Migrazione dei lavori SSIS ETL a AWS

AWS Linee guida prescrittive

AWS Linee guida prescrittive: Migrazione dei lavori SSIS ETL a AWS

Table of Contents

Migrazione dei lavori SSIS ETL a AWS

Durga Prasad e Harpreet Singh, Amazon Web Services ()AWS

Dicembre 2021 (cronologia dei documenti)

Amazon Web Services (AWS) fornisce servizi per lo sviluppo e l'esecuzione di processi di estrazione, trasformazione e caricamento (ETL) o di estrazione, caricamento e trasformazione (ELT) e carichi di lavoro di big data. Microsoft SQL Server Integration Services (SSIS) è uno strumento ETL locale dotato di un'interfaccia grafica per la creazione di lavori ETL. A seconda dell'architettura dello stato di destinazione, puoi utilizzare l'interfaccia AWS Glue grafica o il framework Apache Spark con le librerie Python per creare lavori ETL nel Cloud. AWS

Questa guida descrive i passaggi per la migrazione dei job SSIS ETL/ELT da locale a. AWS Descrive le fasi della migrazione, le migliori pratiche e le raccomandazioni per ridurre lo sforzo di migrazione e migliorare l'esperienza. Le informazioni sono applicabili a qualsiasi AWS architettura di destinazione.

La guida è destinata ai responsabili di programma o di progetto, ai proprietari di prodotti, agli architetti di soluzioni e agli sviluppatori che sono:

- Migrazione da SSIS a AWS per vari motivi, come la modernizzazione o il risparmio sui costi.
- Utilizzo principalmente di AWS servizi per carichi di lavoro ETL e big data.
- Alla ricerca di una soluzione basata sul cloud per sfruttare un modello serverless e prezzi flessibili.

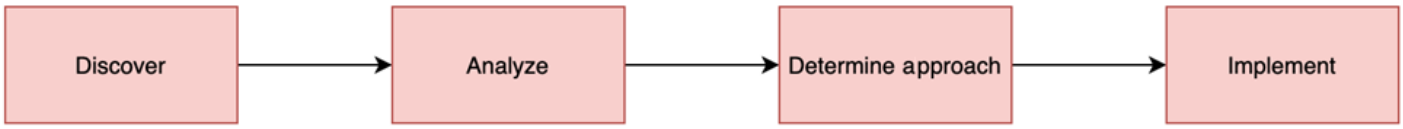
Obiettivi aziendali specifici

La migrazione dei tuoi lavori SSIS sul AWS cloud ti aiuta a raggiungere questi risultati principali:

- Modernizza i processi ETL locali esistenti in modo conveniente
- Rispondi più rapidamente alle esigenze informative dell'organizzazione
- Unisci i processi esistenti ed elimina i processi inutilizzati per ridurre i costi operativi e di manutenzione
- Costruisci una base per soddisfare i futuri requisiti di dati della tua organizzazione

Fasi di migrazione

La migrazione dei processi SSIS ETL sul AWS cloud consiste in quattro fasi principali: scoperta, analisi, determinazione dell'approccio e implementazione.



Queste fasi sono discusse nelle seguenti sezioni:

- Fase di scoperta
- Fase di analisi
- Determinazione dell'approccio alla migrazione
- Fase di implementazione

Fase di scoperta

Nella fase di scoperta, si crea un elenco dei pacchetti SSIS verso i quali si desidera migrare. AWS Diversi team di sviluppo seguono stili, standard e modelli diversi per lo sviluppo di lavori ETL. Ti consigliamo di esaminare i documenti esistenti della tua organizzazione per comprendere questi modelli. Tuttavia, la documentazione è spesso incompleta. È possibile automatizzare l'estrazione di informazioni importanti dagli script ETL. Ciò consente di risparmiare tempo e fatica manuali, riduce gli errori umani e standardizza l'approccio alla migrazione. Ecco alcuni dei dettagli importanti che vorrai estrarre:

- Numero totale di attività del flusso di controllo
- Dettagli delle attività del flusso di controllo
- Numero totale di attività relative al flusso di dati
- Trasformazioni del flusso di dati utilizzate
- Gestori di eventi
- Gestori di connessioni

Utilizzate queste informazioni per comprendere i modelli ETL utilizzati nella vostra organizzazione, per valutarne la complessità e per identificare il AWS servizio appropriato verso cui migrare queste informazioni.

La migrazione di questi dettagli ETL da SSIS rappresenta la maggior parte delle operazioni di migrazione. Tuttavia, proprietà aggiuntive possono fornire informazioni sulle decisioni di progettazione e architettura. Alcune di queste proprietà SSIS sono:

- Checkpoint, utilizzati in SSIS per riavviare i lavori dai punti di errore
- Propagano le variabili, che aiutano un pacchetto SSIS ad avere successo in casi d'uso specifici, anche in caso di errore
- Livelli di isolamento delle transazioni, che controllano la qualità dei dati letti dai database
- Registrazione, per comprendere i tipi di log acquisiti dalla progettazione attuale e le relative posizioni di archiviazione

Il risultato della fase di scoperta può essere un inventario, come illustrato nella tabella seguente.

inventory

count	Package	Flow	Task
1	SSIS_Package_1	control_flow	Microsoft.ExecuteSQLTask
1	SSIS_Package_1	data_flow	Microsoft.BDD
1	SSIS_Package_1	data_flow	Microsoft.ConditionalSplit
2	SSIS_Package_1	data_flow	Microsoft.OLEDBDestination
1	SSIS_Package_1	data_flow	Microsoft.OLEDBSource
1	SSIS_Package_2	control_flow	Microsoft.DbMaintenanceFileCleanupTask
1	SSIS_Package_2	control_flow	Microsoft.ExecuteSQLTask
1	SSIS_Package_2	control_flow	Microsoft.ScriptTask
1	SSIS_Package_2	control_flow	STOCK:FOREACHLOOP
1	SSIS_Package_2	control_flow	STOCK:FORLOOP

Questo inventario potrebbe includere le seguenti informazioni:

- Package: nome del pacchetto SSIS da migrare
- Flusso: flusso di controllo o flusso di dati
- Attività: nome dell'attività del flusso di controllo o del componente del flusso di dati
- Conteggio: numero di volte in cui un'attività è stata utilizzata nel pacchetto SSIS

Fase di analisi

L'analisi delle informazioni della fase di scoperta aiuta a comprendere i modelli e a progettare meglio l'architettura di destinazione. Segui questi suggerimenti per migliorare il risultato dell'analisi:

- L'opzione di registrazione a livello di pacchetto registra gli eventi e acquisisce le informazioni di runtime per ogni componente. Utilizzate la registrazione personalizzata in modo da poter configurare i file di registro in modo da includere informazioni aggiuntive, come ID di esecuzione e altri valori di runtime.
- Reindirizza i record danneggiati in una posizione di archiviazione diversa per l'analisi, la correzione e la rielaborazione.
- Comprendi le strategie di archiviazione ed eliminazione dei dati implementate nell'ambiente SSIS locale.
- Invia avvisi e notifiche utilizzando attività incluse in SSIS (come l'attività Invia posta) o script personalizzati (come l'attività Script).
- Comprendi il comportamento delle trasformazioni nell'ambito per evitare che i dati vengano danneggiati. Ad esempio, la trasformazione Lookup è un equi-join tra due set di dati e il suo confronto fa distinzione tra maiuscole e minuscole.

È possibile mappare ogni voce dell'inventario in base a una complessità stimata, in termini di durata (minuti o ore) o livello (semplice, medio o complesso). Questo inventario esteso, mostrato nella tabella seguente, è il risultato della fase di analisi.

inventory

count	Package	Flow	Task	Effort	Complexity
1	SSIS_Package_1	control_flow	Microsoft.ExecuteSQLTask	30	S
1	SSIS_Package_1	data_flow	Microsoft.BDD	0	N/A
1	SSIS_Package_1	data_flow	Microsoft.ConditionalSplit	30	S
2	SSIS_Package_1	data_flow	Microsoft.OLEDBDestination	60	M
1	SSIS_Package_1	data_flow	Microsoft.OLEDBSource	30	S
1	SSIS_Package_2	control_flow	Microsoft.DbMaintenanceFileCleanupTask	60	M
1	SSIS_Package_2	control_flow	Microsoft.ExecuteSQLTask	30	S
1	SSIS_Package_2	control_flow	Microsoft.ScriptTask	120	C
1	SSIS_Package_2	control_flow	STOCK:FOREACHLOOP	30	S
1	SSIS_Package_2	control_flow	STOCK:FORLOOP	30	S

Determinazione dell'approccio alla migrazione

Per decidere un approccio alla migrazione, si utilizza l'analisi eseguita sui modelli esistenti nella fase precedente. Le future esigenze di dati e analisi della tua organizzazione sono considerazioni altrettanto importanti. Gli strumenti ETL locali tradizionali si occupano di modelli di dati relazionali e dati strutturati. Se hai dati semistrutturati e non strutturati da elaborare, puoi utilizzare AWS servizi come AWS Glue Amazon EMR per la migrazione. Altri fattori che possono influenzare l'approccio alla migrazione includono:

- Sia che si desideri utilizzare un'interfaccia grafica (come AWS Glue Studio) o un framework personalizzato (come le librerie Spark/Python)
- Se disponi di un accesso sicuro a sorgenti e destinazioni locali AWS
- Competenze e formazione richieste al team
- Requisiti di audit e conformità

È possibile scegliere tra tre approcci di migrazione: big bang, phased e lift and shift. La tabella seguente mette a confronto questi tre approcci.

Approccio	Descrizione	Caso d'uso	Vantaggi e svantaggi
Big bang	Migra tutti i pacchetti SSIS entro un periodo di tempo specifico.	• La complessità, l'ambito e l'architettura di destinazione sono chiari. • Il team ha le competenze richieste o la curva di apprendimento è bassa.	• Rischio elevato. • Richiede meno tempo rispetto all'approccio graduale. • Puoi usare AWS Glue Amazon EMR o framework personalizzati.
A fasi	Identifica un pacchetto SSIS per ogni modello e complessità distinti. Migra il pacchetto AWS, testa e confronta i risultati con l'architettura esistente.	• Il tempo non è un vincolo. • Vuoi design diversi per diversi modelli ETL.	• Meno rischioso dell'approccio del big bang ma richiede più tempo e impegno. • Puoi usare AWS Glue Amazon EMR o framework personalizzati.
Solleva e sposta	Migra l'architettura attuale così com'è. AWS	• L'hardware locale non è più supportato. • Non hai le risorse per pianificare immediatamente una migrazione.	• Minori sforzi e tempi di migrazione. • I problemi con la soluzione esistente persistono AWS. • I pacchetti SSIS vengono eseguiti così come sono. Non sono necessari altri strumenti o framework ETL.

Il confronto dei dati sui sistemi di origine e di destinazione è fondamentale per una migrazione di successo. Poiché il sistema di produzione esistente riceve aggiornamenti regolari dai sistemi di origine, questo confronto potrebbe creare confusione. Per questo motivo, quando definisci il tuo approccio alla migrazione, ti consigliamo di decidere anche la tua strategia di convalida dei dati.

- Effettua il backup di tutti i database e i file applicabili dall'ambiente di produzione sul sistema di origine in una data e un'ora specifiche.
- Effettua il backup di tutti i database dall'ambiente di produzione sul sistema di destinazione dopo che tutti i job hanno caricato correttamente i dati dai dati di origine di cui è stato eseguito il backup.
- Ripristina i dati di origine in un ambiente di test ed esegui i nuovi lavori.
- Concordate una percentuale di differenze valide tra i database di origine e di destinazione (vecchi e nuovi). Ad esempio, potresti decidere che una differenza inferiore all'1% è accettabile.
- Elenca tutte le regole di convalida da rispettare.
- Automatizza il confronto il più possibile e rispetta tutte le regole.

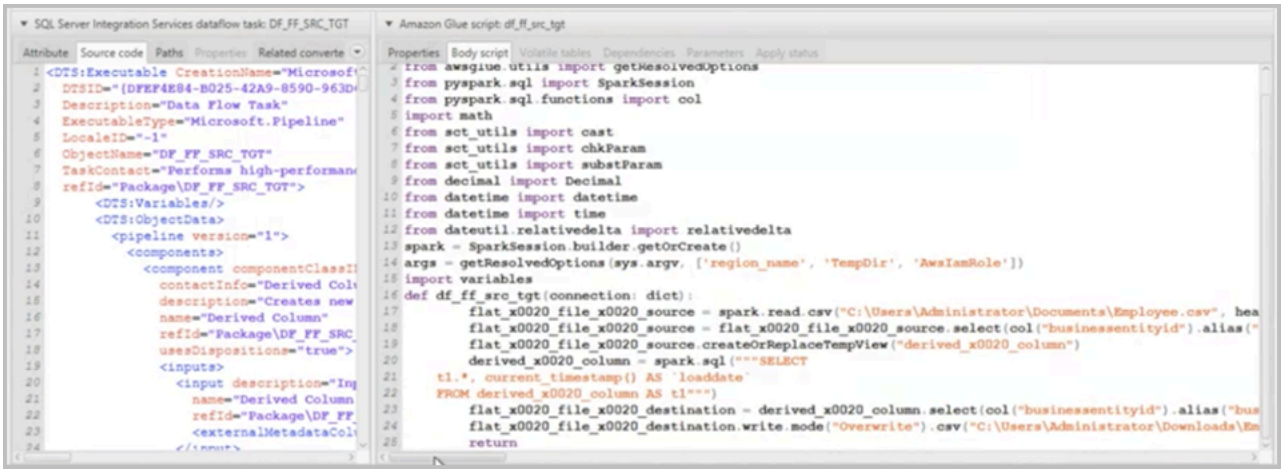
Fase di implementazione

La migrazione che segue il big bang o l'approccio graduale richiede nuovi sviluppi e test. AWS Schema Conversion Tool (AWS SCT) può generare automaticamente AWS Glue lavori dai pacchetti SSIS. Ciò riduce notevolmente i tempi e gli sforzi di migrazione. In alternativa, puoi usare AWS Glue Studio per lo sviluppo basato su interfacce grafiche o creare librerie Spark che puoi eseguire su Amazon EMR o su Amazon EMR. AWS Glue

Le seguenti sezioni forniscono suggerimenti utili per l'utilizzo AWS SCT e Amazon EMR. AWS Glue

AWS SCT

La seguente illustrazione della schermata mostra uno script di AWS Glue lavoro che è stato convertito da. AWS SCT

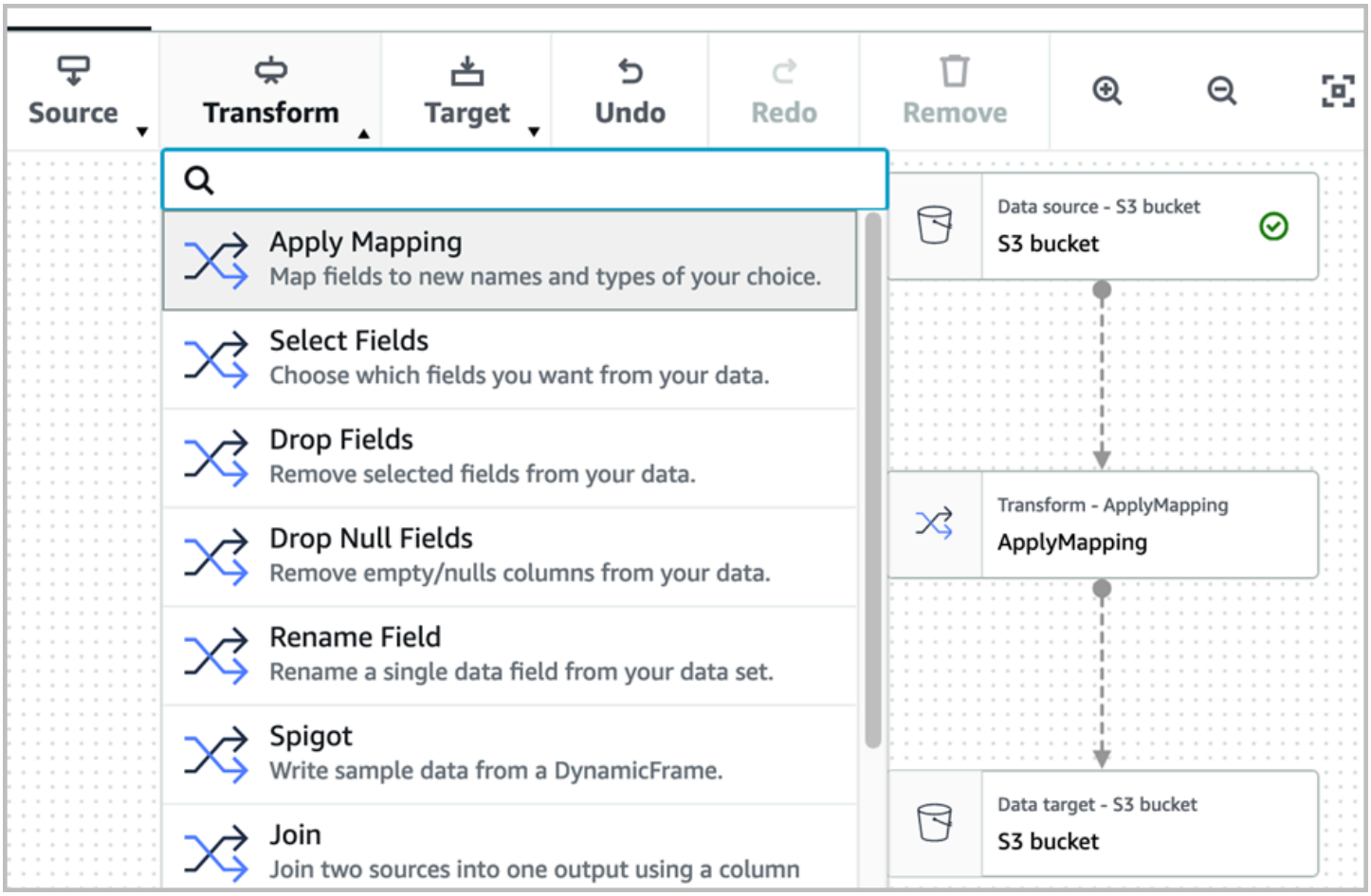


AWS SCT può convertire pacchetti SSIS in AWS Glue lavori in blocco. È possibile modificare lo script per aggiornare la logica esistente o aggiungere una nuova logica, in base al nuovo design. Ti consigliamo di seguire le convenzioni di denominazione degli script AWS SCT convertiti per personalizzare gli script.

Per ulteriori informazioni, consulta Conversione di SSIS all'uso nella documentazione. AWS Glue AWS SCT AWS SCT

AWS Glue

AWS Glue Studio offre un'interfaccia grafica e un'esperienza di sviluppo simili a SSIS, come illustrato nella schermata seguente.



Se preferisci non utilizzare un'interfaccia grafica, puoi anche eseguire i tuoi script personalizzati con le librerie Python richieste dalla console. AWS Glue Per ulteriori informazioni, consulta Fornire script personalizzati nella documentazione. AWS Glue

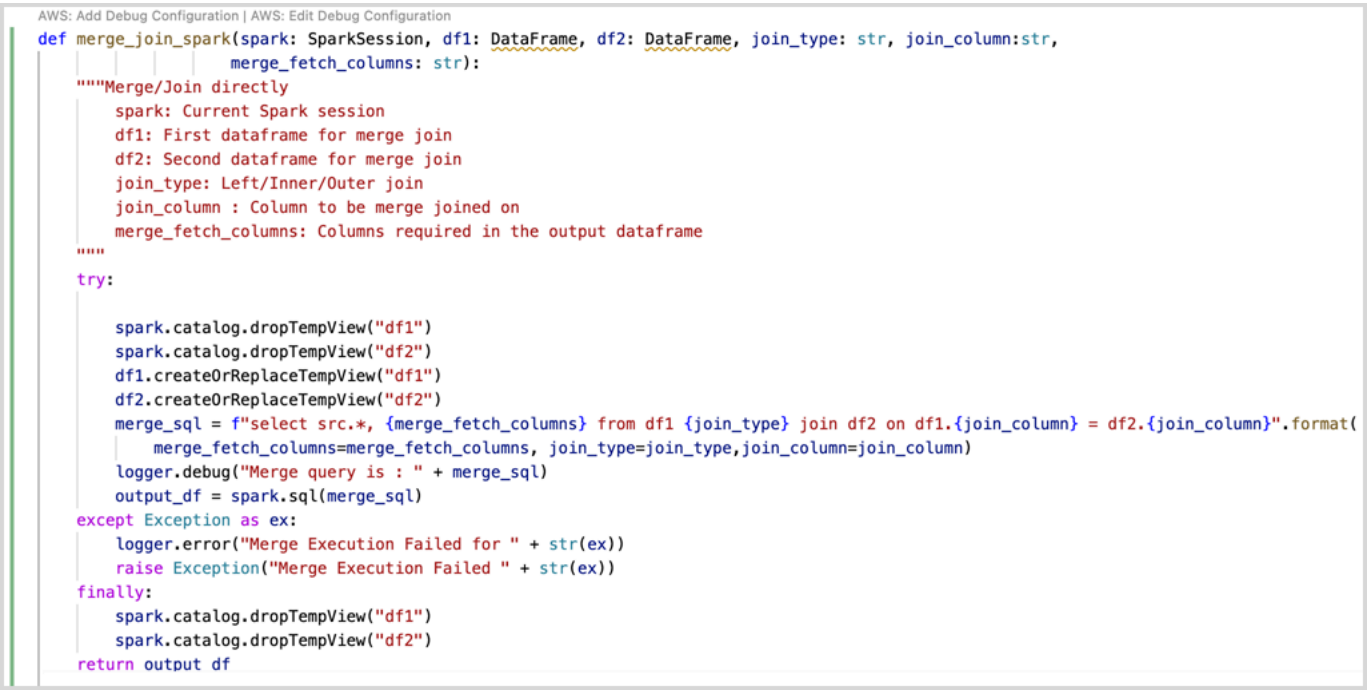
AWS Glue fornisce una serie di trasformazioni integrate per l'elaborazione dei dati. Sono simili alle trasformazioni del flusso di dati SSIS. Segui queste best practice quando migri i tuoi job SSIS ETL utilizzando: AWS Glue

- Preparate una mappatura dalle AWS Glue trasformazioni alle trasformazioni SSIS equivalenti.
- Se le tue trasformazioni non possono essere mappate su AWS Glue trasformazioni, creale usando uno script personalizzato in Python o Scala.
- Per la registrazione personalizzata (ad esempio righe lette, righe scritte o record errati), utilizza script personalizzati oltre ad Amazon. CloudWatch
- Aggiungi un endpoint di sviluppo per sviluppare ed eseguire il debug di script personalizzati a livello locale.

Amazon EMR

È possibile eseguire script personalizzati (scritti in Python o Scala) o librerie Python compilate in cluster EMR, ad esempio. AWS Glue Segui queste best practice:

- Inizia con tipi di istanza ottimizzati per la memoria durante la creazione di cluster EMR con il framework Spark. (SSIS utilizza buffer di memoria).
- Crea metodi Python generici equivalenti a ogni attività o trasformazione SSIS. Ad esempio, nella figura seguente, un metodo che accetta due dataframe come input produce un terzo dataframe che ha come output i record corrispondenti dei due dataframe. Funziona come una trasformazione di merge join.



```
AWS: Add Debug Configuration | AWS: Edit Debug Configuration
def merge_join_spark(spark: SparkSession, df1: DataFrame, df2: DataFrame, join_type: str, join_column:str,
                     merge_fetch_columns: str):
    """Merge/Join directly
        spark: Current Spark session
        df1: First dataframe for merge join
        df2: Second dataframe for merge join
        join_type: Left/Inner/Outer join
        join_column : Column to be merge joined on
        merge_fetch_columns: Columns required in the output dataframe
    """
    try:

        spark.catalog.dropTempView("df1")
        spark.catalog.dropTempView("df2")
        df1.createOrReplaceTempView("df1")
        df2.createOrReplaceTempView("df2")
        merge_sql = f"select src.*, {merge_fetch_columns} from df1 {join_type} join df2 on df1.{join_column} = df2.{join_column}".format(
            merge_fetch_columns=merge_fetch_columns, join_type=join_type,join_column=join_column)
        logger.debug("Merge query is : " + merge_sql)
        output_df = spark.sql(merge_sql)
    except Exception as ex:
        logger.error("Merge Execution Failed for " + str(ex))
        raise Exception("Merge Execution Failed " + str(ex))
    finally:
        spark.catalog.dropTempView("df1")
        spark.catalog.dropTempView("df2")
    return output_df
```

Test in corso

È necessario un framework di test per convalidare la completezza e la correttezza dei dati. Questo framework dovrebbe coprire tutti gli scenari esistenti e tutti i miglioramenti apportati durante la migrazione dei lavori verso. AWS

- Convalida della completezza:
  - Tutti i lavori vengono migrati allo stato di destinazione.
  - Tutte le funzionalità vengono migrate in ogni job.

  - Sono disponibili tutti i tipi di log, inclusi dettagli sull'esecuzione del lavoro, messaggi di errore, record errati e conteggio delle righe.
- Convalida della correttezza:
  - La qualità dei dati è costante negli ambienti esistenti e nuovi.
  - Tutte le colonne di tutte le tabelle corrispondono o le tabelle sono migliorate AWS.
  - Tutte le informazioni di controllo e registrazione corrispondono.

È inoltre necessario verificare che le prestazioni dei lavori migrati corrispondano a quelle dei lavori esistenti.

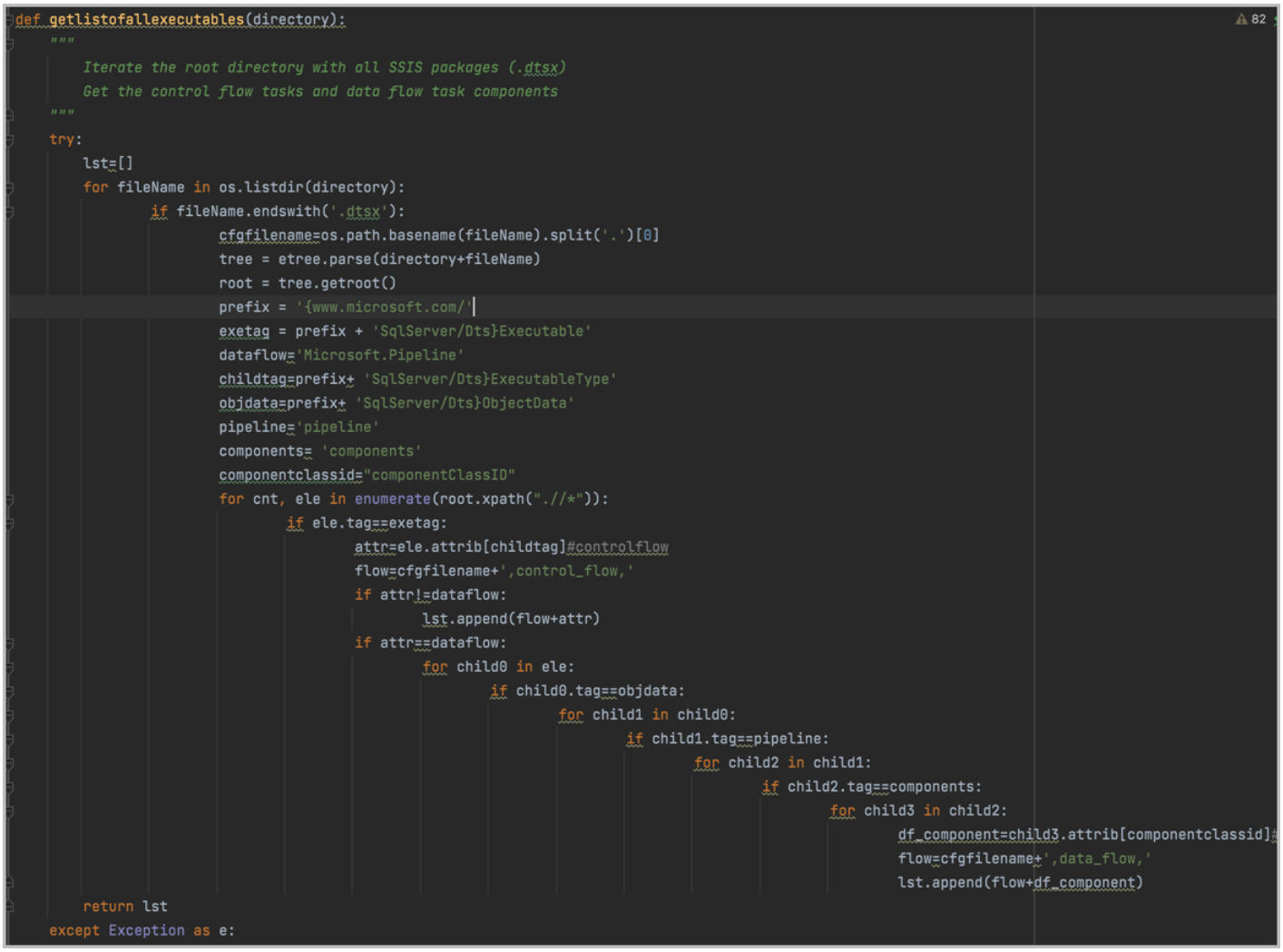
Best practice

Segui queste best practice generali per la migrazione dei job SSIS su: AWS

- Se stai migrando i tuoi database a un servizio AWS gestito come Amazon Relational Database Service (Amazon RDS), descope le attività di manutenzione (come backup, ripristino, aggiornamento delle statistiche o controllo dell'integrità del database) che non sono applicabili o gestite da. AWS
- Crea script e metodi riutilizzabili per standardizzare lo sviluppo e ridurre gli sforzi.
- Testa sempre i lavori migrati su infrastrutture e volumi di dati corrispondenti alle risorse di produzione.
- I pacchetti SSIS sono in formato XML. Crea utilità di analisi XML per automatizzare le attività di migrazione laddove possibile. L'esempio seguente mostra un pacchetto SSIS in formato XML.

```
DTS:LastModifiedProductVersion="11.0.2100.60"
DTS:LocaleID="1033"
DTS:ObjectName="Package"
DTS:PackageType="5"
DTS:VersionBuild="1"
DTS:VersionGUID="{EC16490E-6783-4BE6-92CA-FDD5BC644F88}">
<DTS:Property
  DTS:Name="PackageFormatVersion">6</DTS:Property>
<DTS:ConnectionManagers>
  <DTS:ConnectionManager
```

L'illustrazione seguente mostra un esempio di parser XML.



```
def getlistofallexecutables(directory):
    """
        Iterate the root directory with all SSIS packages (.dtsx)
        Get the control flow tasks and data flow task components
    """
    try:
        lst=[]
        for fileName in os.listdir(directory):
                if fileName.endswith('.dtsx'):
                        cfgfilename=os.path.basename(fileName).split('.')[0]
                        tree = etree.parse(directory+fileName)
                        root = tree.getroot()
                        prefix = '{www.microsoft.com/'
                        exetag = prefix + 'SqlServer/Dts}Executable'
                        dataflow='Microsoft.Pipeline'
                        childtag=prefix+ 'SqlServer/Dts}ExecutableType'
                        objdata=prefix+ 'SqlServer/Dts}ObjectData'
                        pipeline='pipeline'
                        components= 'components'
                        componentclassid="componentClassID"
                        for cnt, ele in enumerate(root.xpath(".//*")):
                                if ele.tag==exetag:
                                        attr=ele.attrib[childtag]#controlflow
                                        flow=cfgfilename+',control_flow,'
                                        if attr!=dataflow:
                                                lst.append(flow+attr)
                                        if attr==dataflow:
                                                for child0 in ele:
                                                        if child0.tag==objdata:
                                                                for child1 in child0:
                                                                        if child1.tag==pipeline:
                                                                                for child2 in child1:
                                                                                        if child2.tag==components:
                                                                                                for child3 in child2:
                                                                                                        df_component=child3.attrib[componentclassid]
                                                                                                        flow=cfgfilename+',data_flow,'
                                                                                                        lst.append(flow+df_component)
        return lst
    except Exception as e:
```

- Utilizza checksum e valori hash per verificare la correttezza e la completezza.
- Se stai creando librerie personalizzate, implementa il threading per eseguire attività in parallelo. Ciò migliora le prestazioni lavorative.
- Disattiva l'ambiente esistente solo dopo che tutti i lavori sono rimasti stabili AWS per un periodo di tempo.
- Usa Amazon CloudWatch per la registrazione, per ridurre gli sforzi di personalizzazione dei log.
- Usa Amazon Simple Notification Service (Amazon SNS) per sostituire le attività personalizzate per l'invio di notifiche. Usa Amazon Simple Email Service (Amazon SES) per sostituire le attività e-mail personalizzate.

Domande frequenti

Questa sezione fornisce le risposte alle domande più frequenti sulla migrazione dei job SSIS ETL a. AWS

Devo migliorare i job SSIS durante la migrazione?

Sì. Implementa contemporaneamente eventuali miglioramenti o correzioni di bug importanti sia nei lavori esistenti che in quelli nuovi. Le modifiche a bassa priorità possono essere implementate dopo la migrazione.

Come posso monitorare i lavori su AWS?

Puoi usare Amazon CloudWatch per monitorare i lavori e inviare avvisi basati su regole.

Quando posso disattivare i miei job SSIS locali?

Si consiglia di mantenere i lavori vecchi e nuovi in parallelo per un periodo di tempo fino a quando le prestazioni, l'accuratezza e la completezza dei dati non siano le stesse in entrambi gli ambienti. È inoltre possibile replicare i sistemi di reporting per collegare e confrontare i report di entrambi gli ambienti. È possibile disattivare i job SSIS quando i report sono identici.

Passaggi successivi

Come passaggio successivo, ti consigliamo di creare un proof of concept per determinare l'approccio di implementazione per l'architettura di destinazione. Per la tua dimostrazione di concetto:

- Crea AWS Glue lavori utilizzando l'interfaccia grafica di AWS Glue Studio.
- Utilizzate AWS Schema Conversion Tool (AWS SCT) per generare AWS Glue script da pacchetti SSIS.
- Crea un framework di migrazione personalizzato utilizzando le librerie Spark.
- Crea un framework di test.
- Identifica strumenti e processi per l'integrazione e l'implementazione continue (CI/CD).

Risorse correlate

- Conversione di SSIS in uso AWS GlueAWS SCT
- Backup e ripristino di Amazon RDS
- AWS Glue documentazione
- AWS Glue FAQs
- Registrazione continua dei lavori AWS Glue
- Documentazione Amazon EMR
- Amazon EMR FAQs
- Monitora le metriche con CloudWatch
- Accelera la convalida della migrazione dei dati su larga scala utilizzando PyDeequ
- AWS Partner competenti in materia di dati e analisi

Cronologia dei documenti

La tabella seguente descrive le modifiche significative apportate a questa guida. Per ricevere notifiche sugli aggiornamenti futuri, puoi abbonarti a un feed RSS.

Modifica	Descrizione	Data
Pubblicazione iniziale	—	6 dicembre 2021

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.