulates Amazon Keyspaces (for Apache Cassandra) keyspace and table
 actions."""

 def __init__(self, keyspaces_client):
 """
 :param keyspaces_client: A Boto3 Amazon Keyspaces client.
 """
 self.keyspaces_client = keyspaces_client
 self.ks_name = None
 self.ks_arn = None
 self.table_name = None

 @classmethod
 def from_client(cls):
 keyspaces_client = boto3.client("keyspaces")
 return cls(keyspaces_client)

 def create_keyspace(self, name):
 """
 Creates a keyspace.

 :param name: The name to give the keyspace.
 :return: The Amazon Resource Name (ARN) of the new keyspace.
 """
 try:
```

```
 response = self.keyspaces_client.create_keyspace(keyspaceName=name)
 self.ks_name = name
 self.ks_arn = response["resourceArn"]
 except ClientError as err:
 logger.error(
 "Couldn't create %s. Here's why: %s: %s",
 name,
 err.response["Error"]["Code"],
 err.response["Error"]["Message"],
)
 raise
 else:
 return self.ks_arn
```

- Per i dettagli sull'API, consulta [CreateKeyspace AWS SDK for Python \(Boto3\) API Reference](#).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. [Utilizzo di questo servizio con un SDK AWS](#) Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

## Utilizzare **CreateTable** con un SDK AWS

Gli esempi di codice seguenti mostrano come utilizzare CreateTable.

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 a new Amazon Keyspaces table.
 /// </summary>
 /// <param name="keyspaceName">The keyspace where the table will be
created.</param>
 /// <param name="schema">The schema for the new table.</param>
 /// <param name="tableName">The name of the new table.</param>
 /// <returns>The Amazon Resource Name (ARN) of the new table.</returns>
 public async Task<string> CreateTable(string keyspaceName, SchemaDefinition
schema, string tableName)
 {
 var request = new CreateTableRequest
 {
 KeyspaceName = keyspaceName,
 SchemaDefinition = schema,
 TableName = tableName,
 PointInTimeRecovery = new PointInTimeRecovery { Status =
PointInTimeRecoveryStatus.ENABLED }
 };

 var response = await _amazonKeyspaces.CreateTableAsync(request);
 return response.ResourceArn;
 }

```

- Per i dettagli sull'API, consulta la [CreateTable](#) sezione AWS SDK per .NET 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](#).

```

 public static void createTable(KeyspacesClient keyClient, String keySpace,
String tableName) {
 try {

```

```
// Set the columns.
ColumnDefinition defTitle = ColumnDefinition.builder()
 .name("title")
 .type("text")
 .build();

ColumnDefinition defYear = ColumnDefinition.builder()
 .name("year")
 .type("int")
 .build();

ColumnDefinition defReleaseDate = ColumnDefinition.builder()
 .name("release_date")
 .type("timestamp")
 .build();

ColumnDefinition defPlot = ColumnDefinition.builder()
 .name("plot")
 .type("text")
 .build();

List<ColumnDefinition> collist = new ArrayList<>();
collist.add(defTitle);
collist.add(defYear);
collist.add(defReleaseDate);
collist.add(defPlot);

// Set the keys.
PartitionKey yearKey = PartitionKey.builder()
 .name("year")
 .build();

PartitionKey titleKey = PartitionKey.builder()
 .name("title")
 .build();

List<PartitionKey> keyList = new ArrayList<>();
keyList.add(yearKey);
keyList.add(titleKey);

SchemaDefinition schemaDefinition = SchemaDefinition.builder()
 .partitionKeys(keyList)
 .allColumns(collist)
 .build();
```

```

 PointInTimeRecovery timeRecovery = PointInTimeRecovery.builder()
 .status(PointInTimeRecoveryStatus.ENABLED)
 .build();

 CreateTableRequest tableRequest = CreateTableRequest.builder()
 .keyspaceName(keySpace)
 .tableName(tableName)
 .schemaDefinition(schemaDefinition)
 .pointInTimeRecovery(timeRecovery)
 .build();

 CreateTableResponse response = keyClient.createTable(tableRequest);
 System.out.println("The table ARN is " + response.resourceArn());

 } catch (KeyspacesException e) {
 System.err.println(e.awsErrorDetails().errorMessage());
 System.exit(1);
 }
}

```

- Per i dettagli sull'API, consulta la [CreateTable](#) sezione AWS SDK for Java 2.x 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 createTable(
 keySpaceVal: String?,
 tableNameVal: String?,
) {
 // Set the columns.
 val defTitle =

```

```
 ColumnDefinition {
 name = "title"
 type = "text"
 }

val defYear =
 ColumnDefinition {
 name = "year"
 type = "int"
 }

val defReleaseDate =
 ColumnDefinition {
 name = "release_date"
 type = "timestamp"
 }

val defPlot =
 ColumnDefinition {
 name = "plot"
 type = "text"
 }

val collList = ArrayList<ColumnDefinition>()
collList.add(defTitle)
collList.add(defYear)
collList.add(defReleaseDate)
collList.add(defPlot)

// Set the keys.
val yearKey =
 PartitionKey {
 name = "year"
 }

val titleKey =
 PartitionKey {
 name = "title"
 }

val keyList = ArrayList<PartitionKey>()
keyList.add(yearKey)
keyList.add(titleKey)
```



```

val schemaDefinition0b =
 SchemaDefinition {
 partitionKeys = keyList
 allColumns = colList
 }

val timeRecovery =
 PointInTimeRecovery {
 status = PointInTimeRecoveryStatus.Enabled
 }

val tableRequest =
 CreateTableRequest {
 keyspaceName = keySpaceVal
 tableName = tableNameVal
 schemaDefinition = schemaDefinition0b
 pointInTimeRecovery = timeRecovery
 }

KeyspacesClient.fromEnvironment { region = "us-east-1" }.use { keyClient ->
 val response = keyClient.createTable(tableRequest)
 println("The table ARN is ${response.resourceArn}")
}
}

```

- Per i dettagli sull'API, [CreateTable](#) consulta AWS SDK for Kotlin 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 KeyspaceWrapper:
 """Encapsulates Amazon Keyspaces (for Apache Cassandra) keyspace and table
 actions."""

```

```
def __init__(self, keyspaces_client):
 """
 :param keyspaces_client: A Boto3 Amazon Keyspaces client.
 """
 self.keyspaces_client = keyspaces_client
 self.ks_name = None
 self.ks_arn = None
 self.table_name = None

 @classmethod
 def from_client(cls):
 keyspaces_client = boto3.client("keyspaces")
 return cls(keyspaces_client)

 def create_table(self, table_name):
 """
 Creates a table in the keyspace.
 The table is created with a schema for storing movie data
 and has point-in-time recovery enabled.

 :param table_name: The name to give the table.
 :return: The ARN of the new table.
 """
 try:
 response = self.keyspaces_client.create_table(
 keyspaceName=self.ks_name,
 tableName=table_name,
 schemaDefinition={
 "allColumns": [
 {"name": "title", "type": "text"},
 {"name": "year", "type": "int"},
 {"name": "release_date", "type": "timestamp"},
 {"name": "plot", "type": "text"},
],
 "partitionKeys": [{"name": "year"}, {"name": "title"}],
 },
 pointInTimeRecovery={"status": "ENABLED"},
)
 except ClientError as err:
 logger.error(
 "Couldn't create table %s. Here's why: %s: %s",
 table_name,
 err.response["Error"]["Code"],
```

```

 err.response["Error"]["Message"],
)
 raise
else:
 return response["resourceArn"]

```

- Per i dettagli sull'API, consulta [CreateTable AWS SDK for Python \(Boto3\) API Reference](#).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. [Utilizzo di questo servizio con un SDK AWS](#) Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

## Utilizzare **DeleteKeyspace** con un SDK AWS

Gli esempi di codice seguenti mostrano come utilizzare DeleteKeyspace.

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 existing keyspace.
/// </summary>
/// <param name="keyspaceName"></param>
/// <returns>A Boolean value indicating the success of the action.</returns>
public async Task<bool> DeleteKeyspace(string keyspaceName)
{

```

```
var response = await _amazonKeyspaces.DeleteKeyspaceAsync(
 new DeleteKeyspaceRequest { KeyspaceName = keyspaceName });
return response.HttpStatusCode == HttpStatusCode.OK;
}
```

- Per i dettagli sull'API, consulta la [DeleteKeyspaces](#) sezione AWS SDK per .NET 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](#).

```
public static void deleteKeyspace(KeyspacesClient keyClient, String
keyspaceName) {
 try {
 DeleteKeyspaceRequest deleteKeyspaceRequest =
DeleteKeyspaceRequest.builder()
 .keyspaceName(keyspaceName)
 .build();

 keyClient.deleteKeyspace(deleteKeyspaceRequest);

 } catch (KeyspacesException e) {
 System.err.println(e.awsErrorDetails().errorMessage());
 System.exit(1);
 }
}
```

- Per i dettagli sull'API, consulta la [DeleteKeyspaces](#) sezione AWS SDK for Java 2.x 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 deleteKeyspace(keyspaceNameVal: String?) {
 val deleteKeyspaceRequest =
 DeleteKeyspaceRequest {
 keyspaceName = keyspaceNameVal
 }

 KeyspacesClient.fromEnvironment { region = "us-east-1" }.use { keyClient ->
 keyClient.deleteKeyspace(deleteKeyspaceRequest)
 }
}
```

- Per i dettagli sull'API, [DeleteKeyspace](#) consulta AWS SDK for Kotlin 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 KeyspaceWrapper:
 """Encapsulates Amazon Keyspaces (for Apache Cassandra) keyspace and table
 actions."""

 def __init__(self, keyspaces_client):
 """
```

```

 :param keyspaces_client: A Boto3 Amazon Keyspaces client.
 """
 self.keyspaces_client = keyspaces_client
 self.ks_name = None
 self.ks_arn = None
 self.table_name = None

 @classmethod
 def from_client(cls):
 keyspaces_client = boto3.client("keyspaces")
 return cls(keyspaces_client)

 def delete_keyspace(self):
 """
 Deletes the keyspace.
 """
 try:
 self.keyspaces_client.delete_keyspace(keyspaceName=self.ks_name)
 self.ks_name = None
 except ClientError as err:
 logger.error(
 "Couldn't delete keyspace %s. Here's why: %s: %s",
 self.ks_name,
 err.response["Error"]["Code"],
 err.response["Error"]["Message"],
)
 raise

```

- Per i dettagli sull'API, consulta [DeleteKeyspace AWS SDK for Python \(Boto3\) API Reference](#).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. [Utilizzo di questo servizio con un SDK AWS](#) Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

## Utilizzare **DeleteTable** con un SDK AWS

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 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 Amazon Keyspaces table.
/// </summary>
/// <param name="keyspaceName">The keyspace containing the table.</param>
/// <param name="tableName">The name of the table to delete.</param>
/// <returns>A Boolean value indicating the success of the action.</returns>
public async Task<bool> DeleteTable(string keyspaceName, string tableName)
{
 var response = await _amazonKeyspaces.DeleteTableAsync(
 new DeleteTableRequest { KeyspaceName = keyspaceName, TableName =
tableName });
 return response.HttpStatusCode == HttpStatusCode.OK;
}
```

- Per i dettagli sull'API, consulta la [DeleteTable](#) sezione AWS SDK per .NET 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](#).

```
public static void deleteTable(KeyspacesClient keyClient, String
keyspaceName, String tableName) {
 try {
 DeleteTableRequest tableRequest = DeleteTableRequest.builder()
 .keyspaceName(keyspaceName)
 .tableName(tableName)
 .build();

 keyClient.deleteTable(tableRequest);

 } catch (KeyspacesException e) {
 System.err.println(e.awsErrorDetails().errorMessage());
 System.exit(1);
 }
}
```

- Per i dettagli sull'API, consulta la [DeleteTable](#) sezione AWS SDK for Java 2.x 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 deleteTable(
```



```

 keyspaceNameVal: String?,
 tableNameVal: String?,
) {
 val tableRequest =
 DeleteTableRequest {
 keyspaceName = keyspaceNameVal
 tableName = tableNameVal
 }

 KeyspacesClient.fromEnvironment { region = "us-east-1" }.use { keyClient ->
 keyClient.deleteTable(tableRequest)
 }
 }
}

```

- Per i dettagli sull'API, [DeleteTable](#) consulta AWS SDK for Kotlin 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 KeyspaceWrapper:
 """Encapsulates Amazon Keyspaces (for Apache Cassandra) keyspace and table
 actions."""

 def __init__(self, keyspaces_client):
 """
 :param keyspaces_client: A Boto3 Amazon Keyspaces client.
 """
 self.keyspaces_client = keyspaces_client
 self.ks_name = None
 self.ks_arn = None
 self.table_name = None

 @classmethod
 def from_client(cls):

```

```
keyspaces_client = boto3.client("keyspaces")
return cls(keyspaces_client)

def delete_table(self):
 """
 Deletes the table from the keyspace.
 """
 try:
 self.keyspaces_client.delete_table(
 keyspaceName=self.ks_name, tableName=self.table_name
)
 self.table_name = None
 except ClientError as err:
 logger.error(
 "Couldn't delete table %s. Here's why: %s: %s",
 self.table_name,
 err.response["Error"]["Code"],
 err.response["Error"]["Message"],
)
 raise
```

- Per i dettagli sull'API, consulta [DeleteTable AWS SDK for Python \(Boto3\) API Reference](#).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. [Utilizzo di questo servizio con un SDK AWS](#) Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

## Utilizzare **GetKeyspace** con un SDK AWS

Gli esempi di codice seguenti mostrano come utilizzare GetKeyspace.

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 data about a keyspace.
/// </summary>
/// <param name="keyspaceName">The name of the keyspace.</param>
/// <returns>The Amazon Resource Name (ARN) of the keyspace.</returns>
public async Task<string> GetKeyspace(string keyspaceName)
{
 var response = await _amazonKeyspaces.GetKeyspaceAsync(
 new GetKeyspaceRequest { KeyspaceName = keyspaceName });
 return response.ResourceArn;
}
```

- Per i dettagli sull'API, consulta la [GetKeyspace](#) sezione AWS SDK per .NET 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](#).

```
public static void checkKeyspaceExistence(KeyspacesClient keyClient, String
keyspaceName) {
 try {
 GetKeyspaceRequest keyspaceRequest = GetKeyspaceRequest.builder()
```

```

 .keyspaceName(keyspaceName)
 .build();

 GetKeyspaceResponse response =
keyClient.getKeyspace(keyspaceRequest);
 String name = response.keyspaceName();
 System.out.println("The " + name + " KeySpace is ready");

 } catch (KeyspacesException e) {
 System.err.println(e.awsErrorDetails().errorMessage());
 System.exit(1);
 }
}

```

- Per i dettagli sull'API, consulta la [GetKeyspace](#) sezione AWS SDK for Java 2.x 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 checkKeyspaceExistence(keyspaceNameVal: String?) {
 val keyspaceRequest =
 GetKeyspaceRequest {
 keyspaceName = keyspaceNameVal
 }
 KeyspacesClient.fromEnvironment { region = "us-east-1" }.use { keyClient ->
 val response: GetKeyspaceResponse =
keyClient.getKeyspace(keyspaceRequest)
 val name = response.keyspaceName
 println("The $name KeySpace is ready")
 }
}

```

- Per i dettagli sull'API, [GetKeyspace](#) consulta AWS SDK for Kotlin 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 KeyspaceWrapper:
 """Encapsulates Amazon Keyspaces (for Apache Cassandra) keyspace and table
 actions."""

 def __init__(self, keyspaces_client):
 """
 :param keyspaces_client: A Boto3 Amazon Keyspaces client.
 """
 self.keyspaces_client = keyspaces_client
 self.ks_name = None
 self.ks_arn = None
 self.table_name = None

 @classmethod
 def from_client(cls):
 keyspaces_client = boto3.client("keyspaces")
 return cls(keyspaces_client)

 def exists_keyspace(self, name):
 """
 Checks whether a keyspace exists.

 :param name: The name of the keyspace to look up.
 :return: True when the keyspace exists. Otherwise, False.
 """
 try:
 response = self.keyspaces_client.get_keyspace(keyspaceName=name)
 self.ks_name = response["keyspaceName"]
 self.ks_arn = response["resourceArn"]
```

```
 exists = True
 except ClientError as err:
 if err.response["Error"]["Code"] == "ResourceNotFoundException":
 logger.info("Keyspace %s does not exist.", name)
 exists = False
 else:
 logger.error(
 "Couldn't verify %s exists. Here's why: %s: %s",
 name,
 err.response["Error"]["Code"],
 err.response["Error"]["Message"],
)
 raise
 return exists
```

- Per i dettagli sull'API, consulta [GetKeyspace AWS](#) SDK for Python (Boto3) API Reference.

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di questo servizio con un SDK AWS](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

## Utilizzare **GetTable** con un SDK AWS

Gli esempi di codice seguenti mostrano come utilizzare GetTable.

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 Amazon Keyspaces table.
/// </summary>
/// <param name="keyspaceName">The keyspace containing the table.</param>
/// <param name="tableName">The name of the Amazon Keyspaces table.</param>
/// <returns>The response containing data about the table.</returns>
public async Task<GetTableResponse> GetTable(string keyspaceName, string
tableName)
{
 var response = await _amazonKeyspaces.GetTableAsync(
 new GetTableRequest { KeyspaceName = keyspaceName, TableName =
tableName });
 return response;
}
```

- Per i dettagli sull'API, consulta la [GetTable](#) sezione AWS SDK per .NET 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](#).

```
public static void checkTable(KeyspacesClient keyClient, String keyspaceName,
String tableName)
 throws InterruptedException {
 try {
 boolean tableStatus = false;
 String status;
 GetTableResponse response = null;
 GetTableRequest tableRequest = GetTableRequest.builder()
 .keyspaceName(keyspaceName)
 .tableName(tableName)
 .build();
```

```

 while (!tableStatus) {
 response = keyClient.getTable(tableRequest);
 status = response.statusAsString();
 System.out.println(". The table status is " + status);

 if (status.compareTo("ACTIVE") == 0) {
 tableStatus = true;
 }
 Thread.sleep(500);
 }

 List<ColumnDefinition> cols =
response.schemaDefinition().allColumns();
 for (ColumnDefinition def : cols) {
 System.out.println("The column name is " + def.name());
 System.out.println("The column type is " + def.type());
 }

 } catch (KeyspacesException e) {
 System.err.println(e.awsErrorDetails().errorMessage());
 System.exit(1);
 }
}

```

- Per i dettagli sull'API, consulta la [GetTable](#) sezione AWS SDK for Java 2.x 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 checkTable(
 keyspaceNameVal: String?,
 tableNameVal: String?,
) {
 var tableStatus = false

```



```

var status: String
var response: GetTableResponse? = null

val tableRequest =
 GetTableRequest {
 keyspaceName = keyspaceNameVal
 tableName = tableNameVal
 }
KeyspacesClient.fromEnvironment { region = "us-east-1" }.use { keyClient ->
 while (!tableStatus) {
 response = keyClient.getTable(tableRequest)
 status = response!!.status.toString()
 println(". The table status is $status")
 if (status.compareTo("ACTIVE") == 0) {
 tableStatus = true
 }
 delay(500)
 }
 val cols: List<ColumnDefinition>? =
response!!.schemaDefinition?.allColumns
 if (cols != null) {
 for (def in cols) {
 println("The column name is ${def.name}")
 println("The column type is ${def.type}")
 }
 }
}
}

```

- Per i dettagli sull'API, [GetTable](#) consulta AWS SDK for Kotlin 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 KeyspaceWrapper:
 """Encapsulates Amazon Keyspaces (for Apache Cassandra) keyspace and table
 actions."""

 def __init__(self, keyspaces_client):
 """
 :param keyspaces_client: A Boto3 Amazon Keyspaces client.
 """
 self.keyspaces_client = keyspaces_client
 self.ks_name = None
 self.ks_arn = None
 self.table_name = None

 @classmethod
 def from_client(cls):
 keyspaces_client = boto3.client("keyspaces")
 return cls(keyspaces_client)

 def get_table(self, table_name):
 """
 Gets data about a table in the keyspace.

 :param table_name: The name of the table to look up.
 :return: Data about the table.
 """
 try:
 response = self.keyspaces_client.get_table(
 keyspaceName=self.ks_name, tableName=table_name
)
 self.table_name = table_name
 except ClientError as err:
 if err.response["Error"]["Code"] == "ResourceNotFoundException":
 logger.info("Table %s does not exist.", table_name)
 self.table_name = None
 response = None
 else:
 logger.error(
 "Couldn't verify %s exists. Here's why: %s: %s",
 table_name,
 err.response["Error"]["Code"],
 err.response["Error"]["Message"],
)

```

```
 raise
 return response
```

- Per i dettagli sull'API, consulta [GetTable AWS SDK for Python \(Boto3\) API Reference](#).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di questo servizio con un SDK AWS](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

## Utilizzare **ListKeyspaces** con un SDK AWS

Gli esempi di codice seguenti mostrano come utilizzare ListKeyspaces.

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>
/// Lists all keyspaces for the account.
/// </summary>
/// <returns>Async task.</returns>
public async Task ListKeyspaces()
{
 var paginator = _amazonKeyspaces.Paginators.ListKeyspaces(new
ListKeyspacesRequest());

 Console.WriteLine("{0, -30}\t{1}", "Keyspace name", "Keyspace ARN");
```

```

 Console.WriteLine(new string('-', Console.WindowWidth));
 await foreach (var keyspace in paginator.Keyspaces)
 {

Console.WriteLine($"{keyspace.KeyspaceName, -30}\t{keyspace.ResourceArn}");
 }
 }
}

```

- Per i dettagli sull'API, consulta la [ListKeyspaces](#) sezione AWS SDK per .NET 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](#).

```

public static void listKeyspacesPaginator(KeyspacesClient keyClient) {
 try {
 ListKeyspacesRequest keyspacesRequest =
ListKeyspacesRequest.builder()
 .maxResults(10)
 .build();

 ListKeyspacesIterable listRes =
keyClient.listKeyspacesPaginator(keyspacesRequest);
 listRes.stream()
 .flatMap(r -> r.keyspaces().stream())
 .forEach(content -> System.out.println(" Name: " +
content.keyspaceName()));
 } catch (KeyspacesException e) {
 System.err.println(e.awsErrorDetails().errorMessage());
 System.exit(1);
 }
}

```

- Per i dettagli sull'API, consulta la [ListKeyspaces](#) sezione AWS SDK for Java 2.x 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 listKeyspacesPaginator() {
 KeyspacesClient.fromEnvironment { region = "us-east-1" }.use { keyClient ->
 keyClient
 .listKeyspacesPaginated(ListKeyspacesRequest {})
 .transform { it.keyspaces?.forEach { obj -> emit(obj) } }
 .collect { obj ->
 println("Name: ${obj.keyspaceName}")
 }
 }
}
```

- Per i dettagli sull'API, [ListKeyspaces](#) consulta AWS SDK for Kotlin 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 KeyspaceWrapper:
 """Encapsulates Amazon Keyspaces (for Apache Cassandra) keyspace and table
 actions."""

 def __init__(self, keyspaces_client):
 """
 :param keyspaces_client: A Boto3 Amazon Keyspaces client.
 """
 self.keyspaces_client = keyspaces_client
 self.ks_name = None
 self.ks_arn = None
 self.table_name = None

 @classmethod
 def from_client(cls):
 keyspaces_client = boto3.client("keyspaces")
 return cls(keyspaces_client)

 def list_keyspaces(self, limit):
 """
 Lists the keyspaces in your account.

 :param limit: The maximum number of keyspaces to list.
 """
 try:
 ks_paginator = self.keyspaces_client.get_paginator("list_keyspaces")
 for page in ks_paginator.paginate(PaginationConfig={"MaxItems":
limit}):
 for ks in page["keyspaces"]:
 print(ks["keyspaceName"])
 print(f"\t{ks['resourceArn']}")
 except ClientError as err:
 logger.error(
 "Couldn't list keyspaces. Here's why: %s: %s",
 err.response["Error"]["Code"],
 err.response["Error"]["Message"],
)
 raise

```

- Per i dettagli sull'API, consulta [ListKeyspaces AWS SDK for Python \(Boto3\) API Reference](#).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di questo servizio con un SDK AWS](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

## Utilizzare **ListTables** con un SDK AWS

Gli esempi di codice seguenti mostrano come utilizzare `ListTables`.

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>
/// Lists the Amazon Keyspaces tables in a keyspace.
/// </summary>
/// <param name="keyspaceName">The name of the keyspace.</param>
/// <returns>A list of TableSummary objects.</returns>
public async Task<List<TableSummary>> ListTables(string keyspaceName)
{
 var response = await _amazonKeyspaces.ListTablesAsync(new
ListTablesRequest { KeyspaceName = keyspaceName });
 response.Tables.ForEach(table =>
 {
 Console.WriteLine($"{table.KeyspaceName}\t{table.TableName}\t{table.ResourceArn}");
 });

 return response.Tables;
}
```

- Per i dettagli sull'API, consulta la [ListTables](#) sezione AWS SDK per .NET 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](#).

```
public static void listTables(KeyspacesClient keyClient, String keyspaceName)
{
 try {
 ListTablesRequest tablesRequest = ListTablesRequest.builder()
 .keyspaceName(keyspaceName)
 .build();

 ListTablesIterable listRes =
 keyClient.listTablesPaginator(tablesRequest);
 listRes.stream()
 .flatMap(r -> r.tables().stream())
 .forEach(content -> System.out.println(" ARN: " +
 content.resourceArn() +
 " Table name: " + content.tableName()));
 } catch (KeyspacesException e) {
 System.err.println(e.awsErrorDetails().errorMessage());
 System.exit(1);
 }
}
```

- Per i dettagli sull'API, consulta la [ListTables](#) sezione AWS SDK for Java 2.x 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 listTables(keyspaceNameVal: String?) {
 val tablesRequest =
 ListTablesRequest {
 keyspaceName = keyspaceNameVal
 }

 KeyspacesClient.fromEnvironment { region = "us-east-1" }.use { keyClient ->
 keyClient
 .listTablesPaginated(tablesRequest)
 .transform { it.tables?.forEach { obj -> emit(obj) } }
 .collect { obj ->
 println(" ARN: ${obj.resourceArn} Table name: ${obj.tableName}")
 }
 }
}
```

- Per i dettagli sull'API, [ListTables](#) consulta AWS SDK for Kotlin 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 KeyspaceWrapper:
```

```

"""Encapsulates Amazon Keyspaces (for Apache Cassandra) keyspace and table
actions."""

def __init__(self, keyspaces_client):
 """
 :param keyspaces_client: A Boto3 Amazon Keyspaces client.
 """
 self.keyspaces_client = keyspaces_client
 self.ks_name = None
 self.ks_arn = None
 self.table_name = None

 @classmethod
 def from_client(cls):
 keyspaces_client = boto3.client("keyspaces")
 return cls(keyspaces_client)

def list_tables(self):
 """
 Lists the tables in the keyspace.
 """
 try:
 table_paginator = self.keyspaces_client.get_paginator("list_tables")
 for page in table_paginator.paginate(keyspaceName=self.ks_name):
 for table in page["tables"]:
 print(table["tableName"])
 print(f"\t{table['resourceArn']}")
 except ClientError as err:
 logger.error(
 "Couldn't list tables in keyspace %s. Here's why: %s: %s",
 self.ks_name,
 err.response["Error"]["Code"],
 err.response["Error"]["Message"],
)
 raise

```

- Per i dettagli sull'API, consulta [ListTables AWS SDK for Python \(Boto3\) API Reference](#).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta [Utilizzo di questo servizio con un SDK AWS](#). Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

## Utilizzare **RestoreTable** con un SDK AWS

Gli esempi di codice seguenti mostrano come utilizzare `RestoreTable`.

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>
/// Restores the specified table to the specified point in time.
/// </summary>
/// <param name="keyspaceName">The keyspace containing the table.</param>
/// <param name="tableName">The name of the table to restore.</param>
/// <param name="timestamp">The time to which the table will be restored.</
param>
/// <returns>The Amazon Resource Name (ARN) of the restored table.</returns>
public async Task<string> RestoreTable(string keyspaceName, string tableName,
string restoredTableName, DateTime timestamp)
{
 var request = new RestoreTableRequest
 {
 RestoreTimestamp = timestamp,
 SourceKeyspaceName = keyspaceName,
 SourceTableName = tableName,
 TargetKeyspaceName = keyspaceName,
 TargetTableName = restoredTableName
 }
}
```

```
};

var response = await _amazonKeyspaces.RestoreTableAsync(request);
return response.RestoredTableARN;
}
```

- Per i dettagli sull'API, consulta la [RestoreTable](#) sezione AWS SDK per .NET 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](#).

```
public static void restoreTable(KeyspacesClient keyClient, String
keyspaceName, ZonedDateTime utc) {
 try {
 Instant myTime = utc.toInstant();
 RestoreTableRequest restoreTableRequest =
RestoreTableRequest.builder()
 .restoreTimestamp(myTime)
 .sourceTableName("Movie")
 .targetKeyspaceName(keyspaceName)
 .targetTableName("MovieRestore")
 .sourceKeyspaceName(keyspaceName)
 .build();

 RestoreTableResponse response =
keyClient.restoreTable(restoreTableRequest);
 System.out.println("The ARN of the restored table is " +
response.restoredTableARN());

 } catch (KeyspacesException e) {
 System.err.println(e.awsErrorDetails().errorMessage());
 System.exit(1);
 }
}
```

```
}
```

- Per i dettagli sull'API, consulta la [RestoreTable](#) sezione AWS SDK for Java 2.x 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 restoreTable(
 keyspaceName: String?,
 utc: ZonedDateTime,
) {
 // Create an aws.smithy.kotlin.runtime.time.Instant value.
 val timeStamp =
 aws.smithy.kotlin.runtime.time
 .Instant(utc.toInstant())
 val restoreTableRequest =
 RestoreTableRequest {
 restoreTimestamp = timeStamp
 sourceTableName = "MovieKotlin"
 targetKeyspaceName = keyspaceName
 targetTableName = "MovieRestore"
 sourceKeyspaceName = keyspaceName
 }

 KeyspacesClient.fromEnvironment { region = "us-east-1" }.use { keyClient ->
 val response = keyClient.restoreTable(restoreTableRequest)
 println("The ARN of the restored table is ${response.restoredTableArn}")
 }
}
```

- Per i dettagli sull'API, [RestoreTable](#) consulta AWS SDK for Kotlin 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 KeyspaceWrapper:
 """Encapsulates Amazon Keyspaces (for Apache Cassandra) keyspace and table
 actions."""

 def __init__(self, keyspaces_client):
 """
 :param keyspaces_client: A Boto3 Amazon Keyspaces client.
 """
 self.keyspaces_client = keyspaces_client
 self.ks_name = None
 self.ks_arn = None
 self.table_name = None

 @classmethod
 def from_client(cls):
 keyspaces_client = boto3.client("keyspaces")
 return cls(keyspaces_client)

 def restore_table(self, restore_timestamp):
 """
 Restores the table to a previous point in time. The table is restored
 to a new table in the same keyspace.

 :param restore_timestamp: The point in time to restore the table. This
 time must be in UTC format.
 :return: The name of the restored table.
 """
 try:
 restored_table_name = f"{self.table_name}_restored"
 self.keyspaces_client.restore_table(
```

```
 sourceKeyspaceName=self.ks_name,
 sourceTableName=self.table_name,
 targetKeyspaceName=self.ks_name,
 targetTableName=restored_table_name,
 restoreTimestamp=restore_timestamp,
)
except ClientError as err:
 logger.error(
 "Couldn't restore table %s. Here's why: %s: %s",
 restore_timestamp,
 err.response["Error"]["Code"],
 err.response["Error"]["Message"],
)
 raise
else:
 return restored_table_name
```

- Per i dettagli sull'API, consulta [RestoreTable AWS SDK for Python \(Boto3\) API Reference](#).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. [Utilizzo di questo servizio con un SDK AWS](#) Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

## Utilizzare **UpdateTable** con un SDK AWS

Gli esempi di codice seguenti mostrano come utilizzare UpdateTable.

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>
/// Updates the movie table to add a boolean column named watched.
/// </summary>
/// <param name="keyspaceName">The keyspace containing the table.</param>
/// <param name="tableName">The name of the table to change.</param>
/// <returns>The Amazon Resource Name (ARN) of the updated table.</returns>
public async Task<string> UpdateTable(string keyspaceName, string tableName)
{
 var newColumn = new ColumnDefinition { Name = "watched", Type =
"boolean" };
 var request = new UpdateTableRequest
 {
 KeyspaceName = keyspaceName,
 TableName = tableName,
 AddColumns = new List<ColumnDefinition> { newColumn }
 };
 var response = await _amazonKeyspaces.UpdateTableAsync(request);
 return response.ResourceArn;
}
```

- Per i dettagli sull'API, consulta la [UpdateTable](#) sezione AWS SDK per .NET 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](#).

```
public static void updateTable(KeyspacesClient keyClient, String keySpace,
String tableName) {
 try {
 ColumnDefinition def = ColumnDefinition.builder()
 .name("watched")
 .type("boolean")
 .build();

 UpdateTableRequest tableRequest = UpdateTableRequest.builder()
 .keyspaceName(keySpace)
 .tableName(tableName)
 .addColumnns(def)
 .build();

 keyClient.updateTable(tableRequest);

 } catch (KeyspacesException e) {
 System.err.println(e.awsErrorDetails().errorMessage());
 System.exit(1);
 }
}
```

- Per i dettagli sull'API, consulta la [UpdateTable](#) sezione AWS SDK for Java 2.x 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 updateTable(
 keySpace: String?,
 tableNameVal: String?,
) {
 val def =
 ColumnDefinition {
 name = "watched"
 type = "boolean"
 }

 val tableRequest =
 UpdateTableRequest {
 keyspaceName = keySpace
 tableName = tableNameVal
 addColumns = listOf(def)
 }

 KeyspacesClient.fromEnvironment { region = "us-east-1" }.use { keyClient ->
 keyClient.updateTable(tableRequest)
 }
}
```

- Per i dettagli sull'API, [UpdateTable](#) consulta AWS SDK for Kotlin 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 KeyspaceWrapper:
 """Encapsulates Amazon Keyspaces (for Apache Cassandra) keyspace and table
 actions."""

 def __init__(self, keyspaces_client):
 """
 :param keyspaces_client: A Boto3 Amazon Keyspaces client.
 """
 self.keyspaces_client = keyspaces_client
 self.ks_name = None
 self.ks_arn = None
 self.table_name = None

 @classmethod
 def from_client(cls):
 keyspaces_client = boto3.client("keyspaces")
 return cls(keyspaces_client)

 def update_table(self):
 """
 Updates the schema of the table.

 This example updates a table of movie data by adding a new column
 that tracks whether the movie has been watched.
 """
 try:
 self.keyspaces_client.update_table(
 keyspaceName=self.ks_name,
 tableName=self.table_name,
 addColumns=[{"name": "watched", "type": "boolean"}],
)
```

```
except ClientError as err:
 logger.error(
 "Couldn't update table %s. Here's why: %s: %s",
 self.table_name,
 err.response["Error"]["Code"],
 err.response["Error"]["Message"],
)
 raise
```

- Per i dettagli sull'API, consulta [UpdateTable AWS SDK for Python \(Boto3\) API Reference](#).

Per un elenco completo delle guide per sviluppatori AWS SDK e degli esempi di codice, consulta. [Utilizzo di questo servizio con un SDK AWS](#) Questo argomento include anche informazioni su come iniziare e dettagli sulle versioni precedenti dell'SDK.

# Librerie e strumenti Amazon Keyspaces (per Apache Cassandra)

Questa sezione fornisce informazioni sulle librerie, esempi di codice e strumenti di Amazon Keyspaces (per Apache Cassandra).

Argomenti

- [Librerie ed esempi](#)
- [Esempi evidenziati e repository degli strumenti di sviluppo](#)

## Librerie ed esempi

[Puoi trovare le librerie open source e gli strumenti di sviluppo di Amazon Keyspaces GitHub nel repository AWSed AWS samples.](#)

## Toolkit per sviluppatori Amazon Keyspaces (per Apache Cassandra)

Questo repository fornisce un'immagine docker con utili strumenti di sviluppo per Amazon Keyspaces. Ad esempio, include un file CQLSHRC con le migliori pratiche, un'espansione di AWS autenticazione opzionale per cqlsh e strumenti di supporto per eseguire attività comuni. Il toolkit è ottimizzato per Amazon Keyspaces, ma funziona anche con i cluster Apache Cassandra.

<https://github.com/aws-samples/amazon-keyspaces-toolkit>.

## Esempi di Amazon Keyspaces (per Apache Cassandra)

Questo repository è il nostro elenco ufficiale di codice di esempio di Amazon Keyspaces. [Il repository è suddiviso in sezioni per lingua \(vedi Esempi\)](#). Ogni lingua ha una propria sottosezione di esempi. Questi esempi illustrano implementazioni e modelli comuni del servizio Amazon Keyspaces che puoi utilizzare per la creazione di applicazioni.

<https://github.com/aws-samples/amazon-keyspaces-examples/>.

## AWS Plugin di autenticazione Signature Version 4 (SigV4)

I plugin consentono di gestire l'accesso ad Amazon Keyspaces AWS Identity and Access Management utilizzando utenti e ruoli (IAM).

Java: <https://github.com/aws/aws-sigv4-auth-cassandra-java-driver-plugin>

Node.js: <https://github.com/aws/aws-sigv4-auth-cassandra-nodejs-driver-plugin>.

Python: <https://github.com/aws/aws-sigv4-auth-cassandra-python-driver-plugin>.

Vai: <https://github.com/aws/aws-sigv4-auth-cassandra-gocql-driver-plugin>.

## Esempi evidenziati e repository degli strumenti di sviluppo

Di seguito è riportata una selezione di utili strumenti di community per Amazon Keyspaces (per Apache Cassandra).

### Buffer del protocollo Amazon Keyspaces

Puoi utilizzare Protocol Buffers (Protobuf) con Amazon Keyspaces per fornire un'alternativa a Apache Cassandra User Defined Types (). UDTs Protobuf è un formato di dati multiplatforma gratuito e open source utilizzato per serializzare dati strutturati. È possibile archiviare i dati Protobuf utilizzando il tipo di dati CQL e il refactor preservando al UDTs contempo BLOB i dati strutturati tra applicazioni e linguaggi di programmazione.

Questo repository fornisce un esempio di codice che si connette ad Amazon Keyspaces, crea una nuova tabella e inserisce una riga contenente un messaggio Protobuf. Quindi la riga viene letta con una forte coerenza.

<https://github.com/aws-samples/amazon-keyspaces-examples/tree/main/java/datastax-v4/protobuf-user-defined-types>

### CloudFormation modello per creare CloudWatch dashboard Amazon per i parametri di Amazon Keyspaces (per Apache Cassandra)

Questo repository fornisce CloudFormation modelli per configurare rapidamente i CloudWatch parametri per Amazon Keyspaces. L'utilizzo di questo modello ti consentirà di iniziare più facilmente fornendo dashboard CloudWatch preconfigurate distribuibili con metriche di uso comune.

<https://github.com/aws-samples/amazon-keyspaces-cloudwatch-cloudformation-templates>.

### Utilizzo di Amazon Keyspaces (per Apache Cassandra) con AWS Lambda

Il repository contiene esempi che mostrano come connettersi ad Amazon Keyspaces da Lambda. Di seguito sono riportati alcuni esempi.

C#/ .NET: <https://github.com/aws-samples/amazon-keyspaces-examples/tree/main/dotnet/datastax-v3/connection-lambda>

Giava: <https://github.com/aws-samples/amazon-keyspaces-examples/tree/main/java/datastax-v4/connection-lambda>

Un altro esempio di Lambda che mostra come distribuire e utilizzare Amazon Keyspaces da un Python Lambda è disponibile nel seguente repository.

<https://github.com/aws-samples/aws-keyspaces-lambda-python>

## Utilizzo di Amazon Keyspaces (per Apache Cassandra) con Spring

Questo è un esempio che mostra come usare Amazon Keyspaces con Spring Boot.

<https://github.com/aws-samples/amazon-keyspaces-examples/tree/main/java/datastax-v4/spring>

## Utilizzo di Amazon Keyspaces (per Apache Cassandra) con Scala

Questo è un esempio che mostra come connettersi ad Amazon Keyspaces utilizzando il plug-in di autenticazione SigV4 con Scala.

<https://github.com/aws-samples/amazon-keyspaces-examples/tree/main/scala/datastax-v4/connection-sigv4>

## Utilizzo di Amazon Keyspaces (per Apache Cassandra) con AWS Glue

Questo è un esempio che mostra come usare Amazon Keyspaces con AWS Glue

<https://github.com/aws-samples/amazon-keyspaces-examples/tree/main/scala/datastax-v4/aws-glue>

## Amazon Keyspaces (per Apache Cassandra) da Cassandra Query Language (CQL) a convertitore CloudFormation

Questo pacchetto implementa uno strumento da riga di comando per convertire gli script di Apache Cassandra Query Language (CQL) in CloudFormation modelli (CloudFormation), che consente agli schemi di Amazon Keyspaces di essere facilmente gestiti in pile CloudFormation

<https://github.com/aws/amazon-keyspaces-cql-to-cfn-converter>.

## Amazon Keyspaces (per Apache Cassandra), aiutanti per il driver Apache Cassandra per Java

Questo repository contiene politiche, esempi e best practice relative ai driver per l'utilizzo del driver DataStax Java con Amazon Keyspaces (per Apache Cassandra).

<https://github.com/aws-samples/amazon-keyspaces-java-driver-helpers>.

## Dimostrazione di compressione istantanea di Amazon Keyspaces (per Apache Cassandra)

Questo repository dimostra come comprimere, archiviare e leggere/scrivere oggetti di grandi dimensioni per prestazioni più veloci e ridurre i costi di throughput e storage.

<https://github.com/aws-samples/amazon-keyspaces-compression-example>.

## Demo di Amazon Keyspaces (per Apache Cassandra) e del codec Amazon S3

Il codec Amazon S3 personalizzato supporta la mappatura trasparente e configurabile dall'utente dei puntatori UUID agli oggetti Amazon S3.

<https://github.com/aws-samples/amazon-keyspaces-large-object-s3-demo>.



# Le migliori pratiche per la progettazione e l'architettura con Amazon Keyspaces (per Apache Cassandra)

Utilizza questa sezione per trovare rapidamente consigli per massimizzare le prestazioni e ridurre al minimo i costi di throughput quando lavori con Amazon Keyspaces.

## Indice

- [Principali differenze e principi di progettazione della progettazione NoSQL](#)
  - [Differenze tra la progettazione di dati relazionali e NoSQL.](#)
  - [Due concetti chiave per la progettazione di NoSQL.](#)
  - [Approccio alla progettazione NoSQL.](#)
- [Ottimizza le connessioni dei driver client per l'ambiente serverless](#)
  - [Come funzionano le connessioni in Amazon Keyspaces](#)
  - [Come configurare le connessioni in Amazon Keyspaces](#)
  - [Come configurare la politica di riprova per le connessioni in Amazon Keyspaces](#)
  - [Come configurare le connessioni tramite endpoint VPC in Amazon Keyspaces](#)
  - [Come monitorare le connessioni in Amazon Keyspaces](#)
  - [Come gestire gli errori di connessione in Amazon Keyspaces](#)
- [Migliori pratiche di modellazione dei dati: consigli per la progettazione di modelli di dati](#)
  - [Come utilizzare le chiavi di partizione in modo efficace in Amazon Keyspaces](#)
    - [Usa lo sharding in scrittura per distribuire uniformemente i carichi di lavoro tra le partizioni](#)
      - [Sharding utilizzando chiavi di partizione composte e valori casuali](#)
      - [Sharding utilizzando chiavi di partizione composte e valori calcolati](#)
- [Ottimizzazione dei costi delle tabelle Amazon Keyspaces](#)
  - [Valutazione dei costi a livello di tabella](#)
    - [Come visualizzare i costi di una singola tabella Amazon Keyspaces](#)
    - [Visualizzazione predefinita di Esploratore dei costi](#)
    - [Come utilizzare e applicare i tag di tabella in Esploratore dei costi](#)
  - [Valuta la modalità di capacità della tua tabella](#)
    - [Modalità di capacità della tabella disponibili](#)
    - [Quando selezionare la modalità di capacità on demand](#)

- [Quando selezionare la modalità di capacità assegnata](#)
- [Fattori aggiuntivi da valutare nella scelta di una modalità di capacità della tabella](#)
- [Valuta le impostazioni di Application Auto Scaling della tua tabella](#)
  - [Informazioni sulle impostazioni dell'Application Auto Scaling](#)
  - [Come identificare le tabelle con un basso utilizzo di destinazione \( \$\leq 50\%\$ \)](#)
  - [Come gestire i carichi di lavoro con varianza stagionale](#)
  - [Come affrontare carichi di lavoro con picchi di lavoro con pattern sconosciuti](#)
  - [Come gestire i carichi di lavoro con applicazioni collegate](#)
- [Identifica le risorse inutilizzate per ottimizzare i costi in Amazon Keyspaces](#)
  - [Come identificare le risorse inutilizzate](#)
  - [Identificazione delle risorse non utilizzate della tabella](#)
  - [Pulizia delle risorse non utilizzate della tabella](#)
  - [Pulizia dei backup di point-in-time ripristino inutilizzati \(PITR\)](#)
- [Valuta i modelli di utilizzo delle tabelle per ottimizzare prestazioni e costi](#)
  - [Riduzione del numero di operazioni ad elevata consistenza di lettura](#)
  - [Abilitazione del Time to Live \(TTL\)](#)
- [Valutare la capacità fornita per un provisioning di dimensioni adeguate](#)
  - [Come recuperare i parametri di consumo dalle tabelle Amazon Keyspaces](#)
  - [Come identificare tabelle Amazon Keyspaces il cui provisioning è insufficiente](#)
  - [Come identificare tabelle Amazon Keyspaces sovraforate](#)

## Principali differenze e principi di progettazione della progettazione NoSQL

I sistemi di database NoSQL come Amazon Keyspaces utilizzano modelli alternativi per la gestione dei dati, come coppie chiave-valore o archiviazione di documenti. Quando si passa da un sistema di gestione di database relazionali a un sistema di database NoSQL come Amazon Keyspaces, è importante comprendere le differenze principali e gli approcci di progettazione specifici.

### Argomenti

- [Differenze tra la progettazione di dati relazionali e NoSQL.](#)
- [Due concetti chiave per la progettazione di NoSQL.](#)

- [Approccio alla progettazione NoSQL.](#)

## Differenze tra la progettazione di dati relazionali e NoSQL.

I sistemi di gestione dei database relazionali (RDBMS) e i database NoSQL, hanno diversi vantaggi e svantaggi:

- Nei sistemi RDBMS, si possono eseguire query sui dati in modo flessibile ma le query sono relativamente costose e non si ridimensionano bene in situazioni di traffico elevato (consulta [the section called “Modellazione dei dati”](#)).
- In un database NoSQL come Amazon Keyspaces, i dati possono essere interrogati in modo efficiente in un numero limitato di modi, al di fuori dei quali le query possono essere costose e lente.

Queste differenze rendono la progettazione del database diversa tra i due sistemi:

- Nei sistemi RDBMS, si progetta tenendo a mente la flessibilità senza doversi preoccupare dei dettagli dell'implementazione o delle prestazioni. L'ottimizzazione delle query in genere non ha ripercussioni sulla progettazione dello schema, ma la normalizzazione è importante.
- In Amazon Keyspaces, progetti lo schema in modo specifico per rendere le query più comuni e importanti il più velocemente ed economicamente possibile. Le tue strutture di dati sono programmate per i requisiti specifici dei casi d'uso del tuo business.

## Due concetti chiave per la progettazione di NoSQL.

La progettazione di un NoSQL richiede un approccio diverso rispetto alla progettazione di un RDBMS. Per un sistema RDBMS puoi creare un modello di dati normalizzati senza pensare ai modelli di accesso. Puoi estenderlo in seguito quando si presentano nuovi quesiti e requisiti di query. Puoi organizzare ogni tipo di dati nella sua tabella.

In che modo la progettazione NoSQL è diversa

- Al contrario, non dovresti iniziare a progettare il tuo schema per Amazon Keyspaces finché non conosci le domande a cui deve rispondere. È essenziale capire da subito i problemi di business e i casi d'uso dell'applicazione.
- È necessario mantenere il minor numero possibile di tabelle in un'applicazione Amazon Keyspaces. Avere meno tabelle mantiene le cose più scalabili, richiede una minore gestione delle

autorizzazioni e riduce il sovraccarico per l'applicazione Amazon Keyspaces. Può anche aiutare a mantenere i costi di backup complessivi più bassi.

## Approccio alla progettazione NoSQL.

Il primo passaggio nella progettazione di un'applicazione Amazon Keyspaces consiste nell'identificare i modelli di query specifici che il sistema deve soddisfare.

In particolare, è importante capire le tre proprietà fondamentali dei modelli di accesso all'applicazione prima di iniziare:

- **Dimensione dei dati:** sapere quanti dati verranno archiviati e richiesti contemporaneamente aiuta a determinare il modo più efficace per partizionare i dati.
- **Forma dei dati:** invece di modificare i dati quando viene elaborata una query (come nei sistemi RDBMS), un database NoSQL organizza i dati in modo che la loro forma nel database corrisponda al soggetto della query. Questo è un fattore chiave nell'aumentare la velocità e la scalabilità.
- **Velocità dei dati:** Amazon Keyspaces si ridimensiona aumentando il numero di partizioni fisiche disponibili per elaborare le query e distribuendo in modo efficiente i dati tra tali partizioni. Conoscere in anticipo quali saranno i picchi di carico delle query può aiutare a determinare come partizionare i dati per utilizzare al meglio la capacità. I/O

Dopo aver identificato i requisiti specifici della query, puoi organizzare i dati in casi ai principi generali che definiscono le prestazioni:

- **Raggruppamento dei dati correlati.** Studi di ricerca sull'ottimizzazione della tabella di routing effettuati 20 anni fa hanno dimostrato che "l'ubicazione di riferimento" era il fattore più importante nel velocizzare il tempo di risposta, ovvero mantenere i dati correlati in un'unica posizione. Al giorno d'oggi, ciò vale anche nei sistemi NoSQL, dove tenere i dati correlati a distanza ravvicinata ha un impatto importante sui costi e sulle prestazioni. Invece di distribuire gli elementi con dati correlati su tabelle multiple, devi tenere gli elementi correlati il più vicino possibile nel tuo sistema NoSQL.

Come regola generale, è necessario mantenere il minor numero possibile di tabelle in un'applicazione Amazon Keyspaces.

Le eccezioni sono i casi in cui sono coinvolti dati di serie temporali a volumi elevati o i set di dati che hanno modelli di accesso molto diversi. Una tabella singola con indici invertiti possono

solitamente abilitare query semplici per creare ed estrarre strutture di dati gerarchici complesse richieste dalla tua applicazione.

- Utilizzo dell'ordine di selezione. Gli elementi correlati possono essere raggruppati e le query possono essere eseguite facilmente se la progettazione della chiave prevede che siano raggruppati. Questa è una strategia di progettazione NoSQL importante.
- Distribuzione delle query. È inoltre importante che un volume elevato di query non si concentri su una sola parte del database, dove può superare I/O la capacità. Invece, devi progettare chiavi di dati per distribuire il traffico il più possibile in maniera uniforme tra le partizioni, evitando "hot spot".

Questi principi generali si traducono in alcuni modelli di progettazione comuni che puoi utilizzare per modellare i dati in modo efficiente in Amazon Keyspaces.

## Ottimizza le connessioni dei driver client per l'ambiente serverless

Per comunicare con Amazon Keyspaces, puoi utilizzare uno qualsiasi dei driver client Apache Cassandra esistenti di tua scelta. Poiché Amazon Keyspaces è un servizio serverless, ti consigliamo di ottimizzare la configurazione della connessione del driver client per le esigenze di throughput della tua applicazione. Questo argomento introduce le migliori pratiche, tra cui il calcolo del numero di connessioni richieste dall'applicazione, nonché il monitoraggio e la gestione degli errori delle connessioni.

### Argomenti

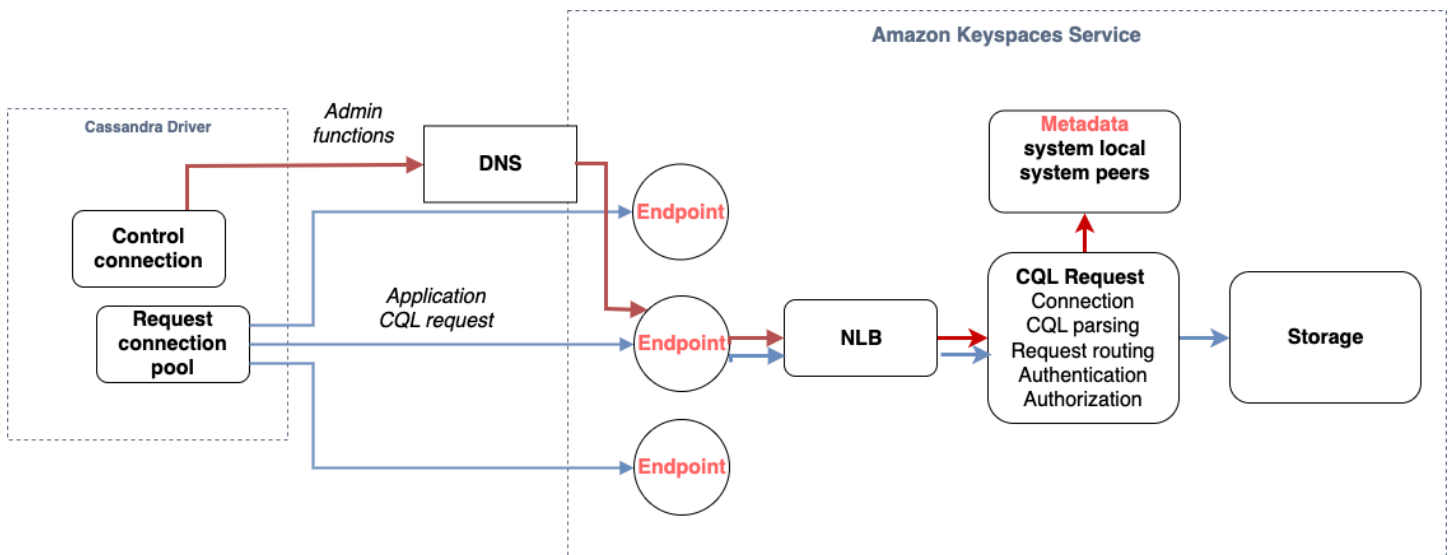
- [Come funzionano le connessioni in Amazon Keyspaces](#)
- [Come configurare le connessioni in Amazon Keyspaces](#)
- [Come configurare la politica di riprova per le connessioni in Amazon Keyspaces](#)
- [Come configurare le connessioni tramite endpoint VPC in Amazon Keyspaces](#)
- [Come monitorare le connessioni in Amazon Keyspaces](#)
- [Come gestire gli errori di connessione in Amazon Keyspaces](#)

## Come funzionano le connessioni in Amazon Keyspaces

Questa sezione offre una panoramica di come funzionano le connessioni dei driver client in Amazon Keyspaces. Poiché una configurazione errata del driver client Cassandra può causare `PerConnectionRequestExceeded` eventi in Amazon Keyspaces, è necessario configurare la

giusta quantità di connessioni nella configurazione del driver client per evitare questi e altri errori di connessione simili.

Durante la connessione ad Amazon Keyspaces, il driver richiede un endpoint seed per stabilire una connessione iniziale. Amazon Keyspaces utilizza DNS per instradare la connessione iniziale verso uno dei tanti endpoint disponibili. Gli endpoint sono collegati a sistemi di bilanciamento del carico di rete che a loro volta stabiliscono una connessione a uno dei gestori di richieste del parco macchine. Dopo aver stabilito la connessione iniziale, il driver client raccoglie informazioni su tutti gli endpoint disponibili dalla tabella `system.peers`. Con queste informazioni, il driver client può creare connessioni aggiuntive agli endpoint elencati. Il numero di connessioni che il driver client può creare è limitato dal numero di connessioni locali specificate nelle impostazioni del driver client. Per impostazione predefinita, la maggior parte dei driver client stabilisce una connessione per endpoint e stabilisce un pool di connessioni con Cassandra e bilancia il carico delle query su quel pool di connessioni. Sebbene sia possibile stabilire più connessioni allo stesso endpoint, tramite il sistema di bilanciamento del carico di rete queste possono essere collegate a molti gestori di richieste diversi. Quando ci si connette tramite l'endpoint pubblico, stabilendo una connessione a ciascuno dei nove endpoint elencati nella `system.peers` tabella si ottengono nove connessioni a diversi gestori di richieste.



## Come configurare le connessioni in Amazon Keyspaces

Amazon Keyspaces supporta fino a 3.000 query CQL per connessione TCP al secondo. Poiché non c'è limite al numero di connessioni che un driver può stabilire, consigliamo di utilizzare come target solo 500 richieste CQL al secondo per connessione per evitare sovraccarichi, picchi di traffico e un migliore bilanciamento del carico. Segui questi passaggi per assicurarti che la connessione del driver sia configurata correttamente per le esigenze dell'applicazione.

Aumenta il numero di connessioni per indirizzo IP che il driver mantiene nel suo pool di connessioni.

- La maggior parte dei driver Cassandra stabilisce un pool di connessioni con Cassandra e bilancia il carico delle query su quel pool di connessioni. Il comportamento predefinito della maggior parte dei driver consiste nello stabilire una singola connessione a ciascun endpoint. Amazon Keyspaces espone nove indirizzi IP peer ai driver, quindi, in base al comportamento predefinito della maggior parte dei driver, si ottengono 9 connessioni. Amazon Keyspaces supporta fino a 3.000 query CQL per connessione TCP al secondo, pertanto il throughput massimo di query CQL di un driver che utilizza le impostazioni predefinite è di 27.000 query CQL al secondo. Se utilizzi le impostazioni predefinite del driver, una singola connessione potrebbe dover elaborare più del throughput massimo di query CQL di 3.000 query CQL al secondo. Ciò potrebbe causare eventi. `PerConnectionRequestExceeded`
- Per evitare `PerConnectionRequestExceeded` eventi, è necessario configurare il driver in modo da creare connessioni aggiuntive per endpoint per distribuire il throughput.
- Come best practice in Amazon Keyspaces, supponi che ogni connessione possa supportare 500 query CQL al secondo.
- Ciò significa che per un'applicazione di produzione che deve supportare circa 27.000 query CQL al secondo distribuite sui nove endpoint disponibili, è necessario configurare sei connessioni per endpoint. Ciò garantisce che ogni connessione elabori non più di 500 richieste al secondo.

Calcola il numero di connessioni per indirizzo IP che devi configurare per il tuo driver in base alle esigenze dell'applicazione.

Per determinare il numero di connessioni da configurare per endpoint per l'applicazione, considera l'esempio seguente. Hai un'applicazione che deve supportare 20.000 query CQL al secondo costituite da 10.000 `INSERTSELECT`, 5.000 e 5.000 operazioni. `DELETE` L'applicazione Java è in esecuzione su tre istanze su Amazon Elastic Container Service (Amazon ECS), dove ogni istanza stabilisce una singola sessione su Amazon Keyspaces. Il calcolo che puoi utilizzare per stimare quante connessioni devi configurare per il tuo driver utilizza il seguente input.

1. Il numero di richieste al secondo che l'applicazione deve supportare.
2. Il numero di istanze disponibili a cui viene sottratta una per tenere conto della manutenzione o dell'errore.
3. Il numero di endpoint disponibili. Se ti connetti tramite endpoint pubblici, hai nove endpoint disponibili. Se utilizzi endpoint VPC, hai da due a cinque endpoint disponibili, a seconda della regione.

4. Utilizza 500 query CQL al secondo per connessione come best practice per Amazon Keyspaces.
5. Arrotonda il risultato.

In questo esempio, la formula ha il seguente aspetto.

```
20,000 CQL queries / (3 instances - 1 failure) / 9 public endpoints / 500 CQL queries
per second = ROUND(2.22) = 3
```

In base a questo calcolo, è necessario specificare tre connessioni locali per endpoint nella configurazione del driver. Per le connessioni remote, configura solo una connessione per endpoint.

## Come configurare la politica di riprova per le connessioni in Amazon Keyspaces

Quando configuri la politica di riprova per una connessione ad Amazon Keyspaces, ti consigliamo di implementare la politica di riprova di Amazon Keyspaces.

`AmazonKeyspacesExponentialRetryPolicy` Questa politica di nuovi tentativi è più adatta a riprovare su connessioni diverse ad Amazon Keyspaces rispetto a quelle del driver. `DefaultRetryPolicy`

Con `AmazonKeyspacesExponentialRetryPolicy`, puoi configurare il numero di tentativi di nuovo tentativo per la connessione che soddisfi le tue esigenze. Per impostazione predefinita, il numero di tentativi di nuovo tentativo `AmazonKeyspacesExponentialRetryPolicy` è impostato su 3.

Un ulteriore vantaggio è che la politica di riprova di Amazon Keyspaces restituisce l'eccezione originale restituita dal servizio, il che indica il motivo per cui il tentativo di richiesta non è riuscito. La politica di riprova predefinita restituisce solo il messaggio generico `NoHostAvailableException`, il che potrebbe nascondere informazioni sull'errore della richiesta.

Per configurare la politica di ripetizione delle richieste utilizzando `AmazonKeyspacesExponentialRetryPolicy`, si consiglia di configurare un numero limitato di tentativi e di gestire le eventuali eccezioni restituite nel codice dell'applicazione.

Per esempi di codice che implementano le politiche di ripetizione dei tentativi, consulta le politiche di [riprova di Amazon Keyspaces su Github](#).



## Come configurare le connessioni tramite endpoint VPC in Amazon Keyspaces

Quando ti connetti tramite endpoint VPC privati, molto probabilmente hai 3 endpoint disponibili. Il numero di endpoint VPC può variare per regione, in base al numero di zone di disponibilità e al numero di sottoreti nel VPC assegnato. La regione Stati Uniti orientali (Virginia settentrionale) ha cinque zone di disponibilità e puoi avere fino a cinque endpoint Amazon Keyspaces. La regione Stati Uniti occidentali (California settentrionale) ha due zone di disponibilità e puoi avere fino a due endpoint Amazon Keyspaces. Il numero di endpoint non influisce sulla scalabilità, ma aumenta il numero di connessioni da stabilire nella configurazione del driver. Analizza l'esempio seguente. La tua applicazione deve supportare 20.000 query CQL ed è in esecuzione su tre istanze su Amazon ECS, dove ogni istanza stabilisce una singola sessione su Amazon Keyspaces. L'unica differenza è il numero di endpoint disponibili nei diversi sistemi. Regioni AWS

Connessioni richieste nella regione Stati Uniti orientali (Virginia settentrionale):

```
20,000 CQL queries / (3 instances - 1 failure) / 5 private VPC endpoints / 500 CQL queries per second = 4 local connections
```

Connessioni richieste nella regione Stati Uniti occidentali (California settentrionale):

```
20,000 CQL queries / (3 instances - 1 failure) / 2 private VPC endpoints / 500 CQL queries per second = 10 local connections
```

### Important

Quando si utilizzano endpoint VPC privati, sono necessarie autorizzazioni aggiuntive per Amazon Keyspaces per rilevare dinamicamente gli endpoint VPC disponibili e popolare la tabella. `system.peers` Per ulteriori informazioni, consulta [the section called “Inserimento delle voci della `system.peers` tabella con informazioni sugli endpoint VPC dell'interfaccia”](#).

Quando si accede ad Amazon Keyspaces tramite un endpoint VPC privato utilizzando un altro Account AWS, è probabile che venga visualizzato un solo endpoint Amazon Keyspaces. Anche in questo caso ciò non influisce sulla portata del throughput possibile verso Amazon Keyspaces, ma potrebbe richiedere un aumento del numero di connessioni nella configurazione del driver. Questo esempio mostra lo stesso calcolo per un singolo endpoint disponibile.

$20,000 \text{ CQL queries} / (3 \text{ instances} - 1 \text{ failure}) / 1 \text{ private VPC endpoints} / 500 \text{ CQL queries per second} = 20 \text{ local connections}$

Per ulteriori informazioni sull'accesso tra account ad Amazon Keyspaces utilizzando un VPC condiviso, consulta [the section called “Configura l'accesso tra account in un VPC condiviso”](#)

## Come monitorare le connessioni in Amazon Keyspaces

Per identificare il numero di endpoint a cui è connessa l'applicazione, è possibile registrare il numero di peer rilevati nella `system.peers` tabella. L'esempio seguente è un esempio di codice Java che stampa il numero di peer dopo che la connessione è stata stabilita.

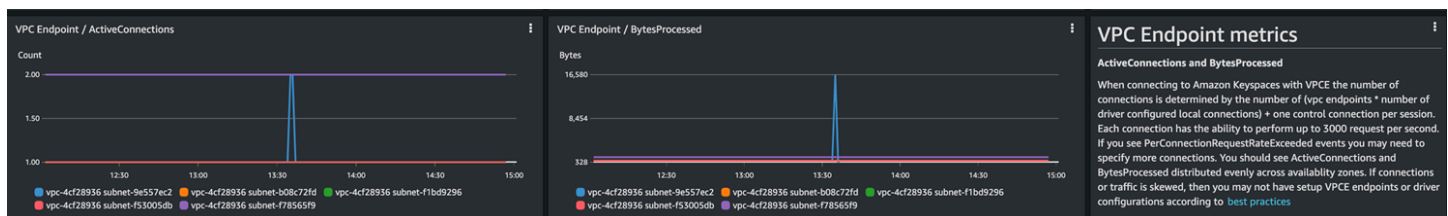
```
ResultSet result = session.execute(new SimpleStatement("SELECT * FROM system.peers"));

logger.info("number of Amazon Keyspaces endpoints:" + result.all().stream().count());
```

### Note

La console o AWS la console CQL non sono distribuite all'interno di un VPC e quindi utilizzano l'endpoint pubblico. Di conseguenza, l'esecuzione della `system.peers` query da applicazioni situate all'esterno del VPCE spesso dà come risultato 9 peer. Può anche essere utile stampare gli indirizzi IP di ciascun peer.

Puoi anche osservare il numero di peer quando usi un endpoint VPC configurando i parametri di Amazon VPCE. CloudWatch In CloudWatch, puoi vedere il numero di connessioni stabilite all'endpoint VPC. I driver Cassandra stabiliscono una connessione per ciascun endpoint per inviare query CQL e una connessione di controllo per raccogliere informazioni sulla tabella di sistema. L'immagine seguente mostra i CloudWatch parametri degli endpoint VPC dopo la connessione ad Amazon Keyspaces con 1 connessione configurata nelle impostazioni del driver. La metrica mostra sei connessioni attive costituite da una connessione di controllo e cinque connessioni (1 per endpoint nelle zone di disponibilità).



Per iniziare a monitorare il numero di connessioni utilizzando un CloudWatch grafico, puoi distribuire questo CloudFormation modello disponibile GitHub nell'archivio dei modelli di [Amazon Keyspaces](#).

## Come gestire gli errori di connessione in Amazon Keyspaces

Quando si supera la quota di 3.000 richieste per connessione, Amazon Keyspaces restituisce un `PerConnectionRequestExceeded` evento e il driver Cassandra riceve un'eccezione `or. WriteTimeout` `ReadTimeout`. Dovresti riprovare questa eccezione con un backoff esponenziale nella tua politica sui tentativi di riprova di Cassandra o nella tua applicazione. È necessario fornire un backoff esponenziale per evitare di inviare richieste aggiuntive.

La politica di riprova predefinita tenta di accedere al piano di `try next host` interrogazione. Poiché Amazon Keyspaces può avere da uno a tre endpoint disponibili durante la connessione all'endpoint VPC, è possibile che `NoHostAvailableException` nei log delle applicazioni vengano visualizzate anche le eccezioni aggiuntive `WriteTimeout` e `ReadTimeout` le eccezioni. Puoi utilizzare le politiche di riprova fornite da Amazon Keyspaces, che riprovano sullo stesso endpoint ma su connessioni diverse.

Puoi trovare esempi di politiche di ripetizione esponenziale per Java nel repository di esempi di codice GitHub Java di Amazon [Keyspaces](#). Puoi trovare esempi di linguaggio aggiuntivi su Github nel repository degli esempi di codice di Amazon [Keyspaces](#).

## Migliori pratiche di modellazione dei dati: consigli per la progettazione di modelli di dati

Una modellazione efficace dei dati è fondamentale per ottimizzare le prestazioni e ridurre al minimo i costi quando si lavora con Amazon Keyspaces (per Apache Cassandra). Questo argomento tratta considerazioni e raccomandazioni chiave per la progettazione di modelli di dati adatti ai modelli di accesso ai dati dell'applicazione.

- **Progettazione della chiave di partizione:** la chiave di partizione svolge un ruolo fondamentale nel determinare come i dati vengono distribuiti tra le partizioni in Amazon Keyspaces. La scelta di una chiave di partizione appropriata può influire in modo significativo sulle prestazioni delle query e sui costi di throughput. Questa sezione illustra le strategie per la progettazione di chiavi di partizione che promuovono una distribuzione uniforme delle attività di lettura e scrittura tra le partizioni.
- **Considerazioni chiave:**
  - **Distribuzione uniforme delle attività:** mira a un'attività di lettura e scrittura uniforme su tutte le partizioni per ridurre al minimo i costi di throughput e sfruttare efficacemente la capacità di burst.

- Schemi di accesso: allinea il design della chiave di partizione ai modelli di accesso ai dati principali dell'applicazione.
- Dimensioni delle partizioni: evitate di creare partizioni troppo grandi, poiché ciò può influire sulle prestazioni e aumentare i costi.

Per visualizzare e progettare modelli di dati più facilmente, puoi utilizzare [NoSQL Workbench](#).

## Argomenti

- [Come utilizzare le chiavi di partizione in modo efficace in Amazon Keyspaces](#)

# Come utilizzare le chiavi di partizione in modo efficace in Amazon Keyspaces

La chiave primaria che identifica in modo univoco ogni riga in una tabella Amazon Keyspaces può essere costituita da una o più colonne chiave di partizione, che determinano in quali partizioni sono archiviati i dati, e una o più colonne di clustering opzionali, che definiscono il modo in cui i dati vengono raggruppati e ordinati all'interno di una partizione.

Poiché la chiave di partizione stabilisce il numero di partizioni in cui sono archiviati i dati e il modo in cui i dati vengono distribuiti tra queste partizioni, il modo in cui è stata scelta la chiave di partizione può avere un impatto significativo sulle prestazioni delle query. In generale, è consigliabile progettare l'applicazione per un'attività uniforme su tutte le partizioni del disco.

La distribuzione uniforme delle attività di lettura e scrittura dell'applicazione su tutte le partizioni consente di ridurre al minimo i costi di throughput e ciò vale sia per le modalità on-demand che per quelle con capacità assegnata. read/write Ad esempio, se si utilizza la modalità di capacità fornita, è possibile determinare i modelli di accesso richiesti dall'applicazione e stimare le unità di capacità di lettura totale (RCU) e le unità di capacità di scrittura (WCU) richieste da ciascuna tabella. Amazon Keyspaces supporta i tuoi modelli di accesso utilizzando il throughput che hai fornito, purché il traffico verso una determinata partizione non superi 3.000 e 1.000. RCUs WCUs

Quando una partizione presenta un throughput di lettura o scrittura elevato e sostenuto, a seconda dei modelli di traffico Amazon Keyspaces può suddividere automaticamente la partizione in due nuove partizioni. Ogni nuova partizione contiene un sottoinsieme delle righe della partizione originale, che distribuisce il throughput in modo uniforme su entrambe le partizioni.

Amazon Keyspaces offre maggiore flessibilità nel provisioning del throughput per partizione fornendo capacità burst, per ulteriori informazioni, consulta. [the section called “Usa la capacità burst”](#)

## Argomenti

- [Usa lo sharding in scrittura per distribuire uniformemente i carichi di lavoro tra le partizioni](#)

## Usa lo sharding in scrittura per distribuire uniformemente i carichi di lavoro tra le partizioni

Un modo per distribuire meglio le scritture su una partizione in Amazon Keyspaces consiste nell'espandere lo spazio. Questa operazione può essere eseguita in modi diversi. Puoi aggiungere una colonna aggiuntiva di chiavi di partizione in cui scrivere numeri casuali per distribuire le righe tra le partizioni. In alternativa, puoi usare un numero calcolato in base al risultato di una query.

### Sharding utilizzando chiavi di partizione composte e valori casuali

Una strategia per distribuire i carichi in modo più uniforme su una partizione consiste nell'aggiungere un'ulteriore colonna chiave di partizione in cui scrivere numeri casuali. In seguito scegli in modo casuale le scritture nello spazio di dimensioni maggiori.

Ad esempio, si consideri la tabella seguente che contiene una singola chiave di partizione che rappresenta una data.

```
CREATE TABLE IF NOT EXISTS tracker.blogs (
 publish_date date,
 title text,
 description int,
 PRIMARY KEY (publish_date));
```

Per distribuire in modo più uniforme questa tabella tra le partizioni, è possibile includere una colonna di chiavi di partizione aggiuntiva shard che memorizza numeri casuali. Ad esempio:

```
CREATE TABLE IF NOT EXISTS tracker.blogs (
 publish_date date,
 shard int,
 title text,
 description int,
 PRIMARY KEY ((publish_date, shard)));
```

Quando inserisci i dati, puoi scegliere un numero casuale tra 1 e 200 per la colonna. `shard`  
In questo modo si ottengono valori di chiave di partizione composti come (2020-07-09, 1)(2020-07-09, 2), e così via. (2020-07-09, 200) In quanto stai scegliendo in modo casuale la chiave di partizione, le scritture alla tabella ogni giorno vengono distribuite in modo uniforme tra le partizioni multiple. Ciò risulta in parallelismo migliore e throughput complessivo maggiore.

Tuttavia, per leggere tutte le righe di un determinato giorno, è necessario interrogare le righe per tutti i frammenti e quindi unire i risultati. Ad esempio, è necessario innanzitutto emettere un'SELECTistruzione per il valore della chiave di partizione. (2020-07-09, 1) Quindi emettete un'altra SELECT dichiarazione per(2020-07-09, 2), e così via, attraverso(2020-07-09, 200). Infine, la tua applicazione dovrebbe unire i risultati di tutte quelle SELECT affermazioni.

Sharding utilizzando chiavi di partizione composte e valori calcolati

Una strategia di scelta casuale può migliorare molto il throughput di scrittura. Tuttavia, è difficile leggere una riga specifica perché non si sa quale valore è stato scritto `shard` nella colonna al momento della scrittura della riga. Per facilitare la lettura delle singole righe, puoi utilizzare una strategia diversa. Invece di usare un numero casuale per distribuire le righe tra le partizioni, usa un numero che puoi calcolare in base a qualcosa su cui vuoi eseguire una query.

Guarda l'esempio precedente, dove una tabella utilizza la giornata di oggi nella chiave di partizione. Supponiamo ora che ogni riga abbia una `title` colonna accessibile e che molto spesso sia necessario trovare le righe per titolo oltre che per data. Prima di scrivere la riga nella tabella, l'applicazione potrebbe calcolare un valore hash basato sul titolo e utilizzarlo per popolare la colonna. `shard` Il calcolo potrebbe dare come risultato un numero compreso tra 1 e 200 distribuito in modo abbastanza uniforme, in maniera simile a cosa viene prodotto dalla strategia casuale.

Probabilmente sarebbe sufficiente un semplice calcolo, ad esempio il prodotto dei valori dei punti di codice UTF-8 per i caratteri del titolo, modulo 200, + 1. Il valore della chiave di partizione composta sarebbe quindi la combinazione della data e del risultato del calcolo.

Con questa strategia, le scritture vengono distribuite in modo uniforme tra i valori delle chiavi di partizione e quindi tra le partizioni fisiche. È possibile eseguire facilmente un'SELECTistruzione per una riga e una data particolari perché è possibile calcolare il valore della chiave di partizione per un valore specifico `title`.

Per leggere tutte le righe di un determinato giorno, è comunque necessario utilizzare SELECT ciascuna (2020-07-09, N) chiave (dove N è compresa tra 1 e 200) e l'applicazione deve quindi unire tutti i risultati. Il beneficio è che eviterai che un singolo valore di chiave di partizione "hot" prenda tutto il carico di lavoro.

# Ottimizzazione dei costi delle tabelle Amazon Keyspaces

Questa sezione illustra le best practice su come ottimizzare i costi per le tabelle Amazon Keyspaces esistenti. Esaminare le seguenti strategie per vedere quale strategia di ottimizzazione dei costi si adatta meglio alle proprie esigenze e affrontarle in modo iterativo. Ogni strategia fornisce una panoramica di ciò che potrebbe influire sui costi, su come cercare opportunità per ottimizzare i costi e linee guida prescrittive su come implementare queste best practice per aiutarti a risparmiare.

## Argomenti

- [Valutazione dei costi a livello di tabella](#)
- [Valuta la modalità di capacità della tua tabella](#)
- [Valuta le impostazioni di Application Auto Scaling della tua tabella](#)
- [Identifica le risorse inutilizzate per ottimizzare i costi in Amazon Keyspaces](#)
- [Valuta i modelli di utilizzo delle tabelle per ottimizzare prestazioni e costi](#)
- [Valutare la capacità fornita per un provisioning di dimensioni adeguate](#)

## Valutazione dei costi a livello di tabella

Lo strumento Cost Explorer disponibile all'interno Console di gestione AWS di consente di visualizzare i costi suddivisi per tipo, ad esempio i costi di lettura, scrittura, archiviazione e backup. Puoi anche visualizzare questi costi riepilogati per periodo, ad esempio per mese o per giorno.

Un problema comune con Cost Explorer è che non è possibile rivedere facilmente i costi di una sola tabella particolare, perché Cost Explorer non consente di filtrare o raggruppare in base ai costi di una tabella specifica. Puoi visualizzare la metrica Billable table size (Byte) di ogni tabella nella console Amazon Keyspaces nella scheda Monitor della tabella. Se sono necessarie ulteriori informazioni relative ai costi per tabella, questa sezione mostra come utilizzare i [tag](#) per eseguire l'analisi dei costi delle singole tabelle in Cost Explorer.

## Argomenti

- [Come visualizzare i costi di una singola tabella Amazon Keyspaces](#)
- [Visualizzazione predefinita di Esploratore dei costi](#)
- [Come utilizzare e applicare i tag di tabella in Esploratore dei costi](#)

## Come visualizzare i costi di una singola tabella Amazon Keyspaces

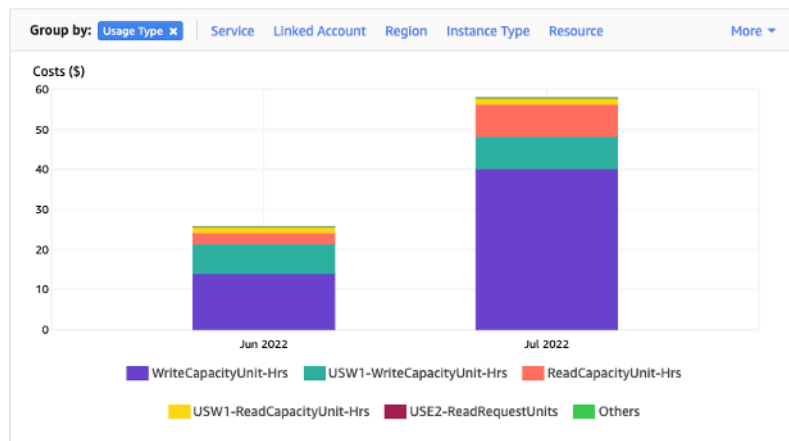
Puoi visualizzare le informazioni di base su una tabella Amazon Keyspaces nella console, tra cui lo schema della chiave primaria, la dimensione fatturabile della tabella e i parametri relativi alla capacità. Puoi utilizzare la dimensione della tabella per calcolare il costo di archiviazione mensile della tabella. Ad esempio, 0,25 USD per GB negli Stati Uniti orientali (Virginia settentrionale).

### Regione AWS

Se la tabella utilizza la modalità di capacità fornita, vengono restituite anche le impostazioni correnti dell'unità di capacità di lettura (RCU) e dell'unità di capacità di scrittura (WCU). È possibile utilizzare queste informazioni per calcolare i costi correnti di lettura e scrittura per la tabella. Tieni presente che questi costi potrebbero cambiare, soprattutto se hai configurato la tabella con la scalabilità automatica di Amazon Keyspaces.

## Visualizzazione predefinita di Esploratore dei costi

La visualizzazione predefinita in Cost Explorer fornisce grafici che mostrano il costo delle risorse consumate, ad esempio velocità effettiva e archiviazione. È possibile scegliere di raggruppare questi costi per periodo, ad esempio i totali per mese o per giorno. È inoltre possibile suddividere e confrontare i costi di archiviazione, lettura, scrittura e altre categorie.



## Come utilizzare e applicare i tag di tabella in Esploratore dei costi

Per impostazione predefinita, Cost Explorer non fornisce un riepilogo dei costi per nessuna tabella specifica, poiché combina i costi di più tabelle in un totale. Tuttavia, puoi utilizzare [l'assegnazione di tag alle risorse AWS](#) per identificare ogni tabella tramite un tag di metadati. I tag sono coppie chiave-valore che è possibile utilizzare per diversi scopi, ad esempio per identificare tutte le risorse appartenenti a un progetto o a un reparto. Per ulteriori informazioni, consulta [the section called "Lavorare con i tag"](#).



Per questo esempio, utilizziamo una tabella con il nome. MyTable

1. Imposta un tag con la chiave di `table_name` e il valore di. MyTable
2. [Attiva il tag in Esploratore dei costi](#), quindi filtra il valore del tag per ottenere maggiore visibilità sui costi di ogni tabella.

#### Note

Potrebbero essere necessari uno o due giorni prima che il tag inizi a comparire in Esploratore dei costi

Puoi impostare i tag dei metadati tu stesso nella console o a livello di codice con CQL, the o SDK. AWS CLI AWS Valuta la possibilità di richiedere l'impostazione di un tag `table_name` come parte del nuovo processo di creazione delle tabelle dell'organizzazione. Per ulteriori informazioni, consulta [the section called "Crea report sull'allocazione dei costi"](#).

## Valuta la modalità di capacità della tua tabella

Questa sezione fornisce una panoramica su come selezionare la modalità di capacità appropriata per la tabella Amazon Keyspaces. Ogni modalità è ottimizzata per soddisfare le esigenze di un carico di lavoro diverso in termini di capacità di risposta alle variazioni della velocità di trasmissione effettiva e di fatturazione dell'utilizzo. Per prendere una decisione è necessario bilanciare questi fattori.

### Argomenti

- [Modalità di capacità della tabella disponibili](#)
- [Quando selezionare la modalità di capacità on demand](#)
- [Quando selezionare la modalità di capacità assegnata](#)
- [Fattori aggiuntivi da valutare nella scelta di una modalità di capacità della tabella](#)

### Modalità di capacità della tabella disponibili

Quando crei una tabella Amazon Keyspaces, devi selezionare la modalità di capacità on demand o provisioning. Per ulteriori informazioni, consulta [the section called "Configura le modalità di capacità di lettura/scrittura"](#).

### Modalità di capacità on demand

La modalità di capacità su richiesta è progettata per eliminare la necessità di pianificare o fornire la capacità della tabella Amazon Keyspaces. In questa modalità, la tabella soddisfa istantaneamente le richieste senza la necessità di aumentare o ridurre le risorse (fino al doppio del precedente picco di throughput della tabella).

Le tabelle on demand vengono fatturate contando il numero di richieste effettive rispetto alla tabella, quindi paghi solo per ciò che utilizzi anziché per ciò che è stato fornito.

### Modalità di capacità assegnata

La modalità di capacità fornita è un modello più tradizionale in cui è possibile definire la capacità disponibile nella tabella per le richieste direttamente o con l'assistenza di Application Auto Scaling. Poiché per la tabella viene assegnata una capacità specifica in un dato momento, la fatturazione si basa sulla capacità assegnata anziché sul numero di richieste. Il superamento della capacità allocata può inoltre far sì che la tabella rifiuti le richieste e riduca l'esperienza degli utenti dell'applicazione.

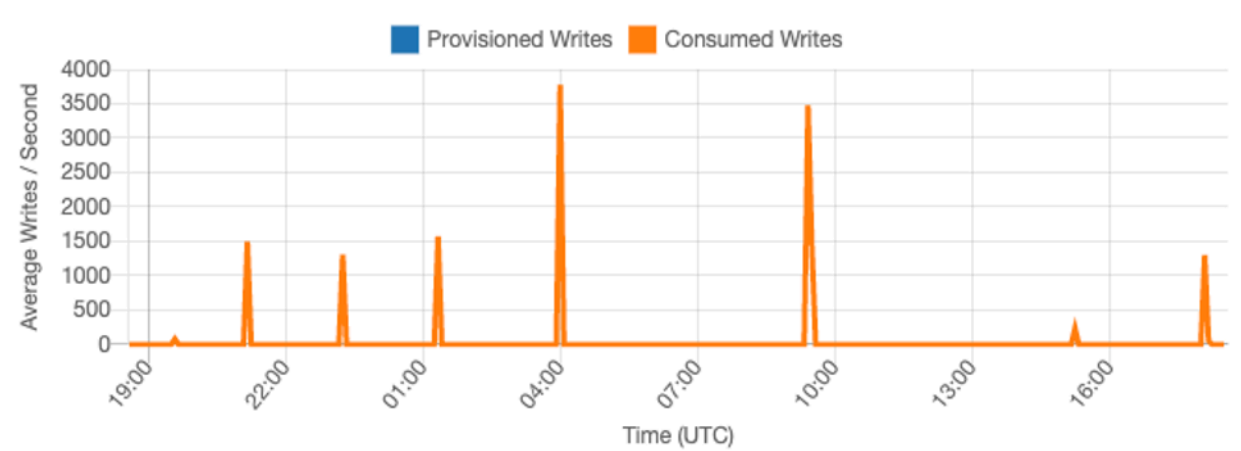
La modalità di capacità fornita richiede un equilibrio tra il non sovra-provisioning o l'insufficiente provisioning della tabella per ottenere entrambi i vantaggi, una bassa incidenza di errori di capacità di throughput insufficiente e l'ottimizzazione dei costi.

### Quando selezionare la modalità di capacità on demand

Per l'ottimizzazione dei costi, la modalità on demand è la scelta migliore quando si ha un carico di lavoro imprevedibile simile a quello mostrato nel grafico seguente.

Questi fattori contribuiscono a questo tipo di carico di lavoro:

- Tempi imprevedibili delle richieste (con conseguenti picchi di traffico)
- Volume variabile di richieste (derivante da carichi di lavoro in batch)
- scende a zero o al di sotto del 18% del picco per una determinata ora (derivante da ambienti di sviluppo o test)



Per i carichi di lavoro con le caratteristiche sopra riportate, l'utilizzo di Application Auto Scaling per mantenere una capacità sufficiente per consentire alla tabella di rispondere ai picchi di traffico può portare a risultati indesiderati. È possibile che la tabella abbia un numero eccessivo di risorse e costi più del necessario, oppure che il provisioning della tabella sia insufficiente e le richieste generino inutili errori di throughput a bassa capacità. In casi come questo, le tabelle su richiesta sono la scelta migliore.

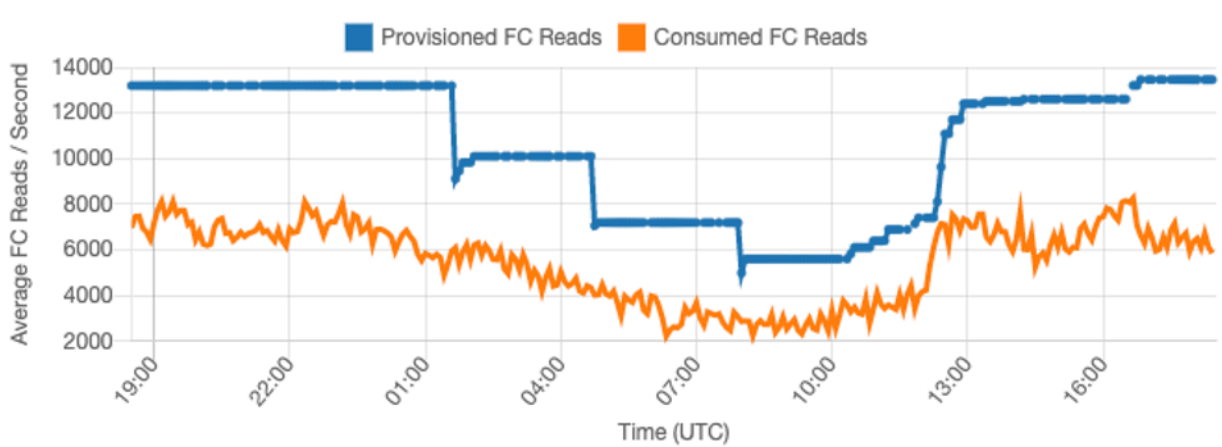
Poiché le tabelle su richiesta vengono fatturate su richiesta, non è necessario fare altro a livello di tabella per ottimizzare i costi. È consigliabile valutare regolarmente le tabelle su richiesta per verificare che il carico di lavoro abbia ancora le caratteristiche sopra indicate. Se il carico di lavoro si è stabilizzato, valuta la possibilità di passare alla modalità provisioning per mantenere l'ottimizzazione dei costi.

## Quando selezionare la modalità di capacità assegnata

Un carico di lavoro ideale per la modalità di provisioning capacity è quello con un modello di utilizzo più prevedibile, come illustrato nel grafico seguente.

I seguenti fattori contribuiscono a un carico di lavoro prevedibile:

- Traffico prevedibile/ciclico per una determinata ora o giorno
- Picchi di traffico limitati e di breve durata



Poiché i volumi di traffico in una determinata ora o giorno sono più stabili, è possibile impostare la capacità assegnata in modo relativamente simile alla capacità effettivamente consumata della tabella. L'ottimizzazione dei costi di una tabella della capacità assegnata consiste in ultima analisi nell'avvicinare il più possibile la capacità fornita (linea blu) alla capacità consumata (linea arancione) senza aumentare `ThrottledRequests` gli eventi della tabella. Lo spazio tra le due linee rappresenta sia uno spreco di capacità che un'assicurazione contro un'esperienza utente negativa dovuta a errori di capacità di throughput insufficiente.

Amazon Keyspaces fornisce Application Auto Scaling per le tabelle di capacità assegnate, che bilancia automaticamente questo valore per tuo conto. Puoi tenere traccia della capacità consumata durante il giorno e configurare la capacità fornita della tabella in base a una manciata di variabili.

### Unità di capacità minima

È possibile impostare la capacità minima di una tabella per limitare il verificarsi di errori di capacità di throughput insufficiente, ma ciò non riduce il costo della tabella. Se la tabella presenta periodi di utilizzo ridotto seguiti da un'improvvisa esplosione di utilizzo elevato, l'impostazione del minimo può impedire che Application Auto Scaling imposti la capacità della tabella su un valore troppo basso.

### Unità di capacità massima

È possibile impostare la capacità massima di una tabella per limitare un dimensionamento della tabella maggiore del previsto. Valuta la possibilità di applicare un valore massimo alle tabelle di sviluppo o di test, dove non è consigliabile eseguire test di carico su larga scala. È possibile impostare un valore massimo per qualsiasi tabella, ma assicuratevi di valutare regolarmente questa impostazione rispetto alla tabella di base quando la utilizzate in produzione, per evitare errori accidentali di capacità di throughput insufficiente.

### Utilizzo di destinazione

L'impostazione dell'utilizzo di destinazione della tabella è il mezzo principale per l'ottimizzazione dei costi per una tabella con capacità assegnata. Se si imposta qui un valore percentuale inferiore, si aumenta l'entità del provisioning eccessivo della tabella, aumentando i costi, ma riducendo il rischio di errori di capacità di throughput insufficiente. L'impostazione di un valore percentuale più elevato consente di ridurre il livello di sovra-provisioning della tabella, ma aumenta il rischio di errori di capacità di throughput insufficiente.

## Fattori aggiuntivi da valutare nella scelta di una modalità di capacità della tabella

Al momento di decidere tra le due modalità di capacità, vi sono alcuni fattori aggiuntivi che vale la pena considerare.

Quando decidi tra le due modalità di tavolo, considera quanto questo sconto aggiuntivo influisca sul costo del tavolo. In molti casi, anche un carico di lavoro relativamente imprevedibile può essere più conveniente se eseguito su una tabella di capacità assegnata in eccesso con capacità riservata.

### Miglioramento della prevedibilità del carico di lavoro

In alcune situazioni, un carico di lavoro può apparentemente avere sia uno schema prevedibile che uno imprevedibile. Sebbene ciò possa essere facilmente supportato con una tabella su richiesta, i costi sarebbero probabilmente inferiori se si riuscissero a migliorare gli schemi imprevedibili del carico di lavoro.

Una delle cause più comuni di questi modelli sono le importazioni in batch. Questo tipo di traffico può spesso superare la capacità di base della tabella a tal punto che si verificherebbero errori di capacità di trasmissione insufficiente se la stessa venisse eseguita. Per mantenere un carico di lavoro come questo in esecuzione su una tabella con capacità assegnata, valuta le seguenti opzioni:

- Se il batch viene eseguito in orari pianificati, è possibile pianificare un aumento della capacità di auto scalabilità dell'applicazione prima dell'esecuzione.
- Se il batch viene eseguito in modo casuale, è consigliabile provare a prolungare il tempo necessario per l'esecuzione anziché eseguirlo il più velocemente possibile.
- Aggiungete un periodo di accelerazione all'importazione, in cui la velocità di importazione inizia in modo ridotto ma aumenta lentamente nell'arco di alcuni minuti fino a quando Application Auto Scaling non ha avuto l'opportunità di iniziare a regolare la capacità della tabella.

## Valuta le impostazioni di Application Auto Scaling della tua tabella

Questa sezione fornisce una panoramica su come valutare le impostazioni Application Auto Scaling delle tabelle Amazon Keyspaces. [Amazon Keyspaces Application Auto Scaling](#) è una funzionalità che gestisce il throughput delle tabelle in base al traffico dell'applicazione e alla metrica di utilizzo prevista. Ciò garantisce che le tabelle abbiano la capacità richiesta per i modelli applicativi.

Il servizio Application Auto Scaling monitora l'utilizzo corrente della tabella e lo confronta con il valore di utilizzo target. TargetValue Ti avvisa se è il momento di aumentare o diminuire la capacità allocata.

### Argomenti

- [Informazioni sulle impostazioni dell'Application Auto Scaling](#)
- [Come identificare le tabelle con un basso utilizzo di destinazione \(<=50%\)](#)
- [Come gestire i carichi di lavoro con varianza stagionale](#)
- [Come affrontare carichi di lavoro con picchi di lavoro con pattern sconosciuti](#)
- [Come gestire i carichi di lavoro con applicazioni collegate](#)

### Informazioni sulle impostazioni dell'Application Auto Scaling

La definizione del valore corretto per l'utilizzo di destinazione, la fase iniziale e i valori finali è un'attività che richiede il coinvolgimento del team operativo. Ciò consente di definire correttamente i valori in base all'utilizzo storico dell'applicazione, utilizzato per attivare le politiche di Application Auto Scaling. L'obiettivo di utilizzo è la percentuale della capacità totale che deve essere soddisfatta durante un periodo di tempo prima che si applichino le regole dell'Application Auto Scaling.

Quando imposti un obiettivo di utilizzo elevato (un obiettivo intorno al 90%), significa che il traffico deve essere superiore al 90% per un periodo di tempo prima che l'Application Auto Scaling venga attivato. Non dovresti utilizzare un target di utilizzo elevato a meno che l'applicazione non sia molto costante e non riceva picchi di traffico.

Quando si imposta un utilizzo molto basso (un obiettivo inferiore al 50%), significa che l'applicazione deve raggiungere il 50% della capacità fornita prima di attivare una policy di Application Auto Scaling. A meno che il traffico delle applicazioni non cresca a un ritmo molto aggressivo, questo di solito si traduce in capacità inutilizzata e risorse sprecate.

## Come identificare le tabelle con un basso utilizzo di destinazione ( $\leq 50\%$ )

Puoi utilizzare AWS CLI o Console di gestione AWS per monitorare e identificare le TargetValues politiche di Application Auto Scaling nelle tue risorse Amazon Keyspaces.

### Note

Quando utilizzi tabelle multiregionali in modalità di capacità fornita con scalabilità automatica di Amazon Keyspaces, assicurati di utilizzare le operazioni dell'API Amazon Keyspaces per configurare la scalabilità automatica. Le operazioni API Application Auto Scaling sottostanti che Amazon Keyspaces chiama per tuo conto non hanno funzionalità multiregionali. Per ulteriori informazioni, consulta [the section called “Visualizza la capacità assegnata e le impostazioni di ridimensionamento automatico per una tabella multiregionale”](#).

## AWS CLI

1. Restituisce l'intero elenco di risorse eseguendo il seguente comando:

```
aws application-autoscaling describe-scaling-policies --service-namespace
cassandra
```

Questo comando restituirà l'elenco completo delle politiche di Application Auto Scaling emesse per qualsiasi risorsa Amazon Keyspaces. Se desideri recuperare solo le risorse da una particolare tabella, puoi aggiungere `--resource-id` parameter. Ad esempio:

```
aws application-autoscaling describe-scaling-policies --service-namespace
cassandra --resource-id "keyspace/keyspace-name/table/table-name"
```

2. Restituisce solo le politiche di ridimensionamento automatico per una particolare tabella eseguendo il seguente comando

```
aws application-autoscaling describe-scaling-policies --service-namespace
cassandra --resource-id "keyspace/keyspace-name/table/table-name"
```

I valori per le policy di Application Auto Scaling sono evidenziati di seguito. È necessario assicurarsi che il valore target sia superiore al 50% per evitare un eccesso di provisioning. Viene visualizzato un risultato simile al seguente:

```

{
 "ScalingPolicies": [
 {
 "PolicyARN": "arn:aws:autoscaling:us-
east-1:111122223333:scalingPolicy:<uuid>:resource/keyspaces/table/table-name-
scaling-policy",
 "PolicyName": "$<full-table-name>",
 "ServiceNamespace": "cassandra",
 "ResourceId": "keyspace/keyspace-name/table/table-name",
 "ScalableDimension": "cassandra:table:WriteCapacityUnits",
 "PolicyType": "TargetTrackingScaling",
 "TargetTrackingScalingPolicyConfiguration": {
 "TargetValue": 70.0,
 "PredefinedMetricSpecification": {
 "PredefinedMetricType": "KeyspacesWriteCapacityUtilization"
 }
 },
 "Alarms": [
 ...
],
 "CreationTime": "2022-03-04T16:23:48.641000+10:00"
 },
 {
 "PolicyARN": "arn:aws:autoscaling:us-
east-1:111122223333:scalingPolicy:<uuid>:resource/keyspaces/table/table-name-
scaling-policy",
 "PolicyName": "$<full-table-name>",
 "ServiceNamespace": "cassandra",
 "ResourceId": "keyspace/keyspace-name/table/table-name",
 "ScalableDimension": "cassandra:table:ReadCapacityUnits",
 "PolicyType": "TargetTrackingScaling",
 "TargetTrackingScalingPolicyConfiguration": {
 "TargetValue": 70.0,
 "PredefinedMetricSpecification": {
 "PredefinedMetricType": "KeyspacesReadCapacityUtilization"
 }
 },
 "Alarms": [
 ...
],
 "CreationTime": "2022-03-04T16:23:47.820000+10:00"
 }
]
}

```



```
}
```

## Console di gestione AWS

1. Accedi a Console di gestione AWS e vai alla pagina del CloudWatch servizio in [Getting Started with](#) the. Console di gestione AWS Seleziona quello appropriato, Regione AWS se necessario.
2. Nel riquadro di navigazione sinistro selezionare Tables (Tabelle). Nella pagina Tables (Tabelle), selezionare il nome della tabella.
3. Nella pagina Dettagli tabella della scheda Capacità, rivedi le impostazioni Application Auto Scaling della tabella.

Se i valori di utilizzo di destinazione sono inferiori o uguali al 50%, è consigliabile esaminare le metriche di utilizzo della tabella per vedere se il [provisioning è insufficiente o eccessivo](#).

## Come gestire i carichi di lavoro con varianza stagionale

Si consideri il seguente scenario: l'applicazione funziona con un valore medio minimo per la maggior parte del tempo, ma il target di utilizzo è basso, quindi l'applicazione può reagire rapidamente agli eventi che si verificano in determinate ore del giorno e la capacità è sufficiente ed evitare limitazioni della larghezza di banda della rete. Questo scenario è comune quando un'applicazione è molto impegnata durante il normale orario di ufficio (dalle 9:00 alle 17:00) ma funziona a un livello base nelle altre ore. Poiché alcuni utenti iniziano a connettersi prima delle 9:00, l'applicazione utilizza questa soglia bassa per aumentare rapidamente e raggiungere la capacità richiesta nelle ore di punta.

Lo scenario potrebbe essere simile al seguente:

- Tra le 17:00 e le 9:00, le unità ConsumedWriteCapacityUnits sono comprese tra 90 e 100
- Gli utenti iniziano a connettersi all'applicazione prima delle 9:00 e le unità di capacità aumentano considerevolmente (il valore massimo che rilevato è 1500 WCU)
- In media, l'utilizzo delle applicazioni varia tra 800 e 1.200 durante l'orario di ufficio

Se lo scenario precedente si applica alla tua applicazione, prendi in considerazione l'utilizzo della [scalabilità automatica dell'applicazione pianificata](#), in cui nella tabella potrebbe ancora essere

configurata una regola Application Auto Scaling, ma con un utilizzo del target meno aggressivo che fornisca la capacità aggiuntiva solo agli intervalli specifici richiesti.

È possibile utilizzare il AWS CLI per eseguire i seguenti passaggi per creare una regola di ridimensionamento automatico pianificata che viene eseguita in base all'ora del giorno e al giorno della settimana.

1. Registra la tua tabella Amazon Keyspaces come destinazione scalabile con. Application Auto Scaling Un target scalabile è una risorsa che Application Auto Scaling può aumentare ridurre orizzontalmente.

```
aws application-autoscaling register-scalable-target \
 --service-namespace cassandra \
 --scalable-dimension cassandra:table:WriteCapacityUnits \
 --resource-id keyspace/keyspace-name/table/table-name \
 --min-capacity 90 \
 --max-capacity 1500
```

2. Impostazione delle operazioni pianificate in base ai requisiti.

Sono necessarie due regole per coprire lo scenario: una per la scalabilità verso l'alto e l'altra per la scalabilità verso il basso. La prima regola per aumentare l'azione pianificata è mostrata nell'esempio seguente.

```
aws application-autoscaling put-scheduled-action \
 --service-namespace cassandra \
 --scalable-dimension cassandra:table:WriteCapacityUnits \
 --resource-id keyspace/keyspace-name/table/table-name \
 --scheduled-action-name my-8-5-scheduled-action \
 --scalable-target-action MinCapacity=800,MaxCapacity=1500 \
 --schedule "cron(45 8 ? * MON-FRI *)" \
 --timezone "Australia/Brisbane"
```

In questo esempio viene mostrata la seconda regola per ridurre l'azione pianificata.

```
aws application-autoscaling put-scheduled-action \
 --service-namespace cassandra \
 --scalable-dimension cassandra:table:WriteCapacityUnits \
 --resource-id keyspace/keyspace-name/table/table-name \
 --scheduled-action-name my-5-8-scheduled-down-action \
 --scalable-target-action MinCapacity=90,MaxCapacity=1500 \
```

```
--schedule "cron(15 17 ? * MON-FRI *)" \
--timezone "Australia/Brisbane"
```

3. Esegui il comando seguente per convalidare che entrambe le regole siano state attivate:

```
aws application-autoscaling describe-scheduled-actions --service-namespace
cassandra
```

Si dovrebbe ottenere un risultato simile a questo:

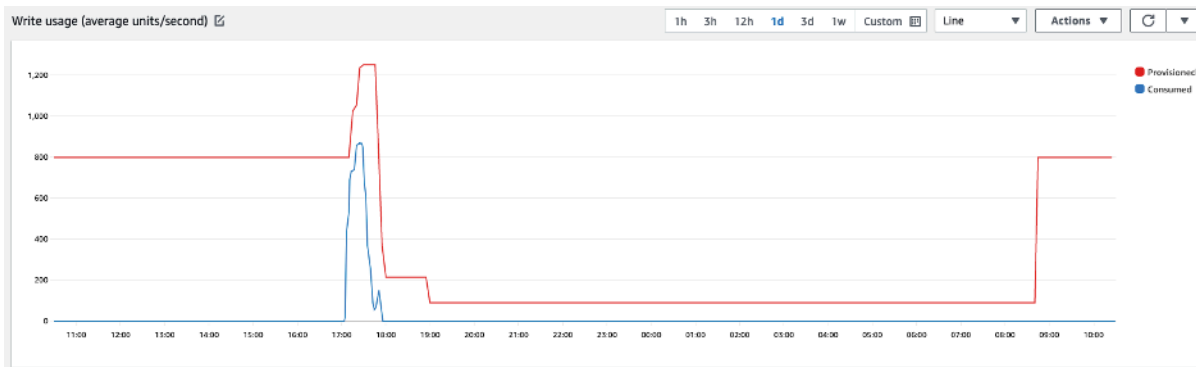
```
{
 "ScheduledActions": [
 {
 "ScheduledActionName": "my-5-8-scheduled-down-action",
 "ScheduledActionARN": "arn:aws:autoscaling:us-
east-1:111122223333:scheduledAction:<uuid>:resource/keyspaces/table/table-
name:scheduledActionName/my-5-8-scheduled-down-action",
 "ServiceNamespace": "cassandra",
 "Schedule": "cron(15 17 ? * MON-FRI *)",
 "Timezone": "Australia/Brisbane",
 "ResourceId": "keyspace/keyspace-name/table/table-name",
 "ScalableDimension": "cassandra:table:WriteCapacityUnits",
 "ScalableTargetAction": {
 "MinCapacity": 90,
 "MaxCapacity": 1500
 },
 "CreationTime": "2022-03-15T17:30:25.100000+10:00"
 },
 {
 "ScheduledActionName": "my-8-5-scheduled-action",
 "ScheduledActionARN": "arn:aws:autoscaling:us-
east-1:111122223333:scheduledAction:<uuid>:resource/keyspaces/table/table-
name:scheduledActionName/my-8-5-scheduled-action",
 "ServiceNamespace": "cassandra",
 "Schedule": "cron(45 8 ? * MON-FRI *)",
 "Timezone": "Australia/Brisbane",
 "ResourceId": "keyspace/keyspace-name/table/table-name",
 "ScalableDimension": "cassandra:table:WriteCapacityUnits",
 "ScalableTargetAction": {
 "MinCapacity": 800,
 "MaxCapacity": 1500
 },
 "CreationTime": "2022-03-15T17:28:57.816000+10:00"
 }
]
}
```

```

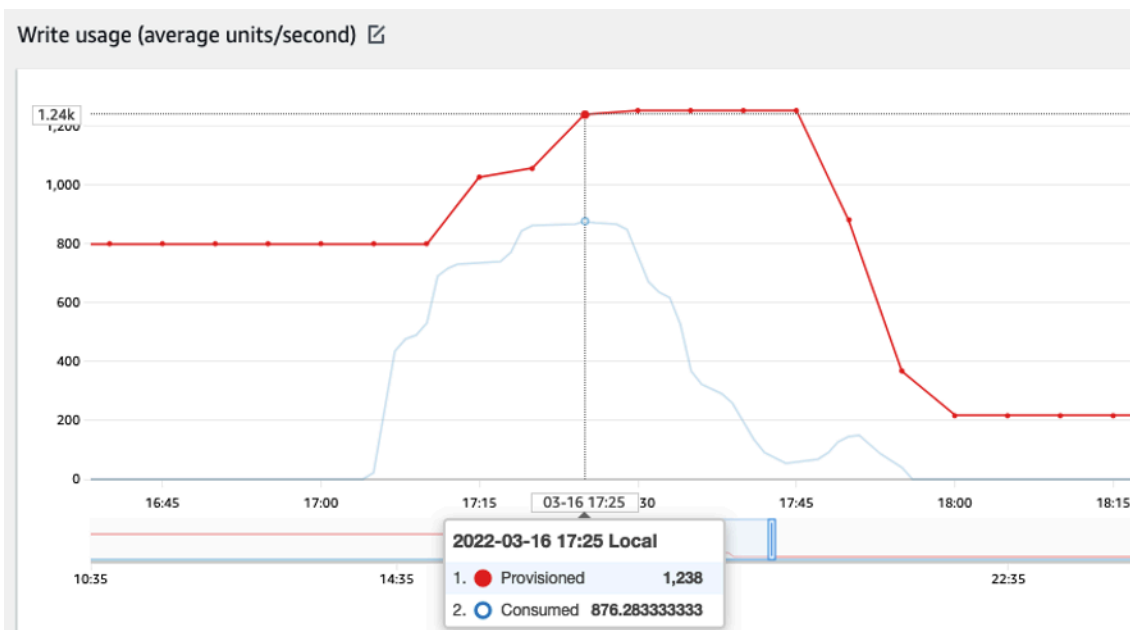
 }
]
}

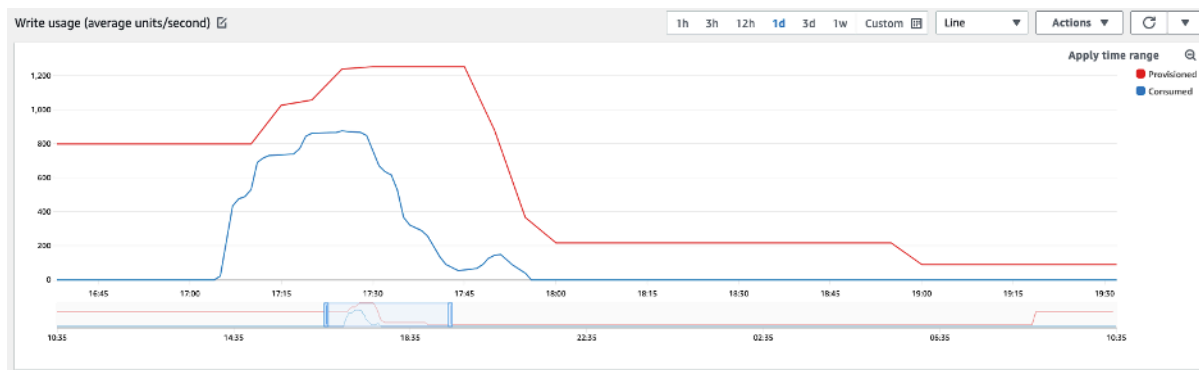
```

L'immagine seguente mostra un carico di lavoro di esempio che mantiene sempre il 70% di utilizzo di destinazione. Notate come le regole di autoscaling siano ancora valide e la velocità effettiva non venga ridotta.



Ingrandendo, possiamo notare che c'è stato un picco nell'applicazione che ha attivato la soglia di dimensionamento automatico del 70%, forzandolo ad attivarsi e a fornire la capacità aggiuntiva richiesta per la tabella. L'azione di ridimensionamento automatico pianificata influirà sui valori massimi e minimi ed è tua responsabilità configurarli.





## Come affrontare carichi di lavoro con picchi di lavoro con pattern sconosciuti

In questo scenario, l'applicazione utilizza un obiettivo di utilizzo molto basso, perché non si conoscono ancora i modelli applicativi e si desidera assicurarsi che il carico di lavoro non subisca errori di velocità effettiva a bassa capacità.

Si consiglia invece di utilizzare la [modalità di capacità on demand](#). Le tabelle on demand sono perfette per carichi di lavoro con picchi di lavoro di cui non si conoscono i pattern di traffico. Con la modalità di capacità on demand, si paga in base alla richiesta per le letture e le scritture dei dati che l'applicazione esegue sulle tabelle. Non è necessario specificare la velocità di lettura e scrittura prevista per l'applicazione, poiché Amazon Keyspaces si adatta istantaneamente ai carichi di lavoro man mano che aumentano o diminuiscono.

## Come gestire i carichi di lavoro con applicazioni collegate

In questo scenario, l'applicazione dipende da altri sistemi, ad esempio scenari di elaborazione in batch in cui è possibile avere grandi picchi di traffico in base agli eventi nella logica dell'applicazione.

Prendi in considerazione lo sviluppo di una logica di auto-scaling delle applicazioni personalizzata che reagisca a quegli eventi in cui puoi aumentare la capacità delle tabelle TargetValues e in base alle tue esigenze specifiche. Potresti trarre vantaggio Amazon EventBridge e utilizzare una combinazione di AWS servizi come  $\lambda$  e Step Functions per rispondere alle tue esigenze applicative specifiche.

## Identifica le risorse inutilizzate per ottimizzare i costi in Amazon Keyspaces

In questa sezione viene fornita una panoramica su come valutare regolarmente le risorse inutilizzate. Man mano che i requisiti delle applicazioni evolvono, devi assicurarti che nessuna risorsa venga inutilizzata e contribuisca a costi non necessari di Amazon Keyspaces. Le procedure descritte di

seguito utilizzano i CloudWatch parametri di Amazon per identificare le risorse inutilizzate e agire per ridurre i costi.

Puoi monitorare Amazon Keyspaces utilizzando CloudWatch, che raccoglie ed elabora dati grezzi da Amazon Keyspaces in metriche leggibili quasi in tempo reale. Queste statistiche vengono conservate per un determinato periodo di tempo, così da consentire l'accesso a informazioni cronologiche per comprendere meglio l'utilizzo. Per impostazione predefinita, i dati metrici di Amazon Keyspaces vengono inviati automaticamente a CloudWatch. Per ulteriori informazioni, consulta [What is Amazon CloudWatch?](#) e [conservazione dei parametri](#) nella Amazon CloudWatch User Guide.

## Argomenti

- [Come identificare le risorse inutilizzate](#)
- [Identificazione delle risorse non utilizzate della tabella](#)
- [Pulizia delle risorse non utilizzate della tabella](#)
- [Pulizia dei backup di point-in-time ripristino inutilizzati \(PITR\)](#)

## Come identificare le risorse inutilizzate

Per identificare le tabelle inutilizzate, puoi dare un'occhiata alle seguenti CloudWatch metriche per un periodo di 30 giorni per capire se ci sono letture o scritture attive su una tabella specifica:

### **ConsumedReadCapacityUnits**

Il numero di unità di capacità di lettura utilizzate nel periodo di tempo specificato, per permettere di tenere traccia di quanta capacità viene utilizzata. Puoi recuperare la capacità di lettura totale consumata per una tabella.

### **ConsumedWriteCapacityUnits**

Il numero di unità di capacità di scrittura utilizzate nel periodo di tempo specificato, per permettere di tenere traccia di quanta capacità viene utilizzata. È possibile recuperare la capacità di scrittura totale consumata per una tabella.

## Identificazione delle risorse non utilizzate della tabella

Amazon CloudWatch è un servizio di monitoraggio e osservabilità che fornisce i parametri della tabella Amazon Keyspaces che puoi utilizzare per identificare le risorse non utilizzate. CloudWatch

le metriche possono essere visualizzate sia tramite Console di gestione AWS AWS Command Line Interface

## AWS Command Line Interface

Per visualizzare le metriche delle tabelle tramite AWS Command Line Interface, puoi utilizzare i seguenti comandi.

1. Per prima cosa, valuta le letture della tabella:

### Note

Se il nome della tabella non è univoco all'interno del tuo account, devi specificare anche il nome del keyspace.

```
aws cloudwatch get-metric-statistics --metric-name
ConsumedReadCapacityUnits --start-time <start-time> --end-time <end-
time> --period <period> --namespace AWS/Cassandra --statistics Sum --
dimensions Name=TableName,Value=<table-name>
```

Per evitare di identificare erroneamente una tabella come non utilizzata, valutare i parametri per un periodo più lungo. Scegliete un intervallo di inizio e fine appropriato, ad esempio 30 giorni, e un periodo appropriato, ad esempio 86400.

Nei dati restituiti, qualsiasi somma superiore a 0 indica che la tabella che stai valutando ha ricevuto traffico di lettura durante quel periodo.

Il risultato seguente mostra una tabella che riceve traffico di lettura nel periodo valutato:

```
{
 "Timestamp": "2022-08-25T19:40:00Z",
 "Sum": 36023355.0,
 "Unit": "Count"
},
{
 "Timestamp": "2022-08-12T19:40:00Z",
 "Sum": 38025777.5,
 "Unit": "Count"
},
```

Il risultato seguente mostra una tabella che non riceve traffico di lettura nel periodo valutato:

```
{
 "Timestamp": "2022-08-01T19:50:00Z",
 "Sum": 0.0,
 "Unit": "Count"
},
{
 "Timestamp": "2022-08-20T19:50:00Z",
 "Sum": 0.0,
 "Unit": "Count"
},
```

2. Quindi, valuta le scritture della tabella:

```
aws cloudwatch get-metric-statistics --metric-name
ConsumedWriteCapacityUnits --start-time <start-time> --end-time <end-
time> --period <period> --namespace AWS/Cassandra --statistics Sum --
dimensions Name=TableName,Value=<table-name>
```

Per evitare di identificare erroneamente una tabella come non utilizzata, consigliamo di valutare i parametri per un periodo più lungo. Scegli un intervallo appropriato di inizio e fine, ad esempio 30 giorni, e un periodo appropriato, ad esempio 86400.

Nei dati restituiti, qualsiasi somma superiore a 0 indica che la tabella che stai valutando ha ricevuto traffico di scrittura durante quel periodo.

Il risultato seguente mostra una tabella che riceve traffico di scrittura nel periodo valutato:

```
{
 "Timestamp": "2022-08-19T20:15:00Z",
 "Sum": 41014457.0,
 "Unit": "Count"
},
{
 "Timestamp": "2022-08-18T20:15:00Z",
 "Sum": 40048531.0,
 "Unit": "Count"
},
```

Il risultato seguente mostra una tabella che non riceve traffico di scrittura nel periodo valutato:



```
{
 "Timestamp": "2022-07-31T20:15:00Z",
 "Sum": 0.0,
 "Unit": "Count"
},
{
 "Timestamp": "2022-08-19T20:15:00Z",
 "Sum": 0.0,
 "Unit": "Count"
},
```

## Console di gestione AWS

I passaggi seguenti consentono di valutare l'utilizzo delle risorse tramite Console di gestione AWS

1. Accedi Console di gestione AWS e vai alla pagina del CloudWatch servizio all'indirizzo <https://console.aws.amazon.com/cloudwatch/>. Se necessario, seleziona l'opzione appropriata Regione AWS in alto a destra della console.
2. Nella barra di navigazione a sinistra, individua la sezione Metriche e scegli Tutte le metriche.
3. L'azione sopra riportata apre una dashboard con due pannelli. Nel pannello superiore, puoi vedere le metriche attualmente rappresentate graficamente. In basso puoi selezionare le metriche disponibili per il grafico. Scegli Amazon Keyspaces nel pannello inferiore.
4. Nel pannello di selezione delle metriche di Amazon Keyspaces, scegli la categoria Table Metrics per mostrare le metriche per le tue tabelle nella regione corrente.
5. Identifica il nome della tabella scorrendo il menu verso il basso, quindi scegli le metriche ConsumedReadCapacityUnits e per la tabella. ConsumedWriteCapacityUnits
6. Scegli la scheda Metriche grafiche (2) e imposta la colonna Statistica su Somma.
7. Per evitare di identificare erroneamente una tabella come inutilizzata, valuta le metriche della tabella per un periodo più lungo. Nella parte superiore del pannello grafico, scegliete un periodo di tempo appropriato, ad esempio 1 mese, per valutare la tabella. Scegli Personalizzato, scegli 1 mese nel menu a discesa e scegli Applica.
8. Valuta i parametri nel grafico della tabella per determinare se viene utilizzata. Parametri superiori a 0 indicano che durante il periodo di tempo preso in considerazione la tabella è stata utilizzata. Un grafico piatto con 0 sia in lettura che in scrittura indica che una tabella non è utilizzata.

## Pulizia delle risorse non utilizzate della tabella

Se sono state identificate risorse non utilizzate della tabella, è possibile ridurne i costi correnti nei seguenti modi.

### Note

Se hai identificato una tabella non utilizzata ma desideri comunque mantenerla disponibile nel caso in cui sia necessario accedervi in futuro, valuta la possibilità di passare alla modalità on demand. Altrimenti, puoi prendere in considerazione l'eliminazione della tabella.

### Modalità di capacità

Amazon Keyspaces addebita i costi per la lettura, la scrittura e l'archiviazione dei dati nelle tabelle Amazon Keyspaces.

Amazon Keyspaces offre [due modalità di capacità](#), che includono opzioni di fatturazione specifiche per l'elaborazione di letture e scritture sulle tabelle: su richiesta e con provisioning. La modalità read/write di capacità controlla il modo in cui ti viene addebitato il throughput di lettura e scrittura e come gestisci la capacità.

Per le tabelle in modalità on demand, non è necessario specificare la velocità effettiva di lettura e scrittura che si prevede l'applicazione esegua. Amazon Keyspaces ti addebita le spese di lettura e scrittura eseguite dall'applicazione sulle tue tabelle in termini di unità di richiesta di lettura e unità di richiesta di scrittura. Se non c'è alcuna attività sul tavolo, non pagherai per la velocità effettiva, ma dovrai comunque sostenere un costo di archiviazione.

### Eliminazione delle tabelle

Se hai scoperto una tabella inutilizzata e desideri eliminarla, valuta la possibilità di eseguire prima un backup o esportare i dati.

I backup eseguiti AWS Backup possono sfruttare lo storage a freddo su più livelli, riducendo ulteriormente i costi. Consulta la documentazione sulla [gestione dei piani di backup](#) per informazioni su come utilizzare un ciclo di vita per spostare il backup nella conservazione a freddo.

Dopo aver eseguito il backup della tabella, puoi scegliere di eliminarla tramite la Console di gestione AWS o tramite l'AWS Command Line Interface.

## Pulizia dei backup di point-in-time ripristino inutilizzati (PITR)

Amazon Keyspaces offre Point-in-time il ripristino, che fornisce backup continui per 35 giorni per aiutarti a proteggerti da scritture o eliminazioni accidentali. I backup PITR hanno dei costi associati.

Consulta la documentazione all'indirizzo [the section called “Backup e ripristino con point-in-time ripristino”](#) per determinare se nelle tabelle sono abilitati backup che potrebbero non essere più necessari.

## Valuta i modelli di utilizzo delle tabelle per ottimizzare prestazioni e costi

Questa sezione fornisce una panoramica su come valutare se stai utilizzando in modo efficiente le tabelle Amazon Keyspaces. Esistono alcuni modelli di utilizzo che non sono ottimali per Amazon Keyspaces e consentono l'ottimizzazione sia dal punto di vista delle prestazioni che dei costi.

### Argomenti

- [Riduzione del numero di operazioni ad elevata consistenza di lettura](#)
- [Abilitazione del Time to Live \(TTL\)](#)

### Riduzione del numero di operazioni ad elevata consistenza di lettura

Amazon Keyspaces ti consente di configurare la [coerenza di lettura](#) in base alla richiesta. Le richieste di lettura sono a consistenza finale per impostazione predefinita. Alla fine, le letture coerenti vengono addebitate a 0,5 RCU per un massimo di 4 KB di dati.

La maggior parte dei carichi di lavoro distribuiti sono flessibili e possono tollerare la consistenza finale. Tuttavia, possono esserci schemi di accesso che richiedono elevata consistenza di lettura. Le letture estremamente coerenti vengono addebitate a 1 RCU per un massimo di 4 KB di dati, raddoppiando sostanzialmente i costi di lettura. Amazon Keyspaces ti offre la flessibilità necessaria per utilizzare entrambi i modelli di coerenza sulla stessa tabella.

È possibile valutare il carico di lavoro e il codice dell'applicazione per verificare se vengono utilizzate letture ad elevata consistenza solo dove necessario.

### Abilitazione del Time to Live (TTL)

[Time to Live \(TTL\)](#) ti aiuta a semplificare la logica delle applicazioni e a ottimizzare il prezzo dello storage facendo scadere automaticamente i dati dalle tabelle. I dati che non ti servono più vengono eliminati automaticamente dalla tabella in base al valore Time to Live che hai impostato.

## Valutare la capacità fornita per un provisioning di dimensioni adeguate

Questa sezione fornisce una panoramica su come valutare se il provisioning è della giusta dimensione sulle tabelle Amazon Keyspaces. Man mano che il carico di lavoro si evolve, è necessario modificare le procedure operative in modo appropriato, soprattutto quando la tabella Amazon Keyspaces è configurata in modalità provisioning e si corre il rischio di un provisioning eccessivo o insufficiente delle tabelle.

Le procedure descritte in questa sezione richiedono informazioni statistiche che devono essere acquisite dalle tabelle Amazon Keyspaces che supportano l'applicazione di produzione. Per comprendere il comportamento dell'applicazione, è necessario definire un periodo di tempo sufficientemente significativo da acquisire la stagionalità dei dati dell'applicazione. Ad esempio, se l'applicazione mostra pattern settimanali, l'utilizzo di un periodo di tre settimane dovrebbe fornire spazio sufficiente per analizzare le esigenze di velocità di trasmissione effettiva dell'applicazione.

Se non sai da dove iniziare, utilizza almeno un mese di utilizzo dei dati per i calcoli seguenti.

Durante la valutazione della capacità, per le tabelle Amazon Keyspaces puoi configurare le unità di capacità di lettura RCUs () e le unità di capacità di scrittura (WCU) in modo indipendente.

### Argomenti

- [Come recuperare i parametri di consumo dalle tabelle Amazon Keyspaces](#)
- [Come identificare tabelle Amazon Keyspaces il cui provisioning è insufficiente](#)
- [Come identificare tabelle Amazon Keyspaces sovrafornite](#)

## Come recuperare i parametri di consumo dalle tabelle Amazon Keyspaces

Per valutare la capacità della tabella, monitora le seguenti CloudWatch metriche e seleziona la dimensione appropriata per recuperare le informazioni della tabella:

| unità di capacità in lettura | Unità di capacità in scrittura |
|------------------------------|--------------------------------|
| ConsumedReadCapacityUnits    | ConsumedWriteCapacityUnits     |
| ProvisionedReadCapacityUnits | ProvisionedWriteCapacityUnits  |
| ReadThrottleEvents           | WriteThrottleEvents            |

È possibile eseguire questa operazione tramite AWS CLI o. Console di gestione AWS

## AWS CLI

Prima di recuperare le metriche di consumo della tabella, devi iniziare acquisendo alcuni punti dati storici utilizzando l'API. CloudWatch

Inizia creando due file: `write-calc.json` e `read-calc.json`. Questi file rappresentano i calcoli per la tabella. È necessario aggiornare alcuni campi, come indicato nella tabella seguente, in modo che corrispondano all'ambiente in uso.

### Note

Se il nome della tabella non è univoco all'interno del tuo account, devi specificare anche il nome del keyspace.

| Nome campo                      | Definizione                                                                       | Esempio                                                |
|---------------------------------|-----------------------------------------------------------------------------------|--------------------------------------------------------|
| <code>&lt;table-name&gt;</code> | Il nome della tabella che stai analizzando                                        | SampleTable                                            |
| <code>&lt;period&gt;</code>     | Il periodo di tempo utilizzato o per valutare l'obiettivo di utilizzo, in secondi | Per un periodo di 1 ora è necessario specificare: 3600 |
| <code>&lt;start-time&gt;</code> | L'inizio dell'intervallo di valutazione, specificato nel formato 01 ISO86         | 2022-02-21T23:00:00                                    |
| <code>&lt;end-time&gt;</code>   | La fine dell'intervallo di valutazione, specificata nel formato 01 ISO86          | 2022-02-22T06:00:00                                    |

Il file di calcolo di scrittura recupera il numero di WCU fornite e consumate nel periodo di tempo per l'intervallo di date specificato. Genera inoltre una percentuale di utilizzo che può essere utilizzata per l'analisi. Il contenuto completo del `write-calc.json` file dovrebbe apparire come nell'esempio seguente.

```

{
 "MetricDataQueries": [
 {
 "Id": "provisionedWCU",
 "MetricStat": {
 "Metric": {
 "Namespace": "AWS/Cassandra",
 "MetricName": "ProvisionedWriteCapacityUnits",
 "Dimensions": [
 {
 "Name": "TableName",
 "Value": "<table-name>"
 }
]
 },
 "Period": <period>,
 "Stat": "Average"
 },
 "Label": "Provisioned",
 "ReturnData": false
 },
 {
 "Id": "consumedWCU",
 "MetricStat": {
 "Metric": {
 "Namespace": "AWS/Cassandra",
 "MetricName": "ConsumedWriteCapacityUnits",
 "Dimensions": [
 {
 "Name": "TableName",
 "Value": "<table-name>""
 }
]
 },
 "Period": <period>,
 "Stat": "Sum"
 },
 "Label": "",
 "ReturnData": false
 },
 {
 "Id": "m1",
 "Expression": "consumedWCU/PERIOD(consumedWCU)",

```

```

 "Label": "Consumed WCUs",
 "ReturnData": false
 },
 {
 "Id": "utilizationPercentage",
 "Expression": "100*(m1/provisionedWCU)",
 "Label": "Utilization Percentage",
 "ReturnData": true
 }
],
"StartTime": "<start-time>",
"EndTime": "<end-time>",
"ScanBy": "TimestampDescending",
"MaxDatapoints": 24
}

```

Il file di calcolo di lettura utilizza metriche simili. Questo file recupera quanti ne RCUs sono stati forniti e consumati durante il periodo di tempo per l'intervallo di date specificato. Il contenuto del `read-calc.json` file dovrebbe apparire come in questo esempio.

```

{
 "MetricDataQueries": [
 {
 "Id": "provisionedRCU",
 "MetricStat": {
 "Metric": {
 "Namespace": "AWS/Cassandra",
 "MetricName": "ProvisionedReadCapacityUnits",
 "Dimensions": [
 {
 "Name": "TableName",
 "Value": "<table-name>"
 }
]
 },
 "Period": <period>,
 "Stat": "Average"
 },
 "Label": "Provisioned",
 "ReturnData": false
 },
 {
 "Id": "consumedRCU",

```

```

"MetricStat": {
 "Metric": {
 "Namespace": "AWS/Cassandra",
 "MetricName": "ConsumedReadCapacityUnits",
 "Dimensions": [
 {
 "Name": "TableName",
 "Value": "<table-name>"
 }
]
 },
 "Period": <period>,
 "Stat": "Sum"
},
"Label": "",
"ReturnData": false
},
{
 "Id": "m1",
 "Expression": "consumedRCU/PERIOD(consumedRCU)",
 "Label": "Consumed RCU",
 "ReturnData": false
},
{
 "Id": "utilizationPercentage",
 "Expression": "100*(m1/provisionedRCU)",
 "Label": "Utilization Percentage",
 "ReturnData": true
}
],
"StartTime": "<start-time>",
"EndTime": "<end-time>",
"ScanBy": "TimestampDescending",
"MaxDatapoints": 24
}

```

Dopo aver creato i file, puoi iniziare a recuperare i dati di utilizzo.

1. Per recuperare i dati di utilizzo della scrittura, emettete il seguente comando:

```
aws cloudwatch get-metric-data --cli-input-json file://write-calc.json
```

2. Per recuperare i dati di utilizzo in lettura, emettete il seguente comando:



```
aws cloudwatch get-metric-data --cli-input-json file://read-calc.json
```

Il risultato per entrambe le query è una serie di punti dati in formato JSON che possono essere utilizzati per l'analisi. I risultati dipendono dal numero di punti dati specificati, dal periodo e dai dati specifici del carico di lavoro. Potrebbe essere come nell'esempio seguente.

```
{
 "MetricDataResults": [
 {
 "Id": "utilizationPercentage",
 "Label": "Utilization Percentage",
 "Timestamps": [
 "2022-02-22T05:00:00+00:00",
 "2022-02-22T04:00:00+00:00",
 "2022-02-22T03:00:00+00:00",
 "2022-02-22T02:00:00+00:00",
 "2022-02-22T01:00:00+00:00",
 "2022-02-22T00:00:00+00:00",
 "2022-02-21T23:00:00+00:00"
],
 "Values": [
 91.55364583333333,
 55.066631944444445,
 2.6114930555555556,
 24.9496875,
 40.947256944444445,
 25.618194444444444,
 0.0
],
 "StatusCode": "Complete"
 }
],
 "Messages": []
}
```

**Note**

Se si specifica un periodo breve e un intervallo di tempo lungo, potrebbe essere necessario modificare il `MaxDatapoints` valore, che per impostazione predefinita è impostato su 24 nello script. Ciò rappresenta un punto dati per ora e 24 al giorno.

## Console di gestione AWS

1. Accedi a Console di gestione AWS e vai alla pagina del CloudWatch servizio in [Getting Started with the Console di gestione AWS](#). Seleziona quello appropriato, Regione AWS se necessario.
2. Individua la sezione Metriche nella barra di navigazione a sinistra e scegli Tutte le metriche.
3. Si apre una dashboard con due pannelli. Il pannello superiore mostra l'immagine e il pannello inferiore contiene le metriche che desideri rappresentare graficamente. Scegli il pannello Amazon Keyspaces.
4. Scegli la categoria Table Metrics dai pannelli secondari. Questo ti mostra le tabelle nella tua cartella corrente. Regione AWS
5. Identificare il nome della tabella scorrendo il menu verso il basso e selezionare le metriche dell'operazione di scrittura: `ConsumedWriteCapacityUnits` e `ProvisionedWriteCapacityUnits`

**Note**

Questo esempio illustra le metriche delle operazioni di scrittura, ma puoi anche utilizzare questi passaggi per visualizzare graficamente le metriche delle operazioni di lettura.

6. Seleziona la scheda Graphed metrics (2) (Metriche nel grafico) per modificare le formule. Per impostazione predefinita, CloudWatch sceglie la funzione statistica Average per i grafici.
7. Dopo aver selezionato entrambe le metriche nel grafico (la casella di controllo a sinistra), selezionare il menu Add math (Aggiungi matematica), seguito da Common (Comune), quindi selezionare la funzione Percentage (Percentuale). Ripetere la procedura due volte.

È la prima volta che si seleziona la funzione Percentuale.

Seconda volta che si seleziona la funzione Percentuale.

8. A questo punto dovresti avere quattro metriche nel menu in basso. Lavoriamo sul calcolo ConsumedWriteCapacityUnits. Per essere coerenti, devi abbinare i nomi a quelli che hai usato nella AWS CLI sezione. Fare clic su m1 ID e modificare questo valore in consumedWCU.
9. Modifica della statistica da Average (Media) a Sum (Somma). Questa azione crea automaticamente un'altra metrica chiamata ANOMALY\_DETECTION\_BAND. Nell'ambito di questa procedura, puoi ignorarla rimuovendo la casella di controllo sulla metrica ad1 appena generata.
10. Ripetere il passaggio 8 per rinominare m2 ID in provisionedWCU. Lasciare la statistica impostata su Average (Media).
11. Scegli l'etichetta Expression1 e aggiorna il valore su m1 e l'etichetta su Consumed. WCU

 Note

Assicurati di aver selezionato solo m1 (casella di controllo a sinistra) e provisionedWCU per visualizzare correttamente i dati. Aggiorna la formula facendo clic su Details (Dettagli) e modificando la formula in consumedWCU/PERIOD(consumedWCU). Questo passaggio potrebbe anche generare un'altra metrica ANOMALY\_DETECTION\_BAND, ma nell'ambito di questa procedura puoi ignorarla.

12. Ora dovreste avere due grafici: uno che indichi il vostro approvvigionamento sulla tabella e l'altro che indichi il consumo. WCU WCU
13. Aggiornare la formula percentuale selezionando il grafico Expression2 (e2). Rinominate le etichette e impostatele su IDs UtilizationPercentage. Rinominare la formula in modo che corrisponda a  $100 * (m1 / provisionedWCU)$ .
14. Rimuovi la casella di controllo da tutte le metriche tranne utilizationPercentage per visualizzare i tuoi modelli di utilizzo. L'intervallo predefinito è impostato su 1 minuto, ma sentitevi liberi di modificarlo se necessario.

I risultati ottenuti dipendono dai dati effettivi del carico di lavoro. Gli intervalli con un utilizzo superiore al 100% sono soggetti a eventi di errore relativi alla bassa capacità di throughput. Amazon Keyspaces offre una capacità di [burst, ma non appena la capacità](#) di burst viene esaurita, qualsiasi elemento superiore al 100% presenta eventi di errore di bassa capacità di throughput.

## Come identificare tabelle Amazon Keyspaces il cui provisioning è insufficiente

Per la maggior parte dei carichi di lavoro, una tabella viene considerata sottodimensionata quando consuma costantemente più dell'80% della capacità assegnata.

La [capacità burst](#) è una funzionalità di Amazon Keyspaces che consente ai clienti di consumare temporaneamente RCUs/WCUs più di quanto originariamente fornito (più del throughput al secondo assegnato definito per la tabella). La capacità di espansione è stata creata per assorbire improvvisi aumenti di traffico dovuti a eventi speciali o picchi di utilizzo. Questa capacità di burst è limitata, per ulteriori informazioni, consulta [the section called "Usa la capacità burst"](#). Non appena la capacità inutilizzata RCUs è esaurita, si possono verificare degli errori di throughput a bassa capacità se si tenta di consumare più capacità di quella fornita. WCUs Quando il traffico delle applicazioni si avvicina al tasso di utilizzo dell'80%, il rischio che si verifichino eventi di errore di throughput a bassa capacità è notevolmente maggiore.

La regola del tasso di utilizzo dell'80% varia in base alla stagionalità dei dati e alla crescita del traffico. Considerare i seguenti scenari:

- Se il traffico è rimasto stabile a un tasso di utilizzo di circa il 90% negli ultimi 12 mesi, la tabella ha la capacità corretta
- Se il traffico delle applicazioni aumenta a un tasso dell'8% mensile in meno di 3 mesi, si arriverà al 100%
- Se il traffico delle applicazioni aumenta a un tasso dell'5% mensile in meno di 4 mesi, si arriverà al 100%

I risultati delle query precedenti forniscono un'immagine del tasso di utilizzo. Utilizzali come guida per valutare ulteriormente altre metriche che possono aiutarti a scegliere di aumentare la capacità della tabella in base alle esigenze (ad esempio: un tasso di crescita mensile o settimanale). Collabora con il tuo team operativo per definire qual è una buona percentuale per il carico di lavoro e le tabelle.

Esistono scenari speciali in cui i dati sono distorti quando vengono analizzati su base giornaliera o settimanale. Ad esempio, con le applicazioni stagionali che presentano picchi di utilizzo durante l'orario di lavoro (ma che poi scendono quasi a zero al di fuori dell'orario di lavoro), è possibile trarre vantaggio dalla [pianificazione dell'auto-scaling](#) delle applicazioni, in cui si specificano le ore del giorno (e i giorni della settimana) per aumentare la capacità fornita, nonché quando ridurla. Invece di puntare a una maggiore capacità per coprire le ore di punta, puoi anche trarre vantaggio dalle configurazioni di [auto-scaling delle tabelle di Amazon Keyspaces](#) se la stagionalità è meno pronunciata.

## Come identificare tabelle Amazon Keyspaces sovrafornite

I risultati delle query ottenuti dagli script precedenti forniscono i dati necessari per eseguire alcune analisi iniziali. Se il set di dati presenta valori di utilizzo inferiori al 20% per diversi intervalli, è possibile che la tabella presenti un provisioning eccessivo. Per definire ulteriormente se è necessario ridurre il numero di WCUs RCUS, è necessario rivedere le altre letture negli intervalli.

Quando la tabella contiene diversi intervalli di utilizzo ridotti, è possibile trarre vantaggio dall'utilizzo delle politiche di Application Auto Scaling, pianificando Application Auto Scaling o semplicemente configurando le politiche di Application Auto Scaling predefinite per la tabella basate sull'utilizzo.

Se hai un carico di lavoro con un basso utilizzo rispetto a un elevato rapporto di accelerazione (Max (ThrottleEvents) /Min (ThrottleEvents) nell'intervallo), ciò potrebbe accadere quando hai un carico di lavoro molto intenso in cui il traffico aumenta in modo significativo in giorni (o ore del giorno) specifici, ma per il resto è costantemente basso. In questi scenari, potrebbe essere utile utilizzare l'[Application Auto Scaling pianificato](#).

# Risoluzione dei problemi di Amazon Keyspaces (per Apache Cassandra)

Questa guida illustra i passaggi per la risoluzione dei problemi in vari scenari quando si lavora con Amazon Keyspaces (per Apache Cassandra). Include informazioni sulla risoluzione di errori generali, problemi di connessione, problemi di gestione della capacità ed errori del Data Definition Language (DDL).

- Errori generali
  - Risoluzione dei problemi relativi a eccezioni di primo livello come `NoNodeAvailableException`, `NoHostAvailableException`, e `AllNodesFailedException`
  - Isolamento degli errori sottostanti dalle eccezioni dei driver Java.
  - Implementazione di politiche di ripetizione dei tentativi e configurazione corretta delle connessioni.
- Problemi di connessione
  - Risoluzione degli errori durante la connessione agli endpoint Amazon Keyspaces `cqlsh` utilizzando i nostri driver client Apache Cassandra.
  - Risoluzione dei problemi delle connessioni degli endpoint VPC, delle connessioni Cassandra-stress e degli errori di configurazione IAM.
  - Gestione delle perdite di connessione durante l'importazione dei dati.
- Errori di gestione della capacità
  - Riconoscimento e risoluzione degli errori di capacità insufficiente relativi a tabelle, partizioni e connessioni.
  - Monitoraggio dei parametri pertinenti di Amazon Keyspaces in Amazon Logs. CloudWatch
  - Ottimizzazione delle connessioni e della velocità effettiva per migliorare le prestazioni.
- Errori DDL (Data Definition Language)
  - Risoluzione degli errori durante la creazione, l'accesso o il ripristino di keyspace e tabelle.
  - Gestione degli errori relativi alle impostazioni Time to Live (TTL) personalizzate, ai limiti delle colonne e alle eliminazioni degli intervalli.
  - Considerazioni relative ai carichi di lavoro di eliminazione intensivi.

Per una guida alla risoluzione dei problemi specifica per l'accesso IAM, consulta [the section called “Risoluzione dei problemi”](#). Per ulteriori informazioni sulle best practice di sicurezza, consulta [the section called “Best practice di sicurezza”](#).

## Argomenti

- [Risoluzione degli errori generali in Amazon Keyspaces](#)
- [Risoluzione degli errori di connessione in Amazon Keyspaces](#)
- [Risoluzione degli errori di gestione della capacità in Amazon Keyspaces](#)
- [Risoluzione degli errori del linguaggio di definizione dei dati in Amazon Keyspaces](#)

# Risoluzione degli errori generali in Amazon Keyspaces

Stai riscontrando errori generali? Ecco alcuni problemi comuni e come risolverli.

## Errori generali

Stai riscontrando una delle seguenti eccezioni di primo livello che possono verificarsi per molte ragioni diverse.

- `NoNodeAvailableException`
- `NoHostAvailableException`
- `AllNodesFailedException`

Queste eccezioni sono generate dal driver del client e possono verificarsi quando si stabilisce la connessione di controllo o quando si eseguono richieste. `read/write/prepare/execute/batch`

Quando l'errore si verifica durante la creazione della connessione di controllo, è segno che tutti i punti di contatto specificati nell'applicazione non sono raggiungibili. Quando l'errore si verifica durante l'esecuzione `read/write/prepare/execute` delle interrogazioni, indica che tutti i tentativi per quella richiesta sono stati esauriti. Ogni nuovo tentativo viene tentato su un nodo diverso quando si utilizza la politica di ripetizione dei tentativi predefinita.

## Come isolare l'errore sottostante dalle eccezioni dei driver Java di primo livello

Questi errori generali possono essere causati da problemi di connessione o durante l'esecuzione di operazioni. `read/write/prepare/execute` Nei sistemi distribuiti è necessario prevedere errori

temporanei e devono essere risolti riprovando la richiesta. Il driver Java non riprova automaticamente quando si verificano errori di connessione, quindi si consiglia di implementare la politica di nuovi tentativi quando si stabilisce la connessione del driver nell'applicazione. Per una panoramica dettagliata delle migliori pratiche di connessione, vedere. [the section called “Connessioni”](#)

Per impostazione predefinita, il driver Java è impostato su false `idempotence` per tutte le richieste, il che significa che il driver Java non riprova automaticamente la `read/write/prepare` richiesta non riuscita. Per impostare `true` e indicare `idempotence` al driver di riprovare le richieste non riuscite, puoi farlo in diversi modi. Ecco un esempio di come è possibile impostare l'idempotenza a livello di codice per una singola richiesta nell'applicazione Java.

```
Statement s = new SimpleStatement("SELECT * FROM my_table WHERE id = 1");
s.setIdempotent(true);
```

Oppure potete impostare l'idempotenza predefinita per l'intera applicazione Java a livello di codice, come illustrato nell'esempio seguente.

```
// Make all statements idempotent by default:
cluster.getConfiguration().getQueryOptions().setDefaultIdempotence(true);
//Set the default idempotency to true in your Cassandra configuration
basic.request.default-idempotence = true
```

Un'altra raccomandazione è quella di creare una politica di riprova a livello di applicazione. In questo caso, l'applicazione deve catturare `NoNodeAvailableException` e riprovare la richiesta. Si consigliano 10 nuovi tentativi con backoff esponenziale a partire da 10 ms e fino a 100 ms con un tempo totale di 1 secondo per tutti i tentativi.

[Un'altra opzione consiste nell'applicare la politica di riprova esponenziale di Amazon Keyspaces quando si stabilisce la connessione al driver Java disponibile su Github.](#)

Verifica di aver stabilito connessioni a più di un nodo quando utilizzi la politica di ripetizione dei tentativi predefinita. Puoi farlo utilizzando la seguente query in Amazon Keyspaces.

```
SELECT * FROM system.peers;
```

Se la risposta a questa query è vuota, significa che stai lavorando con un singolo nodo per Amazon Keyspaces. Se utilizzi la politica di riprova predefinita, non ci saranno nuovi tentativi perché il nuovo tentativo predefinito si verifica sempre su un nodo diverso. Per ulteriori informazioni su come stabilire connessioni tramite endpoint VPC, consulta. [the section called “Connessioni endpoint VPC”](#)



Per un step-by-step tutorial che mostra come stabilire una connessione ad Amazon Keyspaces utilizzando il driver Datastax 4.x Cassandra, consulta. [the section called “Plugin di autenticazione per Java 4.x”](#)

## Risoluzione degli errori di connessione in Amazon Keyspaces

Problemi di connessione? Ecco alcuni problemi comuni e come risolverli.

### Errori di connessione a un endpoint Amazon Keyspaces

Le connessioni non riuscite e gli errori di connessione possono generare messaggi di errore diversi. La sezione seguente illustra gli scenari più comuni.

#### Argomenti

- [Non riesco a connettermi ad Amazon Keyspaces con cqlsh](#)
- [Non riesco a connettermi ad Amazon Keyspaces utilizzando un driver client Cassandra](#)

### Non riesco a connettermi ad Amazon Keyspaces con cqlsh

Stai tentando di connetterti a un endpoint Amazon Keyspaces utilizzando cqlsh e la connessione fallisce con un. **Connection error**

Se tenti di connetterti a una tabella Amazon Keyspaces e cqlsh non è stato configurato correttamente, la connessione non riesce. La sezione seguente fornisce esempi dei problemi di configurazione più comuni che provocano errori di connessione quando si tenta di stabilire una connessione utilizzando cqlsh.

#### Note

Se stai tentando di connetterti ad Amazon Keyspaces da un VPC, sono necessarie autorizzazioni aggiuntive. Per configurare correttamente una connessione utilizzando gli endpoint VPC, segui i passaggi indicati in. [the section called “Connessione con endpoint VPC”](#)

Stai tentando di connetterti ad Amazon Keyspaces usando cqlsh, ma ricevi un errore di connessione. **timed out**

Questo potrebbe accadere se non hai fornito la porta corretta, con il risultato del seguente errore.

```
cqlsh cassandra.us-east-1.amazonaws.com 9140 -u "USERNAME" -p "PASSWORD" --ssl
Connection error: ('Unable to connect to any servers', {'3.234.248.199': error(None,
 "Tried connecting to [('3.234.248.199', 9140)]. Last error: timed out")})
```

Per risolvere il problema, verifica di utilizzare la porta 9142 per la connessione.

Stai tentando di connetterti ad Amazon Keyspaces usando cqlsh, ma ricevi un errore. **Name or service not known**

Questo potrebbe accadere se hai utilizzato un endpoint con errori di ortografia o inesistente. Nell'esempio seguente, il nome dell'endpoint è scritto in modo errato.

```
cqlsh cassandra.us-east-1.amazon.com 9142 -u "USERNAME" -p "PASSWORD" --ssl
Traceback (most recent call last):
 File "/usr/bin/cqlsh.py", line 2458, in <module>
 main(*read_options(sys.argv[1:], os.environ))
 File "/usr/bin/cqlsh.py", line 2436, in main
 encoding=options.encoding)
 File "/usr/bin/cqlsh.py", line 484, in __init__
 load_balancing_policy=WhiteListRoundRobinPolicy([self.hostname]),
 File "/usr/share/cassandra/lib/cassandra-driver-internal-only-3.11.0-bb96859b.zip/
cassandra-driver-3.11.0-bb96859b/cassandra/policies.py", line 417, in __init__
socket.gaierror: [Errno -2] Name or service not known
```

Per risolvere questo problema quando utilizzi endpoint pubblici per connetterti, seleziona un endpoint disponibile e verifica che il nome dell'endpoint non contenga errori. [the section called “Endpoint di servizio”](#) Se utilizzi gli endpoint VPC per la connessione, verifica che le informazioni sull'endpoint VPC siano corrette nella configurazione cqlsh.

Stai tentando di connetterti ad Amazon Keyspaces usando cqlsh, ma ricevi un errore.

### **OperationTimedOut**

Amazon Keyspaces richiede che SSL sia abilitato per le connessioni per garantire una maggiore sicurezza. Il parametro SSL potrebbe mancare se ricevi il seguente errore.

```
cqlsh cassandra.us-east-1.amazonaws.com -u "USERNAME" -p "PASSWORD"
Connection error: ('Unable to connect to any servers', {'3.234.248.192':
 OperationTimedOut('errors=Timed out creating connection (5 seconds),
 last_host=None',)})
```

#

Per risolvere questo problema, aggiungi il seguente flag al comando `cqlsh connection`.

`--ssl`

Stai tentando di connetterti ad Amazon Keyspaces usando `cqlsh` e ricevi un errore. **SSL transport factory requires a valid certfile to be specified**

In questo caso, manca il percorso del SSL/TLS certificato, il che genera il seguente errore.

```
cat .cassandra/cqlshrc
[connection]
port = 9142
factory = cqlshlib.ssl.ssl_transport_factory
#

cqlsh cassandra.us-east-1.amazonaws.com -u "USERNAME" -p "PASSWORD" --ssl
Validation is enabled; SSL transport factory requires a valid certfile to be specified.
Please provide path to the certfile in [ssl] section as 'certfile' option in /
root/.cassandra/cqlshrc (or use [certfiles] section) or set SSL_CERTFILE environment
variable.
#
```

Per risolvere questo problema, aggiungi il percorso al certfile sul tuo computer. Per ulteriori informazioni, consulta [the section called “Come configurare manualmente le cqlsh connessioni per TLS”](#).

```
certfile = path_to_file/keyspaces-bundle.pem
```

Stai tentando di connetterti ad Amazon Keyspaces usando `cqlsh`, ma ricevi un errore. **No such file or directory**

Questo potrebbe accadere se il percorso del file del certificato sul tuo computer è errato, il che genera il seguente errore.

```
cat .cassandra/cqlshrc
[connection]
port = 9142
factory = cqlshlib.ssl.ssl_transport_factory
```

```
[ssl]
validate = true
certfile = /root/wrong_path/keysaces-bundle.pem
#

cqlsh cassandra.us-east-1.amazonaws.com -u "USERNAME" -p "PASSWORD" --ssl
Connection error: ('Unable to connect to any servers', {'3.234.248.192': IOError(2, 'No
such file or directory')})
#
```

Per risolvere questo problema, verifica che il percorso del certfile sul tuo computer sia corretto. Per ulteriori informazioni, consulta [the section called “Come configurare manualmente le cqlsh connessioni per TLS”](#).

Stai tentando di connetterti ad Amazon Keyspaces usando cqlsh, ma ricevi un errore. **[X509] PEM lib**

Questo potrebbe accadere se il pem file del certificato SSL/TLS non è valido, con il risultato del seguente errore.

```
cqlsh cassandra.us-east-1.amazonaws.com -u "USERNAME" -p "PASSWORD" --ssl
Connection error: ('Unable to connect to any servers', {'3.234.248.241':
error(185090057, u"Tried connecting to [('3.234.248.241', 9142)]. Last error: [X509]
PEM lib (_ssl.c:3063)"))})
#
```

Per risolvere questo problema, assicurati di aver scaricato i certificati digitali richiesti. Per ulteriori informazioni, consulta [the section called “Come configurare manualmente le cqlsh connessioni per TLS”](#).

Stai tentando di connetterti ad Amazon Keyspaces usando cqlsh, ma ricevi un errore SSL. **unknown**

Questo potrebbe accadere se il pem file del certificato SSL/TLS è vuoto, il che genera il seguente errore.

```
cqlsh cassandra.us-east-1.amazonaws.com -u "USERNAME" -p "PASSWORD" --ssl
Connection error: ('Unable to connect to any servers', {'3.234.248.220': error(0,
u"Tried connecting to [('3.234.248.220', 9142)]. Last error: unknown error
(_ssl.c:3063)"))})
```

#

Per risolvere questo problema, assicurati di aver scaricato i certificati digitali richiesti. Puoi confermarlo utilizzando i passaggi descritti nell'argomento seguente [the section called “Come configurare manualmente le cqlsh connessioni per TLS”](#).

Stai tentando di connetterti ad Amazon Keyspaces usando cqlsh, ma ricevi un errore. **SSL: CERTIFICATE\_VERIFY\_FAILED**

Questo potrebbe accadere se non è possibile verificare il file del certificato SSL/TLS, con il risultato del seguente errore.

```
Connection error: ('Unable to connect to any servers', {'3.234.248.223':
 error(1, u"Tried connecting to [('3.234.248.223', 9142)]. Last error: [SSL:
 CERTIFICATE_VERIFY_FAILED] certificate verify failed (_ssl.c:727)"))})
```

Per risolvere questo problema, assicurati di aver scaricato i certificati digitali richiesti. Puoi confermarlo utilizzando i passaggi descritti nell'argomento seguente [the section called “Come configurare manualmente le cqlsh connessioni per TLS”](#).

Stai tentando di connetterti ad Amazon Keyspaces usando cqlsh, ma ricevi un errore. **Last error: timed out**

Questo potrebbe accadere se non hai configurato una regola in uscita per Amazon Keyspaces nel tuo gruppo di sicurezza EC2 Amazon, con il risultato del seguente errore.

```
cqlsh cassandra.us-east-1.amazonaws.com 9142 -u "USERNAME" -p "PASSWORD" --ssl
Connection error: ('Unable to connect to any servers', {'3.234.248.206': error(None,
 "Tried connecting to [('3.234.248.206', 9142)]. Last error: timed out")})
#
```

Per confermare che questo problema è causato dalla configurazione dell' EC2 istanza Amazon e non cqlsh, puoi provare a connetterti al tuo keyspace utilizzando AWS CLI, ad esempio, il seguente comando.

```
aws keyspaces list-tables --keyspace-name 'my_keyspace'
```

Se anche questo comando scade, l' EC2 istanza Amazon non è configurata correttamente.

Per confermare di disporre di autorizzazioni sufficienti per accedere ad Amazon Keyspaces, puoi utilizzare AWS CloudShell il comando per connetterti con `cqlsh`. Se tali connessioni vengono stabilite, devi configurare l' EC2 istanza Amazon.

Per risolvere questo problema, verifica che la tua EC2 istanza Amazon disponga di una regola in uscita che consenta il traffico verso Amazon Keyspaces. In caso contrario, devi creare un nuovo gruppo di sicurezza per l' EC2 istanza e aggiungere una regola che consenta il traffico in uscita verso le risorse di Amazon Keyspaces. Per aggiornare la regola in uscita per consentire il traffico verso Amazon Keyspaces, scegli CQLSH/CASSANDRA dal menu a discesa Tipo.

Dopo aver creato il nuovo gruppo di sicurezza con la regola del traffico in uscita, devi aggiungerlo all'istanza. Seleziona l'istanza, quindi scegli Azioni, Sicurezza e quindi Modifica gruppi di sicurezza. Aggiungi il nuovo gruppo di sicurezza con la regola in uscita, ma assicurati che rimanga disponibile anche il gruppo predefinito.

Per ulteriori informazioni su come visualizzare e modificare le regole EC2 in uscita, consulta [Aggiungere regole a un gruppo di sicurezza nella Amazon EC2 User Guide](#).

Stai tentando di connetterti ad Amazon Keyspaces usando `cqlsh`, ma ricevi un errore.

### **Unauthorized**

Questo potrebbe verificarsi se non disponi delle autorizzazioni Amazon Keyspaces nella policy utente di IAM, il che genera il seguente errore.

```
cqlsh cassandra.us-east-1.amazonaws.com 9142 -u "testuser-at-12345678910" -p
"PASSWORD" --ssl
Connection error: ('Unable to connect to any servers', {'3.234.248.241':
AuthenticationFailed('Failed to authenticate to 3.234.248.241: Error from server:
code=2100 [Unauthorized] message="User arn:aws:iam::12345678910:user/testuser has no
permissions."',)})
#
```

Per risolvere questo problema, assicurati che l'utente IAM `testuser-at-12345678910` disponga delle autorizzazioni per accedere ad Amazon Keyspaces. Per esempi di policy IAM che concedono l'accesso ad Amazon Keyspaces, consulta [the section called “Esempi di policy basate sull’identità”](#)

Per una guida alla risoluzione dei problemi specifica per l'accesso IAM, consulta [the section called “Risoluzione dei problemi”](#).

Stai tentando di connetterti ad Amazon Keyspaces usando `cqlsh`, ma ricevi un errore. **Bad credentials**

Questo potrebbe accadere se il nome utente o la password sono errati, con il risultato del seguente errore.

```
cqlsh cassandra.us-east-1.amazonaws.com 9142 -u "USERNAME" -p "PASSWORD" --ssl
Connection error: ('Unable to connect to any servers', {'3.234.248.248':
AuthenticationFailed('Failed to authenticate to 3.234.248.248: Error from server:
code=0100 [Bad credentials] message="Provided username USERNAME and/or password are
incorrect"',))
#
```

Per risolvere il problema, verifica che l'**USERNAME** and **PASSWORD** nel codice corrisponda al nome utente e alla password ottenuti quando hai generato le credenziali [specifiche del servizio](#).

#### Important

Se continui a riscontrare errori durante il tentativo di connessione con cqlsh, esegui nuovamente il comando con l'--debugopzione e includi l'output dettagliato al momento del contatto. [Supporto](#)

## Non riesco a connettermi ad Amazon Keyspaces utilizzando un driver client Cassandra

Le sezioni seguenti mostrano gli errori più comuni durante la connessione con un driver client Cassandra.

Stai tentando di connetterti a una tabella Amazon Keyspaces utilizzando il driver DataStax Java, ma ricevi un **NodeUnavailableException** errore.

Se la connessione su cui viene tentata la richiesta viene interrotta, si verifica il seguente errore.

```
[com.datastax.oss.driver.api.core.NodeUnavailableException: No connection
was available to Node(endPoint=vpce-22ff22f2f22222fff-aa1bb234.cassandra.us-
west-2.vpce.amazonaws.com/11.1.1111.222:9142, hostId=1a23456b-
c77d-8888-9d99-146cb22d6ef6, hashCode=123ca4567)]
```

Per risolvere il problema, individuate il valore del battito cardiaco e abbassatelo a 30 secondi se è più alto.

```
advanced.heartbeat.interval = 30 seconds
```

Quindi cerca il timeout associato e assicurati che il valore sia impostato su almeno 5 secondi.

```
advanced.connection.init-query-timeout = 5 seconds
```

Stai tentando di connetterti a una tabella Amazon Keyspaces utilizzando un driver e il plug-in SigV4, ma ricevi un errore. **AttributeError**

Se le credenziali non sono configurate correttamente, si verifica il seguente errore.

```
cassandra.cluster.NoHostAvailable: ('Unable to connect to any servers',
{'44.234.22.154:9142': AttributeError("'NoneType' object has no attribute
'access_key'")})
```

Per risolvere questo problema, verifica di passare le credenziali associate al tuo utente o ruolo IAM quando usi il plug-in SigV4. Il plugin SigV4 richiede le seguenti credenziali.

- **AWS\_ACCESS\_KEY\_ID**— Specifica una chiave di AWS accesso associata a un utente o ruolo IAM.
- **AWS\_SECRET\_ACCESS\_KEY**— Specifica la chiave segreta associata alla chiave di accesso. Si tratta essenzialmente della "password" per la chiave di accesso.

Per ulteriori informazioni sulle chiavi di accesso e sul plug-in SigV4, consulta [the section called “Crea credenziali IAM per l'autenticazione AWS”](#)

Stai tentando di connetterti a una tabella Amazon Keyspaces utilizzando un driver, ma ricevi un **PartialCredentialsError** errore.

Se **AWS\_SECRET\_ACCESS\_KEY** manca, può verificarsi il seguente errore.

```
cassandra.cluster.NoHostAvailable: ('Unable to connect to any servers',
{'44.234.22.153:9142':
PartialCredentialsError('Partial credentials found in config-file, missing:
aws_secret_access_key')})
```

Per risolvere questo problema, verifica di aver passato sia il **AWS\_ACCESS\_KEY\_ID** plugin SigV4 che il **AWS\_SECRET\_ACCESS\_KEY** quando usi il plugin SigV4. Per ulteriori informazioni sulle chiavi di accesso e sul plug-in SigV4, consulta [the section called “Crea credenziali IAM per l'autenticazione AWS”](#)



Stai tentando di connetterti a una tabella Amazon Keyspaces utilizzando un driver, ma ricevi un **Invalid signature** errore.

Questo potrebbe verificarsi se uno qualsiasi dei componenti richiesti per la firma è errato o non è definito correttamente per la sessione.

- AWS\_ACCESS\_KEY\_ID
- AWS\_SECRET\_ACCESS\_KEY
- AWS\_DEFAULT\_REGION

L'errore seguente è un esempio di chiavi di accesso non valide.

```
cassandra.cluster.NoHostAvailable: ('Unable to connect to any servers',
{'11.234.11.234:9142':
AuthenticationFailed('Failed to authenticate to 11.234.11.234:9142: Error from server:
code=0100
[Bad credentials] message="Authentication failure: Invalid signature"')})
```

Per risolvere questo problema, verifica che le chiavi di accesso e le chiavi di accesso siano Regione AWS state configurate correttamente per consentire al plug-in SigV4 di accedere ad Amazon Keyspaces. Per ulteriori informazioni sulle chiavi di accesso e sul plug-in SigV4, consulta [the section called "Crea credenziali IAM per l'autenticazione AWS"](#)

La mia connessione endpoint VPC non funziona correttamente

Stai tentando di connetterti ad Amazon Keyspaces utilizzando endpoint VPC, ma ricevi errori nella mappa dei token o stai riscontrando un throughput ridotto.

Questo potrebbe accadere se la connessione dell'endpoint VPC non è configurata correttamente.

Per risolvere questi problemi, verifica i seguenti dettagli di configurazione. Per seguire un step-by-step tutorial su come configurare una connessione tramite endpoint VPC di interfaccia per Amazon Keyspaces, consulta [the section called "Connessione con endpoint VPC"](#)

1. Verifica che l'entità IAM utilizzata per connettersi ad Amazon Keyspaces abbia read/write accesso alla tabella utente e all'accesso in lettura alle tabelle di sistema, come mostrato nell'esempio seguente.

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

```

 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "cassandra:Select",
 "cassandra:Modify"
],
 "Resource": [
 "arn:aws:cassandra:us-east-1:111122223333:/keyspace/mykeyspace/table/mytable",
 "arn:aws:cassandra:us-east-1:111122223333:/keyspace/system*"
]
 }
]
 }
}

```

2. Verifica che l'entità IAM utilizzata per connettersi ad Amazon Keyspaces disponga delle autorizzazioni di lettura necessarie per accedere alle informazioni sugli endpoint VPC sulla tua istanza EC2 Amazon, come mostrato nell'esempio seguente.

```

{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "ListVPCEndpoints",
 "Effect": "Allow",
 "Action": [
 "ec2:DescribeNetworkInterfaces",
 "ec2:DescribeVpcEndpoints"
],
 "Resource": "*"
 }
]
}

```

#### Note

Le policy gestite `AmazonKeyspacesFullAccess` includono `AmazonKeyspacesReadOnlyAccess_v2` le autorizzazioni necessarie per consentire ad Amazon Keyspaces di accedere all'istanza EC2 Amazon per leggere informazioni sugli endpoint VPC di interfaccia disponibili.

Per ulteriori informazioni sugli endpoint VPC, consulta [the section called “Utilizzo degli endpoint VPC dell'interfaccia per Amazon Keyspaces”](#)

3. Verificate che la configurazione SSL del driver Java imposti la convalida del nome host su false, come mostrato in questo esempio.

```
hostname-validation = false
```

Per ulteriori informazioni sulla configurazione dei driver, vedere. [the section called “Fase 2: Configurare il driver”](#)

4. Per confermare che l'endpoint VPC è stato configurato correttamente, puoi eseguire la seguente istruzione dall'interno del tuo VPC.

 Note

Non puoi utilizzare l'ambiente di sviluppo locale o l'editor CQL di Amazon Keyspaces per confermare questa configurazione, perché utilizzano l'endpoint pubblico.

```
SELECT peer FROM system.peers;
```

L'output dovrebbe essere simile a questo esempio e restituire da 2 a 6 nodi con IPv4 indirizzi privati quando ci si connette da una IPv4 rete, a seconda della configurazione del VPC e AWS della regione.

```
peer

192.0.2.0.15
192.0.2.0.24
192.0.2.0.13
192.0.2.0.7
192.0.2.0.8

(5 rows)
```

## Non riesco a connettermi tramite **cassandra-stress**

Stai tentando di connetterti ad Amazon Keyspaces usando il **cassandra-stress** comando, ma ricevi un **SSL context** errore.

Ciò accade se provi a connetterti ad Amazon Keyspaces, ma non hai configurato correttamente il TrustStore. Amazon Keyspaces richiede l'uso di Transport Layer Security (TLS) per proteggere le connessioni con i client.

In questo caso, viene visualizzato il seguente errore.

```
Error creating the initializing the SSL Context
```

Per risolvere questo problema, segui le istruzioni per configurare un TrustStore, come illustrato in questo argomento [the section called “Prima di iniziare”](#).

Una volta configurato il TrustStore, dovresti essere in grado di connetterti con il seguente comando.

```
./cassandra-stress user profile=./profile.yaml n=100 "ops(insert=1,select=1)"
cl=LOCAL_QUORUM -node "cassandra.eu-north-1.amazonaws.com" -port native=9142
-transport ssl-alg="PKIX" truststore="./cassandra_truststore.jks" truststore-
password="trustStore_pw" -mode native cql3 user="user_name" password="password"
```

## Non riesco a connettermi utilizzando le identità IAM

Stai tentando di connetterti a una tabella Amazon Keyspaces utilizzando un'identità IAM, ma ricevi un **Unauthorized** errore.

Ciò accade se tenti di connetterti a una tabella Amazon Keyspaces utilizzando un'identità IAM (ad esempio, un utente IAM) senza implementare la policy e fornire prima all'utente le autorizzazioni richieste.

In questo caso, viene visualizzato il seguente errore.

```
Connection error: ('Unable to connect to any servers', {'3.234.248.202':
 AuthenticationFailed('Failed to authenticate to 3.234.248.202:
Error from server: code=2100 [Unauthorized] message="User
arn:aws:iam::1234567890123:user/testuser has no permissions."',))
```

Per risolvere questo problema, verifica le autorizzazioni dell'utente IAM. Per connettersi con un driver standard, un utente deve avere almeno SELECT accesso alle tabelle di sistema, poiché la maggior parte dei driver legge il sistema keyspaces/tables quando stabilisce la connessione.

Ad esempio, le politiche IAM che concedono l'accesso al sistema Amazon Keyspaces e alle tabelle utente, vedi. [the section called “Accesso alle tabelle Amazon Keyspaces”](#)

Per consultare la sezione sulla risoluzione dei problemi specifica di IAM, consulta [the section called “Risoluzione dei problemi”](#).

Sto cercando di importare dati con cqlsh e la connessione alla mia tabella Amazon Keyspaces si interrompe

Stai cercando di caricare dati su Amazon Keyspaces con cqlsh, ma ricevi errori di connessione.

La connessione ad Amazon Keyspaces fallisce dopo che il client cqlsh riceve tre errori consecutivi di qualsiasi tipo dal server. Il client cqlsh ha esito negativo e viene visualizzato il seguente messaggio.

```
Failed to import 1 rows: NoHostAvailable - , will retry later, attempt 3 of 100
```

Per risolvere questo errore, devi assicurarti che i dati da importare corrispondano allo schema della tabella in Amazon Keyspaces. Controlla il file di importazione per verificare eventuali errori di analisi. Puoi provare a utilizzare una singola riga di dati utilizzando un'istruzione INSERT per isolare l'errore.

Il client tenta automaticamente di ristabilire la connessione.

## Risoluzione degli errori di gestione della capacità in Amazon Keyspaces

Problemi con la capacità serverless? Ecco alcuni problemi comuni e come risolverli.

### Errori di capacità serverless

Questa sezione descrive come riconoscere gli errori relativi alla gestione della capacità serverless e come risolverli. Ad esempio, è possibile che si verifichino eventi di capacità insufficiente quando l'applicazione supera la capacità di throughput assegnata.

Poiché Apache Cassandra è un software basato su cluster progettato per funzionare su una flotta di nodi, non presenta eccezioni relative a funzionalità serverless come la capacità di trasmissione. La maggior parte dei driver comprende solo i codici di errore disponibili in Apache Cassandra, quindi Amazon Keyspaces utilizza lo stesso set di codici di errore per mantenere la compatibilità.

Per mappare gli errori di Cassandra agli eventi di capacità sottostanti, puoi utilizzare Amazon CloudWatch per monitorare i parametri pertinenti di Amazon Keyspaces. Gli eventi di capacità

insufficiente che provocano errori sul lato client possono essere classificati in questi tre gruppi in base alla risorsa che causa l'evento:

- **Tabella:** se si sceglie la modalità Provisioned capacity per una tabella e l'applicazione supera il throughput assegnato, è possibile che si verifichino errori di capacità insufficiente. Per ulteriori informazioni, consulta [the section called “Configura le modalità di capacità di lettura/scrittura”](#).
- **Partizione:** è possibile che si verifichino eventi di capacità insufficiente se il traffico verso una determinata partizione supera 3.000 o 1.000. RCUs WCUs Come best practice, consigliamo di distribuire il traffico in modo uniforme tra le partizioni. Per ulteriori informazioni, consulta [the section called “Modellazione dei dati”](#).
- **Connessione:** la velocità effettiva potrebbe essere insufficiente se si supera la quota per il numero massimo di operazioni al secondo, per connessione. Per aumentare la velocità effettiva, è possibile aumentare il numero di connessioni predefinite durante la configurazione della connessione con il driver.

Per informazioni su come configurare le connessioni per Amazon Keyspaces, consulta [the section called “Come configurare le connessioni”](#) Per ulteriori informazioni sull'ottimizzazione delle connessioni sugli endpoint VPC, consulta [the section called “Connessioni endpoint VPC”](#)

Per determinare quale risorsa sta causando l'evento di capacità insufficiente che restituisce l'errore lato client, puoi controllare la dashboard nella console Amazon Keyspaces. Per impostazione predefinita, la console fornisce una visualizzazione aggregata delle metriche più comuni relative alla capacità e al traffico nella sezione Capacità e CloudWatch metriche correlate della scheda Capacità relativa alla tabella.

Per creare la tua dashboard con Amazon CloudWatch, controlla le seguenti metriche di Amazon Keyspaces.

- **PerConnectionRequestRateExceeded**— Richieste ad Amazon Keyspaces che superano la quota per la frequenza di richieste per connessione. Ogni connessione client ad Amazon Keyspaces può supportare fino a 3000 richieste CQL al secondo. Puoi eseguire più di 3000 richieste al secondo creando più connessioni.
- **ReadThrottleEvents**— Richieste ad Amazon Keyspaces che superano la capacità di lettura di una tabella.
- **StoragePartitionThroughputCapacityExceeded**— Richieste a una partizione di storage Amazon Keyspaces che superano la capacità di throughput della partizione. Le partizioni di storage Amazon Keyspaces possono supportare fino a 1000 WCU/WRU al secondo e 3000 RCU/RRU

al secondo. Per mitigare queste eccezioni, ti consigliamo di rivedere il tuo modello di dati per distribuire il read/write traffico su più partizioni.

- **WriteThrottleEvents**— Richieste ad Amazon Keyspaces che superano la capacità di scrittura di una tabella.

Per ulteriori informazioni CloudWatch, consulta [the section called “Monitoraggio con CloudWatch”](#). Per un elenco di tutte le CloudWatch metriche disponibili per Amazon Keyspaces, consulta [the section called “Parametri e dimensioni”](#)

#### Note

Per iniziare con una dashboard personalizzata che mostri tutte le metriche comunemente osservate per Amazon Keyspaces, puoi utilizzare un modello CloudWatch predefinito disponibile nell'AWS archivio degli GitHub esempi.

#### Argomenti

- [Ricevo errori di capacità NoHostAvailable insufficiente dal driver del mio client](#)
- [Ricevo errori di timeout di scrittura durante l'importazione dei dati](#)
- [Non riesco a vedere le dimensioni effettive di archiviazione di un keyspace o di una tabella](#)

### Ricevo errori di capacità **NoHostAvailable** insufficiente dal driver del mio client

Stai visualizzando le nostre eccezioni per una tabella **Read\_Timeout**. **Write\_Timeout**

Il tentativo ripetuto di scrivere o leggere da una tabella Amazon Keyspaces con capacità insufficiente può causare errori lato client specifici del driver.

Utilizzalo CloudWatch per monitorare i parametri di throughput assegnati ed effettivi e gli eventi di capacità insufficiente per la tabella. Ad esempio, una richiesta di lettura che non ha una capacità di throughput sufficiente ha esito negativo con un'**Read\_Timeout** eccezione e viene registrata nella metrica. **ReadThrottleEvents** Una richiesta di scrittura che non ha una capacità di throughput sufficiente ha esito negativo con un'**Write\_Timeout** eccezione e viene registrata nella metrica. **WriteThrottleEvents** Per ulteriori informazioni su questi parametri, consulta [the section called “Parametri e dimensioni”](#).

Per risolvere questi problemi, prendi in considerazione una delle seguenti opzioni.

- Aumentate il throughput assegnato per la tabella, ovvero la quantità massima di capacità di throughput che un'applicazione può consumare. Per ulteriori informazioni, consulta [the section called “Unità di capacità in lettura e unità di capacità in scrittura”](#).
- Consenti al servizio di gestire la capacità di throughput per tuo conto con la scalabilità automatica. Per ulteriori informazioni, consulta [the section called “Gestisci la capacità di throughput con la scalabilità automatica”](#).
- Scegli la modalità di capacità su richiesta per la tabella. Per ulteriori informazioni, consulta [the section called “Configura la modalità di capacità su richiesta”](#).

Se devi aumentare la quota di capacità predefinita per il tuo account, consulta [Quote](#).

Stai riscontrando errori relativi al superamento della capacità di partizione.

Quando viene visualizzato l'errore, la capacità

`StoragePartitionThroughputCapacityExceeded` della partizione viene temporaneamente superata. Ciò potrebbe essere gestito automaticamente dalla capacità adattiva o dalla capacità su richiesta. Ti consigliamo di rivedere il tuo modello di dati per distribuire il read/write traffico su più partizioni per mitigare questi errori. Le partizioni di storage Amazon Keyspaces possono supportare fino a 1000 WCU/WRU al secondo e 3000 RCU/RRU al secondo. Per ulteriori informazioni su come migliorare il modello di dati per distribuire il read/write traffico su più partizioni, consulta [the section called “Modellazione dei dati”](#)

`Write_Timeout` le eccezioni possono anche essere causate da una velocità elevata di operazioni di scrittura simultanee che includono dati statici e non statici nella stessa partizione logica. Se si prevede che il traffico esegua più operazioni di scrittura simultanee che includono dati statici e non statici all'interno della stessa partizione logica, si consiglia di scrivere i dati statici e non statici separatamente. La scrittura dei dati separatamente aiuta anche a ottimizzare i costi di throughput.

Stai riscontrando errori relativi al superamento della frequenza di richiesta di connessione.

Stai vedendo qualcosa a `PerConnectionRequestRateExceeded` causa di una delle seguenti cause.

- Potresti non avere un numero sufficiente di connessioni configurate per sessione.
- Potresti ricevere meno connessioni rispetto ai peer disponibili, perché le autorizzazioni degli endpoint VPC non sono configurate correttamente. Per ulteriori informazioni sulle policy degli endpoint VPC, consulta [the section called “Utilizzo degli endpoint VPC dell'interfaccia per Amazon Keyspaces”](#)



- Se utilizzi un driver 4.x, controlla se la convalida del nome host è abilitata. Il driver abilita la verifica del nome host TLS per impostazione predefinita. Questa configurazione fa sì che Amazon Keyspaces appaia come un cluster a nodo singolo per il driver. Ti consigliamo di disattivare la verifica del nome host.

Ti consigliamo di seguire queste best practice per assicurarti che le connessioni e il throughput siano ottimizzati:

- Configura l'ottimizzazione del throughput delle query CQL.

Amazon Keyspaces supporta fino a 3.000 query CQL per connessione TCP al secondo, ma non c'è limite al numero di connessioni che un driver può stabilire.

La maggior parte dei driver Cassandra open source stabilisce un pool di connessioni con Cassandra e bilancia il carico delle query su quel pool di connessioni. Amazon Keyspaces espone 9 indirizzi IP peer ai driver. Il comportamento predefinito della maggior parte dei driver consiste nello stabilire una singola connessione a ciascun indirizzo IP peer. Pertanto, la velocità massima di trasmissione delle query CQL di un driver che utilizza le impostazioni predefinite sarà di 27.000 query CQL al secondo.

Per aumentare questo numero, si consiglia di aumentare il numero di connessioni per indirizzo IP che il driver mantiene nel proprio pool di connessioni. Ad esempio, impostando il numero massimo di connessioni per indirizzo IP su 2, la velocità effettiva massima del driver raddoppierà fino a 54.000 query CQL al secondo.

- Ottimizza le connessioni a nodo singolo.

Per impostazione predefinita, la maggior parte dei driver Cassandra open source stabilisce una o più connessioni a ogni indirizzo IP pubblicizzato nella `system.peers` tabella quando si stabilisce una sessione. Tuttavia, alcune configurazioni possono comportare la connessione di un driver a un singolo indirizzo IP di Amazon Keyspaces. Ciò può accadere se il driver sta tentando la convalida del nome host SSL dei nodi peer (ad esempio, i driver DataStax Java) o quando si connette tramite un endpoint VPC.

Per ottenere la stessa disponibilità e le stesse prestazioni di un driver con connessioni a più indirizzi IP, ti consigliamo di fare quanto segue:

- Aumentate il numero di connessioni per IP a 9 o più, a seconda della velocità effettiva del client desiderata.

- Crea una politica di nuovi tentativi personalizzata che garantisca che i nuovi tentativi vengano eseguiti sullo stesso nodo. Per ulteriori informazioni, consulta la pagina

[the section called “Come configurare le politiche di nuovo tentativo”](#).

- Se utilizzi endpoint VPC, concedi all'entità IAM utilizzata per la connessione ad Amazon Keyspaces le autorizzazioni di accesso per interrogare il tuo VPC per le informazioni sull'endpoint e sull'interfaccia di rete. Ciò migliora il bilanciamento del carico e aumenta il throughput. read/write Per ulteriori informazioni, consulta [???](#).

## Ricevo errori di timeout di scrittura durante l'importazione dei dati

Stai ricevendo un errore di timeout durante il caricamento dei dati utilizzando il comando. **cqlsh COPY**

```
Failed to import 1 rows: WriteTimeout - Error from server: code=1100 [Coordinator node
timed out waiting for replica nodes' responses]
message="Operation timed out - received only 0 responses." info={'received_responses':
0, 'required_responses': 2, 'write_type': 'SIMPLE', 'consistency':
'LOCAL_QUORUM'}, will retry later, attempt 1 of 100
```

Amazon Keyspaces utilizza le WriteTimeout eccezioni ReadTimeout e per indicare quando una richiesta di scrittura non riesce a causa di una capacità di throughput insufficiente. Per aiutare a diagnosticare eccezioni di capacità insufficienti, Amazon Keyspaces pubblica le seguenti metriche su Amazon. CloudWatch

- WriteThrottleEvents
- ReadThrottledEvents
- StoragePartitionThroughputCapacityExceeded

Per risolvere gli errori di capacità insufficiente durante il caricamento dei dati, riduci la velocità di scrittura per lavoratore o la velocità di acquisizione totale, quindi riprova a caricare le righe. Per ulteriori informazioni, consulta [the section called “Fase 4: Configurare cqlsh COPY FROM le impostazioni”](#). [Per un'opzione di caricamento dei dati più solida, prendi in considerazione l'utilizzo DSBulk, disponibile nel repository. GitHub](#) Per step-by-step istruzioni, vedere [the section called “Caricamento dei dati utilizzando DSBulk”](#).

## Non riesco a vedere le dimensioni effettive di archiviazione di un keyspace o di una tabella

Non è possibile visualizzare la dimensione di archiviazione effettiva del keyspace o della tabella.

Per ulteriori informazioni sulle dimensioni di archiviazione della tabella, consulta [the section called “Valutazione dei costi a livello di tabella”](#). Puoi anche stimare le dimensioni dello spazio di archiviazione iniziando a calcolare la dimensione delle righe in una tabella. Istruzioni dettagliate per il calcolo della dimensione delle righe sono disponibili all'indirizzo [the section called “Stima della dimensione delle righe”](#).

## Risoluzione degli errori del linguaggio di definizione dei dati in Amazon Keyspaces

Hai problemi a creare risorse? Ecco alcuni problemi comuni e come risolverli.

### Errori del linguaggio di definizione dei dati

Amazon Keyspaces esegue operazioni DDL (Data Definition Language) in modo asincrono, ad esempio creando ed eliminando keyspace e tabelle. Se un'applicazione sta tentando di utilizzare la risorsa prima che sia pronta, l'operazione non riesce.

È possibile monitorare lo stato di creazione di nuovi spazi chiave e tabelle in Console di gestione AWS, che indica quando uno spazio chiave o una tabella è in sospeso o attivo. È inoltre possibile monitorare lo stato di creazione di un nuovo keyspace o di una nuova tabella a livello di codice interrogando la tabella dello schema di sistema. Uno spazio chiave o una tabella diventano visibili nello schema di sistema quando sono pronti per l'uso.

#### Note

Per ottimizzare la creazione di keyspace utilizzando CloudFormation, è possibile utilizzare questa utilità per convertire gli script CQL in modelli. CloudFormation [Lo strumento è disponibile nel repository. GitHub](#)

### Argomenti

- [Ho creato un nuovo keyspace, ma non riesco a visualizzarlo o accedervi](#)

- [Ho creato una nuova tabella, ma non riesco a visualizzarla o accedervi](#)
- [Sto cercando di ripristinare una tabella utilizzando Amazon Keyspaces point-in-time recovery \(PITR\), ma il ripristino non riesce](#)
- [Sto cercando di modificare le impostazioni Time INSERT/UPDATE to Live \(TTL\) personalizzate, ma l'operazione non riesce](#)
- [Sto cercando di caricare dati sulla mia tabella Amazon Keyspaces e ricevo un errore sul superamento del numero di colonne](#)
- [Sto cercando di eliminare i dati nella mia tabella Amazon Keyspaces e l'eliminazione non riesce per l'intervallo](#)

## Ho creato un nuovo keyspace, ma non riesco a visualizzarlo o accedervi

Stai ricevendo errori dalla tua applicazione che sta tentando di accedere a un nuovo keyspace.

Se tenti di accedere a uno spazio di chiavi Amazon Keyspaces appena creato che viene ancora creato in modo asincrono, riceverai un errore. L'errore seguente è un esempio.

```
InvalidRequest: Error from server: code=2200 [Invalid query] message="unconfigured keyspace mykeyspace"
```

Il modello di progettazione consigliato per verificare quando un nuovo keyspace è pronto per l'uso consiste nell'eseguire il polling delle tabelle dello schema di sistema Amazon Keyspaces (system\_schema\_mcs.\*).

Per ulteriori informazioni, consulta [the section called “Controlla lo stato di creazione del keyspace”](#).

## Ho creato una nuova tabella, ma non riesco a visualizzarla o accedervi

Stai ricevendo errori dalla tua applicazione che sta tentando di accedere a una nuova tabella.

Se tenti di accedere a una tabella Amazon Keyspaces appena creata che viene ancora creata in modo asincrono, riceverai un errore. Ad esempio, il tentativo di interrogare una tabella che non è ancora disponibile non riesce e genera un errore. `unconfigured table`

```
InvalidRequest: Error from server: code=2200 [Invalid query] message="unconfigured table mykeyspace.mytable"
```

Il tentativo di visualizzare la tabella `sync_table()` non riesce con `unKeyError`.

```
KeyError: 'mytable'
```

Il modello di progettazione consigliato per verificare quando una nuova tabella è pronta per l'uso consiste nel polling delle tabelle dello schema del sistema Amazon Keyspaces (`system_schema_mcs.*`).

Questo è l'output di esempio per una tabella in fase di creazione.

```
user-at-123@cqlsh:system_schema_mcs> select table_name,status from
system_schema_mcs.tables where keyspace_name='example_keyspace' and
table_name='example_table';

table_name | status
-----+-----
example_table | CREATING

(1 rows)
```

Questo è l'output di esempio per una tabella attiva.

```
user-at-123@cqlsh:system_schema_mcs> select table_name,status from
system_schema_mcs.tables where keyspace_name='example_keyspace' and
table_name='example_table';

table_name | status
-----+-----
example_table | ACTIVE

(1 rows)
```

Per ulteriori informazioni, consulta [the section called “Controlla lo stato di creazione della tabella”](#).

Sto cercando di ripristinare una tabella utilizzando Amazon Keyspaces point-in-time recovery (PITR), ma il ripristino non riesce

Se stai cercando di ripristinare una tabella Amazon Keyspaces con point-in-time ripristino (PITR) e vedi che il processo di ripristino inizia ma non viene completato correttamente, potresti non aver

configurato tutte le autorizzazioni richieste necessarie al processo di ripristino per questa particolare tabella.

Oltre alle autorizzazioni utente, Amazon Keyspaces potrebbe richiedere le autorizzazioni per eseguire azioni durante il processo di ripristino per conto del tuo principale. Questo è il caso se la tabella è crittografata con una chiave gestita dal cliente o se utilizzi politiche IAM che limitano il traffico in entrata.

Ad esempio, se utilizzi le chiavi di condizione nella tua policy IAM per limitare il traffico di origine a endpoint o intervalli IP specifici, l'operazione di ripristino ha esito negativo. Per consentire ad Amazon Keyspaces di eseguire l'operazione di ripristino della tabella per conto del tuo principale, devi aggiungere una chiave di condizione `aws:ViaAWSService` globale nella policy IAM.

Per ulteriori informazioni sulle autorizzazioni per ripristinare le tabelle, consulta [the section called “Configura le autorizzazioni IAM per il ripristino”](#)

Sto cercando di modificare le impostazioni Time INSERT/UPDATE to Live (TTL) personalizzate, ma l'operazione non riesce

Se stai tentando di inserire o aggiornare un valore TTL personalizzato, l'operazione potrebbe non riuscire con il seguente errore.

```
TTL is not yet supported.
```

Per specificare valori TTL personalizzati per righe o colonne utilizzando INSERT le nostre UPDATE operazioni, devi prima abilitare TTL per la tabella. È possibile abilitare il TTL per una tabella utilizzando la `ttl` proprietà personalizzata.

Per ulteriori informazioni sull'attivazione delle impostazioni TTL personalizzate per le tabelle, vedere [the section called “Aggiorna il TTL personalizzato della tabella”](#)

Sto cercando di caricare dati sulla mia tabella Amazon Keyspaces e ricevo un errore sul superamento del numero di colonne

Stai caricando dati e hai superato il numero di colonne che possono essere aggiornate contemporaneamente.

Questo errore si verifica quando lo schema della tabella supera la dimensione massima di 350 KB. Per ulteriori informazioni, consulta [Quote](#).

## Sto cercando di eliminare i dati nella mia tabella Amazon Keyspaces e l'eliminazione non riesce per l'intervallo

Stai tentando di eliminare i dati tramite chiave di partizione e ricevi un errore di eliminazione dell'intervallo.

Questo errore si verifica quando si tenta di eliminare più di 1.000 righe in un'unica operazione di eliminazione.

Range delete requests are limited by the amount of items that can be deleted in a single range.

Per ulteriori informazioni, consulta [the section called “Eliminazione dell'intervallo”](#).

Per eliminare più di 1.000 righe all'interno di una singola partizione, considera le seguenti opzioni.

- Elimina per partizione: se la maggior parte delle partizioni è inferiore a 1.000 righe, puoi provare a eliminare i dati per partizione. Se le partizioni contengono più di 1.000 righe, prova invece a eliminarli tramite la colonna di clustering.
- Elimina per colonna di raggruppamento: se il modello contiene più colonne di clustering, puoi utilizzare la gerarchia di colonne per eliminare più righe. Le colonne di clustering sono una struttura annidata ed è possibile eliminare molte righe agendo sulla colonna di primo livello.
- Elimina per singola riga: è possibile scorrere le righe ed eliminare ogni riga utilizzando la relativa chiave primaria completa (colonne di partizione e colonne di raggruppamento).
- Come best practice, valuta la possibilità di suddividere le righe tra le partizioni: in Amazon Keyspaces, ti consigliamo di distribuire la velocità effettiva tra le partizioni delle tabelle. Ciò distribuisce i dati e l'accesso in modo uniforme tra le risorse fisiche, garantendo la migliore velocità di trasmissione. Per ulteriori informazioni, consulta [the section called “Modellazione dei dati”](#).

Prendi in considerazione anche i seguenti consigli quando pianifichi operazioni di eliminazione per carichi di lavoro pesanti.

- Con Amazon Keyspaces, le partizioni possono contenere un numero praticamente illimitato di righe. Ciò consente di scalare le partizioni in modo «più ampio» rispetto alla tradizionale guida di Cassandra di 100 MB. Non è raro che le serie temporali o i registri crescano nel tempo oltre un gigabyte di dati.

- Con Amazon Keyspaces, non ci sono strategie di compattazione o lapidi da prendere in considerazione quando è necessario eseguire operazioni di eliminazione per carichi di lavoro pesanti. Puoi eliminare tutti i dati che desideri senza influire sulle prestazioni di lettura.



# Monitoraggio di Amazon Keyspaces (per Apache Cassandra)

Il monitoraggio è una parte importante per mantenere l'affidabilità, la disponibilità e le prestazioni di Amazon Keyspaces e delle altre AWS soluzioni. AWS fornisce i seguenti strumenti di monitoraggio per monitorare Amazon Keyspaces, segnalare quando qualcosa non va e intraprendere azioni automatiche se necessario:

- Amazon Keyspaces offre una dashboard preconfigurata Console di gestione AWS che mostra la latenza e gli errori aggregati in tutte le tabelle dell'account.
- Amazon CloudWatch monitora AWS le tue risorse e le applicazioni su cui esegui AWS in tempo reale. Puoi raccogliere e monitorare i parametri con dashboard personalizzati. Ad esempio, puoi creare una linea di base per le normali prestazioni di Amazon Keyspaces nel tuo ambiente misurando le prestazioni in diversi momenti e in diverse condizioni di carico. Durante il monitoraggio di Amazon Keyspaces, archivia i dati di monitoraggio storici in modo da poterli confrontare con i dati sulle prestazioni correnti, identificare modelli di prestazioni normali e anomalie prestazionali e ideare metodi per risolvere i problemi. Per stabilire una linea di base, è necessario monitorare almeno gli errori di sistema. Per ulteriori informazioni, consulta la [Amazon CloudWatch User Guide](#).
- Amazon CloudWatch alarms monitora una singola metrica in un periodo di tempo specificato ed esegue una o più azioni in base al valore della metrica rispetto a una determinata soglia per diversi periodi di tempo. Ad esempio, se utilizzi Amazon Keyspaces in modalità provisioning con scaling automatico delle applicazioni, l'azione è una notifica inviata da Amazon Simple Notification Service (Amazon SNS) per valutare una politica di Application Auto Scaling.

CloudWatch gli allarmi non richiamano azioni semplicemente perché si trovano in uno stato particolare. Lo stato deve essere cambiato e restare costante per un numero specificato di periodi. Per ulteriori informazioni, consulta [Monitoraggio di Amazon Keyspaces con Amazon CloudWatch](#).

- Amazon CloudWatch Logs ti consente di monitorare, archiviare e accedere ai tuoi file di log da tabelle Amazon Keyspaces e CloudTrail altre fonti. CloudWatch I log possono monitorare le informazioni 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 CloudTrail acquisisce le chiamate API e gli eventi correlati effettuati da o per conto del tuo Account AWS e fornisce i file di log a un bucket Simple Storage Service (Amazon S3) specificato.

Puoi identificare quali utenti e account hanno richiamato AWS, l'indirizzo IP di origine da cui sono state effettuate le chiamate e quando sono avvenute. Per ulteriori informazioni, consulta la [AWS CloudTrail Guida per l'utente di](#).

Amazon EventBridge è un servizio di bus eventi senza server che semplifica la connessione delle applicazioni con dati provenienti da una varietà di fonti. EventBridge fornisce un flusso di dati in tempo reale dalle tue applicazioni, applicazioni Software-as-a-Service (SaaS) e AWS servizi e indirizza tali dati verso destinazioni come Lambda. In questo modo puoi monitorare gli eventi che si verificano nei servizi e creare architetture basate su eventi. Per ulteriori informazioni, consulta la [Amazon EventBridge User Guide](#).

#### Argomenti

- [Monitoraggio di Amazon Keyspaces con Amazon CloudWatch](#)
- [Registrazione delle chiamate API Amazon Keyspaces con AWS CloudTrail](#)

## Monitoraggio di Amazon Keyspaces con Amazon CloudWatch

Puoi monitorare Amazon Keyspaces utilizzando Amazon CloudWatch, che raccoglie dati grezzi e li elabora in parametri leggibili quasi in tempo reale. Queste statistiche vengono conservate per un periodo di 15 mesi, per permettere l'accesso alle informazioni storiche e offrire una prospettiva migliore sulle prestazioni del servizio o dell'applicazione web.

È anche possibile impostare allarmi che controllano determinate soglie e inviare notifiche o intraprendere azioni quando queste soglie vengono raggiunte. Per ulteriori informazioni, consulta la [Amazon CloudWatch User Guide](#).

#### Note

Per iniziare rapidamente con una CloudWatch dashboard preconfigurata che mostra metriche comuni per Amazon Keyspaces, puoi utilizzare un CloudFormation modello disponibile da <https://github.com/aws-samples/amazon-keyspaces-cloudwatch-cloudformation-templates>

#### Argomenti

- [Come posso usare i parametri di Amazon Keyspaces?](#)
- [Metriche e dimensioni di Amazon Keyspaces](#)

- [Visualizzazione dei parametri di Amazon Keyspaces in CloudWatch](#)
- [Creazione di CloudWatch allarmi per monitorare Amazon Keyspaces](#)

## Come posso usare i parametri di Amazon Keyspaces?

Le metriche riportate da Amazon Keyspaces forniscono informazioni che puoi analizzare in diversi modi. L'elenco seguente mostra alcuni usi comuni dei parametri. Questi suggerimenti sono solo introduttivi e non costituiscono un elenco completo. [Per ulteriori informazioni sui parametri e sulla fidelizzazione, consulta Metrics.](#)

| In che modo?                                                                                        | Parametri rilevanti                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-----------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Come posso determinare se si sono verificati errori di sistema?                                     | È possibile monitorare <code>SystemErrors</code> per determinare se eventuali richieste hanno generato un codice di errore del server. In genere, questo parametro deve essere uguale a zero. Se non lo è, potresti voler indagare.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| Come posso confrontare la capacità di lettura media assegnata con la capacità di lettura consumata? | <p>Per monitorare la capacità di lettura media assegnata e la capacità di lettura consumata</p> <ol style="list-style-type: none"> <li>1. Imposta il periodo <code>ProvisionedReadCapacityUnits</code> per <code>ConsumedReadCapacityUnits</code> e l'intervallo che desideri monitorare.</li> <li>2. Modifica la statistica per <code>ConsumedReadCapacityUnits</code> da <code>Average</code> a <code>Sum</code>.</li> <li>3. Crea una nuova espressione matematica vuota.</li> <li>4. Nella sezione Dettagli della nuova espressione matematica, inserisci l'ID di <b><code>ConsumedReadCapacityUnits</code></b> e dividi la metrica per la funzione CloudWatch <code>PERIOD</code> della metrica (<code>metric_id / (PERIOD (metric_id))</code>).</li> <li>5. Deseleziona. <code>ConsumedReadCapacityUnits</code></li> </ol> <p>È ora possibile confrontare la capacità di lettura media consumata con la capacità assegnata. <a href="#">Per ulteriori informazioni</a></p> |

| In che modo?                                                                                            | Parametri rilevanti                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|---------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                                         | <a href="#">sulle funzioni aritmetiche di base e su come creare una serie temporale, consulta Using metric Math.</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| Come posso confrontare la capacità di scrittura assegnata media con la capacità di scrittura consumata? | <p>Per monitorare la capacità di scrittura media assegnata e la capacità di scrittura consumata</p> <ol style="list-style-type: none"> <li>1. Imposta il periodo <code>ProvisionedWriteCapacityUnits</code> per <code>ConsumedWriteCapacityUnits</code> e l'intervallo che desideri monitorare.</li> <li>2. Modifica la statistica per <code>ConsumedWriteCapacityUnits</code> da <code>aAverage. Sum</code></li> <li>3. Crea una nuova espressione matematica vuota.</li> <li>4. Nella sezione Dettagli della nuova espressione matematica, inserisci l'ID di <b><code>ConsumedWriteCapacityUnits</code></b> e dividi la metrica per la funzione CloudWatch <code>PERIOD</code> della metrica (<code>metric_id/ (PERIOD (metric_id))</code>).</li> <li>5. Deseleziona. <code>ConsumedWriteCapacityUnits</code></li> </ol> <p>È ora possibile confrontare la capacità di scrittura media consumata con la capacità assegnata. <a href="#">Per ulteriori informazioni sulle funzioni aritmetiche di base e su come creare una serie temporale, consulta Using metric Math.</a></p> |

## Metriche e dimensioni di Amazon Keyspaces

Quando interagisci con Amazon Keyspaces, invia parametri e dimensioni ad Amazon. CloudWatch Tutte le metriche vengono aggregate e riportate ogni minuto. Le seguenti sezioni offrono una panoramica di tutti i parametri disponibili per Amazon Keyspaces.

### Argomenti

- [Parametri di Amazon Keyspaces](#)
- [Dimensioni per i parametri di Amazon Keyspaces](#)
- [Metriche per Amazon Keyspaces change data capture \(CDC\)](#)

- [Le dimensioni di Amazon Keyspaces modificano i parametri di acquisizione dei dati \(CDC\)](#)

## Parametri di Amazon Keyspaces

Amazon CloudWatch aggrega i parametri di Amazon Keyspaces a intervalli di un minuto.

Non tutte le statistiche, come Average o Sum, si applicano a tutti i parametri. Tuttavia, tutti questi valori sono disponibili tramite la console Amazon Keyspaces o utilizzando la CloudWatch console o AWS SDKs per tutte le metriche. AWS CLI Nella tabella seguente ciascun parametro presenta un elenco di statistiche valide applicabile a quel parametro.

| Parametro                  | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| AccountMaxTableLevelReads  | <p>Il numero massimo di unità di capacità di lettura che possono essere utilizzate da una tabella di un account. Per le tabelle su richiesta, questo limite limita il numero massimo di unità di richiesta di lettura che una tabella può utilizzare.</p> <p>Unità: Count</p> <p>Statistiche valide:</p> <ul style="list-style-type: none"> <li>• <b>Maximum</b>— Il numero massimo di unità di capacità di lettura che possono essere utilizzate da una tabella dell'account.</li> </ul> |
| AccountMaxTableLevelWrites | <p>Il numero massimo di unità di capacità di scrittura che possono essere utilizzate da una tabella di un account. Per le tabelle su richiesta, questo limite limita il numero massimo di unità di richiesta di scrittura che una tabella può utilizzare.</p> <p>Unità: Count</p> <p>Statistiche valide:</p>                                                                                                                                                                              |

| Parametro                                        | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|--------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                  | <ul style="list-style-type: none"><li>• <b>Maximum</b>— Il numero massimo di unità di capacità di scrittura che possono essere utilizzate da una tabella dell'account.</li></ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>AccountProvisionedReadCapacityUtilization</b> | <p>La percentuale di unità di capacità di lettura assegnata utilizzate dall'account.</p> <p>Unità: Percent</p> <p>Statistiche valide:</p> <ul style="list-style-type: none"><li>• <b>Maximum</b>: la percentuale massima di unità di capacità di lettura assegnata utilizzate dall'account.</li><li>• <b>Minimum</b>: la percentuale minima di unità di capacità di lettura assegnata utilizzate dall'account.</li><li>• <b>Average</b>: la percentuale minima di unità di capacità di lettura assegnata utilizzate dall'account. Il parametro viene pubblicato ad intervalli di cinque minuti. Pertanto, se si modificano rapidamente le unità di capacità di lettura assegnata, questa statistica potrebbe non corrispondere alla media reale.</li></ul> |

| Parametro                                               | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|---------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>AccountProvisionedWriteCapacityUtilization</code> | <p>La percentuale di unità di capacità di scrittura assegnata utilizzate dall'account.</p> <p>Unità: Percent</p> <p>Statistiche valide:</p> <ul style="list-style-type: none"> <li>• <b>Maximum</b>: la percentuale massima di unità di capacità di scrittura assegnata utilizzate dall'account.</li> <li>• <b>Minimum</b>: la percentuale minima di unità di capacità di scrittura assegnata utilizzate dall'account.</li> <li>• <b>Average</b>: la percentuale media di unità di capacità di scrittura assegnata utilizzate dall'account. Il parametro viene pubblicato ad intervalli di cinque minuti. Pertanto, se si modificano rapidamente le unità di capacità di scrittura assegnata, questa statistica potrebbe non corrispondere alla media reale.</li> </ul> |
| <code>BillableTableSizeInBytes</code>                   | <p>La dimensione fatturabile della tabella in byte. È la somma delle dimensioni codificate di tutte le righe della tabella. Questa metrica consente di tenere traccia dei costi di archiviazione delle tabelle nel tempo.</p> <p>Unità: Bytes</p> <p>Dimensioni: <code>Keyspace</code>, <code>TableName</code></p> <p>Statistiche valide:</p> <ul style="list-style-type: none"> <li>• <b>Maximum</b>— La dimensione massima di archiviazione della tabella.</li> <li>• <b>Minimum</b>— La dimensione minima di archiviazione della tabella.</li> <li>• <b>Average</b>— La dimensione media di archiviazione della tabella. Questa metrica viene calcolata su intervalli di 4-6 ore.</li> </ul>                                                                         |

| Parametro                                   | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|---------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>ConditionalCheckFailedRequests</code> | <p>Il numero di richieste di scrittura LWT (Lightweight Transaction) non riuscite. Le operazioni INSERT, UPDATE e DELETE consentono di fornire una condizione logica che deve essere considerata true affinché l'operazione possa procedere. Se questa condizione risulta falsa, <code>ConditionalCheckFailedRequests</code> viene incrementata di uno. I controlli delle condizioni che danno come risultato falso consumano unità di capacità di scrittura in base alla dimensione della riga. Per ulteriori informazioni, consulta <a href="#">the section called “Stima del consumo di capacità di LWT”</a>.</p> <p>Unità: Count</p> <p>Dimensioni: Keyspace, TableName</p> <p>Statistiche valide:</p> <ul style="list-style-type: none"><li>• Minimum</li><li>• Maximum</li><li>• Average</li><li>• SampleCount</li><li>• Sum</li></ul> |



| Parametro                                                                                                                                                                                                                                                                                                                                                                           | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ConsumedReadCapacityUnits                                                                                                                                                                                                                                                                                                                                                           | <p>Il numero di unità con capacità di lettura consumate nel periodo di tempo specificato. Per ulteriori informazioni, vedere la modalità di <a href="#">capacità di lettura/scrittura</a>.</p> <div><p> <b>Note</b></p><p>Per comprendere l'utilizzo medio del throughput al secondo, utilizza la Sum statistica per calcolare la velocità effettiva consumata per il periodo di un minuto. Quindi dividi la somma per il numero di secondi in un minuto (60) per calcolare la media ConsumedReadCapacityUnits al secondo (riconoscendo che questa media non evidenzia eventuali picchi ampi ma brevi nell'attività di lettura verificatisi durante quel minuto). Per ulteriori informazioni sul confronto della capacità di lettura media consumata con la capacità di lettura assegnata, vedere <a href="#">the section called “Utilizzo delle metriche”</a></p></div> |
| Unità: Count                                                                                                                                                                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| Dimensioni: Keyspace, TableName                                                                                                                                                                                                                                                                                                                                                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| Statistiche valide:                                                                                                                                                                                                                                                                                                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <ul style="list-style-type: none"><li>• <b>Minimum</b>— Il numero minimo di unità di capacità di lettura utilizzate da ogni singola richiesta alla tabella.</li><li>• <b>Maximum</b>— Il numero massimo di unità di capacità di lettura consumate da ogni singola richiesta alla tabella.</li><li>• <b>Average</b>: la capacità di lettura media per richiesta consumata.</li></ul> |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |

| Parametro | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|           | <div> <b>Note</b><br/>Il valore <code>Average</code> è influenzato dai periodi di inattività in cui il valore del campione sarà zero.</div> <ul style="list-style-type: none"><li>• <code>Sum</code>: le unità di capacità di lettura totali consumate . Questa è la statistica più utile per il parametro <code>ConsumedReadCapacityUnits</code> .</li><li>• <code>SampleCount</code> — Il numero di richieste ad Amazon Keyspaces, anche se non è stata consumata alcuna capacità di lettura.</li></ul> <div> <b>Note</b><br/>Il valore <code>SampleCount</code> è influenzato dai periodi di inattività in cui il valore del campione sarà zero.</div> |

| Parametro                               | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>ConsumedWriteCapacityUnits</code> | <p>Il numero di unità con capacità di scrittura consumate nel periodo di tempo specificato. È possibile recuperare la capacità di scrittura totale consumata per una tabella. Per ulteriori informazioni, vedete la modalità di capacità di <a href="#">lettura/scrittura</a>.</p> <div><p> <b>Note</b></p><p>Per comprendere l'utilizzo medio del throughput al secondo, utilizza la Sum statistica per calcolare la velocità effettiva consumata per il periodo di un minuto. Quindi dividi la somma per il numero di secondi in un minuto (60) per calcolare la media <code>ConsumedWriteCapacityUnits</code> al secondo (riconoscendo che questa media non evidenzia eventuali picchi ampi ma brevi nell'attività di scrittura verificatisi durante quel minuto). Per ulteriori informazioni sul confronto della capacità di scrittura media consumata con la capacità di scrittura assegnata, vedere <a href="#">the section called “Utilizzo delle metriche”</a></p></div> <p>Unità: Count</p> <p>Dimensioni: Keyspace, TableName</p> <p>Statistiche valide:</p> <ul style="list-style-type: none"><li>• <b>Minimum</b>— Il numero minimo di unità con capacità di scrittura utilizzate da ogni singola richiesta alla tabella.</li><li>• <b>Maximum</b>— Il numero massimo di unità di capacità di scrittura utilizzate da ogni singola richiesta alla tabella.</li><li>• <b>Average</b>: la capacità di scrittura media per richiesta consumata.</li></ul> |

| Parametro | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|           | <div>  <b>Note</b><br/>           Il valore <code>Average</code> è influenzato dai periodi di inattività in cui il valore del campione sarà zero.         </div> <ul style="list-style-type: none"> <li>• <code>Sum</code>: le unità di capacità di scrittura totali consumate . Questa è la statistica più utile per il parametro <code>ConsumedWriteCapacityUnits</code> .</li> <li>• <code>SampleCount</code> — Il numero di richieste ad Amazon Keyspaces, anche se non è stata consumata alcuna capacità di scrittura.</li> </ul> <div>  <b>Note</b><br/>           Il valore <code>SampleCount</code> è influenzato dai periodi di inattività in cui il valore del campione sarà zero.         </div> |

| Parametro                                  | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|--------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MaxProvisionedTableReadCapacityUtilization | <p>La percentuale di unità di capacità di lettura assegnate utilizzate dalla tabella di lettura più elevata di un account.</p> <p>Unità: Percent</p> <p>Statistiche valide:</p> <ul style="list-style-type: none"><li>• <b>Maximum:</b> la percentuale massima di unità di capacità di lettura assegnata utilizzate dalla tabella di lettura assegnata più elevata dell'account.</li><li>• <b>Minimum:</b> la percentuale minima di unità di capacità di lettura assegnata utilizzate dalla tabella di lettura assegnata più elevata dell'account.</li><li>• <b>Average:</b> la percentuale media di unità di capacità di lettura assegnata utilizzate dalla tabella di lettura assegnata più elevata dell'account. Il parametro viene pubblicato ad intervalli di cinque minuti. Pertanto, se si modificano rapidamente le unità di capacità di lettura assegnata, questa statistica potrebbe non corrispondere alla media reale.</li></ul> |

| Parametro                                   | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|---------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MaxProvisionedTableWriteCapacityUtilization | <p>La percentuale di capacità di scrittura assegnata utilizzata dalla tabella di scrittura più assegnata di un account.</p> <p>Unità: Percent</p> <p>Statistiche valide:</p> <ul style="list-style-type: none"><li>• <b>Maximum</b>— La percentuale massima di unità di capacità di scrittura assegnate utilizzate dalla tabella di scrittura più assegnata dell'account.</li><li>• <b>Minimum</b>— La percentuale minima di unità di capacità di scrittura assegnate utilizzate dalla tabella di scrittura più fornita dell'account.</li><li>• <b>Average</b>— La percentuale media di unità di capacità di scrittura assegnate utilizzate dalla tabella di scrittura con la più alta disponibilità dell'account. Il parametro viene pubblicato ad intervalli di cinque minuti. Pertanto, se si modificano rapidamente le unità di capacità di scrittura assegnata, questa statistica potrebbe non corrispondere alla media reale.</li></ul> |

| Parametro                                     | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-----------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>PerConnectionRequestRateExceeded</code> | <p>Richieste ad Amazon Keyspaces che superano la quota di velocità di richiesta per connessione. Ogni connessione client ad Amazon Keyspaces può supportare fino a 3000 richieste CQL al secondo. I client possono creare più connessioni per aumentare la velocità effettiva.</p> <p>Quando si utilizza la replica in più regioni, anche ogni scrittura replicata contribuisce a questa quota. Come best practice, consigliamo di aumentare il numero di connessioni alle tabelle per evitare errori. <code>PerConnectionRequestRateExceeded</code> Non c'è limite al numero di connessioni che puoi avere in Amazon Keyspaces.</p> <p>Unità: Count</p> <p>Dimensioni: Keyspace, TableName, Operation</p> <p>Statistiche valide:</p> <ul style="list-style-type: none"><li>• <code>SampleCount</code></li><li>• <code>Sum</code></li></ul> |

| Parametro                                 | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>ProvisionedReadCapacityUnits</code> | <p>Il numero di unità di capacità di lettura assegnate per una tabella.</p> <p>La <code>TableName</code> dimensione restituisce il valore <code>ProvisionedReadCapacityUnits</code> per la tabella.</p> <p>Unità: Count</p> <p>Dimensioni: Keyspace, TableName</p> <p>Statistiche valide:</p> <ul style="list-style-type: none"><li>• <b>Minimum:</b> l'impostazione più bassa per la capacità di lettura assegnata. Se si utilizza <code>ALTER TABLE</code> per aumentare la capacità di lettura, questo parametro mostra il valore più basso di <code>ReadCapacityUnits</code> assegnato durante questo periodo di tempo.</li><li>• <b>Maximum:</b> l'impostazione più alta per la capacità di lettura assegnata. Se si utilizza <code>ALTER TABLE</code> per aumentare la capacità di lettura, questo parametro mostra il valore più alto di <code>ReadCapacityUnits</code> assegnato durante questo periodo di tempo.</li><li>• <b>Average:</b> la capacità media di lettura assegnata. Il parametro <code>ProvisionedReadCapacityUnits</code> viene pubblicato a intervalli di cinque minuti. Pertanto, se si modificano rapidamente le unità di capacità di lettura assegnata, questa statistica potrebbe non corrispondere alla media reale.</li></ul> |



| Parametro                                  | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|--------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>ProvisionedWriteCapacityUnits</code> | <p>Il numero di unità di capacità di scrittura assegnate per una tabella.</p> <p>La <code>TableName</code> dimensione restituisce il valore <code>ProvisionedWriteCapacityUnits</code> per la tabella.</p> <p>Unità: Count</p> <p>Dimensioni: Keyspace, TableName</p> <p>Statistiche valide:</p> <ul style="list-style-type: none"><li>• <b>Minimum:</b> l'impostazione più bassa per la capacità di scrittura assegnata. Se si utilizza <code>ALTER TABLE</code> per aumentare la capacità di scrittura, questo parametro mostra il valore più basso di <code>WriteCapacityUnits</code> assegnato durante questo periodo di tempo.</li><li>• <b>Maximum:</b> l'impostazione più alta per la capacità di scrittura assegnata. Se si utilizza <code>ALTER TABLE</code> per aumentare la capacità di scrittura, questo parametro mostra il valore più alto di <code>WriteCapacityUnits</code> assegnato durante questo periodo di tempo.</li><li>• <b>Average:</b> la capacità media di scrittura assegnata. Il parametro <code>ProvisionedWriteCapacityUnits</code> viene pubblicato a intervalli di cinque minuti. Pertanto, se si modificano rapidamente le unità di capacità di scrittura assegnata, questa statistica potrebbe non corrispondere alla media reale.</li></ul> |

| Parametro          | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ReadThrottleEvents | <p>Richieste ad Amazon Keyspaces che superano la capacità di lettura prevista per una tabella o quote a livello di account, quote di richiesta per connessione o quote a livello di partizione.</p> <p>Unità: Count</p> <p>Dimensioni: Keyspace, TableName, Operation</p> <p>Statistiche valide:</p> <ul style="list-style-type: none"><li>• SampleCount</li><li>• Sum</li></ul>                                                           |
| ReplicationLatency | <p>Questa metrica si applica solo agli spazi chiave multiregione e misura il tempo impiegato per la replica o deletes da una tabella di replica a un'altra tabella di replica updates in uno spazio di chiavi multiregionale. inserts</p> <p>Unità: Millisecond</p> <p>Dimensioni: TableName, ReceivingRegion</p> <p>Statistiche valide:</p> <ul style="list-style-type: none"><li>• Average</li><li>• Maximum</li><li>• Minimum</li></ul> |

| Parametro                 | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|---------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ReturnedItemCountBySelect | <p>Il numero di righe restituite dalle query a più righe durante il periodo di tempo specificato. <code>SELECT</code> Le <code>SELECT</code> interrogazioni a più righe sono interrogazioni che non contengono la chiave primaria completa, ad esempio scansioni complete di tabelle e interrogazioni con intervalli.</p> <p>Il numero di righe restituite non è necessariamente uguale al numero di righe valutate. Ad esempio, supponiamo di aver richiesto un valore <code>SELECT * ALLOW FILTERING</code> in una tabella contenente 100 righe, ma di aver specificato una <code>WHERE</code> clausola che restringe i risultati in modo da restituire solo 15 righe. In questo caso, la risposta da <code>SELECT</code> conterrebbe una <code>ScanCount</code> riga restituita da 100 e una <code>Count</code> da 15.</p> <p>Unità: Count</p> <p>Dimensioni: Keyspace, TableName, Operation</p> <p>Statistiche valide:</p> <ul style="list-style-type: none"><li>• Minimum</li><li>• Maximum</li><li>• Average</li><li>• SampleCount</li><li>• Sum</li></ul> |

| Parametro                                               | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|---------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>StoragePartitionThroughputCapacityExceeded</code> | <p>Richieste a una partizione di storage Amazon Keyspaces che superano la capacità di throughput della partizione. Le partizioni di storage Amazon Keyspaces possono supportare fino a 1000 WCU/WRU al secondo e 3000 RCU/RRU al secondo. Ti consigliamo di rivedere il tuo modello di dati per distribuire il read/write traffico su più partizioni per mitigare queste eccezioni.</p> <div> <b>Note</b><p>Le partizioni logiche di Amazon Keyspaces possono estendersi su più partizioni di storage e hanno dimensioni praticamente illimitate.</p></div> <p>Unità: Count</p> <p>Dimensioni: Keyspace, TableName, Operation</p> <p>Statistiche valide:</p> <ul style="list-style-type: none"><li>• SampleCount</li><li>• Sum</li></ul> |
| <code>SuccessfulRequestCount</code>                     | <p>Il numero di richieste riuscite elaborate nel periodo di tempo specificato.</p> <p>Unità: Count</p> <p>Dimensioni: Keyspace, TableName, Operation</p> <p>Statistiche valide:</p> <ul style="list-style-type: none"><li>• SampleCount</li></ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

| Parametro                | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|--------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SuccessfulRequestLatency | <p>Le richieste riuscite ad Amazon Keyspaces durante il periodo di tempo specificato. <code>SuccessfulRequestLatency</code> può fornire due diversi tipi di informazioni:</p> <ul style="list-style-type: none"><li>• Il tempo trascorso per le richieste riuscite (<code>Minimum</code>, <code>Maximum</code>, <code>Sum</code> oppure <code>Average</code>).</li><li>• Il numero di richieste eseguite correttamente (<code>SampleCount</code> ).</li></ul> <p><code>SuccessfulRequestLatency</code> riflette l'attività solo all'interno di Amazon Keyspaces e non tiene conto della latenza di rete o dell'attività lato client.</p> <p>Unità: <code>Milliseconds</code></p> <p>Dimensioni: <code>Keyspace</code>, <code>TableName</code>, <code>Operation</code></p> <p>Statistiche valide:</p> <ul style="list-style-type: none"><li>• <code>Minimum</code></li><li>• <code>Maximum</code></li><li>• <code>Average</code></li><li>• <code>SampleCount</code></li></ul> |

| Parametro                   | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SystemErrors                | <p>Le richieste ad Amazon Keyspaces che generano un messaggio <code>ServerError</code> durante il periodo di tempo specificato. A <code>ServerError</code> di solito indica un errore interno del servizio.</p> <p>Unità: Count</p> <p>Dimensioni: Keyspace, TableName, Operation</p> <p>Statistiche valide:</p> <ul style="list-style-type: none"> <li>• Sum</li> <li>• SampleCount</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| SystemReconciliationDeletes | <p>Le unità utilizzate per eliminare i dati archiviati quando i timestamp lato client sono abilitati. <code>SystemReconciliationDelete</code> Ciascuna di esse offre una capacità sufficiente per eliminare o aggiornare fino a 1 KB di dati per riga. Ad esempio, per aggiornare una riga che memorizza 2,5 KB di dati ed eliminare contemporaneamente una o più colonne all'interno della riga sono necessarie 3 <code>SystemReconciliationDeletes</code>. Oppure, per eliminare un'intera riga che contiene 3,5 KB di dati sono necessari 4 <code>SystemReconciliationDeletes</code>.</p> <p>Unità: Count</p> <p>Dimensioni: Keyspace, TableName</p> <p>Statistiche valide:</p> <ul style="list-style-type: none"> <li>• Sum— Il numero totale di prodotti <code>SystemReconciliationDeletes</code> consumati in un periodo di tempo.</li> </ul> |

| Parametro  | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TTLDeletes | <p>Le unità utilizzate per eliminare o aggiornare i dati di seguito utilizzando Time to Live (TTL). TTLDelete</p> <p>Ciascuna offre una capacità sufficiente per eliminare o aggiornare fino a 1 KB di dati per riga. Ad esempio, per aggiornare una riga che memorizza 2,5 KB di dati ed eliminare contemporaneamente una o più colonne all'interno della riga sono necessarie 3 eliminazioni TTL. In alternativa, per eliminare un'intera riga che contiene 3,5 KB di dati sono necessarie 4 eliminazioni TTL.</p> <p>Unità: Count</p> <p>Dimensioni: Keyspace, TableName</p> <p>Statistiche valide:</p> <ul style="list-style-type: none"><li>• Sum— Il numero totale di prodotti TTLDeletes consumati in un periodo di tempo.</li></ul> |

| Parametro           | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| UserErrors          | <p>Richieste ad Amazon Keyspaces che generano un <code>InvalidRequest</code> errore durante il periodo di tempo specificato.</p> <p><code>InvalidRequest</code> gli errori di solito rappresentano un errore lato client, ad esempio una combinazione non valida di parametri, un tentativo di aggiornare una tabella inesistente o una firma di richiesta errata.</p> <p><code>UserErrors</code> rappresentano l'aggregato di richieste non valide per la corrente e la corrente. Regione AWS Account AWS</p> <p>Unità: Count</p> <p>Dimensioni: Keyspace, TableName, Operation</p> <p>Statistiche valide:</p> <ul style="list-style-type: none"><li>• Sum</li><li>• SampleCount</li></ul> |
| WriteThrottleEvents | <p>Richieste ad Amazon Keyspaces che superano la capacità di scrittura assegnata per una tabella o quote a livello di account, quote di richiesta per connessione o quote a livello di partizione.</p> <p>Unità: Count</p> <p>Dimensioni: Keyspace, TableName, Operation</p> <p>Statistiche valide:</p> <ul style="list-style-type: none"><li>• SampleCount</li><li>• Sum</li></ul>                                                                                                                                                                                                                                                                                                         |



## Dimensioni per i parametri di Amazon Keyspaces

Le metriche per Amazon Keyspaces sono qualificate in base ai valori dell'account, del nome della tabella o dell'operazione. Puoi utilizzare la CloudWatch console per recuperare i dati di Amazon Keyspaces lungo una qualsiasi delle dimensioni nella tabella seguente.

| Dimensione | Descrizione                                                                                                                                                                                                                                                                       |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Keyspace   | Questa dimensione limita i dati a uno spazio chiave specifico. Questo valore può essere qualsiasi spazio chiave nella regione corrente e nella regione. Account AWS                                                                                                               |
| Operation  | Questa dimensione limita i dati a una delle operazioni CQL di Amazon Keyspaces, ad esempio le INSERT operazioni o. SELECT                                                                                                                                                         |
| TableName  | Questa dimensione limita i dati a una tabella specifica. Questo valore può essere qualsiasi nome di tabella nella regione corrente e nella regione corrente. Account AWS Se il nome della tabella non è univoco all'interno dell'account, è necessario specificare ancheKeyspace. |

## Metriche per Amazon Keyspaces change data capture (CDC)

Nel tuo CloudWatch account sono disponibili le seguenti metriche che monitorano le chiamate API all'endpoint Amazon Keyspaces change data capture (CDC).

| Parametro            | Descrizione                                                                                                                                                                                                                                                                     |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ReturnedRecordsCount | <p>Il numero di record di stream restituiti dalle GetRecords operazioni durante il periodo di tempo specificato.</p> <p>Unità: Count</p> <p>Dimensioni: Keyspace, TableName, Operation</p> <p>Statistiche valide:</p> <ul style="list-style-type: none"><li>• Minimum</li></ul> |

| Parametro                             | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|---------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                       | <ul style="list-style-type: none"> <li>• <code>Maximum</code></li> <li>• <code>Average</code></li> <li>• <code>SampleCount</code></li> <li>• <code>Sum</code></li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <code>SuccessfulRequestLatency</code> | <p>Le richieste riuscite ad Amazon Keyspaces durante il periodo di tempo specificato. <code>SuccessfulRequestLatency</code> può fornire due diversi tipi di informazioni:</p> <ul style="list-style-type: none"> <li>• Il tempo trascorso per le richieste riuscite (<code>Minimum</code>, <code>Maximum</code>, <code>Sum</code> oppure <code>Average</code>).</li> <li>• Il numero di richieste eseguite correttamente (<code>SampleCount</code> ).</li> </ul> <p><code>SuccessfulRequestLatency</code> riflette l'attività solo all'interno di Amazon Keyspaces CDC APIs e non tiene conto della latenza di rete o dell'attività lato client.</p> <p>Unità: <code>Milliseconds</code></p> <p>Dimensioni: <code>Keyspace</code>, <code>TableName</code>, <code>Operation</code></p> <p>Statistiche valide:</p> <ul style="list-style-type: none"> <li>• <code>Minimum</code></li> <li>• <code>Maximum</code></li> <li>• <code>Average</code></li> <li>• <code>SampleCount</code></li> </ul> |

| Parametro    | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SystemErrors | <p>Le richieste ad Amazon Keyspaces CDC che generano un codice di HTTP 500 stato durante il periodo di tempo specificato. Un codice HTTP 500 di stato di solito indica un errore interno del servizio.</p> <p>Unità: Count</p> <p>Dimensioni: Keyspace, TableName, Operation</p> <p>Statistiche valide:</p> <ul style="list-style-type: none"><li>• Sum</li><li>• SampleCount</li></ul>                                                                             |
| UserErrors   | <p>Richieste ad Amazon Keyspaces CDC che generano un codice di HTTP 400 stato durante il periodo di tempo specificato.</p> <p>UserErrors rappresenta l'aggregato di HTTP 400 errori per Amazon Keyspaces CDC nella versione corrente e in quella Regione AWS corrente. Account AWS</p> <p>Unità: Count</p> <p>Dimensioni: Keyspace, TableName, Operation</p> <p>Statistiche valide:</p> <ul style="list-style-type: none"><li>• Sum</li><li>• SampleCount</li></ul> |

## Le dimensioni di Amazon Keyspaces modificano i parametri di acquisizione dei dati (CDC)

Le metriche per Amazon Keyspaces sono qualificate in base ai valori dell'account, del nome della tabella o dell'operazione. Puoi utilizzare la CloudWatch console per recuperare i dati di Amazon Keyspaces lungo una qualsiasi delle dimensioni nella tabella seguente.

| Dimensione               | Descrizione                                                                                                                                                                                                                                                                                      |
|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>StreamLabel</code> | Questa dimensione limita i dati a uno specifico flusso di acquisizione dei dati di modifica (CDC) quando <code>KeyspaceName</code> e <code>TableName</code> quando sono specificati.                                                                                                             |
| <code>Keyspace</code>    | Questa dimensione limita i dati a uno spazio chiave specifico. Questo valore può essere qualsiasi spazio chiave nella regione corrente e nella regione corrente. Account AWS                                                                                                                     |
| <code>TableName</code>   | Questa dimensione limita i dati a una tabella specifica. Questo valore può essere qualsiasi nome di tabella nella regione corrente e nella regione corrente Account AWS. Se il nome della tabella non è univoco all'interno dell'account, è necessario specificare anche <code>Keyspace</code> . |
| <code>Operation</code>   | Questa dimensione limita i dati a una delle seguenti operazioni dell'API Amazon Keyspaces Streams: <code>GetRecords</code> , <code>GetStream</code> , <code>GetShardIterator</code> e <code>ListStreams</code> .                                                                                 |

## Visualizzazione dei parametri di Amazon Keyspaces in CloudWatch

Quando interagisci con Amazon Keyspaces, invia le seguenti metriche e dimensioni ad Amazon CloudWatch. Tutte le metriche vengono aggregate e riportate ogni minuto. Puoi utilizzare le seguenti procedure per visualizzare i parametri per Amazon Keyspaces.

Per visualizzare le metriche utilizzando la console CloudWatch

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. Se necessario, modifica Regione AWS. Nella barra di navigazione, scegli la regione in cui risiedono AWS le tue risorse. Per ulteriori informazioni, consulta [Endpoint del servizio AWS](#).
3. Nel riquadro di navigazione, seleziona Parametri.
4. Nella scheda Tutte le metriche, scegli AWS/Cassandra.

Per visualizzare le metriche utilizzando la CLI AWS

- Al prompt dei comandi utilizza il comando seguente.

```
aws cloudwatch list-metrics --namespace "AWS/Cassandra"
```

## Creazione di CloudWatch allarmi per monitorare Amazon Keyspaces

Puoi creare un CloudWatch allarme Amazon per Amazon Keyspaces che invia un messaggio Amazon Simple Notification Service (Amazon SNS) quando l'allarme cambia stato. Un allarme monitora un singolo parametro per un periodo di tempo specificato. L'allarme esegue una o più operazioni basate sul valore del parametro relativo a una soglia prestabilita per un certo numero di periodi. L'azione è una notifica inviata a un argomento di Amazon SNS o a una policy di Application Auto Scaling.

Quando utilizzi Amazon Keyspaces in modalità provisioning con Application Auto Scaling, il servizio crea due coppie CloudWatch di allarmi per tuo conto. Ogni coppia rappresenta i limiti superiore e inferiore per le impostazioni di throughput assegnate e consumate. Questi CloudWatch allarmi vengono attivati quando l'utilizzo effettivo della tabella si discosta dall'utilizzo previsto per un periodo di tempo prolungato. Per ulteriori informazioni sugli CloudWatch allarmi creati da Application Auto Scaling, consulta [the section called “Come funziona il ridimensionamento automatico di Amazon Keyspaces”](#)

Gli allarmi richiamano azioni solo per cambiamenti di stato sostenuti. CloudWatch gli allarmi non richiamano azioni semplicemente perché si trovano in uno stato particolare. Lo stato deve essere cambiato e restare costante per un numero specificato di periodi.

Per ulteriori informazioni sulla creazione di CloudWatch allarmi, consulta [Using Amazon CloudWatch alarms](#) nella Amazon CloudWatch User Guide.

# Registrazione delle chiamate API Amazon Keyspaces con AWS CloudTrail

Amazon Keyspaces è integrato con AWS CloudTrail un servizio che fornisce una registrazione delle azioni intraprese da un utente, un ruolo o un AWS servizio in Amazon Keyspaces. CloudTrail acquisisce le chiamate API Data Definition Language (DDL) e le chiamate API Data Manipulation Language (DML) per Amazon Keyspaces come eventi. Le chiamate acquisite includono chiamate dalla console Amazon Keyspaces e chiamate programmatiche alle operazioni dell'API Amazon Keyspaces.

Se crei un trail, puoi abilitare la distribuzione continua di CloudTrail eventi a un bucket Amazon Simple Storage Service (Amazon S3), inclusi gli eventi per Amazon Keyspaces.

Se non configuri un percorso, puoi comunque visualizzare gli eventi supportati più recenti sulla CloudTrail console nella cronologia degli eventi. Utilizzando le informazioni raccolte da CloudTrail, puoi determinare la richiesta che è stata effettuata ad Amazon Keyspaces, l'indirizzo IP da cui è stata effettuata la richiesta, chi ha effettuato la richiesta, quando è stata effettuata e ulteriori dettagli.

Per ulteriori informazioni CloudTrail, consulta la [Guida per l'AWS CloudTrail utente](#).

## Argomenti

- [Configurazione delle voci dei file di log di Amazon Keyspaces in CloudTrail](#)
- [Informazioni su Amazon Keyspaces Data Definition Language \(DDL\) in CloudTrail](#)
- [Informazioni su Amazon Keyspaces Data Manipulation Language \(DML\) in CloudTrail](#)
- [Comprendere le voci dei file di log di Amazon Keyspaces](#)

## Configurazione delle voci dei file di log di Amazon Keyspaces in CloudTrail

Ogni azione dell'API Amazon Keyspaces registrata CloudTrail include parametri di richiesta espressi nel linguaggio di query CQL. Per ulteriori informazioni, consulta [Riferimento al linguaggio CQL](#).

Puoi visualizzare, cercare e scaricare gli eventi recenti in Account AWS. Per ulteriori informazioni, consulta [Visualizzazione degli eventi con CloudTrail la cronologia degli eventi](#).

Per una registrazione continua degli eventi nel tuo Account AWS, compresi gli eventi per Amazon Keyspaces, 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 nella Guida per l'utente di AWS CloudTrail :

- [Panoramica della creazione di un percorso](#)
- [CloudTrail servizi e integrazioni supportati](#)
- [Configurazione delle notifiche Amazon SNS per CloudTrail](#)
- [Ricezione di file di CloudTrail registro da più regioni](#)
- [Ricezione di file di CloudTrail registro da più account](#)

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 credenziali utente root o AWS Identity and Access Management (IAM).
- 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](#).

## Informazioni su Amazon Keyspaces Data Definition Language (DDL) in CloudTrail

CloudTrail è abilitato sul tuo account al Account AWS momento della creazione dell'account. Quando si verifica un'attività DDL in Amazon Keyspaces, tale attività viene automaticamente registrata come CloudTrail evento insieme AWS ad altri eventi di servizio nella cronologia degli eventi. La tabella seguente mostra le istruzioni DDL registrate per Amazon Keyspaces.

| CloudTrail<br><b>eventName</b> | Dichiarazione | Azione CQL      | AWS Azione SDK |
|--------------------------------|---------------|-----------------|----------------|
| CreateKeyspace                 | DDL           | CREATE KEYSPACE | CreateKeyspace |
| AlterKeyspace                  | DDL           | ALTER KEYSPACE  | UpdateKeyspace |

| CloudTrail eventName | Dichiarazione | Azione CQL                        | AWS Azione SDK                                  |
|----------------------|---------------|-----------------------------------|-------------------------------------------------|
| DropKeyspace         | DDL           | DROP KEYSPACE                     | DeleteKeyspace                                  |
| CreateTable          | DDL           | CREATE TABLE                      | CreateTable                                     |
| DropTable            | DDL           | DROP TABLE                        | DeleteTable                                     |
| AlterTable           | DDL           | ALTER TABLE                       | UpdateTable ,<br>TagResource ,<br>UntagResource |
| CreateUdt            | DDL           | CREATE TYPE                       | CreateType                                      |
| DropUdt              | DDL           | DROP TYPE                         | DeleteType                                      |
| GetStream            | DDL           | nessuna azione CQL<br>disponibile | GetStream                                       |
| ListStreams          | DDL           | nessuna azione CQL<br>disponibile | ListStreams                                     |

## Informazioni su Amazon Keyspaces Data Manipulation Language (DML) in CloudTrail

Per abilitare la registrazione delle istruzioni DML di Amazon Keyspaces CloudTrail con, devi prima abilitare la registrazione dell'attività dell'API del piano dati. CloudTrail Puoi iniziare a registrare gli eventi DML di Amazon Keyspaces in percorsi nuovi o esistenti scegliendo di registrare l'attività per la tabella Cassandra del tipo di evento dati utilizzando CloudTrail la console o impostando il valore `AWS::Cassandra::Table` utilizzando `resources.type` la AWS CLI o le operazioni API. CloudTrail Per ulteriori informazioni, consulta [Registrazione degli eventi di dati](#).

Per ulteriori informazioni e un esempio che mostra come creare allarmi per eventi relativi ai dati, consulta il seguente post sul blog AWS Database [Using DML auditing for Amazon Keyspaces \(per Apache Cassandra\)](#).

La tabella seguente mostra gli eventi relativi ai dati per cui vengono registrati. CloudTrail Cassandra table



| CloudTrail<br>eventName | Dichiarazione | Azione CQL                        | AWS Azioni SDK                                                                                         |
|-------------------------|---------------|-----------------------------------|--------------------------------------------------------------------------------------------------------|
| Select                  | DML           | SELECT                            | GetKeyspace ,<br>GetTable, GetType,<br>ListKeyspaces , ListTables , ListTypes<br>, ListTagsForResource |
| Insert                  | DML           | INSERT                            | nessuna azione AWS<br>SDK disponibile                                                                  |
| Aggiornamento           | DML           | UPDATE                            | nessuna azione AWS<br>SDK disponibile                                                                  |
| Eliminazione            | DML           | DELETE                            | nessuna azione AWS<br>SDK disponibile                                                                  |
| GetShardIterator        | DML           | nessuna azione CQL<br>disponibile | GetShardIterator                                                                                       |
| GetRecords              | DML           | nessuna azione CQL<br>disponibile | GetRecords                                                                                             |

## Comprendere le voci dei file di log di Amazon Keyspaces

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 mostra le azioni `CreateKeyspace`, `DropKeyspace`, `CreateTable`, e `DropTable`:

```
{
```

```

"Records": [
 {
 "eventVersion": "1.05",
 "userIdentity": {
 "type": "AssumedRole",
 "principalId": "AKIAIOSFODNN7EXAMPLE:alice",
 "arn": "arn:aws:sts::111122223333:assumed-role/users/alice",
 "accountId": "111122223333",
 "sessionContext": {
 "sessionIssuer": {
 "type": "Role",
 "principalId": "AKIAIOSFODNN7EXAMPLE",
 "arn": "arn:aws:iam::111122223333:role/Admin",
 "accountId": "111122223333",
 "userName": "Admin"
 },
 "webIdFederationData": {},
 "attributes": {
 "mfaAuthenticated": "false",
 "creationDate": "2020-01-15T18:47:56Z"
 }
 }
 },
 "eventTime": "2020-01-15T18:53:04Z",
 "eventSource": "cassandra.amazonaws.com",
 "eventName": "CreateKeyspace",
 "awsRegion": "us-east-1",
 "sourceIPAddress": "10.24.34.01",
 "userAgent": "Cassandra Client/ProtocolV4",
 "requestParameters": {
 "rawQuery": "\n\tCREATE KEYSPACE \"mykeyspace\"\n\tWITH\n\t\tREPLICATION =
{'class': 'SingleRegionStrategy'}\n\t\t",
 "keyspaceName": "mykeyspace"
 },
 "responseElements": null,
 "requestID": "bfa3e75d-bf4d-4fc0-be5e-89d15850eb41",
 "eventID": "d25beae8-f611-4229-877a-921557a07bb9",
 "readOnly": false,
 "resources": [
 {
 "accountId": "111122223333",
 "type": "AWS::Cassandra::Keyspace",
 "ARN": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/mykeyspace/"
 }
]
 }
]

```

```

],
"eventType": "AwsApiCall",
"apiVersion": "3.4.4",
"recipientAccountId": "111122223333",
"managementEvent": true,
"eventCategory": "Management",
"tlsDetails": {
 "tlsVersion": "TLSv1.2",
 "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
 "clientProvidedHostHeader": "cassandra.us-east-1.amazonaws.com"
},
{
 "eventVersion": "1.05",
 "userIdentity": {
 "type": "AssumedRole",
 "principalId": "AKIAIOSFODNN7EXAMPLE:alice",
 "arn": "arn:aws:sts::111122223333:assumed-role/users/alice",
 "accountId": "111122223333",
 "sessionContext": {
 "sessionIssuer": {
 "type": "Role",
 "principalId": "AKIAIOSFODNN7EXAMPLE",
 "arn": "arn:aws:iam::111122223333:role/Admin",
 "accountId": "111122223333",
 "userName": "Admin"
 },
 "webIdFederationData": {},
 "attributes": {
 "mfaAuthenticated": "false",
 "creationDate": "2020-01-15T18:47:56Z"
 }
 }
 },
 "eventTime": "2020-01-15T19:28:39Z",
 "eventSource": "cassandra.amazonaws.com",
 "eventName": "DropKeyspace",
 "awsRegion": "us-east-1",
 "sourceIPAddress": "10.24.34.01",
 "userAgent": "Cassandra Client/ProtocolV4",
 "requestParameters": {
 "rawQuery": "DROP KEYSPACE \"mykeyspace\"",
 "keyspaceName": "mykeyspace"
 },
 "responseElements": null,

```

```

 "requestID": "66f3d86a-56ae-4c29-b46f-abcd489ed86b",
 "eventID": "e5aebac-e1dd-41e3-a515-84fe6aaabd7b",
 "readOnly": false,
 "resources": [
 {
 "accountId": "111122223333",
 "type": "AWS::Cassandra::Keyspace",
 "ARN": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/mykeyspace/"
 }
],
 "eventType": "AwsApiCall",
 "apiVersion": "3.4.4",
 "recipientAccountId": "111122223333",
 "managementEvent": true,
 "eventCategory": "Management",
 "tlsDetails": {
 "tlsVersion": "TLSv1.2",
 "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
 "clientProvidedHostHeader": "cassandra.us-east-1.amazonaws.com"
 },
 {
 "eventVersion": "1.05",
 "userIdentity": {
 "type": "AssumedRole",
 "principalId": "AKIAIOSFODNN7EXAMPLE:alice",
 "arn": "arn:aws:sts::111122223333:assumed-role/users/alice",
 "accountId": "111122223333",
 "sessionContext": {
 "sessionIssuer": {
 "type": "Role",
 "principalId": "AKIAIOSFODNN7EXAMPLE",
 "arn": "arn:aws:iam::111122223333:role/Admin",
 "accountId": "111122223333",
 "userName": "Admin"
 },
 "webIdFederationData": {},
 "attributes": {
 "mfaAuthenticated": "false",
 "creationDate": "2020-01-15T18:47:56Z"
 }
 }
 },
 "eventTime": "2020-01-15T18:55:24Z",
 "eventSource": "cassandra.amazonaws.com",

```

```

 "eventName": "CreateTable",
 "awsRegion": "us-east-1",
 "sourceIPAddress": "10.24.34.01",
 "userAgent": "Cassandra Client/ProtocolV4",
 "requestParameters": {
 "rawQuery": "\n\tCREATE TABLE \"mykeyspace\".\"mytable\"(\n\t\t\"ID\" int,\n\t\t\"username\" text,\n\t\t\"email\" text,\n\t\t\"post_type\" text,\n\t\tPRIMARY\n\t\tKEY((\"ID\", \"username\", \"email\")))",
 "keyspaceName": "mykeyspace",
 "tableName": "mytable"
 },
 "responseElements": null,
 "requestID": "5f845963-70ea-4988-8a7a-2e66d061aacb",
 "eventID": "fe0dbd2b-7b34-4675-a30c-740f9d8d73f9",
 "readOnly": false,
 "resources": [
 {
 "accountId": "111122223333",
 "type": "AWS::Cassandra::Table",
 "ARN": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/mykeyspace/table/mytable"
 }
],
 "eventType": "AwsApiCall",
 "apiVersion": "3.4.4",
 "recipientAccountId": "111122223333",
 "managementEvent": true,
 "eventCategory": "Management",
 "tlsDetails": {
 "tlsVersion": "TLSv1.2",
 "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
 "clientProvidedHostHeader": "cassandra.us-east-1.amazonaws.com"
 },
 {
 "eventVersion": "1.05",
 "userIdentity": {
 "type": "AssumedRole",
 "principalId": "AKIAIOSFODNN7EXAMPLE:alice",
 "arn": "arn:aws:sts::111122223333:assumed-role/users/alice",
 "accountId": "111122223333",
 "sessionContext": {
 "sessionIssuer": {
 "type": "Role",
 "principalId": "AKIAIOSFODNN7EXAMPLE",

```

```

 "arn": "arn:aws:iam::111122223333:role/Admin",
 "accountId": "111122223333",
 "userName": "Admin"
 },
 "webIdFederationData": {},
 "attributes": {
 "mfaAuthenticated": "false",
 "creationDate": "2020-01-15T18:47:56Z"
 }
}
},
"eventTime": "2020-01-15T19:27:59Z",
"eventSource": "cassandra.amazonaws.com",
"eventName": "DropTable",
"awsRegion": "us-east-1",
"sourceIPAddress": "10.24.34.01",
"userAgent": "Cassandra Client/ProtocolV4",
"requestParameters": {
 "rawQuery": "DROP TABLE \"mykeyspace\".\"mytable\"",
 "keyspaceName": "mykeyspace",
 "tableName": "mytable"
},
"responseElements": null,
"requestID": "025501b0-3582-437e-9d18-8939e9ef262f",
"eventID": "1a5cbcdc-4e38-4889-8475-3eab98de0ffd",
"readOnly": false,
"resources": [
 {
 "accountId": "111122223333",
 "type": "AWS::Cassandra::Table",
 "ARN": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/mykeyspace/table/
mytable"
 }
],
"eventType": "AwsApiCall",
"apiVersion": "3.4.4",
"recipientAccountId": "111122223333",
"managementEvent": true,
"eventCategory": "Management",
"tlsDetails": {
 "tlsVersion": "TLSv1.2",
 "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
 "clientProvidedHostHeader": "cassandra.us-east-1.amazonaws.com"
}
}

```

```
]
}
```

Il seguente file di registro mostra un esempio di SELECT istruzione.

```
{
 "eventVersion": "1.09",
 "userIdentity": {
 "type": "IAMUser",
 "principalId": "AKIAIOSFODNN7EXAMPLE",
 "arn": "arn:aws:iam::111122223333:user/alice",
 "accountId": "111122223333",
 "userName": "alice"
 },
 "eventTime": "2023-11-17T10:38:04Z",
 "eventSource": "cassandra.amazonaws.com",
 "eventName": "Select",
 "awsRegion": "us-east-1",
 "sourceIPAddress": "10.24.34.01",
 "userAgent": "Cassandra Client/ProtocolV4",
 "requestParameters": {
 "keyspaceName": "my_keyspace",
 "tableName": "my_table",
 "conditions": [
 "pk = **(Redacted)",
 "ck < 3**(Redacted)0",
 "region = 't**(Redacted)t'"
],
 "select": [
 "pk",
 "ck",
 "region"
],
 "allowFiltering": true
 },
 "responseElements": null,
 "requestID": "6d83bbf0-a3d0-4d49-b1d9-e31779a28628",
 "eventID": "e00552d3-34e9-4092-931a-912c4e08ba17",
 "readOnly": true,
 "resources": [
 {
 "accountId": "111122223333",
 "type": "AWS::Cassandra::Table",
```

```

 "ARN": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/my_keyspace/
table/my_table"
 },
 "eventType": "AwsApiCall",
 "apiVersion": "3.4.4",
 "managementEvent": false,
 "recipientAccountId": "111122223333",
 "eventCategory": "Data",
 "tlsDetails": {
 "tlsVersion": "TLSv1.3",
 "cipherSuite": "TLS_AES_128_GCM_SHA256",
 "clientProvidedHostHeader": "cassandra.us-east-1.amazonaws.com"
 }
}

```

Il seguente file di registro mostra un esempio di INSERT istruzione.

```

{
 "eventVersion": "1.09",
 "userIdentity": {
 "type": "IAMUser",
 "principalId": "AKIAIOSFODNN7EXAMPLE",
 "arn": "arn:aws:iam::111122223333:user/alice",
 "accountId": "111122223333",
 "userName": "alice"
 },
 "eventTime": "2023-12-01T22:11:43Z",
 "eventSource": "cassandra.amazonaws.com",
 "eventName": "Insert",
 "awsRegion": "us-east-1",
 "sourceIPAddress": "10.24.34.01",
 "userAgent": "Cassandra Client/ProtocolV4",
 "requestParameters": {
 "keyspaceName": "my_keyspace",
 "tableName": "my_table",
 "primaryKeys": {
 "pk": "**(Redacted)",
 "ck": "1**(Redacted)8"
 },
 "columnNames": [
 "pk",
 "ck",

```



```

 "region"
],
 "updateParameters": {
 "TTL": "2**(Redacted)0"
 }
 }
 },
 "responseElements": null,
 "requestID": "edf8af47-2f87-4432-864d-a960ac35e471",
 "eventID": "81b56a1c-9bdd-4c92-bb8e-92776b5a3bf1",
 "readOnly": false,
 "resources": [
 {
 "accountId": "111122223333",
 "type": "AWS::Cassandra::Table",
 "ARN": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/my_keyspace/table/
my_table"
 }
],
 "eventType": "AwsApiCall",
 "apiVersion": "3.4.4",
 "managementEvent": false,
 "recipientAccountId": "111122223333",
 "eventCategory": "Data",
 "tlsDetails": {
 "tlsVersion": "TLSv1.3",
 "cipherSuite": "TLS_AES_128_GCM_SHA256",
 "clientProvidedHostHeader": "cassandra.us-east-1.amazonaws.com"
 }
}

```

Il seguente file di registro mostra un esempio di UPDATE istruzione.

```

{
 "eventVersion": "1.09",
 "userIdentity": {
 "type": "IAMUser",
 "principalId": "AKIAIOSFODNN7EXAMPLE",
 "arn": "arn:aws:iam::111122223333:user/alice",
 "accountId": "111122223333",
 "userName": "alice"
 },
 "eventTime": "2023-12-01T22:11:43Z",

```

```

 "eventSource": "cassandra.amazonaws.com",
 "eventName": "Update",
 "awsRegion": "us-east-1",
 "sourceIPAddress": "10.24.34.01",
 "userAgent": "Cassandra Client/ProtocolV4",
 "requestParameters": {
 "keyspaceName": "my_keyspace",
 "tableName": "my_table",
 "primaryKeys": {
 "pk": "'t**(Redacted)t'",
 "ck": "'s**(Redacted)g'"
 },
 "assignmentColumnNames": [
 "nonkey"
],
 "conditions": [
 "nonkey < 1**(Redacted)7"
]
 },
 "responseElements": null,
 "requestID": "edf8af47-2f87-4432-864d-a960ac35e471",
 "eventID": "81b56a1c-9bdd-4c92-bb8e-92776b5a3bf1",
 "readOnly": false,
 "resources": [
 {
 "accountId": "111122223333",
 "type": "AWS::Cassandra::Table",
 "ARN": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/my_keyspace/table/
my_table"
 }
],
 "eventType": "AwsApiCall",
 "apiVersion": "3.4.4",
 "managementEvent": false,
 "recipientAccountId": "111122223333",
 "eventCategory": "Data",
 "tlsDetails": {
 "tlsVersion": "TLSv1.3",
 "cipherSuite": "TLS_AES_128_GCM_SHA256",
 "clientProvidedHostHeader": "cassandra.us-east-1.amazonaws.com"
 }
 }
}

```

Il seguente file di registro mostra un esempio di DELETE istruzione.

```
{
 "eventVersion": "1.09",
 "userIdentity": {
 "type": "IAMUser",
 "principalId": "AKIAIOSFODNN7EXAMPLE",
 "arn": "arn:aws:iam::111122223333:user/alice",
 "accountId": "111122223333",
 "userName": "alice",
 },
 "eventTime": "2023-10-23T13:59:05Z",
 "eventSource": "cassandra.amazonaws.com",
 "eventName": "Delete",
 "awsRegion": "us-east-1",
 "sourceIPAddress": "10.24.34.01",
 "userAgent": "Cassandra Client/ProtocolV4",
 "requestParameters": {
 "keyspaceName": "my_keyspace",
 "tableName": "my_table",
 "primaryKeys": {
 "pk": "**(Redacted)",
 "ck": "**(Redacted)"
 },
 "conditions": [],
 "deleteColumnNames": [
 "m",
 "s"
],
 "updateParameters": {}
 },
 "responseElements": null,
 "requestID": "3d45e63b-c0c8-48e2-bc64-31afc5b4f49d",
 "eventID": "499da055-c642-4762-8775-d91757f06512",
 "readOnly": false,
 "resources": [
 {
 "accountId": "111122223333",
 "type": "AWS::Cassandra::Table",
 "ARN": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/my_keyspace/table/my_table"
 }
],
 "eventType": "AwsApiCall",
}
```

```
"apiVersion": "3.4.4",
"managementEvent": false,
"recipientAccountId": "111122223333",
"eventCategory": "Data",
"tlsDetails": {
 "tlsVersion": "TLSv1.3",
 "cipherSuite": "TLS_AES_128_GCM_SHA256",
 "clientProvidedHostHeader": "cassandra.us-east-1.amazonaws.com"
}
}
```

# Sicurezza in Amazon Keyspaces (per Apache Cassandra)

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 aspetto come sicurezza del cloud e sicurezza nel cloud:

- **Sicurezza del cloud:** AWS è responsabile della protezione dell'infrastruttura che gestisce AWS i servizi in Cloud AWS. AWS fornisce inoltre servizi che è possibile utilizzare in modo sicuro. L'efficacia della nostra sicurezza è regolarmente testata e verificata da revisori di terze parti come parte dei [programmi di conformitàAWS](#). Per maggiori informazioni sui programmi di conformità che si applicano ad Amazon Keyspaces, consulta la sezione [AWS Servizi rientranti nell'ambito del programma di conformità](#).
- **Sicurezza nel cloud:** la tua responsabilità è determinata dal AWS servizio che utilizzi. L'utente è anche responsabile per altri fattori, tra cui la riservatezza dei dati, i requisiti dell'azienda e leggi e normative applicabili.

Questa documentazione ti aiuterà a capire come applicare il modello di responsabilità condivisa quando usi Amazon Keyspaces. I seguenti argomenti mostrano come configurare Amazon Keyspaces per soddisfare i tuoi obiettivi di sicurezza e conformità. Imparerai anche come utilizzare altri AWS servizi che possono aiutarti a monitorare e proteggere le tue risorse Amazon Keyspaces.

## Argomenti

- [Protezione dei dati in Amazon Keyspaces](#)
- [AWS Identity and Access Management per Amazon Keyspaces](#)
- [Convalida della conformità per Amazon Keyspaces \(per Apache Cassandra\)](#)
- [Resilienza e disaster recovery in Amazon Keyspaces](#)
- [Sicurezza dell'infrastruttura in Amazon Keyspaces](#)
- [Analisi della configurazione e delle vulnerabilità per Amazon Keyspaces](#)
- [Best practice di sicurezza per Amazon Keyspaces](#)

# Protezione dei dati in Amazon Keyspaces

Il modello di [responsabilità AWS condivisa modello](#) di si applica alla protezione dei dati in Amazon Keyspaces (per Apache Cassandra). Come descritto in questo modello, AWS è responsabile della protezione dell'infrastruttura globale che gestisce tutti i Cloud AWS. L'utente è responsabile del controllo dei contenuti ospitati su questa infrastruttura. L'utente è inoltre responsabile della configurazione della protezione e delle attività di gestione per i Servizi AWS utilizzati. Per ulteriori informazioni sulla privacy dei dati, vedi le [Domande frequenti sulla privacy dei dati](#). Per informazioni sulla protezione dei dati in Europa, consulta il post del blog relativo al [Modello di responsabilità condivisa AWS e GDPR](#) nel Blog sulla sicurezza AWS.

Ai fini della protezione dei dati, consigliamo di proteggere Account AWS le credenziali e configurare i singoli utenti con AWS IAM Identity Center o AWS Identity and Access Management (IAM). In tal modo, a ogni utente verranno assegnate solo le autorizzazioni necessarie per svolgere i suoi compiti. Ti suggeriamo, inoltre, di proteggere i dati nei seguenti modi:

- Utilizza l'autenticazione a più fattori (MFA) con ogni account.
- Usa SSL/TLS per comunicare con le risorse. AWS richiede TLS 1.2 ed è consigliato TLS 1.3.
- Configura l'API e la registrazione delle attività degli utenti con AWS CloudTrail. Per informazioni sull'utilizzo dei CloudTrail percorsi per acquisire AWS le attività, consulta [Lavorare con i CloudTrail percorsi](#) nella Guida per l'AWS CloudTrail utente.
- Utilizza soluzioni di AWS crittografia, insieme a tutti i controlli di sicurezza predefiniti all'interno Servizi AWS.
- Utilizza i servizi di sicurezza gestiti avanzati, come Amazon Macie, che aiutano a individuare e proteggere i dati sensibili archiviati in Amazon S3.
- Se hai bisogno di moduli crittografici convalidati FIPS 140-3 per accedere 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 \(FIPS\) 140-3](#).

Ti consigliamo di non inserire mai informazioni riservate o sensibili, ad esempio gli indirizzi e-mail dei clienti, nei tag o nei campi di testo in formato libero, ad esempio nel campo Nome. Ciò include quando lavori con Amazon Keyspaces o altro Servizi AWS utilizzando la console, l'API o. AWS CLI AWS SDKs I dati inseriti nei tag o nei campi di testo in formato libero utilizzati per i nomi possono essere utilizzati per i la fatturazione o i log di diagnostica. Quando fornisci un URL a un server esterno, ti suggeriamo vivamente di non includere informazioni sulle credenziali nell'URL per convalidare la tua richiesta al server.

## Argomenti

- [Crittografia inattiva in Amazon Keyspaces](#)
- [Crittografia in transito in Amazon Keyspaces](#)
- [Privacy del traffico di rete in Amazon Keyspaces](#)

## Crittografia inattiva in Amazon Keyspaces

[La crittografia a riposo di Amazon Keyspaces \(per Apache Cassandra\) offre una maggiore sicurezza crittografando tutti i dati inattivi utilizzando le chiavi di crittografia archiviate in \(.AWS Key Management ServiceAWS KMS](#) Questa funzionalità consente di ridurre gli oneri operativi e la complessità associati alla protezione dei dati sensibili. Con la crittografia a riposo, puoi creare applicazioni sensibili alla sicurezza che soddisfano rigorosi requisiti di conformità e normativi per la protezione dei dati.

La crittografia a riposo di Amazon Keyspaces crittografa i dati utilizzando Advanced Encryption Standard (AES-256) a 256 bit. Questo aiuta a proteggere i dati dall'accesso non autorizzato allo storage sottostante.

Amazon Keyspaces crittografa e decrittografa i dati in tabelle e flussi in modo trasparente. Amazon Keyspaces utilizza la crittografia a busta e una gerarchia di chiavi per proteggere le chiavi di crittografia dei dati. Si integra con l'archiviazione e AWS KMS la gestione della chiave di crittografia principale. Per ulteriori informazioni sulla gerarchia delle chiavi di crittografia, vedere. [the section called “Come funziona”](#) Per ulteriori informazioni su AWS KMS concetti come la crittografia degli involucri, consulta i [concetti AWS KMS dei servizi di gestione nella Guida](#) per gli AWS Key Management Service sviluppatori.

Quando crei una nuova tabella, puoi scegliere una delle seguenti AWS KMS chiavi (chiavi KMS):

- Chiave di proprietà di AWS — Questo è il tipo di crittografia predefinito. La chiave è di proprietà di Amazon Keyspaces (senza costi aggiuntivi).
- Chiave gestita dal cliente: questa chiave è memorizzata nel tuo account e viene creata, posseduta e gestita da te. Hai il pieno controllo della chiave gestita dal cliente (a AWS KMS pagamento).

Amazon Keyspaces crittografa automaticamente i flussi di change data capture (CDC) con la stessa chiave della tabella sottostante. Per ulteriori informazioni su CDC, consulta. [the section called “Lavorare con i flussi CDC”](#)

È possibile passare dalla chiave gestita dal Chiave di proprietà di AWS cliente a quella gestita dal cliente in qualsiasi momento. È possibile specificare una chiave gestita dal cliente quando si crea una nuova tabella o si modifica la chiave KMS di una tabella esistente utilizzando la console o utilizzando a livello di codice le istruzioni CQL. Per scoprire come, consulta [Crittografia dei dati inattivi: come utilizzare le chiavi gestite dal cliente per crittografare le tabelle in Amazon Keyspaces](#).

La crittografia a riposo utilizzando l'opzione predefinita di Chiavi di proprietà di AWS è offerta senza costi aggiuntivi. Tuttavia, per le chiavi gestite dal cliente vengono AWS KMS addebitati dei costi. Per ulteriori informazioni sui prezzi, consulta [Prezzi di AWS KMS](#).

La crittografia a riposo di Amazon Keyspaces è disponibile in tutte le regioni Regioni AWS, incluse AWS Cina (Pechino) e AWS Cina (Ningxia). Per ulteriori informazioni, consulta [Crittografia a riposo: come funziona in Amazon Keyspaces](#).

## Argomenti

- [Crittografia a riposo: come funziona in Amazon Keyspaces](#)
- [Crittografia dei dati inattivi: come utilizzare le chiavi gestite dal cliente per crittografare le tabelle in Amazon Keyspaces](#)

## Crittografia a riposo: come funziona in Amazon Keyspaces

La crittografia a riposo di Amazon Keyspaces (per Apache Cassandra) crittografa i dati utilizzando l'Advanced Encryption Standard (AES-256) a 256 bit. Questo aiuta a proteggere i dati dall'accesso non autorizzato allo storage sottostante. Per impostazione predefinita, tutti i dati dei clienti nelle tabelle di Amazon Keyspaces sono crittografati quando sono inattivi e la crittografia lato server è trasparente, il che significa che non sono necessarie modifiche alle applicazioni.

Encryption at rest si integra con AWS Key Management Service (AWS KMS) per la gestione della chiave di crittografia utilizzata per crittografare le tabelle. Quando crei una nuova tabella o aggiorni una tabella esistente, puoi scegliere una delle seguenti opzioni AWS KMS chiave:

- Chiave di proprietà di AWS — Questo è il tipo di crittografia predefinito. La chiave è di proprietà di Amazon Keyspaces (senza costi aggiuntivi).
- Chiave gestita dal cliente: questa chiave è memorizzata nel tuo account e viene creata, posseduta e gestita da te. Hai il pieno controllo della chiave gestita dal cliente (a AWS KMS pagamento).



## AWS KMS chiave (chiave KMS)

Encryption at rest protegge tutti i dati di Amazon Keyspaces con una AWS KMS chiave. Per impostazione predefinita, Amazon Keyspaces utilizza una [Chiave di proprietà di AWS](#) chiave di crittografia multi-tenant creata e gestita in un account del servizio Amazon Keyspaces.

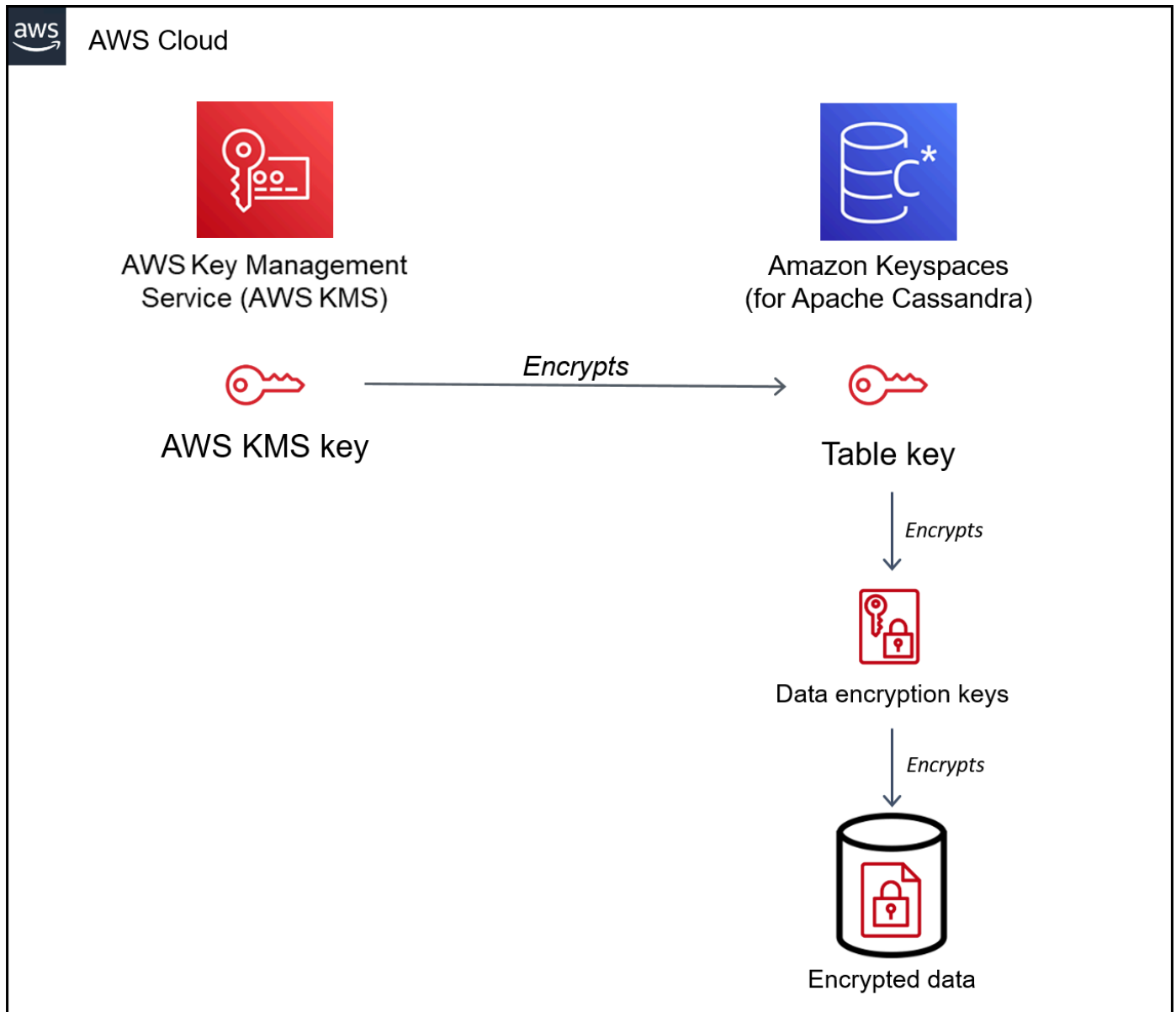
Tuttavia, puoi crittografare le tue tabelle Amazon Keyspaces utilizzando [una chiave gestita dal cliente](#) nel tuo Account AWS. Puoi selezionare una chiave KMS diversa per ogni tabella in un keyspace. La chiave KMS selezionata per una tabella viene utilizzata anche per crittografare tutti i metadati e i backup ripristinabili.

Quando si crea o si aggiorna la tabella, si seleziona la chiave KMS per una tabella. Puoi modificare la chiave KMS per una tabella in qualsiasi momento, nella console Amazon Keyspaces o utilizzando l'[istruzione ALTER TABLE](#). Il processo di cambio delle chiavi KMS è semplice e non richiede tempi di inattività né causa un peggioramento del servizio.

### Gerarchia delle chiavi

Amazon Keyspaces utilizza una gerarchia di chiavi per crittografare i dati. In questa gerarchia di chiavi, la chiave KMS è la chiave principale. Viene utilizzato per crittografare e decrittografare la chiave di crittografia della tabella Amazon Keyspaces. La chiave di crittografia delle tabelle viene utilizzata per crittografare le chiavi di crittografia utilizzate internamente da Amazon Keyspaces per crittografare e decrittografare i dati durante l'esecuzione di operazioni di lettura e scrittura.

Con la gerarchia delle chiavi di crittografia, puoi apportare modifiche alla chiave KMS senza dover crittografare nuovamente i dati o influire sulle applicazioni e sulle operazioni in corso sui dati.



## Chiave della tabella

La chiave della tabella Amazon Keyspaces viene utilizzata come chiave di crittografia delle chiavi. Amazon Keyspaces utilizza la chiave di tabella per proteggere le chiavi interne di crittografia dei dati utilizzate per crittografare i dati archiviati in tabelle, file di log e backup ripristinabili. Amazon Keyspaces genera una chiave di crittografia dei dati unica per ogni struttura sottostante in una tabella. Tuttavia, più righe della tabella potrebbero essere protette dalla stessa chiave di crittografia dei dati.

Quando imposti per la prima volta la chiave KMS su una chiave gestita dal cliente, AWS KMS genera una chiave dati. La chiave AWS KMS dati si riferisce alla chiave della tabella in Amazon Keyspaces.

Quando accedi a una tabella crittografata, Amazon Keyspaces invia una richiesta per utilizzare la chiave KMS AWS KMS per decrittografare la chiave della tabella. Quindi, utilizza la chiave di tabella in chiaro per decrittografare le chiavi di crittografia dei dati di Amazon Keyspaces e utilizza le chiavi di crittografia dei dati in testo semplice per decrittografare i dati della tabella.

Amazon Keyspaces utilizza e archivia la chiave della tabella e le chiavi di crittografia dei dati all'esterno di AWS KMS. Protegge tutte le chiavi con la crittografia [Advanced Encryption Standard](#) (AES) e le chiavi di crittografia a 256 bit. Quindi, memorizza le chiavi crittografate con i dati crittografati in modo che siano disponibili per decrittografare i dati della tabella su richiesta.

## Caching della chiave della tabella

Per evitare di richiedere AWS KMS ogni operazione di Amazon Keyspaces, Amazon Keyspaces memorizza nella cache le chiavi della tabella in testo semplice per ogni connessione in memoria. Se Amazon Keyspaces riceve una richiesta per la chiave della tabella memorizzata nella cache dopo cinque minuti di inattività, invia una nuova richiesta per AWS KMS decrittografare la chiave della tabella. Questa chiamata acquisisce tutte le modifiche apportate alle politiche di accesso della chiave KMS in AWS KMS o AWS Identity and Access Management (IAM) dall'ultima richiesta di decrittografia della chiave della tabella.

## Crittografia envelope

Se modifichi la chiave gestita dal cliente per la tua tabella, Amazon Keyspaces genera una nuova chiave di tabella. Quindi, utilizza la nuova chiave di tabella per crittografare nuovamente le chiavi di crittografia dei dati. Utilizza inoltre la nuova chiave di tabella per crittografare le chiavi di tabella precedenti utilizzate per proteggere i backup ripristinabili. Questo processo è chiamato crittografia a busta. In questo modo è possibile accedere ai backup ripristinabili anche ruotando la chiave gestita dal cliente. Per ulteriori informazioni sulla crittografia delle buste, consulta [Envelope Encryption](#) nella [Developer Guide](#). AWS Key Management Service

## Argomenti

- [AWS chiavi possedute](#)
- [Chiavi gestite dal cliente](#)
- [Note sull'utilizzo di Encryption at Rest](#)

## AWS chiavi possedute

Chiavi di proprietà di AWS non sono archiviati nel tuo Account AWS. Fanno parte di una raccolta di chiavi KMS che AWS possiede e gestisce per essere utilizzate in più Account AWS lingue. AWS i servizi possono essere utilizzati Chiavi di proprietà di AWS per proteggere i tuoi dati.

Non puoi visualizzare, gestire Chiavi di proprietà di AWS, utilizzare o controllare il loro utilizzo. Tuttavia, non è necessario eseguire alcuna operazione o modificare alcun programma per proteggere le chiavi che crittografano i dati.

Non ti viene addebitato un canone mensile o un canone di utilizzo per l'utilizzo di Chiavi di proprietà di AWS e questi non vengono conteggiati nelle AWS KMS quote per il tuo account.

## Chiavi gestite dal cliente

Le chiavi gestite dal cliente sono chiavi Account AWS che crei, possiedi e gestisci. Hai il pieno controllo su queste chiavi KMS.

Utilizza una chiave gestita dal cliente per ottenere le seguenti caratteristiche:

- Crei e gestisci la chiave gestita dal cliente, inclusa l'impostazione e la manutenzione delle [politiche chiave, delle politiche IAM](#) e le [concessioni](#) per il controllo dell'accesso alla chiave gestita dal cliente. È possibile [abilitare e disabilitare](#) la chiave gestita dal cliente, abilitare e disabilitare la [rotazione automatica delle chiavi](#) e [pianificare l'eliminazione della chiave gestita dal cliente](#) quando non è più in uso. Puoi creare tag e alias per le chiavi gestite dal cliente che gestisci.
- L'utente può utilizzare una chiave gestita dal cliente con [materiale di chiave importato](#) o una chiave gestita dal cliente in un [archivio di chiavi personalizzate](#) di cui è proprietario e gestirlo.
- Puoi utilizzare AWS CloudTrail Amazon CloudWatch Logs per tenere traccia delle richieste a cui Amazon Keyspaces invia AWS KMS per tuo conto. Per ulteriori informazioni, consulta [the section called "Fase 6: Configurare il monitoraggio con AWS CloudTrail"](#).

Le chiavi gestite dal cliente [comportano un addebito](#) per ogni chiamata API e a queste chiavi KMS vengono applicate delle AWS KMS quote. Per ulteriori informazioni, consulta le [AWS KMS risorse](#) o richiedi quote.

Quando si specifica una chiave gestita dal cliente come chiave di crittografia principale per una tabella, i backup ripristinabili vengono crittografati con la stessa chiave di crittografia specificata per la tabella al momento della creazione del backup. Se la chiave KMS per la tabella viene ruotata,

l'avvolgimento delle chiavi garantisce che la chiave KMS più recente abbia accesso a tutti i backup ripristinabili.

Amazon Keyspaces deve avere accesso alla tua chiave gestita dal cliente per fornirti l'accesso ai dati delle tabelle. Se lo stato della chiave di crittografia è impostato su `disabilitato` o se ne è pianificata l'eliminazione, Amazon Keyspaces non è in grado di crittografare o decrittografare i dati. Di conseguenza, non sei in grado di eseguire operazioni di lettura e scrittura sulla tabella. Non appena il servizio rileva che la tua chiave di crittografia è inaccessibile, Amazon Keyspaces invia una notifica e-mail per avvisarti.

È necessario ripristinare l'accesso alla chiave di crittografia entro sette giorni, altrimenti Amazon Keyspaces eliminerà automaticamente la tabella. Per precauzione, Amazon Keyspaces crea un backup ripristinabile dei dati della tabella prima di eliminarla. Amazon Keyspaces mantiene il backup ripristinabile per 35 giorni. Dopo 35 giorni, non puoi più ripristinare i dati della tabella. Il backup ripristinabile non ti viene addebitato, ma vengono [applicate le tariffe di ripristino](#) standard.

Puoi utilizzare questo backup ripristinabile per ripristinare i dati in una nuova tabella. Per avviare il ripristino, è necessario abilitare l'ultima chiave gestita dal cliente utilizzata per la tabella e Amazon Keyspaces deve potervi accedere.

#### Note

Quando crei una tabella crittografata utilizzando una chiave gestita dal cliente inaccessibile o pianificata per l'eliminazione prima del completamento del processo di creazione, si verifica un errore. L'operazione di creazione della tabella non riesce e ti viene inviata una notifica via e-mail.

## Note sull'utilizzo di Encryption at Rest

Considera quanto segue quando utilizzi la crittografia a riposo in Amazon Keyspaces.

- La crittografia lato server a riposo è abilitata su tutte le tabelle Amazon Keyspaces e non può essere disabilitata. L'intera tabella è crittografata quando è inattiva, non è possibile selezionare colonne o righe specifiche per la crittografia.
- Per impostazione predefinita, Amazon Keyspaces utilizza una chiave predefinita a servizio singolo (Chiave di proprietà di AWS) per crittografare tutte le tabelle. Se questa chiave non esiste, viene creata per te. Le chiavi predefinite del servizio non possono essere disabilitate.

- La crittografia a riposo crittografa i dati solo quando sono statici (a riposo) su un supporto di archiviazione persistente. Se la sicurezza dei dati è un problema per i dati in transito o per i dati in uso, è necessario adottare misure aggiuntive:
  - Dati in transito: tutti i dati in Amazon Keyspaces sono crittografati in transito. Per impostazione predefinita, le comunicazioni da e verso Amazon Keyspaces sono protette utilizzando la crittografia Secure Sockets Layer (SSL) /Transport Layer Security (TLS).
  - Dati in uso: proteggi i dati prima di inviarli ad Amazon Keyspaces utilizzando la crittografia lato client.
  - Chiavi gestite dal cliente: i dati inattivi nelle tabelle vengono sempre crittografati utilizzando le chiavi gestite dal cliente. Tuttavia, le operazioni che eseguono aggiornamenti atomici di più righe crittografano temporaneamente i dati, utilizzandoli Chiavi di proprietà di AWS durante l'elaborazione. Ciò include le operazioni di eliminazione degli intervalli e le operazioni che accedono simultaneamente a dati statici e non statici.
- [Una singola chiave gestita dal cliente può avere fino a 50.000 concessioni.](#) Ogni tabella Amazon Keyspaces associata a una chiave gestita dal cliente consuma 2 concessioni. Una concessione viene rilasciata quando la tabella viene eliminata. La seconda concessione viene utilizzata per creare un'istantanea automatica della tabella per proteggerla dalla perdita di dati nel caso in cui Amazon Keyspaces perda l'accesso alla chiave gestita dal cliente per errore. Questa concessione viene rilasciata 42 giorni dopo l'eliminazione della tabella.

## Crittografia dei dati inattivi: come utilizzare le chiavi gestite dal cliente per crittografare le tabelle in Amazon Keyspaces

Puoi utilizzare la console o le istruzioni CQL AWS KMS key per specificare le nuove tabelle e aggiornare le chiavi di crittografia delle tabelle esistenti in Amazon Keyspaces. Il seguente argomento descrive come implementare le chiavi gestite dal cliente per tabelle nuove ed esistenti.

### Argomenti

- [Prerequisiti: creare una chiave gestita dal cliente utilizzando AWS KMS e concedere le autorizzazioni ad Amazon Keyspaces](#)
- [Fase 3: Specificare una chiave gestita dal cliente per una nuova tabella](#)
- [Passaggio 4: Aggiornare la chiave di crittografia di una tabella esistente](#)
- [Fase 5: utilizza il contesto di crittografia Amazon Keyspaces nei log](#)
- [Fase 6: Configurare il monitoraggio con AWS CloudTrail](#)

**Prerequisiti:** creare una chiave gestita dal cliente utilizzando AWS KMS e concedere le autorizzazioni ad Amazon Keyspaces

Prima di poter proteggere una tabella Amazon Keyspaces con una [chiave gestita dal cliente](#), devi prima creare la chiave in AWS Key Management Service (AWS KMS) e quindi autorizzare Amazon Keyspaces a utilizzare quella chiave.

**Fase 1:** crea una chiave gestita dal cliente utilizzando AWS KMS

Per creare una chiave gestita dal cliente da utilizzare per proteggere una tabella Amazon Keyspaces, puoi seguire i passaggi in [Creazione di chiavi KMS con crittografia simmetrica](#) utilizzando la console o l'API. AWS

**Fase 2:** Autorizza l'uso della chiave gestita dal cliente

Prima di poter scegliere una [chiave gestita dal cliente](#) per proteggere una tabella Amazon Keyspaces, le politiche su tale chiave gestita dal cliente devono autorizzare Amazon Keyspaces a utilizzarla per tuo conto. Hai il pieno controllo sulle politiche e sulle concessioni relative alla chiave gestita dal cliente. È possibile fornire queste autorizzazioni in una [policy della chiave](#), in una [policy IAM](#) o mediante una [concessione](#).

Amazon Keyspaces non necessita di autorizzazioni aggiuntive per utilizzare l'impostazione predefinita [Chiave di proprietà di AWS](#) per proteggere le tabelle Amazon Keyspaces nel tuo account. AWS

I seguenti argomenti mostrano come configurare le autorizzazioni richieste utilizzando le policy e le concessioni IAM che consentono alle tabelle Amazon Keyspaces di utilizzare una chiave gestita dal cliente.

Argomenti

- [Politica chiave per le chiavi gestite dai clienti](#)
- [Esempi di policy delle chiavi](#)
- [Utilizzo delle sovvenzioni per autorizzare Amazon Keyspaces](#)

Politica chiave per le chiavi gestite dai clienti

Quando selezioni una [chiave gestita dal cliente](#) per proteggere una tabella Amazon Keyspaces, Amazon Keyspaces ottiene l'autorizzazione a utilizzare la chiave gestita dal cliente per conto del principale che effettua la selezione. Tale principale, un utente o un ruolo, deve disporre delle autorizzazioni sulla chiave gestita dal cliente richieste da Amazon Keyspaces.

Amazon Keyspaces richiede almeno le seguenti autorizzazioni su una chiave gestita dal cliente:

- [kms:Encrypt](#)
- [kms:Decrypt](#)
- [kms: ReEncrypt](#) \* (per e) kms: ReEncryptFrom kms: ReEncryptTo
- [kms: GenerateDataKey](#) \* (per kms: GenerateDataKey e kms:) GenerateDataKeyWithoutPlaintext
- [km: DescribeKey](#)
- [km: CreateGrant](#)

### Esempi di policy delle chiavi

Ad esempio, la policy di chiave di esempio riportata di seguito fornisce solo le autorizzazioni necessarie. La policy ha i seguenti effetti:

- Consente ad Amazon Keyspaces di utilizzare la chiave gestita dal cliente nelle operazioni crittografiche e creare sovvenzioni, ma solo quando agisce per conto dei responsabili dell'account che dispongono dell'autorizzazione a utilizzare Amazon Keyspaces. Se i responsabili specificati nell'informativa non sono autorizzati a utilizzare Amazon Keyspaces, la chiamata ha esito negativo, anche se proviene dal servizio Amazon Keyspaces.
- La chiave [kms: ViaService](#) condition consente le autorizzazioni solo quando la richiesta proviene da Amazon Keyspaces per conto dei principali elencati nell'informativa. Tali entità non possono chiamare direttamente queste operazioni. Nota: il valore `kms:ViaService,cassandra.*.amazonaws.com`, ha un asterisco (\*) nella posizione Regione. Amazon Keyspaces richiede l'autorizzazione per essere indipendente da qualsiasi particolare Regione AWS
- Fornisce agli amministratori delle chiavi gestite dal cliente (utenti che possono assumere il db-team ruolo) l'accesso in sola lettura alla chiave gestita dal cliente e l'autorizzazione a revocare le concessioni, incluse le concessioni richieste da [Amazon Keyspaces](#) per proteggere la tabella.
- Fornisce ad Amazon Keyspaces l'accesso in sola lettura alla chiave gestita dal cliente. In questo caso, Amazon Keyspaces può richiamare direttamente queste operazioni. Non deve agire per conto del responsabile del conto.

Prima di utilizzare una politica chiave di esempio, sostituisci i principali di esempio con i principali effettivi del tuo Account AWS

```
{
```



```

 "Id": "key-policy-cassandra",
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "Allow access through Amazon Keyspaces for all principals in the account
that are authorized to use Amazon Keyspaces",
 "Effect": "Allow",
 "Principal": {"AWS": "arn:aws:iam::111122223333:user/db-lead"},
 "Action": [
 "kms:Encrypt",
 "kms:Decrypt",
 "kms:ReEncrypt*",
 "kms:GenerateDataKey*",
 "kms:DescribeKey",
 "kms:CreateGrant"
],
 "Resource": "*",
 "Condition": {
 "StringLike": {
 "kms:ViaService": "cassandra.*.amazonaws.com"
 }
 }
 },
 {
 "Sid": "Allow administrators to view the customer managed key and revoke
grants",
 "Effect": "Allow",
 "Principal": {
 "AWS": "arn:aws:iam::111122223333:role/db-team"
 },
 "Action": [
 "kms:Describe*",
 "kms:Get*",
 "kms:List*",
 "kms:RevokeGrant"
],
 "Resource": "*"
 }
]
 }

```

## Utilizzo delle sovvenzioni per autorizzare Amazon Keyspaces

Oltre alle politiche chiave, Amazon Keyspaces utilizza le concessioni per impostare le autorizzazioni su una chiave gestita dal cliente. Per visualizzare le concessioni della chiave gestita dal cliente nell'account, utilizza l'operazione [ListGrants](#). Amazon Keyspaces non necessita di concessioni o autorizzazioni aggiuntive per utilizzarlo per [Chiave di proprietà di AWS](#) proteggere la tabella.

Amazon Keyspaces utilizza le autorizzazioni di concessione quando esegue attività di manutenzione del sistema in background e di protezione continua dei dati. Utilizza inoltre concessioni per generare le chiavi delle tabelle.

Ogni concessione è specifica di una tabella. Se l'account include più tabelle crittografate con la stessa chiave gestita dal cliente, è prevista una concessione di ogni tipo per ogni tabella. La concessione è vincolata dal [contesto di crittografia Amazon Keyspaces](#), che include il nome della tabella e l'ID. Account AWS La concessione include l'autorizzazione a [ritirarla](#) se non è più necessaria.

Per creare le concessioni, Amazon Keyspaces deve disporre dell'autorizzazione a effettuare `CreateGrant` chiamate per conto dell'utente che ha creato la tabella crittografata.

La policy delle chiavi può anche consentire all'account di [revocare la concessione](#) sulla chiave gestita dal cliente. Tuttavia, se revochi la concessione su una tabella crittografata attiva, Amazon Keyspaces non sarà in grado di proteggere e mantenere la tabella.

### Fase 3: Specificare una chiave gestita dal cliente per una nuova tabella

Segui questi passaggi per specificare la chiave gestita dal cliente su una nuova tabella utilizzando la console Amazon Keyspaces o CQL.

Crea una tabella crittografata utilizzando una chiave gestita dal cliente (console)

1. [Accedi a e apri Console di gestione AWS la console Amazon Keyspaces a casahttps://console.aws.amazon.com/keyspaces/](https://console.aws.amazon.com/keyspaces/).
2. Nel pannello di navigazione, scegli Tabelle, quindi seleziona Crea tabella.
3. Nella pagina Crea tabella, nella sezione Dettagli della tabella, seleziona un keyspace e fornisci un nome per la nuova tabella.
4. Nella sezione Schema, crea lo schema per la tua tabella.
5. Nella sezione Impostazioni tabella, scegli Personalizza impostazioni.

## 6. Continua con le impostazioni di crittografia.

In questo passaggio, si selezionano le impostazioni di crittografia per la tabella.

Nella sezione Encryption at rest, sotto Scegli un AWS KMS key, scegli l'opzione Scegli una chiave KMS diversa (avanzata) e nel campo di ricerca scegli AWS KMS key o inserisci un Amazon Resource Name (ARN).

### Note

Se la chiave selezionata non è accessibile o non dispone delle autorizzazioni richieste, consulta la sezione [Risoluzione dei problemi di accesso tramite chiave](#) nella Guida per gli AWS Key Management Service sviluppatori.

## 7. Scegli Create (Crea) per creare la tabella crittografata.

Crea una nuova tabella utilizzando una chiave gestita dal cliente per la crittografia a riposo (CQL)

Per creare una nuova tabella che utilizza una chiave gestita dal cliente per la crittografia a riposo, è possibile utilizzare l'`CREATE TABLE`istruzione come illustrato nell'esempio seguente. Assicurati di sostituire la chiave ARN con un ARN per una chiave valida con autorizzazioni concesse ad Amazon Keyspaces.

```
CREATE TABLE my_keyspace.my_table(id bigint, name text, place text STATIC, PRIMARY
KEY(id, name)) WITH CUSTOM_PROPERTIES = {
 'encryption_specification':{
 'encryption_type': 'CUSTOMER_MANAGED_KMS_KEY',
 'kms_key_identifier': 'arn:aws:kms:eu-west-1:555555555555:key/11111111-1111-111-1111-111111111111'
 }
};
```

Se ne ricevi una `Invalid Request Exception`, devi confermare che la chiave gestita dal cliente è valida e che Amazon Keyspaces dispone delle autorizzazioni richieste. Per confermare che la chiave è stata configurata correttamente, consulta [Risoluzione dei problemi di accesso tramite chiave nella Guida](#) per gli AWS Key Management Service sviluppatori.

## Passaggio 4: Aggiornare la chiave di crittografia di una tabella esistente

Puoi anche utilizzare la console Amazon Keyspaces o CQL per modificare le chiavi di crittografia di una tabella esistente tra una Chiave di proprietà di AWS chiave KMS gestita dal cliente in qualsiasi momento.

Aggiorna una tabella esistente con la nuova chiave gestita dal cliente (console)

1. [Accedi a e apri Console di gestione AWS la console Amazon Keyspaces a casa](https://console.aws.amazon.com/keyspaces/)<https://console.aws.amazon.com/keyspaces/>.
2. Nel pannello di navigazione, seleziona Tabelle.
3. Scegli la tabella che desideri aggiornare, quindi scegli la scheda Impostazioni aggiuntive.
4. Nella sezione Crittografia a riposo, scegli Gestisci crittografia per modificare le impostazioni di crittografia per la tabella.

In Scegli un AWS KMS key, scegli l'opzione Scegli una chiave KMS diversa (avanzata) e nel campo di ricerca scegli AWS KMS key o inserisci un Amazon Resource Name (ARN).

### Note

Se la chiave selezionata non è valida, consulta la sezione [Risoluzione dei problemi di accesso tramite chiave nella Guida](#) per gli AWS Key Management Service sviluppatori.

In alternativa, puoi sceglierne una Chiave di proprietà di AWS per una tabella crittografata con una chiave gestita dal cliente.

5. Scegli Salva modifiche per salvare le modifiche alla tabella.

Aggiorna la chiave di crittografia utilizzata per una tabella esistente

Per modificare la chiave di crittografia di una tabella esistente, si utilizza l'ALTER TABLE istruzione per specificare una chiave gestita dal cliente per la crittografia a riposo. Assicurati di sostituire la chiave ARN con un ARN per una chiave valida con autorizzazioni concesse ad Amazon Keyspaces.

```
ALTER TABLE my_keyspace.my_table WITH CUSTOM_PROPERTIES = {
 'encryption_specification': {
 'encryption_type': 'CUSTOMER_MANAGED_KMS_KEY',
 'kms_key_identifier': 'arn:aws:kms:eu-west-1:555555555555:key/11111111-1111-111-1111-111111111111'
```

```
 }
};
```

Se ne ricevi una `Invalid Request Exception`, devi confermare che la chiave gestita dal cliente è valida e che Amazon Keyspaces dispone delle autorizzazioni richieste. Per confermare che la chiave è stata configurata correttamente, consulta [Risoluzione dei problemi di accesso tramite chiave nella Guida](#) per gli AWS Key Management Service sviluppatori.

Per riportare la chiave di crittografia all'opzione predefinita di crittografia a riposo con Chiavi di proprietà di AWS, è possibile utilizzare l'`ALTER TABLE` istruzione come illustrato nell'esempio seguente.

```
ALTER TABLE my_keyspace.my_table WITH CUSTOM_PROPERTIES = {
 'encryption_specification': {
 'encryption_type' : 'AWS_OWNED_KMS_KEY'
 }
};
```

Fase 5: utilizza il contesto di crittografia Amazon Keyspaces nei log

Un [contesto di crittografia](#) è un set di coppie chiave-valore che contiene dati arbitrari non segreti. Quando includi un contesto di crittografia in una richiesta di crittografia dei dati, associa AWS KMS crittograficamente il contesto di crittografia ai dati crittografati. lo stesso contesto di crittografia sia necessario per decrittografare i dati.

Amazon Keyspaces utilizza lo stesso contesto di crittografia in tutte le operazioni AWS KMS crittografiche. Se utilizzi una [chiave gestita dal cliente](#) per proteggere la tua tabella Amazon Keyspaces, puoi utilizzare il contesto di crittografia per identificare l'uso della chiave gestita dal cliente nei record e nei log di controllo. [Viene inoltre visualizzato in testo non crittografato nei log, ad esempio in logs for e AWS CloudTrail Amazon Logs. CloudWatch](#)

Nelle sue richieste a AWS KMS, Amazon Keyspaces utilizza un contesto di crittografia con tre coppie chiave-valore.

```
"encryptionContextSubset": {
 "aws:cassandra:keyspaceName": "my_keyspace",
 "aws:cassandra:tableName": "mytable"
 "aws:cassandra:subscriberId": "111122223333"
}
```

- **Keyspace:** la prima coppia chiave-valore identifica lo spazio chiave che include la tabella che Amazon Keyspaces sta crittografando. La chiave è `aws:cassandra:keyspaceName`. Il valore è il nome del keyspace.

```
"aws:cassandra:keyspaceName": "<keyspace-name>"
```

Ad esempio:

```
"aws:cassandra:keyspaceName": "my_keyspace"
```

- **Tabella:** la seconda coppia chiave-valore identifica la tabella che Amazon Keyspaces sta crittografando. La chiave è `aws:cassandra:tableName`. Il valore è il nome della tabella.

```
"aws:cassandra:tableName": "<table-name>"
```

Ad esempio:

```
"aws:cassandra:tableName": "my_table"
```

- **Account:** la terza coppia chiave-valore identifica il Account AWS. La chiave è `aws:cassandra:subscriberId`. Il valore è l'ID dell'account.

```
"aws:cassandra:subscriberId": "<account-id>"
```

Ad esempio:

```
"aws:cassandra:subscriberId": "111122223333"
```

## Fase 6: Configurare il monitoraggio con AWS CloudTrail

Se utilizzi una [chiave gestita dal cliente](#) per proteggere le tue tabelle Amazon Keyspaces, puoi utilizzare AWS CloudTrail i log per tenere traccia delle richieste a cui Amazon Keyspaces invia per tuo conto. AWS KMS

Le `CreateGrant` richieste `GenerateDataKey`, `DescribeKeyDecrypt`, e vengono discusse in questa sezione. Inoltre, Amazon Keyspaces utilizza un' [RetireGrant](#) operazione per rimuovere una concessione quando elimini una tabella.

**Note**

Quando lavori con Amazon Keyspaces, alcune operazioni possono generare CloudTrail eventi con un `invokedBy` campo di `dynamodb.amazonaws.com`. Ciò è previsto e si verifica perché Amazon Keyspaces si integra con Amazon DynamoDB per fornire il proprio servizio.

**GenerateDataKey**

Amazon Keyspaces crea una chiave di tabella unica per crittografare i dati inattivi. Invia una [GenerateDataKey](#) richiesta a AWS KMS che specifica la chiave KMS per la tabella.

L'evento che registra l'operazione `GenerateDataKey` è simile a quello del seguente evento di esempio. L'utente è l'account del servizio Amazon Keyspaces. I parametri includono l'Amazon Resource Name (ARN) della chiave gestita dal cliente, un identificatore di chiave che richiede una chiave a 256 bit e il [contesto di crittografia](#) che identifica lo spazio delle chiavi, la tabella e il Account AWS

```
{
 "eventVersion": "1.08",
 "userIdentity": {
 "type": "AWSService",
 "invokedBy": "AWS Internal"
 },
 "eventTime": "2021-04-16T04:56:05Z",
 "eventSource": "kms.amazonaws.com",
 "eventName": "GenerateDataKey",
 "awsRegion": "us-east-1",
 "sourceIPAddress": "AWS Internal",
 "userAgent": "AWS Internal",
 "requestParameters": {
 "keySpec": "AES_256",
 "encryptionContext": {
 "aws:cassandra:keyspaceName": "my_keyspace",
 "aws:cassandra:tableName": "my_table",
 "aws:cassandra:subscriberId": "123SAMPLE012"
 },
 "keyId": "arn:aws:kms:eu-west-1:555555555555:key/11111111-1111-111-1111-111111111111"
 },
 "responseElements": null,
 "requestID": "5e8e9cb5-9194-4334-aacc-9dd7d50fe246",
```

```

 "eventID": "49fccab9-2448-4b97-a89d-7d5c39318d6f",
 "readOnly": true,
 "resources": [
 {
 "accountId": "123SAMPLE012",
 "type": "AWS::KMS::Key",
 "ARN": "arn:aws:kms:eu-west-1:555555555555:key/11111111-1111-111-1111-111111111111"
 }
],
 "eventType": "AwsApiCall",
 "managementEvent": true,
 "eventCategory": "Management",
 "recipientAccountId": "123SAMPLE012",
 "sharedEventID": "84fbaaf0-9641-4e32-9147-57d2cb08792e"
 }
}

```

## DescribeKey

Amazon Keyspaces utilizza un'[DescribeKey](#) operazione per determinare se la chiave KMS selezionata esiste nell'account e nella regione.

L'evento che registra l'operazione `DescribeKey` è simile a quello del seguente evento di esempio. L'utente è l'account del servizio Amazon Keyspaces. I parametri includono l'ARN della chiave gestita dal cliente e un identificatore di chiave che richiede una chiave a 256 bit.

```

{
 "eventVersion": "1.08",
 "userIdentity": {
 "type": "IAMUser",
 "principalId": "AIDAZ3FNIIVIZZ6H7CFQG",
 "arn": "arn:aws:iam::123SAMPLE012:user/admin",
 "accountId": "123SAMPLE012",
 "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
 "userName": "admin",
 "sessionContext": {
 "sessionIssuer": {},
 "webIdFederationData": {},
 "attributes": {
 "mfaAuthenticated": "false",
 "creationDate": "2021-04-16T04:55:42Z"
 }
 }
 },
}

```



```

 "invokedBy": "AWS Internal"
 },
 "eventTime": "2021-04-16T04:55:58Z",
 "eventSource": "kms.amazonaws.com",
 "eventName": "DescribeKey",
 "awsRegion": "us-east-1",
 "sourceIPAddress": "AWS Internal",
 "userAgent": "AWS Internal",
 "requestParameters": {
 "keyId": "arn:aws:kms:eu-west-1:555555555555:key/11111111-1111-111-1111-111111111111"
 },
 "responseElements": null,
 "requestID": "c25a8105-050b-4f52-8358-6e872fb03a6c",
 "eventID": "0d96420e-707e-41b9-9118-56585a669658",
 "readOnly": true,
 "resources": [
 {
 "accountId": "123SAMPLE012",
 "type": "AWS::KMS::Key",
 "ARN": "arn:aws:kms:eu-west-1:555555555555:key/11111111-1111-111-1111-111111111111"
 }
],
 "eventType": "AwsApiCall",
 "managementEvent": true,
 "eventCategory": "Management",
 "recipientAccountId": "123SAMPLE012"
}

```

## Decrypt

Quando accedi a una tabella Amazon Keyspaces, Amazon Keyspaces deve decrittografare la chiave della tabella in modo da poter decrittografare le chiavi sottostanti nella gerarchia. Quindi decrittografa i dati nella tabella. Per decrittografare la chiave della tabella, Amazon Keyspaces invia una richiesta [Decrypt](#) a AWS KMS che specifica la chiave KMS per la tabella.

L'evento che registra l'operazione Decrypt è simile a quello del seguente evento di esempio. L'utente è il tuo principale che accede alla tabella. Account AWS I parametri includono la chiave crittografata della tabella (come blob di testo cifrato) e il [contesto di crittografia](#) che identifica la tabella e il. Account AWS AWS KMS ricava l'ID della chiave gestita dal cliente dal testo cifrato.

```
{
```

```

 "eventVersion": "1.08",
 "userIdentity": {
 "type": "AWSService",
 "invokedBy": "AWS Internal"
 },
 "eventTime": "2021-04-16T05:29:44Z",
 "eventSource": "kms.amazonaws.com",
 "eventName": "Decrypt",
 "awsRegion": "us-east-1",
 "sourceIPAddress": "AWS Internal",
 "userAgent": "AWS Internal",
 "requestParameters": {
 "encryptionContext": {
 "aws:cassandra:keyspaceName": "my_keyspace",
 "aws:cassandra:tableName": "my_table",
 "aws:cassandra:subscriberId": "123SAMPLE012"
 },
 "encryptionAlgorithm": "SYMMETRIC_DEFAULT"
 },
 "responseElements": null,
 "requestID": "50e80373-83c9-4034-8226-5439e1c9b259",
 "eventID": "8db9788f-04a5-4ae2-90c9-15c79c411b6b",
 "readOnly": true,
 "resources": [
 {
 "accountId": "123SAMPLE012",
 "type": "AWS::KMS::Key",
 "ARN": "arn:aws:kms:eu-west-1:555555555555:key/11111111-1111-1111-1111-111111111111"
 }
],
 "eventType": "AwsApiCall",
 "managementEvent": true,
 "eventCategory": "Management",
 "recipientAccountId": "123SAMPLE012",
 "sharedEventID": "7ed99e2d-910a-4708-a4e3-0180d8dbb68e"
 }
}

```

## CreateGrant

Quando utilizzi una [chiave gestita dal cliente](#) per proteggere la tua tabella Amazon Keyspaces, Amazon Keyspaces utilizza le [concessioni](#) per consentire al servizio di eseguire attività continue di

protezione dei dati, manutenzione e durabilità. Queste sovvenzioni non sono obbligatorie. [Chiavi di proprietà di AWS](#)

Le concessioni create da Amazon Keyspaces sono specifiche per una tabella. Il principale della [CreateGrant](#) richiesta è l'utente che ha creato la tabella.

L'evento che registra l'operazione `CreateGrant` è simile a quello del seguente evento di esempio. I parametri includono l'ARN della chiave gestita dal cliente per la tabella, il principale beneficiario e il principale ritirante (il servizio Amazon Keyspaces) e le operazioni coperte dalla sovvenzione. [Include anche un vincolo che richiede che tutte le operazioni di crittografia utilizzino il contesto di crittografia specificato.](#)

```
{
 "eventVersion": "1.08",
 "userIdentity": {
 "type": "IAMUser",
 "principalId": "AIDAZ3FNIIVIZZ6H7CFQG",
 "arn": "arn:aws:iam::arn:aws:kms:eu-west-1:555555555555:key/11111111-1111-111-1111-111111111111:user/admin",
 "accountId": "arn:aws:kms:eu-west-1:555555555555:key/11111111-1111-111-1111-111111111111",
 "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
 "userName": "admin",
 "sessionContext": {
 "sessionIssuer": {},
 "webIdFederationData": {},
 "attributes": {
 "mfaAuthenticated": "false",
 "creationDate": "2021-04-16T04:55:42Z"
 }
 }
 },
 "invokedBy": "AWS Internal",
 "eventTime": "2021-04-16T05:11:10Z",
 "eventSource": "kms.amazonaws.com",
 "eventName": "CreateGrant",
 "awsRegion": "us-east-1",
 "sourceIPAddress": "AWS Internal",
 "userAgent": "AWS Internal",
 "requestParameters": {
 "keyId": "a7d328af-215e-4661-9a69-88c858909f20",
 "operations": [
 "DescribeKey",
```

```

 "GenerateDataKey",
 "Decrypt",
 "Encrypt",
 "ReEncryptFrom",
 "ReEncryptTo",
 "RetireGrant"
],
 "constraints": {
 "encryptionContextSubset": {
 "aws:cassandra:keyspaceName": "my_keyspace",
 "aws:cassandra:tableName": "my_table",
 "aws:cassandra:subscriberId": "123SAMPLE012"
 }
 },
 "retiringPrincipal": "cassandratest.us-east-1.amazonaws.com",
 "granteePrincipal": "cassandratest.us-east-1.amazonaws.com"
},
"responseElements": {
 "grantId":
"18e4235f1b07f289762a31a1886cb5efd225f069280d4f76cd83b9b9b5501013"
},
"requestID": "b379a767-1f9b-48c3-b731-fb23e865e7f7",
"eventID": "29ee1fd4-28f2-416f-a419-551910d20291",
"readOnly": false,
"resources": [
 {
 "accountId": "123SAMPLE012",
 "type": "AWS::KMS::Key",
 "ARN": "arn:aws:kms:eu-
west-1:555555555555:key/11111111-1111-111-1111-111111111111"
 }
],
"eventType": "AwsApiCall",
"managementEvent": true,
"eventCategory": "Management",
"recipientAccountId": "123SAMPLE012"
}

```

## Crittografia in transito in Amazon Keyspaces

Amazon Keyspaces accetta solo connessioni sicure tramite Transport Layer Security (TLS). La crittografia in transito fornisce un ulteriore livello di protezione dei dati crittografando i dati mentre

viaggiano da e verso Amazon Keyspaces. Le politiche organizzative, le normative di settore o governative e i requisiti di conformità spesso richiedono l'uso della crittografia in transito per aumentare la sicurezza dei dati delle applicazioni quando trasmettono dati sulla rete.

Per informazioni su come crittografare `cqlsh` le connessioni ad Amazon Keyspaces utilizzando TLS, consulta [the section called “Come configurare manualmente le `cqlsh` connessioni per TLS”](#). Per informazioni su come utilizzare la crittografia TLS con i driver client, consulta [the section called “Utilizzo di un driver client Cassandra”](#).

## Privacy del traffico di rete in Amazon Keyspaces

Questo argomento descrive come Amazon Keyspaces (per Apache Cassandra) protegge le connessioni dalle applicazioni locali ad Amazon Keyspaces e tra Amazon Keyspaces e altre risorse all'interno delle stesse. AWS Regione AWS

### Traffico tra servizio e applicazioni e client locali

Sono disponibili due opzioni di connettività tra la rete privata e: AWS

- Una AWS Site-to-Site VPN connessione. Per ulteriori informazioni, consulta [Che cos'è AWS Site-to-Site VPN?](#) nella Guida per l'utente di AWS Site-to-Site VPN .
- Una Direct Connect connessione. Per ulteriori informazioni, consulta [Che cos'è Direct Connect?](#) nella Guida per l'utente di Direct Connect .

In quanto servizio gestito, Amazon Keyspaces (per Apache Cassandra) è protetto dalla AWS sicurezza di rete globale. [Per informazioni sui servizi di AWS sicurezza e su come AWS protegge l'infrastruttura, consulta AWS Cloud Security.](#) Per progettare il tuo AWS ambiente utilizzando le migliori pratiche per la sicurezza dell'infrastruttura, vedi [Infrastructure Protection](#) in Security Pillar AWS Well-Architected Framework.

Utilizzi chiamate API AWS pubblicate per accedere ad Amazon Keyspaces attraverso la rete. I client devono supportare quanto segue:

- Transport Layer Security (TLS). È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Suite di cifratura con Perfect Forward Secrecy (PFS), 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.

Amazon Keyspaces supporta due metodi di autenticazione delle richieste dei client. Il primo metodo utilizza credenziali specifiche del servizio, che sono credenziali basate su password generate per uno specifico utente IAM. Puoi creare e gestire la password utilizzando la console IAM, l'API AWS CLI. AWS Per ulteriori informazioni, consulta [Using IAM with Amazon Keyspaces](#).

Il secondo metodo utilizza un plug-in di autenticazione per il driver DataStax Java open source per Cassandra. Questo plug-in consente [agli utenti, ai ruoli e alle identità federate IAM](#) di aggiungere informazioni di autenticazione alle richieste API di Amazon Keyspaces (per Apache Cassandra) utilizzando il processo [AWS Signature Version 4](#) (SigV4). Per ulteriori informazioni, consulta [the section called “Crea credenziali IAM per l'autenticazione AWS”](#).

## Traffico tra risorse nella stessa regione AWS

Gli endpoint VPC di interfaccia consentono la comunicazione privata tra il tuo cloud privato virtuale (VPC) in esecuzione su Amazon VPC e Amazon Keyspaces. Gli endpoint VPC di interfaccia sono alimentati da AWS PrivateLink, un AWS servizio che consente la comunicazione privata tra e VPCs servizi. AWS PrivateLink consente ciò utilizzando un'interfaccia di rete elastica con dati privati IPs nel tuo VPC in modo che il traffico di rete non lasci la rete Amazon. Gli endpoint VPC di interfaccia non richiedono un gateway Internet, un dispositivo NAT, una connessione VPN o una connessione Direct Connect. Per ulteriori informazioni, consulta [Amazon Virtual Private Cloud and Interface VPC endpoint \(\)](#). AWS PrivateLink Per esempi di policy, consulta [the section called “Utilizzo degli endpoint VPC dell'interfaccia per Amazon Keyspaces”](#).

## AWS Identity and Access Management per Amazon Keyspaces

AWS Identity and Access Management (IAM) è uno strumento Servizio AWS che aiuta un amministratore a controllare in modo sicuro l'accesso alle risorse. AWS Gli amministratori IAM controllano chi può essere autenticato (effettuato l'accesso) e autorizzato (disporre delle autorizzazioni) a utilizzare le risorse Amazon Keyspaces. IAM è uno strumento Servizio AWS che puoi utilizzare senza costi aggiuntivi.

### Argomenti

- [Destinatari](#)
- [Autenticazione con identità](#)
- [Gestione dell'accesso tramite policy](#)
- [Come funziona Amazon Keyspaces con IAM](#)

- [Esempi di policy basate sull'identità di Amazon Keyspaces](#)
- [AWS politiche gestite per Amazon Keyspaces](#)
- [Risoluzione dei problemi relativi all'identità e all'accesso ad Amazon Keyspaces](#)
- [Utilizzo di ruoli collegati ai servizi per Amazon Keyspaces](#)

## 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 (vedi [Risoluzione dei problemi relativi all'identità e all'accesso ad Amazon Keyspaces](#))
- Amministratore del servizio: determina l'accesso degli utenti e invia le richieste di autorizzazione (vedi [Come funziona Amazon Keyspaces con IAM](#))
- Amministratore IAM: scrivi policy per gestire l'accesso (vedi [Esempi di policy basate sull'identità di Amazon Keyspaces](#))

## 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 maggiori informazioni sull'accesso, consultare la sezione [Come accedere a 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 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. Si consiglia vivamente di non utilizzare l'utente root per le attività quotidiane. Per le attività che richiedono le credenziali come utente root, consulta [Attività che richiedono le credenziali dell'utente root](#) nella Guida per l'utente IAM.

## Utenti e gruppi IAM

Un [utente IAM](#) è un'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 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 IAM.

I ruoli IAM sono utili per l'accesso federato degli utenti, le autorizzazioni utente IAM temporanee, l'accesso tra account, l'accesso tra servizi 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 collegandole a identità o risorse. AWS 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 sull'identità possono essere policy in linea (incorporate 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, consultare [Scelta fra policy gestite e policy inline](#) nella Guida per l'utente 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.

## Elenchi di controllo degli accessi ( ) ACLs

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.

Amazon S3 e Amazon VPC sono esempi di servizi che supportano. AWS WAF ACLs Per ulteriori informazioni ACLs, consulta la [panoramica della lista di controllo degli accessi \(ACL\)](#) nella Amazon Simple Storage Service Developer Guide.

## 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: impostano 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 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.
- Le policy di sessione sono policy avanzate che si passano 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 Amazon Keyspaces con IAM

Prima di utilizzare IAM per gestire l'accesso ad Amazon Keyspaces, è necessario comprendere quali funzionalità IAM sono disponibili per l'uso con Amazon Keyspaces. Per avere una visione di alto livello di come Amazon Keyspaces e AWS altri servizi funzionano con IAM, [AWS consulta i servizi che funzionano con](#) IAM nella IAM User Guide.

### Argomenti

- [Politiche basate sull'identità di Amazon Keyspaces](#)
- [Politiche basate sulle risorse di Amazon Keyspaces](#)
- [Autorizzazione basata sui tag Amazon Keyspaces](#)
- [Ruoli IAM di Amazon Keyspaces](#)

## Politiche basate sull'identità di Amazon Keyspaces

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.

Amazon Keyspaces supporta azioni e risorse specifiche e chiavi di condizione. Per informazioni su tutti gli elementi utilizzati in una policy JSON, consulta [Documentazione di riferimento degli elementi delle policy JSON IAM](#) nella Guida per l'utente IAM.

Per visualizzare le risorse e le azioni specifiche del servizio Amazon Keyspaces e le chiavi contestuali delle condizioni che possono essere utilizzate per le policy di autorizzazione IAM, consulta [Actions, resources and condition keys per Amazon Keyspaces \(per Apache Cassandra\)](#) nel Service Authorization Reference.

## Azioni

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 a un criterio. Includere le operazioni in una policy per concedere le autorizzazioni a eseguire l'operazione associata.

Le azioni politiche in Amazon Keyspaces utilizzano il seguente prefisso prima dell'azione: `cassandra:`. Ad esempio, per concedere a qualcuno l'autorizzazione a creare uno spazio di chiavi Amazon Keyspaces con l'istruzione `CREATE CQL` di Amazon Keyspaces, includi l'azione nella sua politica: `cassandra:Create`. Le istruzioni della policy devono includere un elemento `Action` o `NotAction`. Amazon Keyspaces definisce il proprio set di azioni che descrivono le attività che è possibile eseguire con questo servizio.

Per specificare più azioni in una sola istruzione, separa ciascuna di esse con una virgola come mostrato di seguito:

```
"Action": [
 "cassandra:CREATE",
 "cassandra:MODIFY"
]
```

Per visualizzare un elenco delle azioni di Amazon Keyspaces, consulta [Actions Defined by Amazon Keyspaces \(per Apache Cassandra\)](#) nel Service Authorization Reference.

## Resources

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, specificare una risorsa utilizzando il suo [nome della risorsa Amazon \(ARN\)](#). Per le azioni che non supportano le autorizzazioni a livello di risorsa, utilizzare un carattere jolly (\*) per indicare che l'istruzione si applica a tutte le risorse.

```
"Resource": ""
```

In Amazon Keyspaces, gli spazi chiave, le tabelle e gli stream possono essere utilizzati nell'Resourceelemento delle autorizzazioni IAM.

#### Note

Per accedere agli spazi chiave e alle tabelle degli utenti in Amazon Keyspaces, la tua policy IAM deve `cassandra:Select` includere le autorizzazioni sulle tabelle di sistema:

```
arn:${Partition}:cassandra:${Region}:${Account}:/keyspace/system*
```

Ciò si applica ai seguenti scenari:

- AWS Accesso alla console di gestione
- Operazioni relative alle risorse SDK, ad esempio `GetKeyspace`, `GetTableListKeyspaces`, e `ListTables`
- Connessioni standard ai driver del client Apache Cassandra, poiché i driver leggono automaticamente le tabelle di sistema durante l'inizializzazione della connessione

Le tabelle di sistema sono di sola lettura e non possono essere modificate.

La risorsa keyspace di Amazon Keyspaces ha il seguente ARN:

```
arn:${Partition}:cassandra:${Region}:${Account}:/keyspace/${keyspaceName}/
```

La risorsa della tabella Amazon Keyspaces ha il seguente ARN:

```
arn:${Partition}:cassandra:${Region}:${Account}:/keyspace/${keyspaceName}/table/${tableName}
```

La risorsa di flusso Amazon Keyspaces ha il seguente ARN:

```
arn:${Partition}:cassandra:{Region}:${Account}:/keyspace/${keyspaceName}/table/
${tableName}/stream/${streamLabel}
```

Per ulteriori informazioni sul formato di ARNs, consulta [Amazon Resource Names \(ARNs\) e AWS service namespaces](#).

Ad esempio, per specificare mykeyspace lo spazio delle chiavi nella tua dichiarazione, usa il seguente ARN:

```
"Resource": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/mykeyspace/"
```

Per specificare tutti gli spazi chiave che appartengono a un account specifico, usa il carattere jolly (\*):

```
"Resource": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/*"
```

Alcune azioni di Amazon Keyspaces, come quelle per la creazione di risorse, non possono essere eseguite su una risorsa specifica. In questi casi, è necessario utilizzare il carattere jolly (\*).

```
"Resource": ""
```

Ad esempio, per concedere SELECT le autorizzazioni a un principale IAM for mytable inmykeyspace, il principale deve disporre delle autorizzazioni per leggere entrambi, e. mytable keyspace/system\* Per specificare più risorse in una singola istruzione, separale ARNs con virgole.

```
"Resource": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/mykeyspace/table/
mytable",
 "arn:aws:cassandra:us-east-1:111122223333:/keyspace/system"
```

Per visualizzare un elenco dei tipi di risorse Amazon Keyspaces e relativi ARNs, consulta [Resources Defined by Amazon Keyspaces \(per Apache Cassandra\)](#) nel Service Authorization Reference. Per sapere con quali azioni puoi specificare l'ARN di ogni risorsa, consulta [Actions Defined by Amazon Keyspaces \(per Apache Cassandra\)](#).

## Chiavi di condizione

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.

Amazon Keyspaces definisce il proprio set di chiavi di condizione e supporta anche l'utilizzo di alcune chiavi di condizione globali. Per visualizzare tutte le chiavi di condizione AWS globali, consulta le chiavi di [contesto delle condizioni AWS globali](#) nella Guida per l'utente IAM.

Tutte le azioni di Amazon Keyspaces supportano le chiavi `aws:RequestTag/${TagKey}`, `aws:ResourceTag/${TagKey}`, `aws:TagKeys` e `aws:TagKeys` condition. Per ulteriori informazioni, consulta [the section called “Accesso alle risorse Amazon Keyspaces basato su tag”](#).

Per visualizzare un elenco delle chiavi di condizione di Amazon Keyspaces, consulta [Condition Keys for Amazon Keyspaces \(per Apache Cassandra\)](#) nel Service Authorization Reference. Per sapere con quali azioni e risorse puoi utilizzare una chiave di condizione, consulta [Actions Defined by Amazon Keyspaces \(per Apache Cassandra\)](#).

## Esempi

Per visualizzare esempi di policy basate sull'identità di Amazon Keyspaces, consulta [Esempi di policy basate sull'identità di Amazon Keyspaces](#)

## Politiche basate sulle risorse di Amazon Keyspaces

Amazon Keyspaces non supporta politiche basate sulle risorse. Per visualizzare un esempio di una pagina di policy basata su risorse dettagliata, consulta <https://docs.aws.amazon.com/lambda/latest/dg/access-control-resource-based.html>.

## Autorizzazione basata sui tag Amazon Keyspaces

Puoi gestire l'accesso alle tue risorse Amazon Keyspaces utilizzando i tag. Per gestire l'accesso alle risorse in base ai tag, fornisci le informazioni sui tag nell'[elemento condition](#) di una policy utilizzando i tasti `cassandra:ResourceTag/key-name`, `aws:RequestTag/key-name`, o `aws:TagKeys` condition. Per ulteriori informazioni sull'etichettatura delle risorse Amazon Keyspaces, consulta [the section called “Lavorare con i tag”](#)

Per visualizzare policy basate sulle identità di esempio per limitare l'accesso a una risorsa basata su tag su tale risorsa, consulta [Accesso alle risorse Amazon Keyspaces basato su tag](#).

## Ruoli IAM di Amazon Keyspaces

Un [ruolo IAM](#) è un'entità interna all'utente Account AWS che dispone di autorizzazioni specifiche.

Utilizzo di credenziali temporanee con Amazon Keyspaces

Puoi utilizzare credenziali temporanee per effettuare l'accesso utilizzando la federazione, per assumere un ruolo IAM o per assumere un ruolo multi-account. Puoi ottenere credenziali di sicurezza temporanee chiamando operazioni AWS STS API come o. [AssumeRoleGetFederationToken](#)

Amazon Keyspaces supporta l'uso di credenziali temporanee con il plug-in di autenticazione AWS Signature Version 4 (SigV4) disponibile nel repository Github per le seguenti lingue:

- <https://github.com/aws/aws-sigv4-auth-cassandra-java-driver-plugin> Java:.
- Node.js: <https://github.com/aws/aws-sigv4-auth-cassandra-nodejs-driver-plugin>.
- Python: <https://github.com/aws/aws-sigv4-auth-cassandra-python-driver-plugin>.
- Vai: <https://github.com/aws/aws-sigv4-auth-cassandra-gocql-driver-plugin>.

Per esempi e tutorial che implementano il plug-in di autenticazione per accedere ad Amazon Keyspaces a livello di codice, consulta. [the section called “Utilizzo di un driver client Cassandra”](#)

Ruoli collegati ai servizi

[I ruoli collegati ai](#) AWS servizi consentono ai servizi di accedere alle risorse di altri servizi per completare un'azione per tuo conto. I ruoli collegati ai servizi sono visualizzati nell'account IAM e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati ai servizi, ma non può modificarle.

Per informazioni dettagliate sulla creazione o la gestione di ruoli collegati ai servizi Amazon Keyspaces, consulta. [the section called “Uso di ruoli collegati ai servizi”](#)

Ruoli di servizio

Amazon Keyspaces non supporta i ruoli di servizio.

## Esempi di policy basate sull'identità di Amazon Keyspaces

Per impostazione predefinita, gli utenti e i ruoli IAM non dispongono dell'autorizzazione per creare o modificare risorse Amazon Keyspaces. Inoltre, non possono eseguire attività utilizzando la console, CQLSH o l'API AWS CLI. AWS Un amministratore IAM deve creare policy IAM che concedono a utenti e ruoli l'autorizzazione per eseguire operazioni API specifiche sulle risorse specificate di cui

hanno bisogno. L'amministratore deve quindi allegare queste policy a utenti o IAM che richiedono tali autorizzazioni.

Per informazioni su come creare una policy basata su identità IAM utilizzando questi documenti di policy JSON di esempio, consultare [Creazione di policy nella scheda JSON](#) nella Guida per l'utente di IAM.

## Argomenti

- [Best practice delle policy](#)
- [Utilizzo della console Amazon Keyspaces](#)
- [Consentire agli utenti di visualizzare le loro autorizzazioni](#)
- [Accesso alle tabelle Amazon Keyspaces](#)
- [Accesso alle risorse Amazon Keyspaces basato su tag](#)

## Best practice delle policy

Le policy basate sull'identità determinano se qualcuno può creare, accedere o eliminare risorse Amazon Keyspaces nel tuo account. Queste operazioni 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, consultare [Policy gestite da AWS](#) o [Policy gestite da AWS per le funzioni dei processi](#) nella Guida per l'utente 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, consultare [Policy e autorizzazioni in IAM](#) nella Guida per l'utente IAM.
- Condizioni d'uso nelle policy IAM per limitare ulteriormente l'accesso - Per limitare l'accesso a operazioni 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 IAM.

- Utilizzo di IAM Access Analyzer per convalidare le policy IAM e garantire autorizzazioni sicure e funzionali - IAM Access Analyzer convalida le policy nuove ed esistenti in modo che aderiscano alla sintassi della policy IAM (JSON) e alle best practice di IAM. IAM Access Analyzer 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 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 IAM.

Per maggiori informazioni sulle best practice in IAM, consultare [Best practice di sicurezza in IAM](#) nella Guida per l'utente IAM.

## Utilizzo della console Amazon Keyspaces

Amazon Keyspaces non richiede autorizzazioni specifiche per accedere alla console Amazon Keyspaces. Sono necessarie almeno autorizzazioni di sola lettura per elencare e visualizzare i dettagli sulle risorse Amazon Keyspaces presenti nel tuo Account AWS. Se crei una policy basata su identità più restrittiva rispetto alle autorizzazioni minime richieste, la console non funzionerà nel modo previsto per le entità (utenti e ruoli IAM) associate a tale policy.

Le entità hanno a disposizione due policy AWS gestite per l'accesso alla console Amazon Keyspaces.

- [AmazonKeyspacesReadOnlyAccess\\_v2](#) — Questa policy garantisce l'accesso in sola lettura ad Amazon Keyspaces.
- [AmazonKeyspacesFullAccess](#) — Questa politica concede le autorizzazioni per utilizzare Amazon Keyspaces con accesso completo a tutte le funzionalità.

Per ulteriori informazioni sulle politiche gestite di Amazon Keyspaces, consulta [the section called "AWS politiche gestite"](#)

## 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 a livello di codice. 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": "*"
 }
]
}
```

## Accesso alle tabelle Amazon Keyspaces

### Note

Per accedere agli spazi chiave e alle tabelle degli utenti in Amazon Keyspaces, la tua policy IAM deve `cassandra:Select` includere le autorizzazioni sulle tabelle di sistema:

```
arn:${Partition}:cassandra:${Region}:${Account}:/keyspace/system*
```

Ciò si applica ai seguenti scenari:

- AWS Accesso alla console di gestione
- Operazioni relative alle risorse SDK, ad esempio `GetKeyspace`, `GetTableListKeyspaces`, e `ListTables`
- Connessioni standard ai driver del client Apache Cassandra, poiché i driver leggono automaticamente le tabelle di sistema durante l'inizializzazione della connessione

Le tabelle di sistema sono di sola lettura e non possono essere modificate.

Di seguito è riportato un esempio di policy che concede l'accesso in sola lettura (SELECT) alle tabelle di sistema Amazon Keyspaces. Per tutti gli esempi, sostituisci la regione e l'ID dell'account nell'Amazon Resource Name (ARN) con i tuoi.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "cassandra:Select"
],
 "Resource": [
 "arn:aws:cassandra:us-east-1:111122223333:/keyspace/system*"
]
 }
]
}
```

La seguente policy di esempio aggiunge l'accesso in sola lettura alla tabella utente `mytable` nel `keyspace`. `mykeyspace`

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "cassandra:Select"
],
 "Resource": [
 "arn:aws:cassandra:us-east-1:111122223333:/keyspace/mykeyspace/table/mytable",
 "arn:aws:cassandra:us-east-1:111122223333:/keyspace/system*"
]
 }
]
}
```

La seguente politica di esempio assegna read/write l'accesso a una tabella utente e l'accesso in lettura alle tabelle di sistema.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "cassandra:Select",
 "cassandra:Modify"
],
 "Resource": [
 "arn:aws:cassandra:us-east-1:111122223333:/keyspace/mykeyspace/table/mytable",
 "arn:aws:cassandra:us-east-1:111122223333:/keyspace/system*"
]
 }
]
}
```

La seguente politica di esempio consente a un utente di creare tabelle in `mykeyspace` `keyspace`.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "cassandra:Create",
 "cassandra:Select"
],
 "Resource": [
 "arn:aws:cassandra:us-east-1:111122223333:/keyspace/mykeyspace/*",
 "arn:aws:cassandra:us-east-1:111122223333:/keyspace/system*"
]
 }
]
}
```

La seguente politica di esempio assegna l'accesso in lettura alle tabelle di sistema, ma limita l'accesso SELECT (lettura) e MODIFY (scrittura) alla tabella utente. mytable

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "cassandra:Select"
],
 "Resource": [
 "arn:aws:cassandra:us-east-1:111122223333:/keyspace/system*"
]
 },
 {
 "Effect": "Deny",
 "Action": [
 "cassandra:Select",
 "cassandra:Modify"
],
 "Resource": [
 "arn:aws:cassandra:us-east-1:111122223333:/keyspace/mykeyspace/table/mytable"
]
 }
]
}
```

```
}
```

## Accesso alle risorse Amazon Keyspaces basato su tag

Puoi utilizzare le condizioni della tua policy basata sull'identità per controllare l'accesso alle risorse di Amazon Keyspaces in base ai tag. Queste politiche controllano la visibilità degli spazi chiave e delle tabelle nell'account. Tieni presente che le autorizzazioni basate su tag per le tabelle di sistema si comportano in modo diverso quando le richieste vengono effettuate utilizzando l' AWS SDK rispetto alle chiamate API Cassandra Query Language (CQL) tramite i driver Cassandra e gli strumenti di sviluppo.

- Per effettuare `List` richieste di `Get` risorse con l' AWS SDK quando si utilizza l'accesso basato su tag, il chiamante deve avere accesso in lettura alle tabelle di sistema. Ad esempio, sono necessarie le autorizzazioni di `Select` azione per leggere i dati dalle tabelle di sistema tramite l'operazione. `GetTable` Se il chiamante dispone solo dell'accesso basato su tag a una tabella specifica, un'operazione che richiede un accesso aggiuntivo a una tabella di sistema avrà esito negativo.
- Per motivi di compatibilità con il comportamento consolidato dei driver Cassandra, le politiche di autorizzazione basate su tag non vengono applicate quando si eseguono operazioni sulle tabelle di sistema utilizzando chiamate API Cassandra Query Language (CQL) tramite driver Cassandra e strumenti di sviluppo.

L'esempio seguente mostra come è possibile creare una politica che conceda a un utente le autorizzazioni per visualizzare una tabella se la tabella `Owner` contiene il valore del nome utente di quell'utente. In questo esempio si fornisce anche l'accesso in lettura alle tabelle di sistema.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "ReadOnlyAccessTaggedTables",
 "Effect": "Allow",
 "Action": "cassandra:Select",
 "Resource": [
 "arn:aws:cassandra:us-east-1:111122223333:/keyspace/mykeyspace/table/*",
 "arn:aws:cassandra:us-east-1:111122223333:/keyspace/system*"
],
 "Condition": {
 "StringEquals": {
```

```

 "aws:ResourceTag/Owner": "${aws:username}"
 }
}
]
}

```

Puoi allegare questa policy agli utenti IAM nel tuo account. Se un utente denominato `richard-roe` tenta di visualizzare una tabella Amazon Keyspaces, la tabella deve essere contrassegnata con `Owner=richard-roe` o `owner=richard-roe`. In caso contrario, gli viene negato l'accesso. La chiave di tag di condizione `Owner` corrisponde a `Owner` e `owner` perché i nomi delle chiavi di condizione non effettuano la distinzione tra maiuscole e minuscole. Per maggiori informazioni, consultare la sezione [Elementi delle policy JSON di IAM: condizione](#) nella Guida per l'utente IAM.

La seguente politica concede le autorizzazioni a un utente per creare tabelle con tag se la tabella `Owner` contiene il valore del nome utente di quell'utente.

```

{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "CreateTagTableUser",
 "Effect": "Allow",
 "Action": [
 "cassandra:Create",
 "cassandra:TagResource"
],
 "Resource": "arn:aws:cassandra:us-east-1:111122223333:/keyspace/mykeyspace/
table/*",
 "Condition": {
 "StringEquals": {
 "aws:RequestTag/Owner": "${aws:username}"
 }
 }
 }
]
}

```

## AWS politiche gestite per Amazon Keyspaces

Una policy AWS gestita è una policy autonoma creata e amministrata da 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 Ti consigliamo pertanto di ridurre ulteriormente le autorizzazioni definendo [policy gestite dal cliente](#) specifiche per i tuoi casi d'uso.

Non è possibile modificare le autorizzazioni definite nelle politiche gestite. 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 nel 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 politica gestita: AmazonKeyspacesReadOnlyAccess \_v2

È possibile allegare la policy AmazonKeyspacesReadOnlyAccess\_v2 alle identità IAM.

Questa policy garantisce l'accesso in sola lettura ad Amazon Keyspaces e include le autorizzazioni richieste per la connessione tramite endpoint VPC privati.

### Dettagli dell'autorizzazione

Questa policy include le seguenti autorizzazioni:

- **Amazon Keyspaces**— Fornisce accesso in sola lettura ad Amazon Keyspaces.
- **Amazon Keyspaces CDC streams**— Consente ai mandanti di visualizzare gli stream CDC di Amazon Keyspaces.
- **Application Auto Scaling**— Consente ai mandanti di visualizzare le configurazioni da Application Auto Scaling. Ciò è necessario per consentire agli utenti di visualizzare le politiche di ridimensionamento automatico allegate a una tabella.



- **CloudWatch**— Consente ai responsabili di visualizzare i dati metrici e gli allarmi configurati in. CloudWatch Ciò è necessario per consentire agli utenti di visualizzare le dimensioni fatturabili della tabella e gli CloudWatch allarmi configurati per una tabella.
- **AWS KMS**— Consente ai presidi di visualizzare le chiavi configurate in. AWS KMS Ciò è necessario affinché gli utenti possano visualizzare AWS KMS le chiavi che creano e gestiscono nel proprio account per confermare che la chiave assegnata ad Amazon Keyspaces è una chiave di crittografia simmetrica abilitata.
- **Amazon EC2**— Consente ai responsabili di connettersi ad Amazon Keyspaces tramite endpoint VPC di interrogare il VPC sulla tua istanza EC2 Amazon per ottenere informazioni sull'endpoint e sull'interfaccia di rete. Questo accesso in sola lettura all' EC2 istanza Amazon è necessario per consentire ad Amazon Keyspaces di cercare e archiviare gli endpoint VPC dell'interfaccia disponibili nella tabella utilizzata per il bilanciamento del `system.peers` carico di connessione.

[Per rivedere il formato della policy, consulta `\_v2.JSON AmazonKeyspacesReadOnlyAccess`](#)

## AWS politica gestita: AmazonKeyspacesReadOnlyAccess

È possibile allegare la policy AmazonKeyspacesReadOnlyAccess alle identità IAM.

Questa politica garantisce l'accesso in sola lettura ad Amazon Keyspaces.

### Dettagli dell'autorizzazione

Questa policy include le seguenti autorizzazioni:

- **Amazon Keyspaces**— Fornisce accesso in sola lettura ad Amazon Keyspaces.
- **Amazon Keyspaces CDC streams**— Consente ai mandanti di visualizzare gli stream CDC di Amazon Keyspaces.
- **Application Auto Scaling**— Consente ai mandanti di visualizzare le configurazioni da Application Auto Scaling. Ciò è necessario per consentire agli utenti di visualizzare le politiche di ridimensionamento automatico allegate a una tabella.
- **CloudWatch**— Consente ai responsabili di visualizzare i dati metrici e gli allarmi configurati in. CloudWatch Ciò è necessario per consentire agli utenti di visualizzare le dimensioni fatturabili della tabella e gli CloudWatch allarmi configurati per una tabella.

- **AWS KMS**— Consente ai presidi di visualizzare le chiavi configurate in. AWS KMS Ciò è necessario affinché gli utenti possano visualizzare AWS KMS le chiavi che creano e gestiscono nel proprio account per confermare che la chiave assegnata ad Amazon Keyspaces è una chiave di crittografia simmetrica abilitata.

Per rivedere il JSON formato della policy, consulta. [AmazonKeyspacesReadOnlyAccess](#)

## AWS politica gestita: AmazonKeyspacesFullAccess

È possibile allegare la policy AmazonKeyspacesFullAccess alle identità IAM.

Questa politica concede autorizzazioni amministrative che consentono agli amministratori l'accesso illimitato ad Amazon Keyspaces.

### Dettagli dell'autorizzazione

Questa policy include le seguenti autorizzazioni:

- **Amazon Keyspaces**— Consente ai mandanti di accedere a qualsiasi risorsa Amazon Keyspaces ed eseguire tutte le azioni.
- **Application Auto Scaling**— Consente ai responsabili di creare, visualizzare ed eliminare politiche di ridimensionamento automatico per le tabelle Amazon Keyspaces. Ciò è necessario per consentire agli amministratori di gestire le politiche di scalabilità automatica per le tabelle Amazon Keyspaces.
- **CloudWatch**— Consente ai responsabili di visualizzare le dimensioni fatturabili della tabella e di creare, visualizzare ed eliminare CloudWatch allarmi per le politiche di scalabilità automatica di Amazon Keyspaces. Ciò è necessario per consentire agli amministratori di visualizzare le dimensioni fatturabili della tabella e creare una dashboard. CloudWatch
- **IAM**— Consente ad Amazon Keyspaces di creare automaticamente ruoli collegati ai servizi con IAM quando sono attivate le seguenti funzionalità:
  - **Amazon Keyspaces CDC streams**— Quando un amministratore abilita uno stream per una tabella, Amazon Keyspaces crea il ruolo [AWSServiceRoleForAmazonKeyspacesCDC](#) collegato al servizio per pubblicare le CloudWatch metriche nel tuo account per tuo conto.

- **Application Auto Scaling**— Quando un amministratore abilita Application Auto Scaling per una tabella, Amazon Keyspaces crea il ruolo collegato al servizio per eseguire azioni di scalabilità automatica [AWSServiceRoleForApplicationAutoScaling\\_CassandraTable](#) per tuo conto.
- **Amazon Keyspaces multi-Region replication**— Quando un amministratore crea un nuovo keyspace multiregionale o ne aggiunge uno nuovo Regione AWS a uno esistente per una singola regione, Amazon Keyspaces crea il [AWSServiceRoleForAmazonKeyspacesReplication](#) ruolo collegato al servizio per eseguire la replica di tabelle, dati e metadati nelle regioni selezionate per tuo conto.
- **AWS KMS**— Consente ai principali di visualizzare le chiavi configurate in AWS KMS. Ciò è necessario affinché gli utenti possano visualizzare AWS KMS le chiavi che creano e gestiscono nel proprio account per confermare che la chiave assegnata ad Amazon Keyspaces è una chiave di crittografia simmetrica abilitata.
- **Amazon EC2**— Consente ai responsabili di connettersi ad Amazon Keyspaces tramite endpoint VPC di interrogare il VPC sulla tua istanza EC2 Amazon per ottenere informazioni sull'endpoint e sull'interfaccia di rete. Questo accesso in sola lettura all' EC2 istanza Amazon è necessario per consentire ad Amazon Keyspaces di cercare e archiviare gli endpoint VPC dell'interfaccia disponibili nella tabella utilizzata per il bilanciamento del `system.peers` carico di connessione.

Per rivedere il formato della policy, consulta. JSON [AmazonKeyspacesFullAccess](#)

## AWS politica gestita: Keyspaces CDCService RolePolicy

Non è possibile collegare `KeyspacesCDCServiceRolePolicy` alle entità IAM. Questa policy è associata a un ruolo collegato al servizio che consente ad Amazon Keyspaces di eseguire azioni per tuo conto. Per ulteriori informazioni, consulta [the section called “Stream CDC”](#).

Questa politica concede le autorizzazioni necessarie al ruolo collegato al servizio per pubblicare i dati delle metriche di flusso CDC di `AWSServiceRoleForAmazonKeyspacesCDC` Amazon Keyspaces per tuo conto. CloudWatch

### Dettagli dell'autorizzazione

Questa policy include le seguenti autorizzazioni:

- CloudWatch— Consente al service-linked-role [AWSServiceRoleForAmazonKeyspacesCDC](#) di pubblicare i dati metrici dai flussi CDC di Amazon Keyspaces "cloudwatch:namespace": "AWS/Cassandra" nel tuo account per tuo conto. CloudWatch

Per rivedere il JSON formato della politica, consulta [Keyspaces CDCService RolePolicy](#).

## Amazon Keyspaces si aggiorna alle AWS policy gestite

Visualizza i dettagli sugli aggiornamenti delle politiche AWS gestite per Amazon Keyspaces da quando questo servizio ha iniziato a tracciare queste modifiche. Per gli avvisi automatici sulle modifiche apportate alla pagina, iscriviti al feed RSS alla pagina [Cronologia dei documenti](#).

| Modifica                                                                                | Descrizione                                                                                                                                                                                                                                                                                                                                                                                 | Data           |
|-----------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|
| <a href="#">Keyspaces CDCService RolePolicy — Nuova</a> politica                        | Amazon Keyspaces ha aggiunto una nuova policy gestita Keyspaces CDCServiceRolePolicy che concede le autorizzazioni necessarie al ruolo collegato al servizio per AWSServiceRoleForAmazonKeyspacesCDC pubblicare i dati delle metriche di streaming CDC di Amazon Keyspaces per tuo conto. CloudWatch Per ulteriori informazioni, consulta <a href="#">the section called “Stream CDC”</a> . | 02 luglio 2025 |
| <a href="#">AmazonKeyspacesReadOnlyAccess_v2</a> — Aggiornamento a una policy esistente | Amazon Keyspaces ha aggiunto nuove autorizzazioni per consentire ai responsabili IAM di visualizzare i flussi                                                                                                                                                                                                                                                                               | 02 luglio 2025 |

| Modifica                                                                             | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Data           |
|--------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|
|                                                                                      | CDC di Amazon Keyspaces<br>. Per ulteriori informazioni, consulta <a href="#">the section called “Visualizza gli stream CDC”</a> .                                                                                                                                                                                                                                                                                                                                                                                                          |                |
| <a href="#">AmazonKeyspacesReadOnlyAccess</a> : aggiornamento a una policy esistente | Amazon Keyspaces ha aggiunto nuove autorizzazioni per consentire ai responsabili IAM di visualizzare i flussi CDC di Amazon Keyspaces<br>. Per ulteriori informazioni, consulta <a href="#">the section called “Visualizza gli stream CDC”</a> .                                                                                                                                                                                                                                                                                            | 02 luglio 2025 |
| <a href="#">AmazonKeyspacesFullAccess</a> : aggiornamento a una policy esistente     | Amazon Keyspaces ha creato la policy Keyspaces CDCServiceRolePolicy gestita per il ruolo <a href="#">AWSServiceRoleForAmazonKeyspacesCDC</a> collegato al servizio per aggiungere le autorizzazioni necessarie quando un amministratore abilita uno stream per una tabella.<br><br>Amazon Keyspaces utilizza il ruolo collegato al servizio AWSServiceRoleForAmazonKeyspacesCDC per pubblicare i CloudWatch parametri nel tuo account per tuo conto. Per ulteriori informazioni, consulta <a href="#">the section called “Stream CDC”</a> . | 02 luglio 2025 |

| Modifica                                                                         | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Data             |
|----------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| <a href="#">AmazonKeyspacesFullAccess</a> : aggiornamento a una policy esistente | <p>Amazon Keyspaces ha aggiornato il Keyspaces ReplicationService RolePolicy ruolo collegato al servizio <a href="#">AWSServiceRoleForAmazonKeyspacesReplication</a> per aggiungere le autorizzazioni necessarie quando un amministratore ne aggiunge uno nuovo Regione AWS a un keyspace singolo o multiregionale.</p> <p>Amazon Keyspaces utilizza il ruolo collegato al servizio <a href="#">AWSServiceRoleForAmazonKeyspacesReplication</a> per replicare le tabelle, le relative impostazioni e i dati per tuo conto. Per ulteriori informazioni, consulta <a href="#">the section called “Replica in più regioni”</a>.</p> | 19 novembre 2024 |

| Modifica                                                                           | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Data           |
|------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|
| <a href="#">AmazonKeyspacesFullAccess:</a><br>aggiornamento a una policy esistente | <p>Amazon Keyspaces ha aggiunto nuove autorizzazioni per consentire ad Amazon Keyspaces di creare un ruolo collegato ai servizi quando un amministratore aggiunge una nuova regione a uno spazio di chiavi singolo o multiregionale.</p> <p>Amazon Keyspaces utilizza il ruolo collegato al servizio per eseguire attività di replica dei dati per tuo conto. Per ulteriori informazioni, consulta <a href="#">the section called “Replica in più regioni”</a>.</p> | 3 ottobre 2023 |

| Modifica                                                          | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Data              |
|-------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| <a href="#">AmazonKeyspacesReadOnlyAccess_v2 — Nuova politica</a> | <p>Amazon Keyspaces ha creato una nuova policy per aggiungere autorizzazioni di sola lettura per i client che si connettono ad Amazon Keyspaces tramite endpoint VPC di interfaccia per accedere all'istanza Amazon per cercare informazioni di rete. EC2</p> <p>Amazon Keyspaces memorizza gli endpoint VPC di interfaccia disponibili nella <code>system.peers</code> tabella per il bilanciamento del carico di connessione. Per ulteriori informazioni, consulta <a href="#">the section called “Utilizzo di endpoint VPC dell'interfaccia ”</a>.</p> | 12 settembre 2023 |



| Modifica                                                                             | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Data          |
|--------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|
| <a href="#">AmazonKeyspacesFullAccess</a> : aggiornamento a una policy esistente     | <p>Amazon Keyspaces ha aggiunto nuove autorizzazioni per consentire ad Amazon Keyspaces di creare un ruolo collegato al servizio quando un amministratore crea uno spazio chiave multiregionale.</p> <p>Amazon Keyspaces utilizza il ruolo collegato al servizio <code>AWSServiceRoleForAmazonKeyspacesReplication</code> per eseguire attività di replica dei dati per tuo conto. Per ulteriori informazioni, consulta <a href="#">the section called “Replica in più regioni”</a>.</p> | 5 giugno 2023 |
| <a href="#">AmazonKeyspacesReadOnlyAccess</a> : aggiornamento a una policy esistente | <p>Amazon Keyspaces ha aggiunto nuove autorizzazioni per consentire agli utenti di visualizzare le dimensioni fatturabili di una tabella utilizzando CloudWatch.</p> <p>Amazon Keyspaces si integra con Amazon CloudWatch per consentirti di monitorare le dimensioni fatturabili delle tabelle. Per ulteriori informazioni, consulta <a href="#">the section called “Parametri di Amazon Keyspaces”</a>.</p>                                                                            | 7 luglio 2022 |

| Modifica                                                                               | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                   | Data          |
|----------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|
| <a href="#">AmazonKeyspacesFullAccess</a> :<br>aggiornamento a una policy<br>esistente | <p>Amazon Keyspaces ha aggiunto nuove autorizzazioni per consentire agli utenti di visualizzare le dimensioni fatturabili di una tabella utilizzando CloudWatch.</p> <p>Amazon Keyspaces si integra con Amazon CloudWatch per consentirti di monitorare le dimensioni fatturabili delle tabelle. Per ulteriori informazioni, consulta <a href="#">the section called “Parametri di Amazon Keyspaces”</a>.</p> | 7 luglio 2022 |

| Modifica                                                                             | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Data          |
|--------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|
| <a href="#">AmazonKeyspacesReadOnlyAccess</a> : aggiornamento a una policy esistente | <p>Amazon Keyspaces ha aggiunto nuove autorizzazioni per consentire agli utenti di visualizzare le AWS KMS chiavi che sono state configurate per la crittografia Amazon Keyspaces quando sono inutilizzate.</p> <p>La crittografia a riposo di Amazon Keyspaces si integra con la protezione e la gestione delle chiavi di crittografia utilizzate AWS KMS per crittografare i dati inattivi. Per visualizzare la AWS KMS chiave configurata per Amazon Keyspaces, sono state aggiunte autorizzazioni di sola lettura.</p> | 1 giugno 2021 |

| Modifica                                                                           | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Data          |
|------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|
| <a href="#">AmazonKeyspacesFullAccess:</a><br>aggiornamento a una policy esistente | <p>Amazon Keyspaces ha aggiunto nuove autorizzazioni per consentire agli utenti di visualizzare le AWS KMS chiavi che sono state configurate per la crittografia Amazon Keyspaces quando sono inutilizzate.</p> <p>La crittografia a riposo di Amazon Keyspaces si integra con la protezione e la gestione delle chiavi di crittografia utilizzate AWS KMS per crittografare i dati inattivi. Per visualizzare la AWS KMS chiave configurata per Amazon Keyspaces, sono state aggiunte autorizzazioni di sola lettura.</p> | 1 giugno 2021 |
| Amazon Keyspaces ha iniziato a tracciare le modifiche                              | Amazon Keyspaces ha iniziato a tracciare le modifiche per le sue politiche AWS gestite.                                                                                                                                                                                                                                                                                                                                                                                                                                    | 1 giugno 2021 |

## Risoluzione dei problemi relativi all'identità e all'accesso ad Amazon Keyspaces

Utilizza le seguenti informazioni per aiutarti a diagnosticare e risolvere i problemi più comuni che potresti riscontrare quando lavori con Amazon Keyspaces e IAM.

### Argomenti

- [Non sono autorizzato a eseguire un'azione in Amazon Keyspaces](#)
- [Ho modificato un utente o un ruolo IAM e le modifiche non hanno avuto effetto immediato](#)

- [Non riesco a ripristinare una tabella utilizzando Amazon Keyspaces point-in-time recovery \(PITR\)](#)
- [Non sono autorizzato a eseguire iam: PassRole](#)
- [Sono un amministratore e desidero consentire ad altri di accedere ad Amazon Keyspaces](#)
- [Voglio consentire a persone esterne a me di accedere Account AWS alle mie risorse Amazon Keyspaces](#)

## Non sono autorizzato a eseguire un'azione in Amazon Keyspaces

Se ti Console di gestione AWS dice che non sei autorizzato a eseguire un'azione, devi contattare l'amministratore per ricevere assistenza. L'amministratore è la persona da cui si sono ricevuti il nome utente e la password.

Il seguente errore di esempio si verifica quando l'utente mateojackson IAM tenta di utilizzare la console per visualizzare i dettagli su a *table* ma non dispone cassandra:*Select* delle autorizzazioni per la tabella.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
cassandra:Select on resource: mytable
```

In questo caso, Mateo richiede al suo amministratore di aggiornare le policy per poter accedere alla risorsa *mytable* utilizzando l'azione cassandra:*Select*.

## Ho modificato un utente o un ruolo IAM e le modifiche non hanno avuto effetto immediato

Le modifiche alle policy IAM possono richiedere fino a 10 minuti per avere effetto per le applicazioni con connessioni esistenti e stabilite ad Amazon Keyspaces. Le modifiche alle policy IAM hanno effetto immediato quando le applicazioni stabiliscono una nuova connessione. Se hai apportato modifiche a un utente o a un ruolo IAM esistente e non hanno avuto effetto immediato, attendi 10 minuti o disconnettiti e riconnettiti ad Amazon Keyspaces.

## Non riesco a ripristinare una tabella utilizzando Amazon Keyspaces point-in-time recovery (PITR)

Se stai tentando di ripristinare una tabella Amazon Keyspaces con point-in-time ripristino (PITR) e vedi che il processo di ripristino inizia, ma non viene completato correttamente, potresti non aver configurato tutte le autorizzazioni richieste necessarie per il processo di ripristino. Devi contattare il

tuo amministratore per ricevere assistenza e chiedere a quella persona di aggiornare le tue politiche per consentirti di ripristinare una tabella in Amazon Keyspaces.

Oltre alle autorizzazioni utente, Amazon Keyspaces può richiedere le autorizzazioni per eseguire azioni durante il processo di ripristino per conto del tuo principale. Questo è il caso se la tabella è crittografata con una chiave gestita dal cliente o se utilizzi politiche IAM che limitano il traffico in entrata. Ad esempio, se utilizzi le chiavi di condizione nella tua policy IAM per limitare il traffico di origine a endpoint o intervalli IP specifici, l'operazione di ripristino non riesce. Per consentire ad Amazon Keyspaces di eseguire l'operazione di ripristino della tabella per conto del tuo principale, devi aggiungere una chiave di condizione `aws:ViaAWSService` globale nella policy IAM.

Per ulteriori informazioni sulle autorizzazioni per ripristinare le tabelle, consulta [the section called “Configura le autorizzazioni IAM per il ripristino”](#)

## Non sono autorizzato a eseguire iam: PassRole

Se ricevi un messaggio di errore indicante che non sei autorizzato a eseguire l'azione `iam:PassRole`, le tue politiche devono essere aggiornate per consentirti di trasferire un ruolo ad Amazon Keyspaces.

Alcuni Servizi AWS consentono di trasferire 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.

Il seguente errore di esempio si verifica quando un utente IAM denominato `marymajor` tenta di utilizzare la console per eseguire un'azione in Amazon Keyspaces. 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.

## Sono un amministratore e desidero consentire ad altri di accedere ad Amazon Keyspaces

Per consentire ad altri di accedere ad Amazon Keyspaces, devi concedere l'autorizzazione alle persone o alle applicazioni che devono accedervi. Se lo utilizzi AWS IAM Identity Center per gestire persone e applicazioni, assegna set di autorizzazioni a utenti o gruppi per definirne il livello di accesso. I set di autorizzazioni creano e assegnano automaticamente le policy IAM ai ruoli IAM associati alla persona o all'applicazione. Per ulteriori informazioni, consulta [Set di autorizzazioni](#) nella Guida per l'AWS IAM Identity Center utente.

Se non utilizzi IAM Identity Center, devi creare entità IAM (utenti o ruoli) per le persone o le applicazioni che necessitano di accesso. Devi quindi allegare una policy all'entità che concede loro le autorizzazioni corrette in Amazon Keyspaces. Dopo aver concesso le autorizzazioni, fornisci le credenziali all'utente o allo sviluppatore dell'applicazione. Utilizzeranno tali credenziali per accedere. AWS Per ulteriori informazioni sulla creazione di utenti, gruppi, policy e autorizzazioni IAM, consulta [IAM Identities](#) and [Policies and permissions in IAM nella IAM](#) User Guide.

## Voglio consentire a persone esterne a me di accedere Account AWS alle mie risorse Amazon Keyspaces

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, consultare gli argomenti seguenti:

- Per sapere se Amazon Keyspaces supporta queste funzionalità, consulta. [Come funziona Amazon Keyspaces con IAM](#)
- Per sapere come fornire l'accesso alle tue risorse su tutto Account AWS ciò che possiedi, consulta [Fornire l'accesso a un utente IAM in un altro Account AWS 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, consultare [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.

## Utilizzo di ruoli collegati ai servizi per Amazon Keyspaces

[Amazon Keyspaces \(per Apache Cassandra\) utilizza ruoli collegati ai servizi AWS Identity and Access Management \(IAM\)](#). Un ruolo collegato ai servizi è un tipo unico di ruolo IAM collegato direttamente ad Amazon Keyspaces. I ruoli collegati ai servizi sono predefiniti da Amazon Keyspaces e includono tutte le autorizzazioni richieste dal servizio per chiamare altri servizi per tuo conto. AWS

Per informazioni su altri servizi che supportano i ruoli collegati ai servizi, consulta i [AWS servizi che funzionano con IAM e cerca i servizi con](#) Sì nella colonna Ruoli collegati ai servizi. Scegli Sì in corrispondenza di un link per visualizzare la documentazione relativa al ruolo collegato al servizio per tale servizio.

### Argomenti

- [Utilizzo dei ruoli per la scalabilità automatica delle applicazioni Amazon Keyspaces](#)
- [Utilizzo dei ruoli per la replica multiregionale di Amazon Keyspaces](#)
- [Utilizzo dei ruoli per gli stream CDC di Amazon Keyspaces](#)

## Utilizzo dei ruoli per la scalabilità automatica delle applicazioni Amazon Keyspaces

[Amazon Keyspaces \(per Apache Cassandra\) utilizza ruoli collegati ai servizi AWS Identity and Access Management \(IAM\)](#). Un ruolo collegato ai servizi è un tipo unico di ruolo IAM collegato direttamente ad Amazon Keyspaces. I ruoli collegati ai servizi sono predefiniti da Amazon Keyspaces e includono tutte le autorizzazioni richieste dal servizio per chiamare altri servizi per tuo conto. AWS

Un ruolo collegato al servizio semplifica la configurazione di Amazon Keyspaces perché non è necessario aggiungere manualmente le autorizzazioni necessarie. Amazon Keyspaces definisce le autorizzazioni dei suoi ruoli collegati ai servizi e, se non diversamente definito, solo Amazon Keyspaces può assumerne i ruoli. Le autorizzazioni definite includono la policy di attendibilità e la policy delle autorizzazioni che non può essere collegata a nessun'altra entità IAM.

È possibile eliminare un ruolo collegato ai servizi solo dopo aver eliminato le risorse correlate. In questo modo proteggi le tue risorse Amazon Keyspaces perché non puoi rimuovere inavvertitamente l'autorizzazione ad accedere alle risorse.

### Autorizzazioni di ruolo collegate ai servizi per Amazon Keyspaces

Amazon Keyspaces utilizza il ruolo collegato al servizio denominato `AWSServiceRoleForApplicationAutoScaling_CassandraTable` per consentire ad Application Auto Scaling di chiamare Amazon Keyspaces e Amazon per tuo conto. CloudWatch



Il ruolo `AWSServiceRoleForApplicationAutoScaling_CassandraTable` collegato al servizio prevede che i seguenti servizi assumano il ruolo:

- `cassandra.application-autoscaling.amazonaws.com`

La politica di autorizzazione dei ruoli consente ad Application Auto Scaling di completare le seguenti azioni sulle risorse Amazon Keyspaces specificate:

- Operazione: `cassandra:Select` su `arn:*:cassandra:*:*:/keyspace/system/table/*`
- Operazione: `cassandra:Select` sulla risorsa `arn:*:cassandra:*:*:/keyspace/system_schema/table/*`
- Operazione: `cassandra:Select` sulla risorsa `arn:*:cassandra:*:*:/keyspace/system_schema_mcs/table/*`
- Operazione: `cassandra:Alter` sulla risorsa `arn:*:cassandra:*:*:""`

## Creazione di un ruolo collegato ai servizi per Amazon Keyspaces

Non è necessario creare manualmente un ruolo collegato ai servizi per la scalabilità automatica di Amazon Keyspaces. Quando abiliti la scalabilità automatica di Amazon Keyspaces su una tabella con Console di gestione AWS, CQL, o l' AWS CLI AWS API, Application Auto Scaling crea il ruolo collegato al servizio per te.

Se elimini questo ruolo collegato ai servizi, è possibile ricrearlo seguendo lo stesso processo utilizzato per ricreare il ruolo nell'account. Quando abiliti la scalabilità automatica di Amazon Keyspaces per una tabella, Application Auto Scaling crea nuovamente il ruolo collegato al servizio per te.

### Important

Questo ruolo collegato al servizio può apparire nell'account, se è stata completata un'operazione in un altro servizio che utilizza le caratteristiche supportate da questo ruolo. Per ulteriori informazioni, consulta A [new role appeared in my](#). Account AWS

## Modifica di un ruolo collegato al servizio per Amazon Keyspaces

Amazon Keyspaces non consente di modificare il ruolo collegato al `AWSServiceRoleForApplicationAutoScaling_CassandraTable` servizio. Dopo aver creato un ruolo

collegato al servizio, non potrai modificarne il nome perché varie entità potrebbero farvi riferimento. È possibile tuttavia modificarne la descrizione utilizzando IAM. Per ulteriori informazioni, consulta [Modifica di un ruolo collegato ai servizi](#) nella Guida per l'utente di IAM.

## Eliminazione di un ruolo collegato al servizio per Amazon Keyspaces

Se non è più necessario utilizzare una funzionalità o un servizio che richiede un ruolo collegato al servizio, ti consigliamo di eliminare il ruolo. In questo modo non hai un'entità non utilizzata che non viene monitorata o gestita attivamente. Tuttavia, devi prima disabilitare il ridimensionamento automatico su tutte le tabelle dell'account Regioni AWS prima di poter eliminare manualmente il ruolo collegato al servizio. Per disabilitare il ridimensionamento automatico sulle tabelle Amazon Keyspaces, consulta [the section called “Disattiva la scalabilità automatica di Amazon Keyspaces per una tabella”](#)

### Note

Se il ridimensionamento automatico di Amazon Keyspaces utilizza il ruolo quando tenti di modificare le risorse, l'annullamento della registrazione potrebbe non riuscire. In questo caso, attendi alcuni minuti e quindi ripeti l'operazione.

Per eliminare manualmente il ruolo collegato ai servizi mediante IAM

Utilizza la console IAM AWS CLI, o l' AWS API per eliminare il ruolo collegato al servizio. `AWSServiceRoleForApplicationAutoScaling_CassandraTable` Per ulteriori informazioni, consulta [Eliminazione del ruolo collegato ai servizi](#) nella Guida per l'utente di IAM.

### Note

Per eliminare il ruolo collegato al servizio utilizzato dalla scalabilità automatica di Amazon Keyspaces, devi prima disabilitare la scalabilità automatica su tutte le tabelle dell'account.

## Regioni supportate per i ruoli collegati ai servizi Amazon Keyspaces

Amazon Keyspaces supporta l'utilizzo di ruoli collegati ai servizi in tutte le regioni in cui il servizio è disponibile. Per ulteriori informazioni, consulta [Service endpoints for Amazon Keyspaces](#).

## Utilizzo dei ruoli per la replica multiregionale di Amazon Keyspaces

[Amazon Keyspaces \(per Apache Cassandra\) utilizza ruoli collegati ai servizi AWS Identity and Access Management \(IAM\)](#). Un ruolo collegato ai servizi è un tipo unico di ruolo IAM collegato direttamente ad Amazon Keyspaces. I ruoli collegati ai servizi sono predefiniti da Amazon Keyspaces e includono tutte le autorizzazioni richieste dal servizio per chiamare altri servizi per tuo conto. AWS

Un ruolo collegato al servizio semplifica la configurazione di Amazon Keyspaces perché non è necessario aggiungere manualmente le autorizzazioni necessarie. Amazon Keyspaces definisce le autorizzazioni dei suoi ruoli collegati ai servizi e, se non diversamente definito, solo Amazon Keyspaces può assumerne i ruoli. Le autorizzazioni definite includono la policy di attendibilità e la policy delle autorizzazioni che non può essere collegata a nessun'altra entità IAM.

È possibile eliminare un ruolo collegato ai servizi solo dopo aver eliminato le risorse correlate. In questo modo proteggi le tue risorse Amazon Keyspaces perché non puoi rimuovere inavvertitamente l'autorizzazione ad accedere alle risorse.

### Autorizzazioni di ruolo collegate ai servizi per Amazon Keyspaces

Amazon Keyspaces utilizza il ruolo collegato al servizio denominato per consentire ad `AWSServiceRoleForAmazonKeyspacesReplication` Amazon Keyspaces di aggiungerne di nuovi a un keyspace Regioni AWS per tuo conto e di replicare le tabelle e tutti i relativi dati e impostazioni nella nuova regione. Il ruolo consente inoltre ad Amazon Keyspaces di replicare le scritture su tabelle in tutte le regioni per tuo conto.

Il ruolo `AWSServiceRoleForAmazonKeyspacesReplication` collegato al servizio prevede che i seguenti servizi assumano il ruolo:

- `replication.cassandra.amazonaws.com`

La politica di autorizzazione dei ruoli denominata `KeyspacesReplicationServiceRolePolicy` consente ad Amazon Keyspaces di completare le seguenti azioni:

- Operazione: `cassandra:Select`
- Operazione: `cassandra:SelectMultiRegionResource`
- Operazione: `cassandra:Modify`
- Operazione: `cassandra:ModifyMultiRegionResource`
- Operazione: `cassandra:AlterMultiRegionResource`

- Azione: `application-autoscaling:RegisterScalableTarget` — Amazon Keyspaces utilizza le autorizzazioni di auto scaling dell'applicazione quando aggiungi una replica a una singola tabella regionale in modalità provisioning con la scalabilità automatica abilitata.
- Operazione: `application-autoscaling:DeregisterScalableTarget`
- Operazione: `application-autoscaling:DescribeScalableTargets`
- Operazione: `application-autoscaling:PutScalingPolicy`
- Operazione: `application-autoscaling:DescribeScalingPolicies`
- Operazione: `cassandra:Alter`
- Operazione: `cloudwatch>DeleteAlarms`
- Operazione: `cloudwatch:DescribeAlarms`
- Operazione: `cloudwatch:PutMetricAlarm`

Sebbene il ruolo collegato al servizio Amazon Keyspaces `AWSService`

`RoleForAmazonKeyspacesReplication` fornisca le autorizzazioni: «Action:» per l'Amazon Resource Name (ARN) «arn: \*» specificato nella policy, Amazon Keyspaces fornisce l'ARN del tuo account.

Le autorizzazioni per creare il ruolo collegato al servizio sono incluse nella policy gestita. `AWSService` `RoleForAmazonKeyspacesReplication` `AmazonKeyspacesFullAccess` Per ulteriori informazioni, consulta [the section called “AmazonKeyspacesFullAccess”](#).

Per consentire a utenti, gruppi o ruoli di creare, modificare o eliminare un ruolo orientato ai servizi, devi configurare le autorizzazioni. Per ulteriori informazioni, consulta [Autorizzazioni del ruolo collegato ai servizi](#) nella Guida per l'utente di IAM.

### Creazione di un ruolo collegato ai servizi per Amazon Keyspaces

Non puoi creare manualmente un ruolo collegato al servizio. Quando crei uno spazio chiave multiregionale nell' AWS API Console di gestione AWS, Amazon Keyspaces crea automaticamente il ruolo collegato al servizio. AWS CLI

Se elimini questo ruolo collegato ai servizi, è possibile ricrearlo seguendo lo stesso processo utilizzato per ricreare il ruolo nell'account. Quando crei uno spazio di chiavi multiregionale, Amazon Keyspaces crea nuovamente il ruolo collegato al servizio per te.

### Modifica di un ruolo collegato al servizio per Amazon Keyspaces

Amazon Keyspaces non consente di modificare il ruolo collegato al `AWSService`

`RoleForAmazonKeyspacesReplication` servizio. Dopo aver creato un ruolo collegato al servizio,

non potrai modificarne il nome perché varie entità potrebbero farvi riferimento. È possibile tuttavia modificarne la descrizione utilizzando IAM. Per ulteriori informazioni, consulta [Modifica di un ruolo collegato ai servizi](#) nella Guida per l'utente di IAM.

## Eliminazione di un ruolo collegato al servizio per Amazon Keyspaces

Se non è più necessario utilizzare una funzionalità o un servizio che richiede un ruolo collegato al servizio, ti consigliamo di eliminare il ruolo. In questo modo non sarà più presente un'entità non utilizzata che non viene monitorata e gestita attivamente. Tuttavia, devi prima eliminare tutti gli spazi chiave multiregionali dell'account Regioni AWS prima di poter eliminare manualmente il ruolo collegato al servizio.

## Pulizia di un ruolo collegato ai servizi

Prima di poter utilizzare IAM per eliminare un ruolo collegato al servizio, devi prima eliminare tutti gli spazi chiave e le tabelle multiregionali utilizzati dal ruolo.

### Note

Se il servizio Amazon Keyspaces utilizza il ruolo quando tenti di eliminare le risorse, l'eliminazione potrebbe non riuscire. In questo caso, attendi alcuni minuti e quindi ripeti l'operazione.

Per eliminare le risorse Amazon Keyspaces utilizzate da AWSService RoleForAmazonKeyspacesReplication (console)

1. [Accedi a e apri Console di gestione AWS la console Amazon Keyspaces a casa](https://console.aws.amazon.com/keyspaces/)<https://console.aws.amazon.com/keyspaces/>.
2. Scegliete Keyspaces dal pannello a sinistra.
3. Seleziona tutti gli spazi chiave multiregionali dall'elenco.
4. Scegli Elimina, conferma l'eliminazione e scegli Elimina spazi chiave.

Puoi anche eliminare gli spazi chiave multiregione a livello di codice utilizzando uno dei seguenti metodi.

- L'istruzione Cassandra Query Language (CQL). [???](#)
- L'operazione [delete-keyspace](#) della CLI. AWS

- Il [DeleteKeyspace](#) funzionamento dell'API Amazon Keyspaces.

## Eliminazione manuale del ruolo collegato ai servizi

Utilizza la console IAM AWS CLI, l'AWS API per eliminare il ruolo collegato al AWSService RoleForAmazonKeyspacesReplication servizio. Per ulteriori informazioni, consulta [Eliminazione del ruolo collegato al servizio](#) nella Guida per l'utente di IAM.

## Regioni supportate per i ruoli collegati ai servizi Amazon Keyspaces

Amazon Keyspaces non supporta l'utilizzo di ruoli collegati ai servizi in tutte le regioni in cui il servizio è disponibile. Puoi utilizzare il AWSService RoleForAmazonKeyspacesReplication ruolo nelle seguenti regioni.

| Nome della Regione           | Identità della regione | Support in Amazon Keyspaces |
|------------------------------|------------------------|-----------------------------|
| US East (N. Virginia)        | us-east-1              | Sì                          |
| Stati Uniti orientali (Ohio) | us-east-2              | Sì                          |
| US West (N. California)      | us-west-1              | Sì                          |
| US West (Oregon)             | us-west-2              | Sì                          |
| Asia Pacific (Mumbai)        | ap-south-1             | Sì                          |
| Asia Pacifico (Osaka-Locale) | ap-northeast-3         | Sì                          |
| Asia Pacifico (Seul)         | ap-northeast-2         | Sì                          |
| Asia Pacifico (Singapore)    | ap-southeast-1         | Sì                          |
| Asia Pacifico (Sydney)       | ap-southeast-2         | Sì                          |
| Asia Pacifico (Tokyo)        | ap-northeast-1         | Sì                          |
| Canada (Central)             | ca-central-1           | Sì                          |
| Europe (Frankfurt)           | eu-central-1           | Sì                          |

| Nome della Regione                     | Identità della regione | Support in Amazon Keyspaces |
|----------------------------------------|------------------------|-----------------------------|
| Europa (Irlanda)                       | eu-west-1              | Sì                          |
| Europe (London)                        | eu-west-2              | Sì                          |
| Europe (Paris)                         | eu-west-3              | Sì                          |
| Africa (Cape Town)                     | af-south-1             | Sì                          |
| Sud America (São Paulo)                | sa-east-1              | Sì                          |
| AWS GovCloud (Stati Uniti orientali)   | us-gov-east-1          | No                          |
| AWS GovCloud (Stati Uniti occidentali) | us-gov-west-1          | No                          |

## Utilizzo dei ruoli per gli stream CDC di Amazon Keyspaces

[Amazon Keyspaces \(per Apache Cassandra\) utilizza ruoli collegati ai servizi AWS Identity and Access Management \(IAM\)](#). Un ruolo collegato ai servizi è un tipo unico di ruolo IAM collegato direttamente ad Amazon Keyspaces. I ruoli collegati ai servizi sono predefiniti da Amazon Keyspaces e includono tutte le autorizzazioni richieste dal servizio per chiamare altri servizi per tuo conto. AWS

Un ruolo collegato al servizio semplifica la configurazione di Amazon Keyspaces perché non è necessario aggiungere manualmente le autorizzazioni necessarie. Amazon Keyspaces definisce le autorizzazioni dei suoi ruoli collegati ai servizi e, se non diversamente definito, solo Amazon Keyspaces può assumerne i ruoli. Le autorizzazioni definite includono la policy di attendibilità e la policy delle autorizzazioni che non può essere collegata a nessun'altra entità IAM.

Non puoi eliminare il ruolo collegato al servizio.

### Autorizzazioni di ruolo collegate ai servizi per Amazon Keyspaces

Amazon Keyspaces utilizza il ruolo collegato al servizio denominato `AWSServiceRoleForAmazonKeyspacesCDC` per consentire agli stream di Amazon Keyspaces CDC di pubblicare i parametri nel tuo account per CloudWatch tuo conto.

Il ruolo collegato al servizio `AWSService RoleForAmazonKeyspaces CDC` prevede che il seguente servizio assuma il ruolo:

- `cassandra-streams.amazonaws.com`

La politica di autorizzazione dei ruoli denominata [CDCServiceRolePolicyKeyspaces](#) consente ad Amazon Keyspaces di completare le seguenti azioni sulle risorse nel namespace: CloudWatch AWS/Cassandra

- Operazione: `cloudwatch:PutMetricData` su \*

Il `AWSServiceRoleForAmazonKeyspacesCDC` fornisce le autorizzazioni: Azione: `cloudwatch:PutMetricData` su tutte le risorse che soddisfano la seguente condizione:  
`"cloudwatch:namespace": "AWS/Cassandra"`

Per ulteriori informazioni su Keyspaces `CDCServiceRolePolicy`, vedere [the section called “Keyspaces CDCService RolePolicy”](#)

Per abilitare gli stream CDC per una tabella, che crea automaticamente il ruolo `AWSServiceRoleForAmazonKeyspacesCDC` collegato al servizio, il principale IAM necessita delle seguenti autorizzazioni.

```
{
 "Sid": "KeyspacesCDCServiceLinkedRole",
 "Effect": "Allow",
 "Action": "iam:CreateServiceLinkedRole",
 "Resource": "arn:aws:iam::*:role/aws-service-role/cassandra-streams.amazonaws.com/AWSServiceRoleForAmazonKeyspacesCDC",
 "Condition": {
 "StringLike": {
 "iam:AWSServiceName": "cassandra-streams.amazonaws.com"
 }
 }
}
```

Le autorizzazioni per creare il ruolo CDC collegato al servizio sono incluse nella policy gestita. `AWSServiceRoleForAmazonKeyspacesAmazonKeyspacesFullAccess` Per ulteriori informazioni, consulta [the section called “AmazonKeyspacesFullAccess”](#).

Creazione di un ruolo collegato ai servizi per Amazon Keyspaces

Non è necessario creare manualmente un ruolo collegato al servizio per gli stream CDC di Amazon Keyspaces. Quando abiliti gli stream Amazon Keyspaces CDC su una tabella con CQL Console di gestione AWS, o l'API, AWS Amazon Keyspaces crea il ruolo collegato al servizio per te. AWS CLI



Se elimini questo ruolo collegato ai servizi, è possibile ricrearlo seguendo lo stesso processo utilizzato per ricreare il ruolo nell'account. Quando abiliti gli stream Amazon Keyspaces CDC per una tabella, Amazon Keyspaces crea nuovamente il ruolo collegato al servizio per te.

### Modifica di un ruolo collegato al servizio per Amazon Keyspaces

Amazon Keyspaces non consente di modificare il ruolo collegato al servizio AWSService RoleForAmazonKeyspaces CDC. Dopo aver creato un ruolo collegato al servizio, non potrai modificarne il nome perché varie entità potrebbero farvi riferimento. È possibile tuttavia modificarne la descrizione utilizzando IAM. Per ulteriori informazioni, consulta [Modifica di un ruolo collegato ai servizi](#) nella Guida per l'utente di IAM.

### Regioni supportate per i ruoli collegati ai servizi Amazon Keyspaces

Amazon Keyspaces supporta l'utilizzo di ruoli collegati ai servizi in tutte le regioni in cui il servizio è disponibile. Per ulteriori informazioni, consulta [AWS Regioni ed endpoint](#).

## Convalida della conformità per Amazon Keyspaces (per Apache Cassandra)

I revisori di terze parti valutano la sicurezza e la conformità di Amazon Keyspaces (per Apache Cassandra) come parte di più programmi di conformità. AWS Ciò include:

- ISO/IEC 27001:2013, 27017:2015, 27018:2019, and ISO/IEC9001:2015. Per ulteriori informazioni, consulta le certificazioni e i [AWS servizi ISO e CSA STAR](#).
- System and Organization Controls (SOC)
- Payment Card Industry (PCI)
- Programma federale di gestione dei rischi e delle autorizzazioni (FedRAMP) High
- Health Insurance Portability and Accountability Act (HIPAA)

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 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 e disaster recovery in Amazon Keyspaces

L'infrastruttura AWS globale è costruita attorno a zone di disponibilità. Regioni AWS Regioni AWS 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.

Amazon Keyspaces replica i dati automaticamente tre volte in più zone di AWS disponibilità all'interno della stessa Regione AWS per garantire durabilità e disponibilità elevata.

Per ulteriori informazioni sulle Regioni AWS zone di disponibilità, consulta l'infrastruttura [AWS globale](#).

Oltre all'infrastruttura AWS globale, Amazon Keyspaces offre diverse funzionalità per supportare le tue esigenze di resilienza e backup dei dati.

### Replica in più regioni

Amazon Keyspaces offre la replica in più regioni se è necessario replicare dati o applicazioni su distanze geografiche maggiori. Puoi replicare le tue tabelle Amazon Keyspaces tra Regioni AWS diverse tabelle a tua scelta. Per ulteriori informazioni, consulta [the section called “Replica multi regione”](#).

### Point-in-time recupero (PITR)

PITR aiuta a proteggere le tabelle Amazon Keyspaces da operazioni di scrittura o cancellazione accidentali fornendo backup continui dei dati delle tabelle. Per ulteriori informazioni, consulta [Point-in-time recovery for Amazon Keyspaces](#).

## Sicurezza dell'infrastruttura in Amazon Keyspaces

In quanto servizio gestito, Amazon Keyspaces (per Apache Cassandra) è protetto dalla AWS sicurezza di rete globale. [Per informazioni sui servizi di AWS sicurezza e su come AWS protegge](#)

[l'infrastruttura, consulta AWS Cloud Security](#). Per progettare il tuo AWS ambiente utilizzando le migliori pratiche per la sicurezza dell'infrastruttura, vedi [Infrastructure Protection](#) in Security Pillar AWS Well-Architected Framework.

Utilizzi chiamate API AWS pubblicate per accedere ad Amazon Keyspaces attraverso la rete. I client devono supportare quanto segue:

- Transport Layer Security (TLS). È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Suite di cifratura con Perfect Forward Secrecy (PFS), 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.

Amazon Keyspaces supporta due metodi di autenticazione delle richieste dei client. Il primo metodo utilizza credenziali specifiche del servizio, che sono credenziali basate su password generate per uno specifico utente IAM. Puoi creare e gestire la password utilizzando la console IAM, l'API AWS CLI. AWS Per ulteriori informazioni, consulta [Using IAM with Amazon Keyspaces](#).

Il secondo metodo utilizza un plug-in di autenticazione per il driver DataStax Java open source per Cassandra. Questo plug-in consente [agli utenti, ai ruoli e alle identità federate IAM](#) di aggiungere informazioni di autenticazione alle richieste API di Amazon Keyspaces (per Apache Cassandra) utilizzando il processo [AWS Signature Version 4 \(SigV4\)](#). Per ulteriori informazioni, consulta [the section called “Crea credenziali IAM per l'autenticazione AWS”](#).

Puoi chiamare queste operazioni API da qualsiasi posizione di rete, ma Amazon Keyspaces supporta politiche di accesso basate sulle risorse, che possono includere restrizioni basate sull'indirizzo IP di origine. Puoi anche utilizzare le policy di Amazon Keyspaces per controllare l'accesso da endpoint Amazon Virtual Private Cloud (Amazon VPC) specifici o specifici VPCs. In effetti, questo isola l'accesso alla rete a una determinata risorsa Amazon Keyspaces solo dal VPC specifico all'interno della rete. AWS

Puoi utilizzare un endpoint VPC di interfaccia per impedire che il traffico tra Amazon VPC e Amazon Keyspaces esca dalla rete Amazon. Gli endpoint VPC di interfaccia sono basati su una AWS tecnologia che consente la comunicazione privata tra AWS i servizi utilizzando un'interfaccia di rete elastica con private nel IPs tuo Amazon VPC. AWS PrivateLink Per ulteriori informazioni, consulta [the section called “Utilizzo di endpoint VPC dell'interfaccia”](#).

## Utilizzo di Amazon Keyspaces con endpoint VPC di interfaccia

Gli endpoint VPC di interfaccia consentono la comunicazione privata tra il tuo cloud privato virtuale (VPC) in esecuzione su Amazon VPC e Amazon Keyspaces. Gli endpoint VPC di interfaccia sono alimentati da AWS PrivateLink, un AWS servizio che consente la comunicazione privata tra e VPCs servizi. AWS

AWS PrivateLink consente ciò utilizzando un'interfaccia di rete elastica con indirizzi IP privati nel tuo VPC in modo che il traffico di rete non lasci la rete Amazon. Gli endpoint VPC dell'interfaccia non richiedono gateway Internet, dispositivo NAT, connessione VPN o connessione Direct Connect . Per ulteriori informazioni, consulta [Amazon Virtual Private Cloud](#) and [Interface VPC endpoint \(\)](#).AWS PrivateLink

### Argomenti

- [Utilizzo degli endpoint VPC dell'interfaccia per Amazon Keyspaces](#)
- [Inserimento delle voci della system.peers tabella con informazioni sugli endpoint VPC dell'interfaccia](#)
- [Controllo dell'accesso agli endpoint VPC di interfaccia per Amazon Keyspaces](#)
- [Disponibilità](#)
- [Policy degli endpoint VPC e ripristino di Amazon point-in-time Keyspaces \(PITR\)](#)
- [Errori e avvertenze comuni](#)

## Utilizzo degli endpoint VPC dell'interfaccia per Amazon Keyspaces

Puoi creare un endpoint VPC di interfaccia in modo che il traffico tra Amazon Keyspaces e le tue risorse Amazon VPC inizi a fluire attraverso l'endpoint VPC dell'interfaccia. [Per iniziare, segui i passaggi per creare un endpoint di interfaccia](#). Successivamente, modifica il gruppo di sicurezza associato all'endpoint creato nel passaggio precedente e configura una regola in entrata per la porta 9142. Per ulteriori informazioni, consulta [Aggiungere, rimuovere e](#) aggiornare le regole.

Per un step-by-step tutorial su come configurare una connessione ad Amazon Keyspaces tramite un endpoint VPC, consulta. [the section called “Connessione con endpoint VPC”](#) Per informazioni su come configurare l'accesso tra account diversi per le risorse Amazon Keyspaces separate dalle applicazioni in modo Account AWS diverso in un VPC, consulta. [the section called “Configura l'accesso tra più account”](#)

## Inserimento delle voci della **system.peers** tabella con informazioni sugli endpoint VPC dell'interfaccia

I driver Apache Cassandra utilizzano la `system.peers` tabella per richiedere informazioni sui nodi relative al cluster. I driver Cassandra utilizzano le informazioni sui nodi per bilanciare il carico delle connessioni e riprovare le operazioni. Amazon Keyspaces inserisce automaticamente nove voci nella `system.peers` tabella per i client che si connettono tramite l'endpoint pubblico.

Per fornire ai client che si connettono tramite endpoint VPC di interfaccia con funzionalità simili, Amazon Keyspaces compila la `system.peers` tabella del tuo account con una voce per ogni zona di disponibilità in cui è disponibile un endpoint VPC. Per cercare e archiviare gli endpoint VPC di interfaccia disponibili nella tabella `system.peers`, Amazon Keyspaces richiede che tu conceda all'entità IAM utilizzata per connettersi ad Amazon Keyspaces le autorizzazioni di accesso per interrogare il tuo VPC per le informazioni sull'endpoint e sull'interfaccia di rete.

### Important

La compilazione della `system.peers` tabella con gli endpoint VPC dell'interfaccia disponibili migliora il bilanciamento del carico e aumenta il throughput. read/write. È consigliato a tutti i client che accedono ad Amazon Keyspaces utilizzando endpoint VPC di interfaccia ed è necessario per Apache Spark.

Per concedere all'entità IAM utilizzata per connettersi ad Amazon Keyspaces le autorizzazioni per cercare le informazioni necessarie sugli endpoint VPC dell'interfaccia, puoi aggiornare il ruolo o la policy utente IAM esistente o creare una nuova policy IAM come mostrato nell'esempio seguente.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "ListVPCEndpoints",
 "Effect": "Allow",
 "Action": [
 "ec2:DescribeNetworkInterfaces",
 "ec2:DescribeVpcEndpoints"
],
 "Resource": "*"
 }
]
}
```

```
]
}
```

#### Note

Le policy gestite `AmazonKeyspacesFullAccess` includono `AmazonKeyspacesReadOnlyAccess_v2` le autorizzazioni necessarie per consentire ad Amazon Keyspaces di accedere all'istanza EC2 Amazon per leggere informazioni sugli endpoint VPC di interfaccia disponibili.

Per confermare che la policy sia stata impostata correttamente, consulta la `system.peers` tabella per visualizzare le informazioni di rete. Se la `system.peers` tabella è vuota, potrebbe indicare che la policy non è stata configurata correttamente o che hai superato la quota di frequenza delle richieste `DescribeNetworkInterfaces` e le azioni `DescribeVPCEndpoints` API. `DescribeVPCEndpoints` rientra nella `Describe*` categoria ed è considerata un'azione non mutante. `DescribeNetworkInterfaces` rientra nel sottoinsieme delle azioni non mutanti non filtrate e non impaginate e si applicano quote diverse. Per ulteriori informazioni, consulta le [dimensioni dei bucket di token e le tariffe di ricarica](#) nell'Amazon EC2 API Reference.

Se vedi una tabella vuota, riprova qualche minuto dopo per escludere problemi relativi alla quota delle tariffe di richiesta. Per verificare di aver configurato correttamente gli endpoint VPC, consulta [the section called “Errori di connessione agli endpoint VPC”](#). Se la tua query restituisce risultati dalla tabella, la tua policy è stata configurata correttamente.

## Controllo dell'accesso agli endpoint VPC di interfaccia per Amazon Keyspaces

Con le policy degli endpoint VPC, puoi controllare l'accesso alle risorse in due modi:

- **Policy IAM:** puoi controllare le richieste, gli utenti o i gruppi autorizzati ad accedere ad Amazon Keyspaces tramite un endpoint VPC specifico. Puoi farlo utilizzando una [chiave di condizione](#) nella policy associata a un utente, gruppo o ruolo IAM.
- **Policy VPC:** puoi controllare quali endpoint VPC hanno accesso alle tue risorse Amazon Keyspaces associando loro delle policy. Per limitare l'accesso a uno spazio chiave o a una tabella specifici per consentire il traffico proveniente solo da un endpoint VPC specifico, modifica la policy IAM esistente che limita l'accesso alle risorse e aggiungi quell'endpoint VPC.

Di seguito sono riportati alcuni esempi di policy per gli endpoint per l'accesso alle risorse di Amazon Keyspaces.

- Esempio di policy IAM: limita tutti gli accessi a una tabella Amazon Keyspaces specifica a meno che il traffico non provenga dall'endpoint VPC specificato: questa policy di esempio può essere associata a un utente, ruolo o gruppo IAM. Limita l'accesso a una tabella Amazon Keyspaces specificata a meno che il traffico in entrata non provenga da un endpoint VPC specificato.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "UserOrRolePolicyToDenyAccess",
 "Action": "cassandra:*",
 "Effect": "Deny",
 "Resource": [
 "arn:aws:cassandra:us-east-1:111122223333:/keyspace/mykeyspace/table/mytable",
 "arn:aws:cassandra:us-east-1:111122223333:/keyspace/system*"
],
 "Condition": { "StringNotEquals" : { "aws:sourceVpce": "vpce-abc123" } }
 }
]
}
```

#### Note

Per limitare l'accesso a una tabella specifica, devi includere anche l'accesso alle tabelle di sistema. Le tabelle di sistema sono di sola lettura.

- Esempio di policy VPC: accesso in sola lettura: questa policy di esempio può essere collegata a un endpoint VPC. (Per ulteriori informazioni, consulta [Controllare l'accesso alle risorse Amazon VPC](#)). Limita le azioni all'accesso in sola lettura alle risorse Amazon Keyspaces tramite l'endpoint VPC a cui è collegato.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "ReadOnly",
```

```

 "Principal": "*",
 "Action": [
 "cassandra:Select"
],
 "Effect": "Allow",
 "Resource": "*"
 }
]
}

```

- Esempio di policy VPC: limita l'accesso a una tabella Amazon Keyspaces specifica: questa policy di esempio può essere collegata a un endpoint VPC. Limita l'accesso a una tabella specifica tramite l'endpoint VPC a cui è collegata.

```

{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "RestrictAccessToTable",
 "Principal": "*",
 "Action": "cassandra:*",
 "Effect": "Allow",
 "Resource": [
 "arn:aws:cassandra:us-east-1:111122223333:/keyspace/mykeyspace/table/mytable",
 "arn:aws:cassandra:us-east-1:111122223333:/keyspace/system*"
]
 }
]
}

```

#### Note

Per limitare l'accesso a una tabella specifica, è necessario includere anche l'accesso alle tabelle di sistema. Le tabelle di sistema sono di sola lettura.

## Disponibilità

Amazon Keyspaces supporta l'utilizzo di endpoint VPC di interfaccia in tutti i luoghi in Regioni AWS cui il servizio è disponibile. Per ulteriori informazioni, consulta [???](#).



## Policy degli endpoint VPC e ripristino di Amazon point-in-time Keyspaces (PITR)

Se utilizzi policy IAM con [chiavi di condizione](#) per limitare il traffico in entrata, l'operazione di ripristino della tabella potrebbe non riuscire. Ad esempio, se limiti il traffico di origine a endpoint VPC specifici utilizzando chiavi di `aws:SourceVpce` condizione, l'operazione di ripristino della tabella ha esito negativo. Per consentire ad Amazon Keyspaces di eseguire un'operazione di ripristino per conto del tuo principale, devi aggiungere una chiave di `aws:ViaAWSService` condizione alla tua policy IAM. La chiave di `aws:ViaAWSService` condizione consente l'accesso quando un AWS servizio effettua una richiesta utilizzando le credenziali del principale. Per ulteriori informazioni, consulta [IAM JSON Policy elements: Condition key](#) nella IAM User Guide. La seguente policy ne è un esempio.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "CassandraAccessForVPCE",
 "Effect": "Allow",
 "Action": "cassandra:*",
 "Resource": "*",
 "Condition": {
 "Bool": {
 "aws:ViaAWSService": "false"
 },
 "StringEquals": {
 "aws:SourceVpce": [
 "vpce-12345678901234567"
]
 }
 }
 },
 {
 "Sid": "CassandraAccessForAwsService",
 "Effect": "Allow",
 "Action": "cassandra:*",
 "Resource": "*",
 "Condition": {
 "Bool": {
 "aws:ViaAWSService": "true"
 }
 }
 }
]
}
```

```
}
```

## Errori e avvertenze comuni

Se utilizzi Amazon Virtual Private Cloud e ti connetti ad Amazon Keyspaces, potresti visualizzare il seguente avviso.

```
Control node cassandra.us-east-1.amazonaws.com/1.111.111.111:9142 has an entry
for itself in system.peers: this entry will be ignored. This is likely due to a
misconfiguration;
please verify your rpc_address configuration in cassandra.yaml on all nodes in your
cluster.
```

Questo avviso si verifica perché la `system.peers` tabella contiene le voci per tutti gli endpoint Amazon VPC che Amazon Keyspaces dispone delle autorizzazioni per la visualizzazione, incluso l'endpoint Amazon VPC tramite il quale sei connesso. Puoi tranquillamente ignorare questo avviso.

Per altri errori, vedere [the section called “Errori di connessione agli endpoint VPC”](#).

## Utilizzo dei flussi CDC di Amazon Keyspaces con endpoint VPC di interfaccia

Gli endpoint VPC di interfaccia consentono la comunicazione privata tra il tuo cloud privato virtuale (VPC) in esecuzione su Amazon VPC e Amazon Keyspaces. Gli endpoint VPC di interfaccia sono alimentati da AWS PrivateLink, un AWS servizio che consente la comunicazione privata tra e VPCs servizi. AWS

AWS PrivateLink consente ciò utilizzando un'interfaccia di rete elastica con indirizzi IP privati nel tuo VPC in modo che il traffico di rete non lasci la rete Amazon. Gli endpoint VPC dell'interfaccia non richiedono gateway Internet, dispositivo NAT, connessione VPN o connessione Direct Connect . Per ulteriori informazioni, consulta [Amazon Virtual Private Cloud](#) and [Interface VPC endpoint \(\)](#).AWS PrivateLink

### Argomenti

- [Utilizzo degli endpoint VPC dell'interfaccia per i flussi CDC di Amazon Keyspaces](#)
- [Endpoint VPC dell'interfaccia CDC di Amazon Keyspaces \(CDC\)](#)
- [Crea un endpoint VPC con interfaccia Streams CDC di Amazon Keyspaces](#)

- [Aggiornamento di un endpoint VPC con interfaccia Amazon Keyspaces CDC Streams](#)
- [Elenca i flussi utilizzando un endpoint VPC con interfaccia CDC Streams di Amazon Keyspaces](#)
- [Crea una policy per un endpoint VPC con interfaccia Amazon Keyspaces CDC Streams](#)

## Utilizzo degli endpoint VPC dell'interfaccia per i flussi CDC di Amazon Keyspaces

Puoi utilizzare un endpoint VPC di interfaccia in modo che il traffico tra i flussi CDC di Amazon Keyspaces e le tue risorse Amazon VPC inizi a fluire attraverso l'endpoint VPC dell'interfaccia. Puoi utilizzare le policy degli endpoint VPC per limitare l'accesso ai tuoi stream CDC.

Per ulteriori informazioni sugli stream CDC di Amazon Keyspaces, consulta [the section called “Lavorare con i flussi CDC”](#)

### Endpoint VPC dell'interfaccia CDC di Amazon Keyspaces (CDC)

Quando crei un endpoint di interfaccia, Amazon Keyspaces CDC Streams genera due tipi di nomi DNS specifici dell'endpoint per lo stream: regionale e zonale.

#### Regionale

Il nome DNS regionale include le seguenti informazioni:

- un ID endpoint Amazon VPC univoco
- un identificatore di servizio
- il Regione AWS
- il `vpce.amazonaws.com` suffisso

Per un endpoint Amazon VPC con l'ID `vpce-1a2b3c4d`, il nome DNS generato potrebbe essere simile al seguente esempio: `vpce-1a2b3c4d-5e6f.cassandra-streams.us-east-1.vpce.amazonaws.com`

#### Zonale

Il nome DNS zonale include la [zona di disponibilità](#) oltre alle informazioni nel nome DNS regionale. Il nome DNS generato per l'endpoint Amazon VPC con l'ID `vpce-1a2b3c4d` sarebbe simile al seguente esempio, si noti che ora include Regione AWS la zona di disponibilità: `vpce-1a2b3c4d-5e6f-us-east-1a.cassandra-streams.us-east-1.vpce.amazonaws.com`

Puoi utilizzare questa opzione se la tua architettura isola le zone di disponibilità. Ad esempio, puoi utilizzarla per il contenimento degli errori o per ridurre i costi di trasferimento dei dati a livello regionale.

#### Note

Per ottenere un'affidabilità ottimale, consigliamo di distribuire il servizio su un minimo di tre zone di disponibilità.

## Crea un endpoint VPC con interfaccia Streams CDC di Amazon Keyspaces

Puoi utilizzare l'SDK AWS CLI o l' AWS SDK per accedere alle operazioni dell'API Amazon Keyspaces CDC Streams tramite gli endpoint dell'interfaccia Amazon Keyspaces CDC Streams. Per un elenco completo di tutte le operazioni API disponibili, consulta [Amazon Keyspaces Streams API Reference](#).

Per ulteriori informazioni su come creare endpoint VPC, consulta [creare un endpoint di interfaccia nella](#) Amazon VPC User Guide.

Per creare un endpoint VPC, puoi utilizzare la sintassi nell'esempio seguente.

```
aws ec2 create-vpc-endpoint \
 --region us-east-1 \
 --service-name api.aws.us-east-1.cassandra-streams \
 --vpc-id client-vpc-id \
 --subnet-ids client-subnet-id \
 --vpc-endpoint-type Interface \
 --security-group-ids client-sg-id
```

## Aggiornamento di un endpoint VPC con interfaccia Amazon Keyspaces CDC Streams

Per aggiornare un endpoint VPC, puoi utilizzare la sintassi nell'esempio seguente.

```
aws ec2 modify-vpc-endpoint \
 --region us-east-1 \
 --vpc-endpoint-id client-vpc-id \
 --policy-document policy-document \ #example optional parameter
 --add-security-group-ids security-group-ids \ #example optional parameter
```

## Elenca i flussi utilizzando un endpoint VPC con interfaccia CDC Streams di Amazon Keyspaces

Per elencare gli stream che utilizzano un endpoint VPC, puoi utilizzare la sintassi nell'esempio seguente. Assicurati di sostituire la regione e il nome DNS dell'ID dell'endpoint VPC con le tue informazioni.

```
aws keyspacesstreams \
 --endpoint https://vpce-1a2b3c4d-5e6f.cassandra-streams.us-east-1.vpce.amazonaws.com \
 --region us-east-1 \
 list-streams
```

## Crea una policy per un endpoint VPC con interfaccia Amazon Keyspaces CDC Streams

Puoi allegare una policy per gli endpoint al tuo endpoint Amazon VPC che controlla l'accesso ai flussi CDC di Amazon Keyspaces. Questa policy specifica le informazioni riportate di seguito:

- Il principio AWS Identity and Access Management (IAM) che può eseguire azioni
- Le azioni che possono essere eseguite
- Le risorse sui cui si possono eseguire le azioni

Per limitare l'accesso a flussi Amazon Keyspaces CDC specifici per consentire solo servizi specifici AWS nel tuo accesso ad Amazon VPC, puoi utilizzare il seguente esempio.

La seguente policy di flusso concede l'accesso a qualsiasi principale IAM per le azioni `cassandra:GetStream` e `cassandra:GetRecords` per lo stream specificato `2025-02-20T11:22:33.444` collegato alla risorsa appartenente all'account. `/keyspace/mykeyspace/table/mytable/123456788901` Per utilizzare questa policy sugli endpoint, assicurati di sostituire la regione, l'ID dell'account e la risorsa con l'etichetta stream.

```
{
 "Version": "2012-10-17",
 "Id": "Policy1216114807515",
 "Statement": [
 { "Sid": "Access-to-specific-stream-only",
 "Principal": "*",
```

```
"Action": [
 "cassandra:GetStream",
 "cassandra:GetRecords"
],
"Effect": "Allow",
"Resource": ["arn:aws:cassandra:us-east-1:111122223333:/keyspace/mykeyspace/
table/mytable/stream/2025-02-20T11:22:33.444"]
}
]
```

#### Note

Amazon Keyspaces non supporta gli endpoint Gateway per gli stream CDC.

## Analisi della configurazione e delle vulnerabilità per Amazon Keyspaces

AWS gestisce attività di sicurezza di base come l'applicazione di patch al sistema operativo guest (OS) e al database, la configurazione del firewall e il disaster recovery. Queste procedure sono state riviste e certificate dalle terze parti appropriate. Per ulteriori dettagli, consulta le seguenti risorse :

- Modello di [responsabilità condivisa Modello](#) di di
- [Amazon Web Services: panoramica dei processi di sicurezza](#) (whitepaper)

## Best practice di sicurezza per Amazon Keyspaces

Amazon Keyspaces (per Apache Cassandra) offre una serie di funzionalità di sicurezza da prendere in considerazione durante lo sviluppo e l'implementazione delle proprie politiche di sicurezza. Le seguenti best practice sono linee guida generali e non rappresentano una soluzione di sicurezza completa. Poiché queste best practice potrebbero non essere appropriate o sufficienti per l'ambiente, gestiscile come considerazioni utili anziché prescrizioni.

### Argomenti

- [Best practice di sicurezza preventiva per Amazon Keyspaces](#)
- [Best practice di sicurezza per i Detective per Amazon Keyspaces](#)

## Best practice di sicurezza preventiva per Amazon Keyspaces

Le seguenti best practice di sicurezza sono considerate preventive perché possono aiutarti ad anticipare e prevenire gli incidenti di sicurezza in Amazon Keyspaces.

### Usa la crittografia a riposo

[Amazon Keyspaces crittografa a riposo tutti i dati utente archiviati nelle tabelle utilizzando chiavi di crittografia archiviate in AWS Key Management Service \(AWS KMS\)](#). Questo fornisce un livello aggiuntivo di protezione dei dati dagli accessi non autorizzati allo storage sottostante.

Per impostazione predefinita, Amazon Keyspaces utilizza una Chiave di proprietà di AWS per crittografare tutte le tabelle. Se questa chiave non esiste, viene creata per te. Le chiavi predefinite del servizio non possono essere disabilitate.

In alternativa, puoi utilizzare una [chiave gestita dal cliente](#) per la crittografia di Rest. Per ulteriori informazioni, consulta [Amazon Keyspaces Encryption at Rest](#).

### Usa i ruoli IAM per autenticare l'accesso ad Amazon Keyspaces

Affinché utenti, applicazioni e altri AWS servizi possano accedere ad Amazon Keyspaces, devono includere AWS credenziali valide nelle loro AWS richieste API. Non è necessario archiviare AWS le credenziali direttamente nell'applicazione o nell'istanza. EC2 Si tratta di credenziali a lungo termine che non vengono automaticamente ruotate e, pertanto, potrebbero avere un impatto aziendale significativo se compromesse. Un ruolo IAM consente di ottenere le chiavi di accesso temporanee che possono essere utilizzate per accedere ai servizi e alle risorse AWS .

Per ulteriori informazioni, consulta [IAM Roles](#) (Ruoli IAM).

### Usa le policy IAM per l'autorizzazione di base di Amazon Keyspaces

Quando concedi le autorizzazioni, decidi chi le ottiene, per quali Amazon Keyspaces APIs ottengono le autorizzazioni e le azioni specifiche che desideri consentire su tali risorse. L'implementazione del privilegio minimo è fondamentale per ridurre i rischi per la sicurezza e l'impatto che possono derivare da errori o intenzioni malevole.

Associa le policy di autorizzazione alle identità IAM (ovvero utenti, gruppi e ruoli) e quindi concedi le autorizzazioni per eseguire operazioni sulle risorse Amazon Keyspaces.

Puoi farlo usando quanto segue:

- [AWS politiche gestite \(predefinite\)](#)

- [Policy gestite dal cliente](#)

Utilizzo di condizioni di policy IAM per il controllo granulare degli accessi

Quando concedi le autorizzazioni in Amazon Keyspaces, puoi specificare le condizioni che determinano l'effetto di una politica di autorizzazioni. L'implementazione del privilegio minimo è fondamentale per ridurre i rischi per la sicurezza e l'impatto che può derivare da errori o intenzioni malevole.

Puoi specificare le condizioni durante la concessione delle autorizzazioni utilizzando una policy IAM. Ad esempio, puoi eseguire le operazioni seguenti:

- Concedi le autorizzazioni per consentire agli utenti l'accesso in sola lettura a spazi chiave o tabelle specifici.
- Concedi le autorizzazioni per consentire a un utente l'accesso in scrittura a una determinata tabella, in base all'identità di quell'utente.

Per ulteriori informazioni, consulta Esempi di policy [basate sull'identità](#).

Valutazione della crittografia lato client

Se memorizzi dati sensibili o riservati in Amazon Keyspaces, potresti voler crittografare tali dati il più vicino possibile alla loro origine in modo che siano protetti per tutto il loro ciclo di vita. Grazie alla crittografia dei dati sensibili in transito e inattivi, puoi accertarti che i dati di testo non crittografato non siano disponibili per terze parti.

## Best practice di sicurezza per i Detective per Amazon Keyspaces

Le seguenti best practice di sicurezza sono considerate investigative perché possono aiutarti a rilevare potenziali punti deboli e incidenti di sicurezza.

Usa AWS CloudTrail per monitorare l'utilizzo delle AWS Key Management Service chiavi (AWS KMS)  
AWS KMS

Se utilizzi una [AWS KMS chiave gestita dal cliente](#) per la crittografia a riposo, l'utilizzo di questa chiave viene AWS CloudTrail registrato. CloudTrail fornisce visibilità sull'attività degli utenti registrando le azioni intraprese sul tuo account. CloudTrail registra informazioni importanti su ogni azione, tra cui chi ha effettuato la richiesta, i servizi utilizzati, le azioni eseguite, i parametri delle azioni e gli elementi di risposta restituiti dal AWS servizio. Queste informazioni aiutano a tenere traccia delle modifiche apportate alle AWS risorse e a risolvere i problemi operativi. CloudTrail semplifica la garanzia della conformità alle politiche interne e agli standard normativi.



È possibile utilizzare CloudTrail per controllare l'utilizzo delle chiavi. CloudTrail crea file di registro che contengono una cronologia delle chiamate AWS API e degli eventi correlati per il tuo account. Questi file di registro includono tutte le richieste AWS KMS API effettuate utilizzando la console e gli strumenti della riga di comando, oltre a quelle effettuate tramite AWS servizi integrati. AWS SDKs È possibile utilizzare questi file di registro per ottenere informazioni su quando è stata utilizzata la AWS KMS chiave, l'operazione richiesta, l'identità del richiedente, l'indirizzo IP da cui proviene la richiesta e così via. Per ulteriori informazioni, consulta [Registrazione di chiamate API di AWS Key Management Service con AWS CloudTrail](#) nella [Guida per l'utente di AWS CloudTrail](#).

Utilizzato CloudTrail per monitorare le operazioni DDL (Data Definition Language) di Amazon Keyspaces

CloudTrail fornisce visibilità sull'attività degli utenti registrando le azioni intraprese sul tuo account. CloudTrail registra informazioni importanti su ogni azione, tra cui chi ha effettuato la richiesta, i servizi utilizzati, le azioni eseguite, i parametri delle azioni e gli elementi di risposta restituiti dal AWS servizio. Queste informazioni consentono di tenere traccia delle modifiche apportate alle AWS risorse e di risolvere i problemi operativi. CloudTrail semplifica la garanzia della conformità alle politiche interne e agli standard normativi.

Tutte le [operazioni DDL](#) di Amazon Keyspaces vengono registrate automaticamente. CloudTrail Le operazioni DDL consentono di creare e gestire gli spazi chiave e le tabelle di Amazon Keyspaces.

Quando si verifica un'attività in Amazon Keyspaces, tale attività viene registrata in un CloudTrail evento insieme ad altri eventi di AWS servizio nella cronologia degli eventi. Per ulteriori informazioni, consulta [Registrazione delle operazioni di Amazon Keyspaces utilizzando](#). AWS CloudTrail Puoi visualizzare, cercare e scaricare eventi recenti in. Account AWS Per ulteriori informazioni, consulta [Visualizzazione degli eventi con cronologia degli CloudTrail eventi](#) nella Guida AWS CloudTrail per l'utente.

[Per una registrazione continua degli eventi nel tuo Account AWS, compresi gli eventi per Amazon Keyspaces, crea un percorso.](#) Un trail consente di CloudTrail inviare file di log a un bucket Amazon Simple Storage Service (Amazon S3). Per impostazione predefinita, quando crei un percorso sulla console, il percorso si applica a tutti. Regioni AWS Il percorso registra gli eventi di tutte le regioni nella partizione AWS e distribuisce i file di log nel bucket S3 specificato. Inoltre, puoi configurare altri AWS servizi per analizzare ulteriormente e agire in base ai dati sugli eventi raccolti nei CloudTrail log.

## Etichetta le tue risorse Amazon Keyspaces per l'identificazione e l'automazione

Puoi assegnare metadati alle tue AWS risorse sotto forma di tag. Ogni tag è una semplice etichetta composta da una chiave definita dal cliente e da un valore opzionale che può semplificare la gestione, la ricerca e il filtraggio delle risorse.

Il tagging consente l'implementazione di gruppi controllati. Anche se non ci sono tipi di tag inerenti, è possibile suddividere le risorse in base a scopo, proprietario, ambiente o altri criteri. Di seguito vengono mostrati alcuni esempi:

- **Accesso:** utilizzato per controllare l'accesso alle risorse di Amazon Keyspaces in base ai tag. Per ulteriori informazioni, consulta [the section called “Autorizzazione basata sui tag Amazon Keyspaces”](#).
- **Sicurezza:** utilizzato per determinare requisiti come le impostazioni di protezione dei dati.
- **Riservatezza:** un identificatore per lo specifico livello di riservatezza dei dati supportato da una risorsa.
- **Ambiente:** utilizzato per differenziare tra infrastruttura di sviluppo, test e produzione.

Per ulteriori informazioni, consulta [Strategie di AWS etichettatura](#) e [Aggiungere tag ed etichette alle risorse](#).

# Riferimento al linguaggio CQL per Amazon Keyspaces (per Apache Cassandra)

Dopo esserti connesso a un endpoint Amazon Keyspaces, usi Cassandra Query Language (CQL) per lavorare con il tuo database. CQL è simile in molti modi a Structured Query Language (SQL).

- **Elementi CQL:** questa sezione tratta gli elementi fondamentali di CQL supportati in Amazon Keyspaces, inclusi identificatori, costanti, termini e tipi di dati. Spiega concetti come tipi di stringhe, tipi numerici, tipi di raccolta e altro ancora.
- **Data Definition Language (DDL):** le istruzioni DDL vengono utilizzate per gestire strutture di dati come keyspace e tabelle in Amazon Keyspaces. Questa sezione descrive le istruzioni per la creazione, la modifica e l'eliminazione di keyspace e tabelle, nonché per il ripristino delle tabelle da un backup. point-in-time
- **Data Manipulation Language (DML):** le istruzioni DML vengono utilizzate per gestire i dati all'interno delle tabelle. Questa sezione descrive le istruzioni per la selezione, l'inserimento, l'aggiornamento e l'eliminazione dei dati. Spiega inoltre funzionalità di interrogazione avanzate come l'utilizzo IN dell'operatore, l'ordinamento dei risultati e l'impaginazione.
- **Funzioni integrate:** Amazon Keyspaces supporta una varietà di funzioni scalari integrate che puoi utilizzare nelle istruzioni CQL. Questa sezione fornisce una panoramica di queste funzioni, inclusi esempi del loro utilizzo.

In questo argomento, troverai sintassi dettagliate, esempi e best practice per utilizzare CQL in modo efficace in Amazon Keyspaces.

## Argomenti

- [Elementi del Cassandra Query Language \(CQL\) in Amazon Keyspaces](#)
- [Istruzioni DDL \(linguaggio di definizione dei dati\) in Amazon Keyspaces](#)
- [Istruzioni DML \(linguaggio di manipolazione dei dati\) in Amazon Keyspaces](#)
- [Funzioni integrate in Amazon Keyspaces](#)

# Elementi del Cassandra Query Language (CQL) in Amazon Keyspaces

Scopri gli elementi del Cassandra Query Language (CQL) supportati da Amazon Keyspaces, inclusi identificatori, costanti, termini e tipi di dati.

## Argomenti

- [Identificatori](#)
- [Costanti](#)
- [Termini](#)
- [Tipi di dati](#)
- [Codifica JSON dei tipi di dati Amazon Keyspaces](#)

## Identificatori

Gli identificatori (o nomi) vengono utilizzati per identificare tabelle, colonne e altri oggetti. Un identificatore può essere citato o non citato. Vale quanto segue.

```
identifier ::= unquoted_identifier | quoted_identifier
unquoted_identifier ::= re('[a-zA-Z][a-zA-Z0-9_]*)'
quoted_identifier ::= ''' (any character where " can appear if doubled)+ '''
```

## Costanti

Sono definite le seguenti costanti.

```
constant ::= string | integer | float | boolean | uuid | blob | NULL
string ::= '\'' (any character where ' can appear if doubled)+ '\''
 ::= '$$' (any character other than '$$') '$$'
integer ::= re('-?[0-9]+')
float ::= re('-?[0-9]+(\.[0-9]*)?([eE][+-]?[0-9+]?)') | NAN | INFINITY
boolean ::= TRUE | FALSE
uuid ::= hex{8}-hex{4}-hex{4}-hex{4}-hex{12}
hex ::= re("[0-9a-fA-F]")
blob ::= '0' ('x' | 'X') hex+
```

## Termini

Un termine indica il tipo di valori supportati. I termini sono definiti come segue.

```
term ::= constant | literal | function_call | arithmetic_operation |
 type_hint | bind_marker
literal ::= collection_literal | tuple_literal
function_call ::= identifier '(' [term (',' term)*] ')'
arithmetic_operation ::= '-' term | term ('+' | '-' | '*' | '/' | '%') term
```

## Tipi di dati

Amazon Keyspaces supporta i seguenti tipi di dati:

### Tipo stringa

| Tipo di dati         | Description                                                                            |
|----------------------|----------------------------------------------------------------------------------------|
| <code>ascii</code>   | Rappresenta una stringa di caratteri ASCII.                                            |
| <code>text</code>    | Rappresenta una stringa con codifica UTF-8.                                            |
| <code>varchar</code> | Rappresenta una stringa con codifica UTF-8 (è un alias per <code>varchar.text</code> ) |

### Tipi numerici

| Tipo di dati         | Description                                                                                                                               |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| <code>bigint</code>  | Rappresenta un valore long firmato a 64 bit.                                                                                              |
| <code>counter</code> | Rappresenta un contatore intero con segno a 64 bit. Per ulteriori informazioni, consulta <a href="#">the section called “Contatori”</a> . |
| <code>decimal</code> | Rappresenta un decimale a precisione variabile.                                                                                           |

| Tipo di dati        | Description                                                           |
|---------------------|-----------------------------------------------------------------------|
| <code>double</code> | Rappresenta una virgola mobile IEEE 754 a 64 bit.                     |
| <code>float</code>  | Rappresenta una virgola mobile IEEE 754 a 32 bit.                     |
| <code>int</code>    | Rappresenta un int. firmato a 32 bit.                                 |
| <code>varint</code> | Rappresenta un valore intero compreso nell'intervallo $\pm 10^{38}$ . |

## Contatori

Una `counter` colonna contiene un numero intero con segno a 64 bit. Il valore del contatore viene incrementato o diminuito utilizzando l'[the section called “UPDATE”](#) istruzione e non può essere impostato direttamente. Ciò rende `counter` le colonne utili per tenere traccia dei conteggi. Ad esempio, puoi utilizzare i contatori per tenere traccia del numero di voci in un file di registro o del numero di volte in cui un post è stato visualizzato su un social network. Le seguenti restrizioni si applicano alle `counter` colonne:

- Una colonna di tipo `counter` non può far parte `primary key` di una tabella.
- In una tabella che contiene una o più colonne di tipo `counter`, tutte le colonne della tabella devono essere di tipo `counter`.

Nei casi in cui un controaggiornamento fallisce (ad esempio, a causa di timeout o perdita di connessione con Amazon Keyspaces), il client non sa se il valore del contatore è stato aggiornato. Se l'aggiornamento viene riprovato, l'aggiornamento al valore del contatore potrebbe essere applicato una seconda volta.

## Tipo di BLOB

| Tipo di dati      | Description                 |
|-------------------|-----------------------------|
| <code>blob</code> | Rappresenta byte arbitrari. |

## Tipo booleano

| Tipo di dati | Description               |
|--------------|---------------------------|
| boolean      | Rappresenta true o. false |

## Tipi relativi al tempo

| Tipo di dati | Description                                                                                                                      |
|--------------|----------------------------------------------------------------------------------------------------------------------------------|
| date         | Una stringa nel formato<yyyy>-<mm>-<dd> .                                                                                        |
| timestamp    | Numero intero con segno a 64 bit che rappresenta la data e l'ora dall'epoca (1° gennaio 1970 alle 00:00:00 GMT) in millisecondi. |
| timeuuid     | <a href="#">Rappresenta</a> un UUID della versione 1.                                                                            |

## Tipi di raccolta

| Tipo di dati | Description                                                            |
|--------------|------------------------------------------------------------------------|
| list         | Rappresenta una raccolta ordinata di elementi letterali.               |
| map          | Rappresenta una raccolta non ordinata di coppie chiave-valore.         |
| set          | Rappresenta una raccolta non ordinata di uno o più elementi letterali. |

Si dichiara una colonna di raccolta utilizzando il tipo di raccolta seguito da un altro tipo di dati (ad esempio TEXT o INT) tra parentesi angolate. È possibile creare una colonna con un SET of TEXT oppure creare una coppia MAP di TEXT e INT chiave-valore, come illustrato nell'esempio seguente.

```
SET <TEXT>
MAP <TEXT, INT>
```

Una raccolta non congelata consente di aggiornare ogni singolo elemento della raccolta. I timestamp sul lato client e le impostazioni Time to Live (TTL) vengono memorizzati per i singoli elementi.

Quando utilizzi la FROZEN parola chiave su un tipo di raccolta, i valori della raccolta vengono serializzati in un unico valore immutabile e Amazon Keyspaces li tratta come un BLOB. Questa è una collezione congelata. Un'UPDATEistruzione INSERT o sovrascrive l'intera raccolta congelata. Non è possibile aggiornare singoli elementi all'interno di una raccolta congelata.

I timestamp sul lato client e le impostazioni Time to Live (TTL) si applicano all'intera raccolta congelata, non ai singoli elementi. Frozenle colonne della raccolta possono far parte di una tabella. PRIMARY KEY

Puoi annidare raccolte congelate. Ad esempio, potete definire a MAP entro a SET se MAP sta utilizzando la FROZEN parola chiave, come illustrato nell'esempio seguente.

```
SET <FROZEN> <MAP <TEXT, INT>>>
```

Amazon Keyspaces supporta l'annidamento di un massimo di 8 livelli di raccolte congelate per impostazione predefinita. Per ulteriori informazioni, consulta [the section called “Quote di servizio Amazon Keyspaces”](#). Per ulteriori informazioni sulle differenze funzionali con Apache Cassandra, consulta. [the section called “FROZENcollezioni”](#) Per ulteriori informazioni sulla sintassi CQL, vedere e. [the section called “CREATE TABLE”](#) [the section called “ALTER TABLE”](#)

## Tipo di tupla

Il tipo di tuple dati rappresenta un gruppo limitato di elementi letterali. Puoi usare una tupla come alternativa a. user defined type Non è necessario utilizzare la FROZEN parola chiave per le tuple. Questo perché una tupla è sempre bloccata e non è possibile aggiornare gli elementi singolarmente.

## Altri tipi

| Tipo di dati | Description                                                                    |
|--------------|--------------------------------------------------------------------------------|
| inet         | Una stringa che rappresenta un indirizzo IP, in uno dei due IPv4 IPv6 formati. |



## Statico

In una tabella Amazon Keyspaces con colonne di clustering, puoi utilizzare la `STATIC` parola chiave per creare una colonna statica di qualsiasi tipo.

La seguente dichiarazione ne è un esempio.

```
my_column INT STATIC
```

Per ulteriori informazioni sull'utilizzo delle colonne statiche, vedere [the section called “Stima del consumo di capacità delle colonne statiche”](#).

## Tipi definiti dall'utente () UDTs

Amazon Keyspaces supporta tipi definiti dall'utente (). UDTs Puoi utilizzare qualsiasi tipo di dati Amazon Keyspaces valido per creare un UDT, incluse raccolte e altri tipi esistenti. UDTs Puoi creare UDTs in un keyspace e puoi usarli per definire colonne in qualsiasi tabella nel keyspace.

Per ulteriori informazioni sulla sintassi CQL, vedere. [the section called “Tipi”](#) Per ulteriori informazioni sull'utilizzo di UDTs, vedere. [the section called “Tipi definiti dall'utente \(\) UDTs”](#)

Per verificare quanti UDTs sono supportati per keyspace, i livelli di nidificazione supportati e altri valori e quote predefiniti correlati, consulta. UDTs [the section called “Quote e valori predefiniti per i tipi definiti dall'utente \(UDTs\) in Amazon Keyspaces”](#)

## Codifica JSON dei tipi di dati Amazon Keyspaces

Amazon Keyspaces offre le stesse mappature dei tipi di dati JSON di Apache Cassandra. La tabella seguente descrive i tipi di dati che Amazon Keyspaces accetta nelle `INSERT` JSON istruzioni e i tipi di dati utilizzati da Amazon Keyspaces quando restituisce dati con l'istruzione. `SELECT` JSON

Per tipi di dati a campo singolo come `float`, `int`, e `UUID` date, puoi anche inserire dati come file. `string` Per i tipi di dati e le raccolte composti, ad esempio `tuplemap`, `elist`, puoi anche inserire dati come JSON o come codificati. `JSON string`

| Tipo di dati JSON | Tipi di dati accettati nelle dichiarazioni <b>INSERT JSON</b> | Tipi di dati restituiti nelle <b>SELECT JSON</b> istruzioni | Note                                                                                         |
|-------------------|---------------------------------------------------------------|-------------------------------------------------------------|----------------------------------------------------------------------------------------------|
| ascii             | string                                                        | string                                                      | Utilizza l'escape \u dei caratteri JSON.                                                     |
| bigint            | integer, string                                               | integer                                                     | La stringa deve essere un numero intero valido a 64 bit.                                     |
| blob              | string                                                        | string                                                      | La stringa deve iniziare con 0x seguita da un numero pari di cifre esadecimali.              |
| boolean           | boolean, string                                               | boolean                                                     | La stringa deve essere una delle due opzioni true. false                                     |
| date              | string                                                        | string                                                      | Formato della data YYYY-MM-DD , fuso orario UTC.                                             |
| decimal           | integer, float, string                                        | float                                                       | Può superare la precisione in virgola mobile IEEE-754 a 32 o 64 bit nel decoder lato client. |
| double            | integer, float, string                                        | float                                                       | La stringa deve essere un numero intero o un float valido.                                   |
| float             | integer, float, string                                        | float                                                       | La stringa deve essere un numero                                                             |

| Tipo di dati JSON     | Tipi di dati accettati nelle dichiarazioni <b>INSERT JSON</b> | Tipi di dati restituiti nelle <b>SELECT JSON</b> istruzioni | Note                                                             |
|-----------------------|---------------------------------------------------------------|-------------------------------------------------------------|------------------------------------------------------------------|
|                       |                                                               |                                                             | intero o un float valido.                                        |
| <code>inet</code>     | <code>string</code>                                           | <code>string</code>                                         | IPv4 o indirizzo. IPv6                                           |
| <code>int</code>      | <code>integer, string</code>                                  | <code>integer</code>                                        | La stringa deve essere un numero intero valido a 32 bit.         |
| <code>list</code>     | <code>list, string</code>                                     | <code>list</code>                                           | Utilizza la rappresentazione nativa dell'elenco JSON.            |
| <code>map</code>      | <code>map, string</code>                                      | <code>map</code>                                            | Utilizza la rappresentazione cartografica JSON nativa.           |
| <code>smallint</code> | <code>integer, string</code>                                  | <code>integer</code>                                        | La stringa deve essere un numero intero valido a 16 bit.         |
| <code>set</code>      | <code>list, string</code>                                     | <code>list</code>                                           | Utilizza la rappresentazione nativa dell'elenco JSON.            |
| <code>text</code>     | <code>string</code>                                           | <code>string</code>                                         | Utilizza l'escape dei caratteri JSON. <code>\u</code>            |
| <code>time</code>     | <code>string</code>                                           | <code>string</code>                                         | Ora del giorno nel formato <code>HH-MM-SS[.fffffffffff]</code> . |

| Tipo di dati JSON | Tipi di dati accettati nelle dichiarazioni <b>INSERT JSON</b> | Tipi di dati restituiti nelle <b>SELECT JSON</b> istruzioni | Note                                                                                                                                                           |
|-------------------|---------------------------------------------------------------|-------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| timestamp         | integer, string                                               | string                                                      | Un Timestamp . Le costanti di stringa consentono di memorizzare i timestamp come date. Vengono restituiti i timbri datari con formato. YYYY-MM-DD HH:MM:SS.SSS |
| timeuuid          | string                                                        | string                                                      | UUID di tipo 1. Vedi <a href="#">constants</a> per il formato UUID.                                                                                            |
| tinyint           | integer, string                                               | integer                                                     | La stringa deve essere un numero intero valido a 8 bit.                                                                                                        |
| tuple             | list, string                                                  | list                                                        | Utilizza la rappresentazione nativa dell'elenco JSON.                                                                                                          |
| UDT               | map, string                                                   | map                                                         | Utilizza la rappresentazione cartografica JSON nativa con i nomi dei campi come chiavi.                                                                        |
| uuid              | string                                                        | string                                                      | Vedi <a href="#">constants</a> per il formato UUID.                                                                                                            |
| varchar           | string                                                        | string                                                      | Utilizza l'escape dei caratteri JSON. \u                                                                                                                       |

| Tipo di dati JSON | Tipi di dati accettati nelle dichiarazioni <b>INSERT JSON</b> | Tipi di dati restituiti nelle <b>SELECT JSON</b> istruzioni | Note                                                                                         |
|-------------------|---------------------------------------------------------------|-------------------------------------------------------------|----------------------------------------------------------------------------------------------|
| varint            | integer, string                                               | integer                                                     | Lunghezza variabile ; potrebbe superare numeri interi a 32 o 64 bit nel decoder lato client. |

## Istruzioni DDL (linguaggio di definizione dei dati) in Amazon Keyspaces

Il linguaggio di definizione dei dati (DDL) è l'insieme di istruzioni Cassandra Query Language (CQL) che usi per gestire le strutture di dati in Amazon Keyspaces (per Apache Cassandra), come keyspace e tabelle. Utilizzi DDL per creare queste strutture di dati, modificarle dopo la creazione e rimuoverle quando non sono più in uso. Amazon Keyspaces esegue operazioni DDL in modo asincrono. Per ulteriori informazioni su come confermare il completamento di un'operazione asincrona, consulta. [the section called “Creazione ed eliminazione asincrona di spazi chiave e tabelle”](#)

Sono supportate le seguenti istruzioni DDL:

- [CREA KEYSPACE](#)
- [MODIFICA KEYSPACE](#)
- [ELIMINA KEYSPACE](#)
- [USA](#)
- [CREA TABELLA](#)
- [ALTER TABLE](#)
- [RIPRISTINA TABELLA](#)
- [DROP TABLE](#)
- [CREA TIPO](#)
- [TIPO DI GOCCIA](#)

### Argomenti

- [Keyspaces](#)
- [Tabelle](#)
- [Tipi definiti dall'utente \(\) UDTs](#)

## Keyspaces

Un keyspace raggruppa tabelle correlate rilevanti per una o più applicazioni. In termini di sistema di gestione di database relazionali (RDBMS), gli spazi chiave sono più o meno simili a database, tablespace o costrutti simili.

### Note

In Apache Cassandra, i keyspace determinano il modo in cui i dati vengono replicati tra più nodi di storage. Tuttavia, Amazon Keyspaces è un servizio completamente gestito: i dettagli del relativo livello di storage vengono gestiti per tuo conto. Per questo motivo, gli spazi chiave in Amazon Keyspaces sono solo costrutti logici e non sono correlati allo storage fisico sottostante.

Per informazioni sui limiti e i vincoli di quota per gli spazi chiave di Amazon Keyspaces, consulta.

### [Quote](#)

Dichiarazioni per gli spazi chiave

- [CREA KEYSPACE](#)
- [ALTERA LO SPAZIO DEI TASTI](#)
- [RILASCIAMO LO SPAZIO CHIAVE](#)
- [USE](#)

## CREA KEYSPACE

Usa l'`CREATE KEYSPACE`istruzione per creare un nuovo keyspace.

Sintassi

```
create_keyspace_statement ::=
CREATE KEYSPACE [IF NOT EXISTS] keyspace_name
```

### WITH *options*

Dove:

- *keyspace\_name* è il nome dello spazio chiave da creare.
- le opzioni sono una o più delle seguenti:
  - REPLICATION— Una mappa che indica la strategia di replica per il keyspace:
    - SingleRegionStrategy— Per uno spazio chiave a regione singola. (Obbligatorio)
    - NetworkTopologyStrategy— Specificarne almeno due. Regioni AWS Il fattore di replica per ogni regione è tre. (Facoltativo)
  - DURABLE\_WRITES— Le scritture su Amazon Keyspaces sono sempre durevoli, quindi questa opzione non è richiesta. Tuttavia, se specificato, il valore deve essere *true*.
  - TAGS— Un elenco di tag di coppia chiave-valore da allegare alla risorsa al momento della creazione. (Facoltativo)

### Esempio

Create uno spazio chiave come segue.

```
CREATE KEYSPACE my_keyspace
 WITH REPLICATION = {'class': 'SingleRegionStrategy'} and TAGS ={'key1':'val1',
'key2':'val2'} ;
```

Per creare uno spazio chiave multiregionale, specificatene *NetworkTopologyStrategy* e includetene almeno due. Regioni AWS Il fattore di replica per ogni regione è tre.

```
CREATE KEYSPACE my_keyspace
 WITH REPLICATION = {'class': 'NetworkTopologyStrategy', 'us-east-1': '3', 'ap-
southeast-1': '3', 'eu-west-1': '3'};
```

## ALTERA LO SPAZIO DEI TASTI

È possibile utilizzare l'*ALTER KEYSPACE WITH* istruzione per le seguenti opzioni

- REPLICATION— Utilizzate questa opzione per aggiungere una nuova Regione AWS replica a un keyspace. È possibile aggiungere una nuova regione a una regione singola o a uno spazio chiave multiregione.

- TAGS— Utilizzate questa opzione per aggiungere o rimuovere tag da uno spazio chiave.

## Sintassi

```
alter_keyspace_statement ::=
 ALTER KEYSPACE keyspace_name
 WITH options
```

## Dove:

- *keyspace\_name* è il nome dello spazio chiave da modificare.
- le opzioni sono una delle seguenti:
  - ADD | DROP TAGS— Un elenco di tag di coppia chiave-valore da aggiungere o rimuovere dallo spazio delle chiavi.
  - REPLICATION— Una mappa che indica la strategia di replica per il keyspace;
    - class— NetworkTopologyStrategy definisce lo spazio chiave come uno spazio chiave multiregionale.
    - region— Specificarne un aggiuntivo Regione AWS per questo keyspace. Il fattore di replica per ogni regione è tre.
    - CLIENT\_SIDE\_TIMESTAMPS— L'impostazione predefinita è DISABLED. È possibile modificare lo stato solo in ENABLED.

## Esempi

Modificate un keyspace come mostrato nell'esempio seguente per aggiungere tag.

```
ALTER KEYSPACE my_keyspace ADD TAGS {'key1': 'val1', 'key2': 'val2'};
```

Per aggiungere una terza regione a uno spazio chiave multiregionale, è possibile utilizzare la seguente istruzione.

```
ALTER KEYSPACE my_keyspace
WITH REPLICATION = {
 'class': 'NetworkTopologyStrategy',
 'us-east-1': '3',
 'us-west-2': '3',
 'us-west-1': '3'
```



```
} AND CLIENT_SIDE_TIMESTAMPS = {'status': 'ENABLED'};
```

## RILASCIAMO LO SPAZIO CHIAVE

Utilizzate l'`DROP KEYSPACE` istruzione per rimuovere uno spazio chiave, inclusi tutti i suoi contenuti, ad esempio le tabelle.

### Sintassi

```
drop_keyspace_statement ::=
DROP KEYSPACE [IF EXISTS] keyspace_name
```

### Dove:

- `keyspace_name` è il nome dello spazio chiave da eliminare.

### Esempio

```
DROP KEYSPACE my_keyspace;
```

## USE

Utilizzate l'`USE` istruzione per definire lo spazio chiave corrente. Ciò consente di fare riferimento a oggetti associati a uno spazio chiave specifico, ad esempio tabelle e tipi, senza utilizzare il nome completo che include il prefisso `keyspace`.

### Sintassi

```
use_statement ::=
USE keyspace_name
```

### Dove:

- `keyspace_name` è il nome del `keyspace` da utilizzare.

### Esempio

```
USE my_keyspace;
```

# Tabelle

Le tabelle sono le strutture di dati principali in Amazon Keyspaces. I dati in una tabella sono organizzati in righe e colonne. Un sottoinsieme di queste colonne viene utilizzato per determinare il partizionamento (e in ultima analisi il posizionamento dei dati) mediante la specificazione di una chiave di partizione.

Un altro set di colonne può essere definito in colonne di clustering, il che significa che possono partecipare come predicati all'esecuzione delle query.

Per impostazione predefinita, vengono create nuove tabelle con capacità di throughput su richiesta. È possibile modificare la modalità di capacità per tabelle nuove ed esistenti. Per ulteriori informazioni sulle modalità di throughput read/write della capacità, vedere [the section called “Configura le modalità di capacità di lettura/scrittura”](#).

Per le tabelle in modalità provisioning, è possibile configurare in modo opzionale. AUTOSCALING\_SETTINGS Per ulteriori informazioni sulla scalabilità automatica di Amazon Keyspaces e sulle opzioni disponibili, consulta. [the section called “Configura il ridimensionamento automatico su una tabella esistente”](#)

Per informazioni sui limiti e i vincoli di quota per le tabelle Amazon Keyspaces, consulta. [Quote](#)

Dichiarazioni per le tabelle

- [CREATE TABLE](#)
- [ALTER TABLE](#)
- [RIPRISTINARE LA TABELLA](#)
- [DROP TABLE](#)

## CREATE TABLE

Utilizzate l'CREATE TABLEistruzione per creare una nuova tabella.

Sintassi

```
create_table_statement ::= CREATE TABLE [IF NOT EXISTS] table_name
 '('
 column_definition
 (',' column_definition)*
```

```

 [',' PRIMARY KEY '(' primary_key ')']
 ')' [WITH table_options]

column_definition ::= column_name cql_type [FROZEN][STATIC][PRIMARY KEY]

primary_key ::= partition_key [',' clustering_columns]

partition_key ::= column_name
 | '(' column_name (',' column_name)* ')'

clustering_columns ::= column_name (',' column_name)*

table_options ::= [table_options]
 | CLUSTERING ORDER BY '(' clustering_order
 ')' [AND table_options]
 | cdc
 | CUSTOM_PROPERTIES
 | AUTOSCALING_SETTINGS
 | default_time_to_live
 | TAGS

clustering_order ::= column_name (ASC | DESC) (',' column_name (ASC | DESC))*

```

Dove:

- *table\_name* è il nome della tabella da creare. Il nome completo include il prefisso keyspace. In alternativa, è possibile impostare lo spazio chiave corrente con l'USE istruzione keyspace.
- *column\_definition* è composto da quanto segue:
  - *column\_name*— Il nome della colonna.
  - *cql\_type*— Un tipo di dati Amazon Keyspaces (vedi [Tipi di dati](#)).
  - *FROZEN*— Indica questa colonna definita dall'utente o di tipo collection (ad esempio, LISTSET, oMAP) come congelata. Una raccolta congelata viene serializzata in un unico valore immutabile e trattata come una. BLOB Per ulteriori informazioni, consulta [the section called “Tipi di raccolta”](#).
  - *STATIC*— Indica questa colonna come statica. Le colonne statiche memorizzano i valori condivisi da tutte le righe della stessa partizione.
  - *PRIMARY KEY*— Indica questa colonna come chiave primaria della tabella.
- *primary\_key* è composto da quanto segue:
  - *partition\_key*

- `clustering_columns`
- `partition_key`:
  - La chiave di partizione può essere una singola colonna o un valore composto da due o più colonne. La parte relativa alla chiave di partizione della chiave primaria è obbligatoria e determina in che modo Amazon Keyspaces archivia i dati.
- `clustering_columns`:
  - La parte opzionale della colonna di clustering della chiave primaria determina il modo in cui i dati vengono raggruppati e ordinati all'interno di ciascuna partizione.
- `table_options` sono costituite da quanto segue:
  - `CLUSTERING ORDER BY`— Il `CLUSTERING ORDER` predefinito su una tabella è composto dalle chiavi di raggruppamento nella direzione di ordinamento ASC (crescente). Specificalo per ignorare il comportamento di ordinamento predefinito.
  - `cdc`— Un valore booleano che specifica se Amazon Keyspaces crea un flusso CDC (change data capture) per la tabella. Il valore predefinito è `false`. Per specificare `view_type` quando abilitare uno stream, imposta il `with. cdc_specification CUSTOM_PROPERTIES`
  - `CUSTOM_PROPERTIES`— Una mappa di impostazioni specifiche di Amazon Keyspaces.
    - `capacity_mode`: specifica la modalità di capacità di velocità effettiva di lettura/scrittura per la tabella. Le opzioni sono `throughput_mode: PAY_PER_REQUEST` e `throughput_mode: PROVISIONED`. La modalità di capacità fornita richiede e come input. `read_capacity_units write_capacity_units` Il valore predefinito è `throughput_mode: PAY_PER_REQUEST`.
    - `cdc_specification`: specifica il flusso `view_type` CDC. Le opzioni sono:
      - `NEW_AND_OLD_IMAGES`— entrambe le versioni della riga, prima e dopo la modifica. Questa è l'impostazione predefinita.
      - `NEW_IMAGE`— la versione della riga dopo la modifica.
      - `OLD_IMAGE`— la versione della riga prima della modifica.
      - `KEYS_ONLY`— le chiavi di partizione e di clustering della riga che è stata modificata.

Per ulteriori informazioni sugli stream CDC, vedere. [the section called “Lavorare con i flussi CDC”](#) Per alcuni esempi di codice, consulta [the section called “Abilita uno stream CDC per una nuova tabella”](#).

- `client_side_timestamps`: specifica se i timestamp lato client sono abilitati o disabilitati per la tabella. Le opzioni sono `{'status': 'enabled'}` e `{'status': 'disabled'}`.

Se non è specificato, l'impostazione predefinita è. `status:disabled` Dopo aver abilitato i timestamp lato client per una tabella, questa impostazione non può essere disabilitata.

- `encryption_specification`: specifica le opzioni di crittografia per la crittografia a riposo. Se non è specificato, l'impostazione predefinita è `encryption_type:AWS_OWNED_KMS_KEY`. L'opzione di crittografia la chiave gestita dal cliente richiede la AWS KMS chiave in formato Amazon Resource Name (ARN) come input: `kms_key_identifier:ARN`.  
`kms_key_identifier:ARN`
- `point_in_time_recovery`: specifica se il point-in-time ripristino è abilitato o disabilitato per la tabella. Le opzioni sono `status:enabled` e `status:disabled`. Se non è specificato, l'impostazione predefinita è `status:disabled`.
- `replica_updates`: specifica le impostazioni di una tabella multiarea specifiche di un. Regione AWS Per una tabella con più regioni, è possibile configurare la capacità di lettura della tabella in modo diverso per. Regione AWS È possibile farlo configurando i seguenti parametri. Per maggiori informazioni ed esempi, consulta [the section called “Crea una tabella multiregionale in modalità provisioning”](#).
- `region`— La replica Regione AWS della tabella con le seguenti impostazioni:
  - `read_capacity_units`
- `TTL`: abilita le impostazioni personalizzate Time to Live per la tabella. Per abilitare, usa `status:enabled`. Il valore predefinito è `status:disabled`. Dopo averlo TTL abilitato, non puoi disabilitarlo per la tabella.
- `AUTOSCALING_SETTINGS` include le seguenti impostazioni opzionali per le tabelle in modalità provisioning. Per maggiori informazioni ed esempi, consulta [the section called “Crea una nuova tabella con ridimensionamento automatico”](#).
- `provisioned_write_capacity_autoscaling_update`:
  - `autoscaling_disabled`— Per abilitare la scalabilità automatica per la capacità di scrittura, impostare il valore su. `false` Il valore predefinito è `true`. (Facoltativo)
  - `minimum_units`— Il livello minimo di velocità di scrittura che la tabella dovrebbe essere sempre pronta a supportare. Il valore deve essere compreso tra 1 e la quota massima di throughput al secondo prevista per l'account (40.000 per impostazione predefinita).
  - `maximum_units`— Il livello massimo di velocità di scrittura che la tabella dovrebbe essere sempre pronta a supportare. Il valore deve essere compreso tra 1 e la quota massima di throughput al secondo prevista per l'account (40.000 per impostazione predefinita).
  - `scaling_policy`— Amazon Keyspaces supporta la politica di tracciamento degli obiettivi. L'obiettivo di auto scaling è la capacità di scrittura assegnata alla tabella.

- `target_tracking_scaling_policy_configuration`— Per definire la politica di tracciamento del target, è necessario definire il valore target. Per ulteriori informazioni sul tracciamento del target e sui periodi di cooldown, consulta [Target Tracking Scaling Policies](#) nella Application Auto Scaling User Guide.
- `target_value`— Il tasso di utilizzo previsto della tabella. La scalabilità automatica di Amazon Keyspaces assicura che il rapporto tra capacità consumata e capacità fornita rimanga pari o vicino a questo valore. `target_value` viene definito in percentuale. Un doppio tra 20 e 90. (Obbligatorio)
- `scale_in_cooldown`— Un periodo di recupero in secondi tra le attività di ridimensionamento che consente alla tabella di stabilizzarsi prima che inizi un'altra scala di attività. Se non viene fornito alcun valore, il valore predefinito è 0. (Facoltativo)
- `scale_out_cooldown`— Un periodo di recupero in secondi tra le attività di ridimensionamento che consente alla tabella di stabilizzarsi prima che inizi un'altra attività di scalabilità orizzontale. Se non viene fornito alcun valore, il valore predefinito è 0. (Facoltativo)
- `disable_scale_in`: A boolean che specifica se `scale-in` è disabilitato o abilitato per la tabella. Questo parametro è disabilitato per impostazione predefinita. Per attivarlo `scale-in`, imposta il booleano valore su `FALSE`. Ciò significa che la capacità viene automaticamente ridotta per una tabella per tuo conto. (Facoltativo)
- `provisioned_read_capacity_autoscaling_update`:
  - `autoscaling_disabled`— Per abilitare la scalabilità automatica per la capacità di lettura, impostare il valore su `false`. Il valore predefinito è `true`. (Facoltativo)
  - `minimum_units`— Il livello minimo di velocità effettiva che la tabella dovrebbe essere sempre pronta a supportare. Il valore deve essere compreso tra 1 e la quota massima di throughput al secondo prevista per l'account (40.000 per impostazione predefinita).
  - `maximum_units`— Il livello massimo di velocità effettiva che la tabella dovrebbe essere sempre pronta a supportare. Il valore deve essere compreso tra 1 e la quota massima di throughput al secondo prevista per l'account (40.000 per impostazione predefinita).
  - `scaling_policy`— Amazon Keyspaces supporta la politica di tracciamento degli obiettivi. L'obiettivo della scalabilità automatica è la capacità di lettura assegnata alla tabella.
  - `target_tracking_scaling_policy_configuration`— Per definire la politica di tracciamento del target, è necessario definire il valore target. Per ulteriori informazioni sul tracciamento del target e sui periodi di cooldown, consulta [Target Tracking Scaling Policies](#) nella Application Auto Scaling User Guide.

- `target_value`— Il tasso di utilizzo previsto della tabella. La scalabilità automatica di Amazon Keyspaces assicura che il rapporto tra capacità consumata e capacità fornita rimanga pari o vicino a questo valore. `target_value` viene definito in percentuale. Un doppio tra 20 e 90. (Obbligatorio)
- `scale_in_cooldown`— Un periodo di recupero in secondi tra le attività di ridimensionamento che consente alla tabella di stabilizzarsi prima che inizi un'altra scala di attività. Se non viene fornito alcun valore, il valore predefinito è 0. (Facoltativo)
- `scale_out_cooldown`— Un periodo di recupero in secondi tra le attività di ridimensionamento che consente alla tabella di stabilizzarsi prima che inizi un'altra attività di scalabilità orizzontale. Se non viene fornito alcun valore, il valore predefinito è 0. (Facoltativo)
- `disable_scale_in`: A boolean che specifica se `scale-in` è disabilitato o abilitato per la tabella. Questo parametro è disabilitato per impostazione predefinita. Per attivarlo `scale-in`, imposta il boolean valore su `FALSE`. Ciò significa che la capacità viene automaticamente ridotta per una tabella per tuo conto. (Facoltativo)
- `replica_updates`: specifica le impostazioni di ridimensionamento automatico Regione AWS specifiche di una tabella multiregione. Per una tabella con più regioni, è possibile configurare la capacità di lettura della tabella in modo diverso per. Regione AWS. È possibile farlo configurando i seguenti parametri. Per maggiori informazioni ed esempi, consulta [the section called “Aggiornamento della capacità assegnata e delle impostazioni di ridimensionamento automatico per una tabella multiregionale”](#).
- `region`— La replica Regione AWS della tabella con le seguenti impostazioni:
  - `provisioned_read_capacity_autoscaling_update`
  - `autoscaling_disabled`— Per abilitare il ridimensionamento automatico della capacità di lettura della tabella, impostate il valore su. `false` Il valore predefinito è `true`. (Facoltativo)

 Note

Il ridimensionamento automatico per una tabella con più regioni deve essere abilitato o disabilitato per tutte le repliche della tabella.

- `minimum_units`— Il livello minimo di velocità di lettura che la tabella dovrebbe essere sempre pronta a supportare. Il valore deve essere compreso tra 1 e la quota massima di throughput al secondo prevista per l'account (40.000 per impostazione predefinita).

- `maximum_units`— Il livello massimo di velocità di lettura che la tabella dovrebbe essere sempre pronta a supportare. Il valore deve essere compreso tra 1 e la quota massima di throughput al secondo prevista per l'account (40.000 per impostazione predefinita).
- `scaling_policy`— Amazon Keyspaces supporta la politica di tracciamento degli obiettivi. L'obiettivo della scalabilità automatica è la capacità di lettura assegnata alla tabella.
- `target_tracking_scaling_policy_configuration`— Per definire la politica di tracciamento del target, è necessario definire il valore target. Per ulteriori informazioni sul tracciamento del target e sui periodi di cooldown, consulta [Target Tracking Scaling Policies](#) nella Application Auto Scaling User Guide.
- `target_value`— Il tasso di utilizzo previsto della tabella. La scalabilità automatica di Amazon Keyspaces assicura che il rapporto tra la capacità di lettura consumata e la capacità di lettura assegnata rimanga pari o vicino a questo valore. `target_value` viene definito in percentuale. Un doppio tra 20 e 90. (Obbligatorio)
- `scale_in_cooldown`— Un periodo di recupero in secondi tra le attività di ridimensionamento che consente alla tabella di stabilizzarsi prima che inizi un'altra scala di attività. Se non viene fornito alcun valore, il valore predefinito è 0. (Facoltativo)
- `scale_out_cooldown`— Un periodo di recupero in secondi tra le attività di ridimensionamento che consente alla tabella di stabilizzarsi prima che inizi un'altra attività di scalabilità orizzontale. Se non viene fornito alcun valore, il valore predefinito è 0. (Facoltativo)
- `disable_scale_in`: A boolean che specifica se `scale-in` è disabilitato o abilitato per la tabella. Questo parametro è disabilitato per impostazione predefinita. Per attivarlo `scale-in`, imposta il booleano valore su `FALSE`. Ciò significa che la capacità di lettura viene automaticamente ridotta per una tabella per conto dell'utente. (Facoltativo)
- `default_time_to_live`— L'impostazione predefinita del Time to Live in secondi per il tavolo.
- `TAGS`— Un elenco di tag di coppia chiave-valore da allegare alla risorsa al momento della creazione.
- `clustering_order` è composto da quanto segue:
  - `column_name`— Il nome della colonna.



- *ASC / DESC*— Imposta il modificatore di ordine ascendente (ASC) o discendente (DESC). Se non è specificato, l'ordine predefinito è ASC.

## Esempio

```
CREATE TABLE IF NOT EXISTS my_keyspace.my_table (
 id text,
 name text,
 region text,
 division text,
 project text,
 role text,
 pay_scale int,
 vacation_hrs float,
 manager_id text,
 PRIMARY KEY (id,division))
WITH CUSTOM_PROPERTIES={
 'capacity_mode':{
 'throughput_mode':
'PROVISIONED', 'read_capacity_units': 10, 'write_capacity_units': 20
 },
 'point_in_time_recovery':{'status':
'enabled'},
 'encryption_specification':{
 'encryption_type':
'CUSTOMER_MANAGED_KMS_KEY',

 'kms_key_identifier':'arn:aws:kms:eu-west-1:555555555555:key/11111111-1111-111-1111-111111111111'
 }
}
AND CLUSTERING ORDER BY (division ASC)
AND TAGS={'key1':'val1', 'key2':'val2'}
AND default_time_to_live = 3024000;
```

In una tabella che utilizza colonne di clustering, le colonne non raggruppate possono essere dichiarate statiche nella definizione della tabella. Per ulteriori informazioni sulle colonne statiche, vedere [the section called “Stima del consumo di capacità delle colonne statiche”](#)

## Esempio

```
CREATE TABLE my_keyspace.my_table (
```

```
id int,
name text,
region text,
division text,
project text STATIC,
PRIMARY KEY (id,division));
```

È possibile creare una tabella con una colonna che utilizza un tipo definito dall'utente (UDT). La prima istruzione negli esempi crea un tipo, la seconda istruzione crea una tabella con una colonna che utilizza il tipo.

### Esempio

```
CREATE TYPE my_keyspace."udt""N@ME" (my_field int);
CREATE TABLE my_keyspace.my_table (my_col1 int pri key, my_col2 "udt""N@ME");
```

## ALTER TABLE

Utilizzate l'`ALTER TABLE` istruzione per aggiungere nuove colonne, aggiungere tag o modificare le proprietà personalizzate della tabella.

### Sintassi

```
alter_table_statement ::= ALTER TABLE table_name

 [ADD (column_definition | column_definition_list)]
 [[ADD | DROP] TAGS {'key1':'val1', 'key2':'val2'}]
 [WITH table_options [, ...]] ;

column_definition ::= column_name cql_type
```

### Dove:

- *table\_name* è il nome della tabella da modificare.
- *column\_definition* è il nome della colonna e del tipo di dati da aggiungere.
- *column\_definition\_list* è un elenco di colonne separate da virgole inserite tra parentesi.
- *table\_option* sono costituiti da quanto segue:

- ***AUTOSCALING\_SETTINGS*** include le impostazioni di ridimensionamento automatico opzionali per le tabelle predisposte. Per la sintassi e le descrizioni dettagliate, vedere. [the section called “CREATE TABLE”](#) Per alcuni esempi, consulta [the section called “Configura il ridimensionamento automatico su una tabella esistente”](#).
- ***cdc***— Un valore booleano che specifica se Amazon Keyspaces crea un flusso CDC (change data capture) per la tabella. Il valore predefinito è `false`. Per specificare `view_type` quando abilitare uno stream, imposta il `with. cdc_specification CUSTOM_PROPERTIES`
- ***CUSTOM\_PROPERTIES***— Una mappa di impostazioni specifiche per Amazon Keyspaces.
  - ***capacity\_mode***: specifica la modalità di capacità di velocità effettiva di lettura/scrittura per la tabella. Le opzioni sono `throughput_mode:PAY_PER_REQUEST` e `throughput_mode:PROVISIONED`. La modalità di capacità fornita richiede e come input. `read_capacity_units write_capacity_units` Il valore predefinito è `throughput_mode:PAY_PER_REQUEST`.
  - ***cdc\_specification***: specifica il flusso `view_type` CDC. Le opzioni sono:
    - ***NEW\_AND\_OLD\_IMAGES***— entrambe le versioni della riga, prima e dopo la modifica. Questa è l'impostazione predefinita.
    - ***NEW\_IMAGE***— la versione della riga dopo la modifica.
    - ***OLD\_IMAGE***— la versione della riga prima della modifica.
    - ***KEYS\_ONLY***— le chiavi di partizione e di clustering della riga che è stata modificata.

Per ulteriori informazioni sugli stream CDC, vedere. [the section called “Lavorare con i flussi CDC”](#) Per alcuni esempi di codice, consulta [the section called “Abilita uno stream CDC per una tabella esistente”](#).

- ***client\_side\_timestamps***: specifica se i timestamp lato client sono abilitati o disabilitati per la tabella. Le opzioni sono `{'status': 'enabled'}` e `{'status': 'disabled'}`. Se non è specificato, l'impostazione predefinita è. `status:disabled` Dopo aver abilitato i timestamp lato client per una tabella, questa impostazione non può essere disabilitata.
- ***encryption\_specification***: specifica l'opzione di crittografia per la crittografia a riposo. Le opzioni sono `encryption_type:AWS_OWNED_KMS_KEY` e `encryption_type:CUSTOMER_MANAGED_KMS_KEY`. L'opzione di crittografia la chiave gestita dal cliente richiede la AWS KMS chiave in formato Amazon Resource Name (ARN) come input. `kms_key_identifier:ARN`

- `point_in_time_recovery`: specifica se il point-in-time ripristino è abilitato o disabilitato per la tabella. Le opzioni sono `status:enabled` e `status:disabled`. Il valore predefinito è `status:disabled`.
- `replica_updates`: specifica le impostazioni Regione AWS specifiche di una tabella con più regioni. Per una tabella con più regioni, è possibile configurare la capacità di lettura della tabella in modo diverso per Regione AWS. È possibile farlo configurando i seguenti parametri. Per maggiori informazioni ed esempi, consulta [the section called “Aggiornamento della capacità assegnata e delle impostazioni di ridimensionamento automatico per una tabella multiregionale”](#).
- `region`— La replica Regione AWS della tabella con le seguenti impostazioni:
  - `read_capacity_units`
- `ttl`: abilita le impostazioni personalizzate Time to Live per la tabella. Per abilitare, usa `status:enabled`. Il valore predefinito è `status:disabled`. Dopo averlo `ttl` abilitato, non puoi disabilitarlo per la tabella.
- `default_time_to_live`: l'impostazione predefinita del Time to Live in secondi per la tabella.
- `TAGS` è un elenco di tag di coppia chiave-valore da allegare alla risorsa.

 Note

Con `ALTER TABLE`, puoi modificare solo una singola proprietà personalizzata. Non è possibile combinare più di un comando `ALTER TABLE` nella stessa istruzione.

## Esempi

L'istruzione seguente mostra come aggiungere una colonna a una tabella esistente.

```
ALTER TABLE mykeyspace.mytable ADD (ID int);
```

Questa istruzione mostra come aggiungere due colonne di raccolta a una tabella esistente:

- Una colonna di raccolta congelata `col_frozen_list` che contiene una raccolta congelata annidata
- Una colonna di raccolta non congelata `col_map` che contiene una raccolta congelata annidata

```
ALTER TABLE my_Table ADD(col_frozen_list FROZEN<LIST<FROZEN<SET<TEXT>>>>, col_map
MAP<INT, FROZEN<SET<INT>>>>);
```

L'esempio seguente mostra come aggiungere una colonna che utilizza un tipo definito dall'utente (UDT) a una tabella.

```
ALTER TABLE my_keyspace.my_table ADD (my_column, my_udt);
```

Per modificare la modalità di capacità di una tabella e specificare le unità di capacità di lettura e scrittura, è possibile utilizzare la seguente istruzione.

```
ALTER TABLE mykeyspace.mytable WITH CUSTOM_PROPERTIES={'capacity_mode':
{'throughput_mode': 'PROVISIONED', 'read_capacity_units': 10, 'write_capacity_units':
20}};
```

L'istruzione seguente specifica una chiave KMS gestita dal cliente per la tabella.

```
ALTER TABLE mykeyspace.mytable WITH CUSTOM_PROPERTIES={
 'encryption_specification':{
 'encryption_type': 'CUSTOMER_MANAGED_KMS_KEY',
 'kms_key_identifier': 'arn:aws:kms:eu-west-1:555555555555:key/11111111-1111-111-1111-111111111111'
 }
};
```

Per abilitare point-in-time il ripristino di una tabella, è possibile utilizzare la seguente istruzione.

```
ALTER TABLE mykeyspace.mytable WITH CUSTOM_PROPERTIES={'point_in_time_recovery':
{'status': 'enabled'}};
```

Per impostare un valore Time to Live predefinito in secondi per una tabella, è possibile utilizzare la seguente istruzione.

```
ALTER TABLE my_table WITH default_time_to_live = 2592000;
```

Questa istruzione abilita le impostazioni Time to Live personalizzate per una tabella.

```
ALTER TABLE mytable WITH CUSTOM_PROPERTIES={'ttl':{'status': 'enabled'}};
```

## RIPRISTINARE LA TABELLA

Usa l'`RESTORE TABLE` istruzione per ripristinare una tabella in un determinato momento. Questa istruzione richiede che point-in-time il ripristino sia abilitato su una tabella. Per ulteriori informazioni, consulta [the section called “Backup e ripristino con point-in-time ripristino”](#).

### Sintassi

```
restore_table_statement ::=
 RESTORE TABLE restored_table_name FROM TABLE source_table_name
 [WITH table_options [, ...]];
```

### Dove:

- *restored\_table\_name* è il nome della tabella ripristinata.
- *source\_table\_name* è il nome della tabella di origine.
- *table\_options* è composto da quanto segue:
  - *restore\_timestamp* è l'ora del punto di ripristino in formato ISO 8601. Se non è specificato, viene utilizzato il timestamp corrente.
  - *CUSTOM\_PROPERTIES*— Una mappa di impostazioni specifiche per Amazon Keyspaces.
    - *capacity\_mode*: specifica la modalità di capacità di velocità effettiva di lettura/scrittura per la tabella. Le opzioni sono `throughput_mode:PAY_PER_REQUEST` e `throughput_mode:PROVISIONED`. La modalità di capacità fornita richiede e come input. `read_capacity_units` `write_capacity_units` L'impostazione predefinita è l'impostazione corrente della tabella di origine.
    - *encryption\_specification*: specifica l'opzione di crittografia per la crittografia a riposo. Le opzioni sono `encryption_type:AWS_OWNED_KMS_KEY` e `encryption_type:CUSTOMER_MANAGED_KMS_KEY`. L'opzione di crittografia la chiave gestita dal cliente richiede la AWS KMS chiave in formato Amazon Resource Name (ARN) come input: `kms_key_identifier:ARN` Per ripristinare una tabella crittografata con una chiave gestita dal cliente su una tabella crittografata con un Chiave di proprietà di AWS, Amazon Keyspaces richiede l'accesso alla AWS KMS chiave della tabella di origine.
    - *point\_in\_time\_recovery*: specifica se il point-in-time ripristino è abilitato o disabilitato per la tabella. Le opzioni sono `status:enabled` e `status:disabled`. A differenza di quando si creano nuove tabelle, lo stato predefinito per le tabelle ripristinate è `status:enabled` dovuto al fatto che l'impostazione viene ereditata dalla tabella di origine. Per disabilitare PITR per le tabelle ripristinate, è necessario `status:disabled` impostarlo in modo esplicito.

- **replica\_updates**: specifica le impostazioni Regione AWS specifiche di una tabella multiregionale. Per una tabella con più regioni, è possibile configurare la capacità di lettura della tabella in modo diverso per. Regione AWS. È possibile farlo configurando i seguenti parametri.
  - **region**— La replica Regione AWS della tabella con le seguenti impostazioni:
    - **read\_capacity\_units**
- **AUTOSCALING\_SETTINGS** include le impostazioni di ridimensionamento automatico opzionali per le tabelle predisposte. Per una sintassi e una descrizione dettagliate, vedere. [the section called “CREATE TABLE”](#)
- **TAGS** è un elenco di tag di coppia chiave-valore da allegare alla risorsa.

#### Note

Le tabelle eliminate possono essere ripristinate solo al momento dell'eliminazione.

## Esempio

```
RESTORE TABLE mykeyspace.mytable_restored from table mykeyspace.my_table
WITH restore_timestamp = '2020-06-30T04:05:00+0000'
AND custom_properties = {'point_in_time_recovery':{'status':'disabled'},
 'capacity_mode':{'throughput_mode': 'PROVISIONED', 'read_capacity_units': 10,
 'write_capacity_units': 20}}
AND TAGS={'key1':'val1', 'key2':'val2'};
```

## DROP TABLE

Utilizzate l'**DROP TABLE** istruzione per rimuovere una tabella dallo spazio delle chiavi.

### Sintassi

```
drop_table_statement ::=
 DROP TABLE [IF EXISTS] table_name
```

### Dove:

- **IF EXISTS** **DROP TABLE** impedisce di fallire se la tabella non esiste. (Facoltativo)

- *table\_name* è il nome della tabella da eliminare.

## Esempio

```
DROP TABLE my_keyspace.my_table;
```

## Tipi definiti dall'utente () UDTs

UDT: un raggruppamento di campi e tipi di dati che puoi utilizzare per definire una singola colonna in Amazon Keyspaces. I tipi di dati validi UDTs sono tutti i tipi di dati Cassandra supportati, comprese le raccolte e altri UDTs che hai già creato nello stesso keyspace. Per ulteriori informazioni sui tipi di dati Cassandra supportati, consulta [the section called “Supporto per i tipi di dati Cassandra”](#)

```
user_defined_type ::= udt_name
udt_name ::= [keyspace_name '.'] identifier
```

### Dichiarazioni per i tipi

- [CREATE TYPE](#)
- [DROP TYPE](#)

## CREATE TYPE

Utilizzate l'`CREATE TYPE` istruzione per creare un nuovo tipo.

### Sintassi

```
create_type_statement ::= CREATE TYPE [IF NOT EXISTS] udt_name
 '(' field_definition (',' field_definition) * ')'
 field_definition ::= identifier cql_type
```

### Dove:

- `IF NOT EXISTS` `CREATE TYPE` impedisce di fallire se il tipo esiste già. (Facoltativo)
- *udt\_name* è il nome completo dell'UDT in formato type, ad esempio. `my_keyspace.my_type`  
Se si definisce lo spazio chiave corrente con l'`USE` istruzione, non è necessario specificare il nome dello spazio chiave.
- *field\_definition* consiste in un nome e un tipo.



La tabella seguente mostra esempi di nomi UDT consentiti. La prima colonna mostra come inserire il nome quando si crea il tipo, la seconda colonna mostra come Amazon Keyspaces formatta il nome internamente. Amazon Keyspaces prevede il nome formattato per operazioni come `GetType`

| Nome inserito             | Nome formattato         | Nota                                                                                                                             |
|---------------------------|-------------------------|----------------------------------------------------------------------------------------------------------------------------------|
| MY_UDT                    | my_udt                  | Senza virgolette, Amazon Keyspaces converte tutti i caratteri maiuscoli in minuscoli.                                            |
| "MY_UDT"                  | MY_UDT                  | Con le virgolette doppie, Amazon Keyspaces rispetta i caratteri maiuscoli e rimuove le virgolette dal nome formattato.           |
| "1234"                    | 1234                    | Con le virgolette doppie, il nome può iniziare con un numero e Amazon Keyspaces rimuove le virgolette dal nome formattato.       |
| "Special_Ch@r@cter s<>!!" | Special_Ch@r@cter s<>!! | Con le virgolette doppie, il nome può contenere caratteri speciali e Amazon Keyspaces rimuove le virgolette dal nome formattato. |
| "nested""quote s"         | nested""quotes          | Amazon Keyspaces rimuove le virgolette doppie esterne e le virgolette di escape dal nome formattato.                             |

## Esempi

```
CREATE TYPE my_keyspace.phone (
 country_code int,
 number text
);
```

Puoi nidificare UDTs se l'UDT annidato è bloccato. Per ulteriori informazioni sui valori predefiniti e sulle quote per i tipi, consulta [the section called “Quote e valori predefiniti di Amazon Keyspaces UDT”](#)

```
CREATE TYPE my_keyspace.user (
 first_name text,
```

```
last_name text,
phones FROZEN<phone>
);
```

Per altri esempi di codice che mostrano come creare UDTs, consulta [the section called “Tipi definiti dall'utente \(\) UDTs”](#).

## DROP TYPE

Utilizzate l'`DROP TYPE`istruzione per eliminare un UDT. È possibile eliminare solo un tipo che non è utilizzato da un altro tipo o tabella.

### Sintassi

```
drop_type_statement ::= DROP TYPE [IF EXISTS] udt_name
```

Dove:

- `IF EXISTS``DROP TYPE`impedisce di fallire se il tipo non esiste. (Facoltativo)
- *udt\_name*è il nome completo dell'UDT in formato type, ad esempio. `my_keyspace.my_type`  
Se si definisce lo spazio chiave corrente con l'`USE`istruzione, non è necessario specificare il nome dello spazio chiave.

### Esempio

```
DROP TYPE udt_name;
```

## Istruzioni DML (linguaggio di manipolazione dei dati) in Amazon Keyspaces

Il linguaggio di manipolazione dei dati (DML) è l'insieme di istruzioni Cassandra Query Language (CQL) che usi per gestire i dati nelle tabelle Amazon Keyspaces (per Apache Cassandra). Le istruzioni DML vengono utilizzate per aggiungere, modificare o eliminare i dati in una tabella.

Utilizzi anche istruzioni DML per interrogare i dati in una tabella. (Nota che CQL non supporta join o sottoquery.)

### Argomenti

- [SELECT](#)
- [INSERT](#)
- [UPDATE](#)
- [DELETE](#)

## SELECT

Utilizzare un'istruzione SELECT per interrogare i dati.

### Sintassi

```

select_statement ::= SELECT [JSON] (select_clause | '*')
 FROM table_name
 [WHERE 'where_clause']
 [ORDER BY 'ordering_clause']
 [LIMIT (integer | bind_marker)]
 [ALLOW FILTERING]

select_clause ::= selector [AS identifier] (',' selector [AS identifier])
selector ::= column_name
 | term
 | CAST '(' selector AS cql_type ')'
 | function_name '(' [selector (',' selector)*] ')'

where_clause ::= relation (AND relation)*
relation ::= column_name operator term
 TOKEN

operator ::= '=' | '<' | '>' | '<=' | '>=' | IN | CONTAINS | CONTAINS KEY
ordering_clause ::= column_name [ASC | DESC] (',' column_name [ASC | DESC])*
```

### Esempi

```

SELECT name, id, manager_id FROM "myGSGKeyspace".employees_tbl ;

SELECT JSON name, id, manager_id FROM "myGSGKeyspace".employees_tbl ;
```

Per una tabella che mappa i tipi di dati con codifica JSON ai tipi di dati Amazon Keyspaces, consulta [the section called “Codifica JSON dei tipi di dati Amazon Keyspaces”](#)

Utilizzo della parola chiave **IN**

La **IN** parola chiave specifica l'uguaglianza per uno o più valori. Può essere applicata alla chiave di partizione e alla colonna di clustering. I risultati vengono restituiti nell'ordine in cui le chiavi sono presentate nell'**SELECT** istruzione.

## Esempi

```
SELECT * from mykeyspace.mytable WHERE primary.key1 IN (1,2) and clustering.key1 = 2;
SELECT * from mykeyspace.mytable WHERE primary.key1 IN (1,2) and clustering.key1 <= 2;
SELECT * from mykeyspace.mytable WHERE primary.key1 = 1 and clustering.key1 IN (1, 2);
SELECT * from mykeyspace.mytable WHERE primary.key1 <= 2 and clustering.key1 IN (1, 2)
ALLOW FILTERING;
```

Per ulteriori informazioni sulla **IN** parola chiave e su come Amazon Keyspaces elabora l'istruzione, consulta. [the section called “Usa IN SELECT”](#)

## Ordinamento dei risultati

La **ORDER BY** clausola specifica l'ordinamento dei risultati restituiti. Prende come argomenti un elenco di nomi di colonna insieme all'ordinamento per ogni colonna. È possibile specificare le colonne di raggruppamento solo nelle clausole di ordinamento. Le colonne non raggruppate non sono consentite. Le opzioni di ordinamento sono **ASC** per l'ordinamento crescente e **DESC** per l'ordinamento decrescente. Se l'ordinamento viene omesso, viene utilizzato l'ordinamento predefinito della colonna di raggruppamento. Per i possibili tipi di ordinamento, vedere. [the section called “Ordina i risultati”](#)

## Esempio

```
SELECT name, id, division, manager_id FROM "myGSGKeyspace".employees_tbl WHERE id =
'012-34-5678' ORDER BY division;
```

Quando si utilizza **ORDER BY** con la **IN** parola chiave, i risultati vengono ordinati all'interno di una pagina. Il riordino completo con impaginazione disabilitata non è supportato.

## TOKEN

È possibile applicare la **TOKEN** funzione alla **PARTITION KEY** colonna **SELECT** e alle **WHERE** clausole. Con la **TOKEN** funzione, Amazon Keyspaces restituisce le righe in base al valore del token mappato di **PARTITION\_KEY** anziché al valore di. **PARTITION KEY**

**TOKEN** le relazioni non sono supportate con la **IN** parola chiave.

## Esempi

```
SELECT TOKEN(id) from my_table;

SELECT TOKEN(id) from my_table WHERE TOKEN(id) > 100 and TOKEN(id) < 10000;
```

## funzione TTL

È possibile utilizzare la TTL funzione con l'SELECTistruzione per recuperare il tempo di scadenza in secondi memorizzato per una colonna. Se non è impostato alcun TTL valore, la funzione restituiscenull.

## Esempio

```
SELECT TTL(my_column) from my_table;
```

La TTL funzione non può essere utilizzata su colonne a celle multiple come le raccolte.

## Funzione WRITETIME

È possibile utilizzare la WRITETIME funzione con l'SELECTistruzione per recuperare il timestamp archiviato come metadati per il valore di una colonna solo se la tabella utilizza timestamp sul lato client. Per ulteriori informazioni, consulta [the section called “Timestamp lato client”](#).

```
SELECT WRITETIME(my_column) from my_table;
```

La WRITETIME funzione non può essere utilizzata su colonne a celle multiple come le raccolte.

### Note

Per motivi di compatibilità con il comportamento consolidato dei driver Cassandra, le politiche di autorizzazione basate su tag non vengono applicate quando si eseguono operazioni sulle tabelle di sistema utilizzando le chiamate API Cassandra Query Language (CQL) tramite i driver Cassandra e gli strumenti di sviluppo. Per ulteriori informazioni, consulta [the section called “Accesso alle risorse Amazon Keyspaces basato su tag”](#).

## INSERT

Utilizzate l'INSERTistruzione per aggiungere una riga a una tabella.

## Sintassi

```

insert_statement ::= INSERT INTO table_name (names_values | json_clause)
 [IF NOT EXISTS]
 [USING update_parameter (AND update_parameter)*]
names_values ::= names VALUES tuple_literal
json_clause ::= JSON string [DEFAULT (NULL | UNSET)]
names ::= '(' column_name (',' column_name)* ')'

```

## Esempio

```

INSERT INTO "myGSGKeyspace".employees_tbl (id, name, project, region, division, role,
pay_scale, vacation_hrs, manager_id)
VALUES ('012-34-5678', 'Russ', 'NightFlight', 'US', 'Engineering', 'IC', 3, 12.5,
'234-56-7890') ;

```

## Aggiorna i parametri

INSERT supporta i seguenti valori come `update_parameter`:

- **TTL**— Un valore temporale in secondi. Il valore massimo configurabile è 630.720.000 secondi, che equivale a 20 anni.
- **TIMESTAMP**— Un `bigint` valore che rappresenta il numero di microsecondi dall'ora base standard nota come `epoch`: 1° gennaio 1970 alle 00:00:00 GMT. Un timestamp in Amazon Keyspaces deve essere compreso tra 2 giorni passati e 5 minuti futuri.

## Esempio

```

INSERT INTO my_table (userid, time, subject, body, user)
VALUES (B79CB3BA-745E-5D9A-8903-4A02327A7E09, 96a29100-5e25-11ec-90d7-
b5d91eceda0a, 'Message', 'Hello', '205.212.123.123')
USING TTL 259200;

```

## Supporto JSON

Per una tabella che mappa i tipi di dati con codifica JSON ai tipi di dati Amazon Keyspaces, consulta [the section called “Codifica JSON dei tipi di dati Amazon Keyspaces”](#)

Puoi usare la JSON parola chiave per inserire una mappa con JSON codifica in una singola riga. Per le colonne che esistono nella tabella ma sono omesse nell'istruzione JSON insert, utilizza DEFAULT UNSET per preservare i valori esistenti. DEFAULT NULLUtilizzatelo per scrivere un valore NULL in ogni riga di colonne omesse e sovrascrivere i valori esistenti (si applicano le tariffe di scrittura standard). DEFAULT NULLè l'opzione predefinita.

### Esempio

```
INSERT INTO "myGSGKeyspace".employees_tbl JSON '{"id":"012-34-5678",
 "name": "Russ",
 "project": "NightFlight",
 "region": "US",
 "division": "Engineering",
 "role": "IC",
 "pay_scale": 3,
 "vacation_hrs": 12.5,
 "manager_id": "234-56-7890"}';
```

Se i dati JSON contengono chiavi duplicate, Amazon Keyspaces memorizza l'ultimo valore per la chiave (simile a Apache Cassandra). Nell'esempio seguente, dove si trova la chiave duplicata, viene id utilizzato il valore. 234-56-7890

### Esempio

```
INSERT INTO "myGSGKeyspace".employees_tbl JSON '{"id":"012-34-5678",
 "name": "Russ",
 "project": "NightFlight",
 "region": "US",
 "division": "Engineering",
 "role": "IC",
 "pay_scale": 3,
 "vacation_hrs": 12.5,
 "id": "234-56-7890"}';
```

## UPDATE

Utilizzate l'UPDATEistruzione per modificare una riga in una tabella.

### Sintassi

```
update_statement ::= UPDATE table_name
 [USING update_parameter (AND update_parameter)*]
```

```

 SET assignment (',' assignment) *
 WHERE where_clause
 [IF (EXISTS | condition (AND condition) *)]
update_parameter ::= (integer | bind_marker)
assignment ::= simple_selection '=' term
 | column_name '=' column_name ('+' | '-') term
 | column_name '=' list_literal '+' column_name
simple_selection ::= column_name
 | column_name '[' term ']'
 | column_name '.' `field_name`
condition ::= simple_selection operator term

```

## Esempio

```

UPDATE "myGSGKeyspace".employees_tbl SET pay_scale = 5 WHERE id = '567-89-0123' AND
division = 'Marketing' ;

```

Per incrementare `acounter`, utilizzare la seguente sintassi. Per ulteriori informazioni, consulta [the section called “Contatori”](#).

```

UPDATE ActiveUsers SET counter = counter + 1 WHERE user = A70FE1C0-5408-4AE3-
BE34-8733E5K09F14 AND action = 'click';

```

## Aggiorna i parametri

UPDATE supporta i seguenti valori come `update_parameter`:

- **TTL**— Un valore temporale in secondi. Il valore massimo configurabile è 630.720.000 secondi, che equivale a 20 anni.
- **TIMESTAMP**— Un `bigint` valore che rappresenta il numero di microsecondi dall'ora base standard nota come `epoch`: 1° gennaio 1970 alle 00:00:00 GMT. Un timestamp in Amazon Keyspaces deve essere compreso tra 2 giorni passati e 5 minuti futuri.

## Esempio

```

UPDATE my_table (userid, time, subject, body, user)
VALUES (B79CB3BA-745E-5D9A-8903-4A02327A7E09, 96a29100-5e25-11ec-90d7-
b5d91eceda0a, 'Message', 'Hello again', '205.212.123.123')
USING TIMESTAMP '2022-11-03 13:30:54+0400';

```



# DELETE

Utilizzate l'DELETEistruzione per rimuovere una riga da una tabella.

## Sintassi

```
delete_statement ::= DELETE [simple_selection (',' simple_selection)]
 FROM table_name
 [USING update_parameter (AND update_parameter)*]
 WHERE where_clause
 [IF (EXISTS | condition (AND condition)*)]

simple_selection ::= column_name
 | column_name '[' term ']'
 | column_name '.' `field_name`

condition ::= simple_selection operator term
```

Dove:

- *table\_name* è la tabella che contiene la riga che si desidera eliminare.

## Esempio

```
DELETE manager_id FROM "myGSGKeyspace".employees_tbl WHERE id='789-01-2345' AND
division='Executive' ;
```

DELETEsupporta il seguente valore comeupdate\_parameter:

- **TIMESTAMP**— Un bigint valore che rappresenta il numero di microsecondi trascorsi dall'ora base standard nota comeepoch: 1° gennaio 1970 alle 00:00:00 GMT.

# Funzioni integrate in Amazon Keyspaces

Amazon Keyspaces (per Apache Cassandra) supporta una varietà di funzioni integrate che puoi usare nelle istruzioni Cassandra Query Language (CQL).

## Argomenti

- [Funzioni scalari](#)

## Funzioni scalari

Una funzione scalare esegue un calcolo su un singolo valore e restituisce il risultato come valore singolo. Amazon Keyspaces supporta le seguenti funzioni scalari.

| Funzione                      | Description                                                                                              |
|-------------------------------|----------------------------------------------------------------------------------------------------------|
| <code>blobAsType</code>       | Restituisce un valore del tipo di dati specificato.                                                      |
| <code>cast</code>             | Converte un tipo di dati nativo in un altro tipo di dati nativo.                                         |
| <code>currentDate</code>      | Restituisce la data corrente date/time .                                                                 |
| <code>currentTime</code>      | Restituisce la corrente date/time come ora.                                                              |
| <code>currentTimestamp</code> | Restituisce la corrente date/time come timestamp.                                                        |
| <code>currentTimeUUID</code>  | Restituisce la corrente date/time come a. <code>timeuuid</code>                                          |
| <code>fromJson</code>         | Converte la stringa JSON nel tipo di dati della colonna selezionata.                                     |
| <code>maxTimeuuid</code>      | Restituisce il valore più grande possibile <code>timeuuid</code> per il timestamp o la stringa di data.  |
| <code>minTimeuuid</code>      | Restituisce il valore più piccolo possibile <code>timeuuid</code> per il timestamp o la stringa di data. |
| <code>now</code>              | Restituisce un nuovo codice univoco. <code>timeuuid</code>                                               |
| <code>toDate</code>           | Converte un timestamp <code>timeuuid</code> o un timestamp in un tipo di data.                           |
| <code>toJson</code>           | Restituisce il valore della colonna selezionata in formato JSON.                                         |

| Funzione                     | Description                                                                                                                                               |
|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>token</code>           | Restituisce il valore hash della chiave di partizione.                                                                                                    |
| <code>toTimestamp</code>     | Converte una <code>timeuuid</code> o una <code>data</code> in un <code>timestamp</code> .                                                                 |
| <code>TTL</code>             | Restituisce il tempo di scadenza in secondi per una colonna.                                                                                              |
| <code>typeAsBlob</code>      | Converte il tipo di dati specificato in <code>unblob</code> .                                                                                             |
| <code>toUnixTimestamp</code> | Converte un <code>timestamp</code> <code>timeuuid</code> o un <code>timestamp</code> in un <code>bigInt</code>                                            |
| <code>uuid</code>            | Restituisce un UUID casuale della versione 4.                                                                                                             |
| <code>writetime</code>       | Restituisce il timestamp del valore della colonna specificata.                                                                                            |
| <code>dateOf</code>          | (Obsoleto) Estrae il timestamp di <code>a</code> e restituisce il valore come <code>timeuuid data</code> .                                                |
| <code>unixTimestampOf</code> | (Obsoleto) Estrae il timestamp di <code>a</code> e restituisce il valore come <code>timestamp</code> intero non elaborato a 64 <code>timeuuid</code> bit. |

# Quote per Amazon Keyspaces (per Apache Cassandra)

Questa sezione descrive le quote correnti e i valori predefiniti per Amazon Keyspaces (per Apache Cassandra).

## Argomenti

- [Quote di servizio Amazon Keyspaces](#)
- [Aumento o diminuzione della velocità di trasmissione effettiva \(per tabelle assegnate\)](#)
- [Crittografia Amazon Keyspaces a riposo](#)
- [Quote e valori predefiniti per i tipi definiti dall'utente \(UDTs\) in Amazon Keyspaces](#)

## Quote di servizio Amazon Keyspaces

La tabella seguente contiene le quote di Amazon Keyspaces (per Apache Cassandra) e i valori predefiniti. Le informazioni su quali quote possono essere modificate sono disponibili nella console [Service Quotas](#), dove puoi anche richiedere aumenti delle quote. Per ulteriori informazioni sulle quote, contatta [Supporto AWS](#).

| Quota                                          | Description                                                                                                                                                                             | Amazon Keyspaces predefinito |
|------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|
| Numero massimo di spazi chiave per Regione AWS | Il numero massimo di spazi chiave per questo abbonato per regione. È possibile modificare questo valore predefinito nella console <a href="#">Service Quotas</a> .                      | 256                          |
| Numero massimo di tabelle per Regione AWS      | Il numero massimo di tabelle in tutti gli spazi chiave per questo abbonato per regione. È possibile modificare questo valore predefinito nella console <a href="#">Service Quotas</a> . | 256                          |

| Quota                                                                      | Description                                                                                                                                                                                                | Amazon Keyspaces predefinito |
|----------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|
| Quantità massima di dati ripristinati utilizzando ADD REGION le operazioni | La dimensione massima dei dati che possono essere ripristinati contemporaneamente dalle ADD REGION operazioni. Per aumentare la quantità di dati da ripristinare contemporaneamente, contatta il supporto. | 10 TB                        |
| Dimensione massima dello schema della tabella                              | La dimensione massima di uno schema di tabella.                                                                                                                                                            | 350 KB                       |
| Numero massimo di operazioni DDL simultanee                                | Il numero massimo di operazioni DDL simultanee e consentite per questo abbonamento per regione.                                                                                                            | 50                           |
| Numero massimo di interrogazioni per connessione                           | Il numero massimo di query CQL che possono essere elaborate da una singola connessione TCP client al secondo.                                                                                              | 3000                         |
| Dimensione massima della riga                                              | La dimensione massima di una riga, esclusi i dati statici delle colonne. Per informazioni dettagliate, vedi <a href="#">the section called “Stima della dimensione delle righe”</a> .                      | 1 MB                         |

| Quota                                                      | Description                                                                                                                                                                                                                                                                                   | Amazon Keyspaces predefinito |
|------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|
| Numero massimo di colonne nelle istruzioni e INSERT UPDATE | Il numero massimo di colonne consentito in CQL INSERT o nelle UPDATE istruzioni. Un'UPDATEistruzione INSERT or supporta fino a 225 colonne regolari quando Time to Live (TTL) è disattivato. Se il TTL è attivato, è possibile modificare fino a 166 colonne regolari in un'unica operazione. | 225/166                      |
| Numero massimo di dati statici per partizione logica       | La dimensione massima di aggregazione di dati statici in una partizione logica. Per informazioni dettagliate, vedi <a href="#">the section called “Calcola la dimensione statica delle colonne per partizione logica”</a> .                                                                   | 1 MB                         |
| Numero massimo di sottoquery per dichiarazione IN SELECT   | Il numero massimo di sottoquery che è possibile utilizzare per la IN parola chiave in un'istruzione. SELECT È possibile modificare questo valore predefinito nella console <a href="#">Service Quotas</a> .                                                                                   | 100                          |

| Quota                                                                            | Description                                                                                                                                                                                                                                                                                                                           | Amazon Keyspaces predefinito |
|----------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|
| Numero massimo di raccolte congelate annidate per Regione AWS                    | Il numero massimo di raccolte nidificate supportate quando si utilizza la FROZEN parola chiave per una colonna con un tipo di dati di raccolta. Per ulteriori informazioni sulle raccolte congelate, consultate <a href="#">the section called “Tipi di raccolta”</a> . Per aumentare il livello di nidificazione, contatta Supporto. | 8                            |
| Numero massimo di istruzioni in un batch registrato                              | Il numero massimo di istruzioni CQL supportate quando si utilizza un'istruzione batch registrata. Per ulteriori informazioni sulle istruzioni batch, vedere <a href="#">the section called “Usa istruzioni batch”</a>                                                                                                                 | 100                          |
| Numero massimo di istruzioni in un batch registrato utilizzando colonne statiche | Il numero massimo di istruzioni CQL supportate quando si utilizzano colonne statiche in un'istruzione batch registrata. Per ulteriori informazioni sulle istruzioni batch, vedere <a href="#">the section called “Usa istruzioni batch”</a>                                                                                           | 50                           |

| Quota                                                   | Description                                                                                                                                                                                                                                                                      | Amazon Keyspaces predefinito |
|---------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|
| Dimensione massima dell'istruzione batch registrata     | La dimensione massima di un'istruzione batch registrata. Per ulteriori informazioni sulle istruzioni batch, vedere. <a href="#">the section called “Usa istruzioni batch”</a>                                                                                                    | 4 MB                         |
| Numero massimo di istruzioni in un batch non registrato | Il numero massimo di istruzioni CQL supportate quando si utilizza un'istruzione batch non registrata. Per ulteriori informazioni sulle istruzioni batch, vedere. <a href="#">the section called “Usa istruzioni batch”</a>                                                       | 30                           |
| Velocità massima di lettura al secondo                  | La velocità massima di lettura al secondo, unità di richiesta di lettura (RRUs) o unità di capacità di lettura (RCUs), che può essere assegnata a una tabella per regione. È possibile modificare questo valore predefinito nella console <a href="#">Service Quotas</a> .       | 40.000                       |
| Velocità massima di scrittura al secondo                | La velocità massima di scrittura al secondo, unità di richiesta di scrittura (WRUs) o unità di capacità di scrittura (WCUs), che può essere assegnata a una tabella per regione. È possibile modificare questo valore predefinito nella console <a href="#">Service Quotas</a> . | 40.000                       |



| Quota                                                                               | Description                                                                                                                                                                                                                                                                                   | Amazon Keyspaces predefinito |
|-------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|
| Velocità massima di scrittura per una tabella utilizzando change-data-capture (CDC) | La capacità massima di scrittura per una tabella Amazon Keyspaces con CDC abilitato. Questa quota predefinita si applica solo alle tabelle in modalità di capacità fornita. read/write È possibile modificare questo valore predefinito nella console <a href="#">Service Quotas</a> .        | 40.000                       |
| Velocità effettiva di lettura a livello di account (fornita)                        | Il numero massimo di unità di capacità di lettura aggregata (RCUs) allocate per l'account per regione. Questo è applicabile solo per le tabelle in modalità di capacità assegnata. read/write È possibile modificare questo valore predefinito nella console <a href="#">Service Quotas</a> . | 80.000                       |
| Throughput di scrittura a livello di account (fornito)                              | Il numero massimo di unità di capacità di scrittura aggregata (WCU) allocate per l'account per regione. Questo è applicabile solo per le tabelle in modalità di capacità fornita. read/writ e È possibile modificare questo valore predefinito nella console <a href="#">Service Quotas</a> . | 80.000                       |

| Quota                                                         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Amazon Keyspaces predefinito |
|---------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|
| Numero massimo di obiettivi scalabili per regione per account | Il numero massimo di obiettivi scalabili per l'account per regione. Una tabella Amazon Keyspaces conta come una destinazione scalabile se la scalabilità automatica è abilitata per la capacità di lettura e come un'altra destinazione scalabile se la scalabilità automatica è abilitata per la capacità di scrittura. Puoi modificare e questo valore predefinito nella console <a href="#">Service Quotas</a> per Application Auto Scaling scegliendo destinazioni scalabili per Amazon Keyspaces. | 1.500                        |
| Dimensione massima della chiave di partizione                 | La dimensione massima della chiave di partizione composta. Un massimo di 3 byte di spazio di archiviazione aggiuntivo vengono aggiunti alla dimensione non elaborata di ogni colonna inclusa nella chiave di partizionamento per i metadati.                                                                                                                                                                                                                                                           | 2048 byte                    |

| Quota                                                                                        | Description                                                                                                                                                                                                                | Amazon Keyspaces predefinito |
|----------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|
| Dimensione massima della chiave di clustering                                                | La dimensione massima combinata di tutte le colonne di clustering. Un massimo di 4 byte di spazio di archiviazione aggiuntivo vengono aggiunti alla dimensione non elaborata di ogni colonna di clustering per i metadati. | 850 byte                     |
| Numero massimo di ripristini simultanei di tabelle utilizzando Point-in-time Recovery (PITR) | Il numero massimo di ripristini simultanei di tabelle utilizzando PITR per sottoscrittore è 4. È possibile modificare questo valore predefinito nella console <a href="#">Service Quotas</a> .                             | 4                            |
| Quantità massima di dati ripristinati utilizzando il point-in-time ripristino (PITR)         | La dimensione massima dei dati che possono essere ripristinati tramite PITR entro 24 ore. È possibile modificare questo valore predefinito nella console <a href="#">Service Quotas</a> .                                  | 5 TB                         |

## Aumento o diminuzione della velocità di trasmissione effettiva (per tabelle assegnate)

### Aumento della velocità di trasmissione effettiva assegnata

È possibile aumentare ReadCapacityUnits o WriteCapacityUnits con la frequenza necessaria utilizzando la console o l'`ALTER TABLE` istruzione. Le nuove impostazioni non hanno effetto fino a quando l'operazione `ALTER TABLE` non è terminata.

Non puoi superare le quote per account quando aggiungi la capacità assegnata. Inoltre, è possibile aumentare la capacità assegnata per i tavoli in base alle proprie esigenze. Per ulteriori informazioni sulle quote di ogni account, consulta la sezione precedente, [the section called “Quote di servizio Amazon Keyspaces”](#).

## Riduzione della velocità di trasmissione effettiva assegnata

Per ogni tabella di una ALTER TABLE dichiarazione, puoi diminuire ReadCapacityUnits o WriteCapacityUnits (o entrambi). Le nuove impostazioni non hanno effetto fino a quando l'operazione ALTER TABLE non è terminata.

La riduzione è consentita fino a un massimo di quattro volte al giorno. Un giorno è definito in base a Universal Coordinated Time (UTC). Inoltre, se non vi sono state riduzioni nell'ultima ora, sarà concessa una ulteriore riduzione. Ciò porta effettivamente a 27 il numero massimo di riduzioni in un giorno (4 diminuzioni nella prima ora e 1 diminuzione per ciascuna delle successive finestre di 1 ora in un giorno).

## Crittografia Amazon Keyspaces a riposo

Puoi modificare le opzioni di crittografia tra una AWS KMS chiave AWS di proprietà e una AWS KMS chiave gestita dal cliente fino a quattro volte nell'arco di 24 ore, per tabella, a partire dal momento in cui è stata creata la tabella. E se non vi sono state modifiche nelle ultime 6 ore, sarà consentita un'ulteriore modifica. Ciò porta effettivamente a otto il numero massimo di modifiche in un giorno (quattro modifiche nelle prime sei ore e una modifica per ciascuna delle successive finestre di sei ore in un giorno).

È possibile modificare l'opzione di crittografia per utilizzare una AWS KMS chiave AWS proprietaria tutte le volte che è necessario, anche se la quota precedente è stata esaurita.

A meno che tu non ne richieda una quantità maggiore, le quote sono le seguenti. Per richiedere un aumento della quota di servizio, vedere [Supporto](#).

# Quote e valori predefiniti per i tipi definiti dall'utente (UDTs) in Amazon Keyspaces

## Quote e valori predefiniti di Amazon Keyspaces UDT

La tabella seguente contiene quote e valori predefiniti relativi ad Amazon UDTs Keyspaces. Per ulteriori informazioni su queste quote, contatta. Supporto AWS

| Quota                                           | Description                                                                   | Amazon Keyspaces predefinito |
|-------------------------------------------------|-------------------------------------------------------------------------------|------------------------------|
| Numero massimo di per UDTs Regione AWS          | Il numero massimo di spazi chiave per questo abbonato per UDTs regione.       | 256                          |
| Numero massimo di tabelle per UDT               | Il numero massimo di tabelle che possono fare riferimento allo stesso UDT.    | 100                          |
| Numero massimo di UDTs per tabella              | Il numero massimo di riferimenti a UDTs cui una tabella può fare riferimento. | 50                           |
| Livello massimo di nidificazione di UDTs        | La profondità di nidificazione massima supportata per. UDTs                   | 8                            |
| Numero massimo di figli diretti UDTs per UDT    | Il numero massimo di bambini UDTs supportati per un UDT.                      | 10                           |
| Numero massimo di genitori diretti UDTs per UDT | Il numero massimo di genitori UDTs supportato per un UDT.                     | 10                           |
| Dimensione massima dello schema UDT             | La dimensione massima dello schema per un UDT.                                | 25 KB                        |
| Lunghezza massima del nome UDT                  | Il numero massimo di caratteri nel nome UDT.                                  | 48                           |

| Quota                                    | Description                                             | Amazon Keyspaces predefinito |
|------------------------------------------|---------------------------------------------------------|------------------------------|
| Lunghezza massima del nome del campo UDT | Il numero massimo di caratteri in un nome di campo UDT. | 128                          |

# Cronologia dei documenti per Amazon Keyspaces (per Apache Cassandra)

La tabella seguente descrive le modifiche importanti alla documentazione dall'ultima versione di Amazon Keyspaces (per Apache Cassandra). Per ricevere notifiche sugli aggiornamenti di questa documentazione, è possibile sottoscrivere un feed RSS.

- Ultimo aggiornamento della documentazione: 28 ottobre 2025

| Modifica                                                                                     | Descrizione                                                                                                    | Data              |
|----------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------|-------------------|
| <a href="#">Amazon Keyspaces ora supporta i batch registrati</a>                             | Ora puoi combinare più istruzioni in batch registrati in Amazon Keyspaces.                                     | 11 novembre 2025  |
| <a href="#">Modifiche ai certificati TLS di Amazon Keyspaces</a>                             | Istruzioni aggiornate per includere i certificati emessi nell'ambito della catena Amazon Trust Services (ATS). | 25 ottobre 2025   |
| <a href="#">Amazon Keyspaces ora supporta IPv4 e richiede IPv6</a>                           | Supporto e configurazioni degli endpoint dual-stack di Amazon Keyspaces. IPv4 IPv6                             | 30 settembre 2025 |
| <a href="#">NOWe ora UUID sono supportati in tutte le istruzioni CQL in Amazon Keyspaces</a> | Ora puoi utilizzare le UUID funzioni NOW and in tutte le istruzioni CQL in Amazon Keyspaces.                   | 8 settembre 2025  |
| <a href="#">L'adattatore Kinesis di Amazon Keyspaces Streams è ora disponibile</a>           | Ora puoi elaborare gli stream CDC di Amazon Keyspaces con la Kinesis Client Library (KCL).                     | 2 luglio 2025     |
| <a href="#">Nuova politica gestita di Amazon Keyspaces</a>                                   | Amazon Keyspaces ha aggiunto una nuova policy gestita Keyspaces                                                | 2 luglio 2025     |

`CDCServiceRolePolicy` che concede le autorizzazioni necessarie al ruolo collegato al servizio per `AWSServiceRoleForAmazonKeyspacesCDC` pubblicare i dati delle metriche di streaming CDC di Amazon Keyspaces per tuo conto. CloudWatch Per ulteriori informazioni, consulta [Using roles for Amazon Keyspaces CDC Streams](#).

#### [Aggiornamento delle policy gestite da Amazon Keyspaces](#)

Amazon Keyspaces ha aggiunto nuove autorizzazioni alle policy gestite `AmazonKeyspacesReadOnlyAccess_v2` e alle policy `AmazonKeyspacesReadOnlyAccess` gestite per consentire ai responsabili IAM di visualizzare i flussi. Amazon Keyspaces ha aggiornato la policy `AmazonKeyspacesFullAccess` gestita con le autorizzazioni per creare un ruolo collegato al servizio `AWSServiceRoleForAmazonKeyspacesCDC` che pubblica CloudWatch i parametri per conto dei principali.

2 luglio 2025



|                                                                                                                  |                                                                                                                                                                                                                                                                                                                               |                  |
|------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| <a href="#">I flussi CDC (change data capture) di Amazon Keyspaces sono ora disponibili</a>                      | Abilita gli stream CDC per registrare le modifiche nelle tabelle Amazon Keyspaces.                                                                                                                                                                                                                                            | 2 luglio 2025    |
| <a href="#">La replica multiregionale di Amazon Keyspaces ora supporta la regione Africa (Città del Capo).</a>   | Ora puoi definire la regione Africa (Città del Capo) per gli spazi chiave multiregionali.                                                                                                                                                                                                                                     | 31 marzo 2025    |
| <a href="#">La replica multiregionale di Amazon Keyspaces ora supporta un numero illimitato. Regioni AWS</a>     | Ora puoi definire tutte le regioni che desideri per gli spazi chiave multiregionali.                                                                                                                                                                                                                                          | 25 marzo 2025    |
| <a href="#">Supporto di Amazon Keyspaces per tipi definiti dall'utente (UDTs) in spazi chiave multiregionali</a> | Ora puoi creare e utilizzare UDTs spazi chiave multiregionali.                                                                                                                                                                                                                                                                | 24 marzo 2025    |
| <a href="#">Aggiornamento delle policy gestite da Amazon Keyspaces</a>                                           | Amazon Keyspaces ha aggiunto nuove autorizzazioni alla policy AmazonKey spacesFullAccess gestita per consentire ai responsabili IAM di aggiungere e nuove regioni a uno spazio di chiavi esistente. Ciò include un aggiornamento del ruolo collegato al servizio. <a href="#">AWSServiceRoleForAmazonKeyspacesReplication</a> | 19 novembre 2024 |
| <a href="#">Aggiungi repliche a tabelle multiregionali in Amazon Keyspaces</a>                                   | Ora puoi aggiungerne di nuovi Regioni AWS a keyspace singoli e multiregionali esistenti.                                                                                                                                                                                                                                      | 19 novembre 2024 |

[Supporto per tipi definiti dall'utente \(UDTs\) in Amazon Keyspaces](#)

Con il supporto nativo di UDTs Amazon Keyspaces, ora puoi definire strutture di dati nelle tue applicazioni che rappresentano gerarchie di dati del mondo reale.

30 ottobre 2024

[ADD COLUMNsupporto per la replica multiregionale di Amazon Keyspaces](#)

Amazon Keyspaces ora supporta le modifiche allo schema per le tabelle multiregionali.

17 settembre 2024

[Linee guida aggiornate alla migrazione per Amazon Keyspaces](#)

La guida alla migrazione aggiornata descrive come creare un piano di migrazione e per migrare con successo da Apache Cassandra ad Amazon Keyspaces, comprese diverse strategie per le migrazioni offline e online.

28 giugno 2024

[Connect ad Amazon Keyspaces da Amazon Elastic Kubernetes Service](#)

Ora puoi seguire un step-by-step tutorial per connetterti ad Amazon Keyspaces da Amazon EKS.

7 febbraio 2024

[Amazon Keyspaces Supporto per la replica multiregionale per le tabelle predisposte](#)

Amazon Keyspaces ora supporta la modalità di capacità fornita per tabelle multiregionali.

23 gennaio 2024

[Scalabilità automatica di Amazon Keyspaces per tabelle predisposte APIs](#)

Amazon Keyspaces ora offre CQL supporto AWS API per la configurazione della scalabilità automatica con modalità di capacità fornita.

23 gennaio 2024

|                                                                             |                                                                                                                                                                                                                                                                                                                                           |                   |
|-----------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| <a href="#">Attività DML di Amazon Keyspaces inclusa nei log CloudTrail</a> | Ora puoi controllare le chiamate API di Amazon Keyspaces Data Manipulation Language (DML). AWS CloudTrail                                                                                                                                                                                                                                 | 20 dicembre 2023  |
| <a href="#">Supporto Amazon Keyspaces per la parola chiave FROZEN</a>       | Amazon Keyspaces ora supporta la FROZEN parola chiave per i tipi di dati di raccolta.                                                                                                                                                                                                                                                     | 15 novembre 2023  |
| <a href="#">Aggiornamento delle policy gestite da Amazon Keyspaces</a>      | Amazon Keyspaces ha aggiunto nuove autorizzazioni alla policy AmazonKeyspacesFullAccess gestita per consentire ai client che si connettono ad Amazon Keyspaces tramite gli endpoint dell'interfaccia VPC di accedere all' EC2 istanza Amazon per aggiornare la tabella Amazon system.peers Keyspaces con le informazioni di rete dal VPC. | 3 ottobre 2023    |
| <a href="#">Aggiornamento delle policy gestite da Amazon Keyspaces</a>      | Amazon Keyspaces ha creato una nuova policy AmazonKeyspacesReadOnlyAccess_v2 gestita per consentire ai client che si connettono ad Amazon Keyspaces tramite endpoint di interfaccia VPC di accedere all'istanza Amazon per EC2 aggiornare la tabella Amazon Keyspaces system.peers con le informazioni di rete dal VPC.                   | 12 settembre 2023 |

[Le migliori pratiche per creare connessioni in Amazon Keyspaces](#)

Scopri come migliorare e ottimizzare le configurazioni dei driver client in Amazon Keyspaces.

30 giugno 2023

[Gli spazi chiave di sistema sono ora documentati per Amazon Keyspaces](#)

Scopri cosa è memorizzato negli spazi chiave di sistema e come interrogarli per ottenere informazioni utili in Amazon Keyspaces.

21 giugno 2023

[Amazon Keyspaces ora supporta la replica in più regioni](#)

La replica multiregionale di Amazon Keyspaces ti aiuta a mantenere applicazioni distribuite a livello globale fornendo tolleranza agli errori, stabilità e resilienza migliorate.

5 giugno 2023

[Aggiornamento delle policy gestite da Amazon Keyspaces](#)

Amazon Keyspaces ha aggiunto nuove autorizzazioni alla policy AmazonKey spacesFullAccess gestita per consentire ad Amazon Keyspaces di creare un ruolo collegato ai servizi quando un amministratore crea uno spazio chiave multiregionale.

5 giugno 2023

[Supporto Amazon Keyspaces per la parola chiave IN](#)

Amazon Keyspaces ora supporta la IN parola chiave nelle SELECT istruzioni.

25 aprile 2023

[Accesso su più account per Amazon Keyspaces e endpoint VPC di interfaccia](#)

Scopri come implementare l'accesso tra account per Amazon Keyspaces con endpoint VPC.

20 aprile 2023

|                                                                                           |                                                                                                                                                                                                                                                                  |                  |
|-------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| <a href="#">Supporto Amazon Keyspaces per timestamp lato client</a>                       | I timestamp lato client di Amazon Keyspaces sono timestamp a livello di cella compatibili con Cassandra che aiutano le applicazioni distribuite a determinare l'ordine delle operazioni di scrittura quando client diversi apportano modifiche agli stessi dati. | 14 marzo 2023    |
| <a href="#">Guida introduttiva ad Amazon Keyspaces e agli endpoint VPC di interfaccia</a> | In questo step-by-step tutorial, scopri come connetterti ad Amazon Keyspaces da un VPC.                                                                                                                                                                          | 1 marzo 2023     |
| <a href="#">Ottimizzazione dei costi delle tabelle Amazon Keyspaces</a>                   | Sono disponibili best practice e linee guida per aiutarti a identificare strategie per ottimizzare i costi delle tabelle Amazon Keyspaces esistenti.                                                                                                             | 17 febbraio 2023 |
| <a href="#">Ora Murmur3Partitioner è l'impostazione predefinita</a>                       | Ora Murmur3Partitioner è il partizionatore predefinito in Amazon Keyspaces.                                                                                                                                                                                      | 17 novembre 2022 |
| <a href="#">Amazon Keyspaces ora supporta Murmur3Partitioner</a>                          | Murmur3Partitioner È ora disponibile in Amazon Keyspaces.                                                                                                                                                                                                        | 9 novembre 2022  |
| <a href="#">Supporta l'aggiornamento per stringhe vuote e valori blob</a>                 | Amazon Keyspaces ora supporta anche stringhe vuote e valori blob come valori di colonne di clustering.                                                                                                                                                           | 19 ottobre 2022  |

[Amazon Keyspaces è ora disponibile in AWS GovCloud \(US\)](#)

Amazon Keyspaces è ora disponibile in AWS GovCloud (US) Region ed è conforme all'elevata conformità a FedRAMP. [Per informazioni sugli endpoint disponibili, consulta Endpoint FIPS.AWS GovCloud \(US\) Region](#)

4 agosto 2022

[Monitora i costi di storage delle tabelle Amazon Keyspaces con Amazon CloudWatch](#)

Amazon Keyspaces ora ti aiuta a monitorare e tenere traccia dei costi di archiviazione delle tabelle nel tempo con questa metrica. `BillableTableSizeInBytes` CloudWatch

14 giugno 2022

[Amazon Keyspaces ora supporta Terraform](#)

Ora puoi usare Terraform per eseguire operazioni DDL (Data Definition Language) in Amazon Keyspaces.

9 giugno 2022

[Supporto per la funzione Amazon Keyspaces token](#)

Amazon Keyspaces ora ti aiuta a ottimizzare le query delle applicazioni utilizzando la funzione. `token`

19 aprile 2022

[Integrazione di Amazon Keyspaces con Apache Spark](#)

[Amazon Keyspaces ora ti aiuta a leggere e scrivere dati in Apache Spark più facilmente utilizzando il connettore open source Spark Cassandra](#)

19 aprile 2022

-

[Riferimento all'API Amazon Keyspaces](#)

Amazon Keyspaces supporta le operazioni del piano di controllo per gestire keyspaces e tabelle utilizzando SDK e AWS CLI. La guida di riferimento dell'API descrive in dettaglio le operazioni del piano di controllo supportate.

2 marzo 2022

[Come risolvere i problemi di configurazione più comuni quando si utilizza Amazon Keyspaces.](#)

Scopri di più su come risolvere i problemi di configurazione più comuni che potresti riscontrare quando usi Amazon Keyspaces.

22 novembre 2021

[Supporto Amazon Keyspaces per Time to Live \(TTL\).](#)

Amazon Keyspaces Time to Live (TTL) ti aiuta a semplificare la logica delle applicazioni e a ottimizzare il prezzo dello storage facendo scadere automaticamente i dati dalle tabelle.

18 ottobre 2021

[Migrazione dei dati su Amazon DSBulk Keyspaces utilizzando.](#)

Step-by-step tutorial per la migrazione dei dati da Apache Cassandra ad Amazon Keyspaces utilizzando Bulk Loader (). DataStax DSBulk

9 agosto 2021

[Supporto di Amazon Keyspaces per le voci degli endpoint VPC nella tabella system.peers](#)

Amazon Keyspaces consente di compilare la `system.peers` tabella con le informazioni sugli endpoint VPC dell'interfaccia disponibili per migliorare il bilanciamento del carico e aumentare la velocità effettiva `read/write`.

29 luglio 2021

|                                                                                                                 |                                                                                                                                                                                                                                   |                 |
|-----------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| <a href="#">Aggiornamento delle policy gestite da IAM per supportare le chiavi gestite dai clienti. AWS KMS</a> | Le policy gestite da IAM per Amazon Keyspaces ora includono le autorizzazioni per elencare e visualizzare le AWS KMS chiavi gestite dai clienti disponibili archiviate in. AWS KMS                                                | 1 giugno 2021   |
| <a href="#">Supporto Amazon Keyspaces per chiavi gestite AWS KMS dai clienti.</a>                               | Amazon Keyspaces ti consente di assumere il controllo delle AWS KMS chiavi gestite dai clienti archiviate AWS KMS per la crittografia a riposo.                                                                                   | 1 giugno 2021   |
| <a href="#">Supporto Amazon Keyspaces per la sintassi JSON</a>                                                  | Amazon Keyspaces ti aiuta a leggere e scrivere documenti JSON più facilmente e supportando la sintassi JSON per le operazioni INSERT e SELECT.                                                                                    | 21 gennaio 2021 |
| <a href="#">Supporto Amazon Keyspaces per colonne statiche</a>                                                  | Amazon Keyspaces ora ti aiuta ad aggiornare e archiviare dati comuni tra più righe in modo efficiente utilizzando colonne statiche.                                                                                               | 9 novembre 2020 |
| <a href="#">Versione GA del supporto NoSQL Workbench per Amazon Keyspaces</a>                                   | NoSQL Workbench è un'applicazione lato client che ti aiuta a progettare e visualizzare più facilmente modelli di dati non relazionali per Amazon Keyspaces. I client NoSQL Workbench sono disponibili per Windows, macOS e Linux. | 28 ottobre 2020 |



[Versione di anteprima del supporto NoSQL Workbench per Amazon Keyspaces](#)

NoSQL Workbench è un'applicazione lato client che ti aiuta a progettare e visualizzare più facilmente modelli di dati non relazionali per Amazon Keyspaces. I client NoSQL Workbench sono disponibili per Windows, macOS e Linux.

5 ottobre 2020

[Nuovi esempi di codice per l'accesso programmatico ad Amazon Keyspaces](#)

Continuiamo ad aggiungere e esempi di codice per l'accesso programmatico ad Amazon Keyspaces. Sono ora disponibili esempi per i driver Java, Python, Go, C# e Perl Cassandra che supportano Apache Cassandra versione 3.11.2.

17 luglio 2020

[point-in-timeRipristino di Amazon Keyspaces \(PITR\)](#)

Amazon Keyspaces ora offre point-in-time il ripristino (PITR) per proteggere le tabelle da operazioni di scrittura o eliminazione accidentali fornendo backup continui dei dati delle tabelle.

9 luglio 2020

[Disponibilità generale di Amazon Keyspaces](#)

Con Amazon Keyspaces , precedentemente noto durante l'anteprima come Amazon Managed Apache Cassandra Service (MCS), puoi usare il codice Cassandra Query Language (CQL), i driver Cassandra con licenza Apache 2.0 e gli strumenti di sviluppo che già usi oggi.

23 aprile 2020

|                                                                                          |                                                                                                                                                                                                                                                                                              |                |
|------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|
| <a href="#">Scalabilità automatica di Amazon Keyspaces</a>                               | Amazon Keyspaces (per Apache Cassandra) si integra con Application Auto Scaling per aiutarti a fornire capacità di throughput in modo efficiente e per carichi di lavoro variabili in risposta al traffico effettivo delle applicazioni regolando automaticamente la capacità di throughput. | 23 aprile 2020 |
| <a href="#">Endpoint VPC (Virtual Private Cloud) di interfaccia per Amazon Keyspaces</a> | Amazon Keyspaces offre comunicazioni private tra il servizio e il tuo VPC in modo che il traffico di rete non lasci la rete Amazon.                                                                                                                                                          | 16 aprile 2020 |
| <a href="#">Politiche di accesso basate su tag</a>                                       | Ora puoi utilizzare i tag di risorsa nelle policy IAM per gestire l'accesso ad Amazon Keyspaces.                                                                                                                                                                                             | 8 aprile 2020  |
| <a href="#">Tipo di dati del contatore</a>                                               | Amazon Keyspaces ora ti aiuta a coordinare incrementi e decrementi dei valori delle colonne utilizzando contatori.                                                                                                                                                                           | 7 aprile 2020  |
| <a href="#">Tagging di risorse</a>                                                       | Amazon Keyspaces ora consente di etichettare e classificare le risorse utilizzando i tag.                                                                                                                                                                                                    | 31 marzo 2020  |
| <a href="#">CloudFormation supporto</a>                                                  | Amazon Keyspaces ora ti aiuta ad automatizzare la creazione e la gestione delle risorse utilizzando CloudFormation.                                                                                                                                                                          | 25 marzo 2020  |

### [Support per ruoli e policy IAM e autenticazione SigV4](#)

Sono state aggiunte informazioni su come utilizzare [AWS Identity and Access Management \(IAM\)](#) per gestire le autorizzazioni di accesso e implementare politiche di sicurezza per Amazon Keyspaces e su come utilizzare il plug-in di autenticazione per il driver Java per Cassandra per DataStax accedere programmaticamente ad Amazon Keyspaces utilizzando ruoli IAM e identità federate.

17 marzo 2020

### [Modalità di capacità di lettura/scrittura](#)

Amazon Keyspaces ora supporta due modalità di capacità di read/write throughput. La modalità di capacità di lettura/scrittura controlla il modo in cui ti viene addebitato il throughput di lettura e scrittura e come viene gestita la capacità di throughput delle tabelle.

20 febbraio 2020

### [Versione iniziale](#)

Questa documentazione riguarda la versione iniziale di Amazon Keyspaces (per Apache Cassandra).

3 dicembre 2019

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.