Utilizzo di Apache Iceberg su AWS

AWS Guida prescrittiva

AWS Guida prescrittiva: Utilizzo di Apache Iceberg su AWS

Table of Contents

Usare Apache Iceberg su AWS

Amazon Web Services (collaboratori)

Aprile 2024 (cronologia dei documenti)

Apache Iceberg è un formato di tabella open source che semplifica la gestione delle tabelle migliorando al contempo le prestazioni. AWS i servizi di analisi come Amazon EMR AWS Glue, Amazon Athena e Amazon Redshift includono il supporto nativo per Apache Iceberg, in modo da poter creare facilmente data lake transazionali su Amazon Simple Storage Service (Amazon S3). AWS

Questa guida tecnica fornisce indicazioni su come iniziare a usare Apache Iceberg su diversi argomenti e include best practice e consigli per eseguire Apache Iceberg su larga scala AWS servizi, ottimizzando al contempo costi e prestazioni. AWS

Questa guida si applica a chiunque utilizzi Apache Iceberg su AWS, dagli utenti alle prime armi che desiderano iniziare rapidamente a usare Apache Iceberg agli utenti avanzati che desiderano ottimizzare e ottimizzare i carichi di lavoro Apache Iceberg esistenti. AWS

In questa guida:

Data lake moderni

Casi d'uso avanzati nei data lake moderni

I data lake offrono una delle migliori opzioni per l'archiviazione dei dati in termini di costi, scalabilità e flessibilità. Puoi utilizzare un data lake per conservare grandi volumi di dati strutturati e non strutturati a basso costo e utilizzare questi dati per diversi tipi di carichi di lavoro di analisi, dai report di business intelligence all'elaborazione di big data, all'analisi in tempo reale, all'apprendimento automatico e all'intelligenza artificiale generativa (AI), per guidare decisioni migliori.

Nonostante questi vantaggi, i data lake non sono stati inizialmente progettati con funzionalità simili a quelle dei database. Un data lake non fornisce il supporto per la semantica di elaborazione basata sull'atomicità, la coerenza, l'isolamento e la durabilità (ACID), che potrebbe essere necessaria per ottimizzare e gestire efficacemente i dati su larga scala tra centinaia o migliaia di utenti utilizzando una moltitudine di tecnologie diverse. I data lake non forniscono supporto nativo per le seguenti funzionalità:

- Esecuzione di aggiornamenti ed eliminazioni efficienti a livello di record man mano che i dati cambiano nell'azienda
- Gestione delle prestazioni delle query man mano che le tabelle crescono fino a milioni di file e centinaia di migliaia di partizioni
- Garantire la coerenza dei dati tra più scrittori e lettori simultanei
- Prevenzione del danneggiamento dei dati quando le operazioni di scrittura falliscono a metà dell'operazione
- Evoluzione degli schemi di tabelle nel tempo senza riscrivere (parzialmente) i set di dati

Queste sfide sono diventate particolarmente frequenti in casi d'uso come la gestione dell'acquisizione dei dati di modifica (CDC) o in casi d'uso relativi alla privacy, alla cancellazione dei dati e all'inserimento di dati in streaming, che possono portare a tabelle non ottimali.

I data lake che utilizzano le tradizionali tabelle in formato Hive supportano le operazioni di scrittura solo per interi file. Ciò rende gli aggiornamenti e le eliminazioni difficili da implementare, dispendiosi in termini di tempo e costi. Inoltre, i controlli e le garanzie di concorrenza offerti nei sistemi conformi agli ACID sono necessari per garantire l'integrità e la coerenza dei dati.

Per aiutare a superare queste sfide, Apache Iceberg offre funzionalità aggiuntive simili a quelle dei database che semplificano l'ottimizzazione e la gestione dei data lake, supportando al contempo lo storage su sistemi convenienti come Amazon Simple Storage Service (Amazon S3).

Introduzione ad Apache Iceberg

Apache Iceberg è un formato di tabella open source che fornisce funzionalità nelle tabelle dei data lake che storicamente erano disponibili solo nei database o nei data warehouse. È progettato per garantire scalabilità e prestazioni ed è ideale per la gestione di tabelle di oltre centinaia di gigabyte. Alcune delle caratteristiche principali dei tavoli Iceberg sono:

- Eliminare, aggiornare e unire.Iceberg supporta i comandi SQL standard per il data warehousing da utilizzare con le tabelle dei data lake.
- Pianificazione rapida della scansione e filtraggio avanzato. Iceberg archivia metadati come statistiche a livello di partizioni e colonne che possono essere utilizzati dai motori per velocizzare la pianificazione e l'esecuzione delle query.
- Evoluzione completa dello schema. Iceberg supporta l'aggiunta, l'eliminazione, l'aggiornamento o la ridenominazione di colonne senza effetti collaterali.
- Evoluzione delle partizioni. È possibile aggiornare il layout delle partizioni di una tabella man mano che il volume di dati o i modelli di query cambiano. Iceberg supporta la modifica delle colonne su cui è partizionata una tabella, l'aggiunta di colonne o la rimozione di colonne dalle partizioni composite.
- Partizionamento nascosto.Questa funzione impedisce la lettura automatica delle partizioni non necessarie. Ciò elimina la necessità per gli utenti di comprendere i dettagli di partizionamento della tabella o di aggiungere filtri aggiuntivi alle loro query.
- Ripristino della versione. Gli utenti possono correggere rapidamente i problemi ripristinando lo stato precedente alla transazione.
- Viaggio nel tempo. Gli utenti possono interrogare una versione precedente specifica di una tabella.
- Isolamento serializzabile. Le modifiche alle tabelle sono atomiche, quindi i lettori non vedono mai modifiche parziali o non eseguite.
- Scrittori concorrenti. Iceberg utilizza la concorrenza ottimistica per consentire il successo di più transazioni. In caso di conflitto, uno degli autori deve ritentare la transazione.
- Formati di file aperti. Iceberg supporta diversi formati di file open source, tra cui Apache Parquet, ApacheAvro e Apache ORC.

In sintesi, i data lake che utilizzano il formato Iceberg traggono vantaggio dalla coerenza transazionale, dalla velocità, dalla scalabilità e dall'evoluzione dello schema. Per ulteriori informazioni su queste e altre funzionalità di Iceberg, consulta la documentazione di Apache Iceberg.

AWS supporto per Apache Iceberg

Apache Iceberg è supportato dai più diffusi framework di elaborazione dati open source e da Amazon EMR, AWS servizi Amazon Athena,Amazon Redshift e. AWS Glue Il diagramma seguente illustra un'architettura di riferimento semplificata di un data lake basata su Iceberg.

Quanto segue AWS servizi fornisce integrazioni Iceberg native. Ce ne sono altre AWS servizi che possono interagire con Iceberg, indirettamente o impacchettando le librerie Iceberg.

- Amazon S3 è il posto migliore per creare data lake grazie alle sue capacità di durabilità, disponibilità, scalabilità, sicurezza, conformità e audit. Iceberg è stato progettato e realizzato per interagire con Amazon S3 senza problemi e fornisce supporto per molte funzionalità di Amazon S3 elencate nella documentazione di Iceberg.
- Amazon EMR è una soluzione di big data per l'elaborazione di dati su scala petabyte, l'analisi interattiva e l'apprendimento automatico che utilizza framework open source come Apache Spark, Flink, Trino e Hive. Amazon EMR può essere eseguito su cluster Amazon Elastic Compute Cloud

(Amazon EC2) personalizzati, Amazon Elastic Kubernetes Service (Amazon EKS) o Amazon EMR Serverless. AWS Outposts

- Amazon Athena è un servizio di analisi interattivo senza server basato su framework open source. Supporta formati di file e tabelle aperte e offre un modo semplificato e flessibile per analizzare petabyte di dati nel luogo in cui risiedono. Athena fornisce supporto nativo per lettura, viaggio nel tempo, scrittura e query DDL per Iceberg e utilizza il metastore for the AWS Glue Data Catalog Iceberg.
- Amazon Redshift è un data warehouse cloud su scala petabyte che supporta opzioni di distribuzione basate su cluster e serverless. Amazon Redshift Spectrum può eseguire query su tabelle esterne registrate e archiviate su Amazon S3. AWS Glue Data Catalog Redshift Spectrum supporta anche il formato di storage Iceberg.
- AWS Glueè un servizio di integrazione dei dati senza server che semplifica l'individuazione, la preparazione, lo spostamento e l'integrazione di dati provenienti da più fonti per l'analisi, l'apprendimento automatico (ML) e lo sviluppo di applicazioni. AWS Glue 3.0 e versioni successive supportano il framework Iceberg per i data lake. Puoi utilizzarlo AWS Glue per eseguire operazioni di lettura e scrittura sulle tabelle Iceberg in Amazon S3 o lavorare con le tabelle Iceberg utilizzando. AWS Glue Data Catalog Sono supportate anche operazioni aggiuntive come inserimento, aggiornamento, query Spark e scritture Spark.
- AWS Glue Data Catalogfornisce un servizio di catalogo dati compatibile con Hive metastore che supporta le tabelle Iceberg.
- Crawler di AWS Gluefornisce automazioni per registrare le tabelle Iceberg in. AWS Glue Data Catalog
- Amazon SageMaker supporta l'archiviazione di set di funzionalità in Amazon SageMaker Feature Store utilizzando il formato Iceberg.
- AWS Lake Formationfornisce autorizzazioni di controllo degli accessi grossolane e dettagliate per accedere ai dati, incluse le tabelle Iceberg utilizzate da Athena o Amazon Redshift. Per saperne di più sul supporto delle autorizzazioni per le tabelle Iceberg, consulta la documentazione di Lake Formation.

AWS offre un'ampia gamma di servizi che supportano Iceberg, ma coprire tutti questi servizi non rientra nello scopo di questa guida. Le seguenti sezioni trattano Spark (streaming in batch e strutturato) su Amazon EMR AWS Glue e Amazon Athena SQL. La sezione seguente fornisce una rapida panoramica del supporto Iceberg in Athena SQL.

Guida introduttiva alle tabelle Apache Iceberg in Amazon Athena SQL

Amazon Athena fornisce supporto integrato per Apache Iceberg. È possibile utilizzare Iceberg senza passaggi o configurazioni aggiuntivi, ad eccezione della configurazione dei prerequisiti del servizio descritti nella sezione Guida introduttiva della documentazione di Athena. Questa sezione fornisce una breve introduzione alla creazione di tabelle in Athena. Per ulteriori informazioni, consulta Lavorare con le tabelle Apache Iceberg utilizzando Athena SQL più avanti in questa guida.

È possibile creare tabelle Iceberg AWS utilizzando motori diversi. Queste tabelle funzionano perfettamente su tutti i tavoli. AWS servizi Per creare le tue prime tabelle Iceberg con Athena SQL, puoi utilizzare il seguente codice boilerplate.

```
CREATE TABLE <table_name> (
    col_1 string,
    col_2 string,
    col_3 bigint,
    col_ts timestamp)
PARTITIONED BY (col_1, <<<partition_transform>>>(col_ts))
LOCATION 's3://<bucket>/<folder>/<table_name>/'
TBLPROPERTIES (
    'table_type' ='ICEBERG'
)
```

Le sezioni seguenti forniscono esempi di creazione di tabelle Iceberg partizionate e non partizionate in Athena. Per ulteriori informazioni, consulta la sintassi di Iceberg dettagliata nella documentazione di Athena.

Creazione di una tabella non partizionata

L'istruzione di esempio seguente personalizza il codice SQL standard per creare una tabella Iceberg non partizionata in Athena. È possibile aggiungere questa istruzione all'editor di query in Athenaconsole per creare la tabella.

```
CREATE TABLE athena_iceberg_table (
    color string,
    date string,
    name string,
```

```
    price bigint,
    product string,
    ts timestamp)
LOCATION 's3://DOC_EXAMPLE_BUCKET/ice_warehouse/iceberg_db/athena_iceberg_table/'
TBLPROPERTIES (
    'table_type' ='ICEBERG'
)
```

Per step-by-step istruzioni sull'uso dell'editor di query, consulta Guida introduttiva nella documentazione di Athena.

Creazione di una tabella partizionata

L'istruzione seguente crea una tabella partizionata basata sulla data utilizzando il concetto di partizionamento nascosto di Iceberg. Utilizza la `day()` trasformazione per ricavare partizioni giornaliere, utilizzando il `dd-mm-yyyy` formato, da una colonna di timestamp. Iceberg non memorizza questo valore come nuova colonna nel set di dati. Invece, il valore viene derivato al volo quando si scrivono o si interrogano dati.

```
CREATE TABLE athena_iceberg_table_partitioned (
    color string,
    date string,
    name string,
    price bigint,
    product string,
    ts timestamp)
PARTITIONED BY (day(ts))
LOCATION 's3://DOC_EXAMPLE_BUCKET/ice_warehouse/iceberg_db/athena_iceberg_table/'
TBLPROPERTIES (
    'table_type' ='ICEBERG'
)
```

Creazione di una tabella e caricamento dei dati con una singola istruzione CTAS

Negli esempi partizionati e non partizionati delle sezioni precedenti, le tabelle Iceberg vengono create come tabelle vuote. È possibile caricare dati nelle tabelle utilizzando l'istruzione or. `INSERT MERGE` In alternativa, è possibile utilizzare un'`CREATE TABLE AS SELECT (CTAS)`istruzione per creare e caricare dati in una tabella Iceberg in un unico passaggio.

CTAS è il modo migliore in Athena per creare una tabella e caricare dati in un'unica istruzione. L'esempio seguente illustra come utilizzare CTAS per creare una tabella Iceberg (`iceberg_ctas_table`) da una tabella Hive/Parquet () esistente in Athena. `hive_table`

```
CREATE TABLE iceberg_ctas_table WITH (
  table_type = 'ICEBERG',
  is_external = false,
  location = 's3://DOC_EXAMPLE_BUCKET/ice_warehouse/iceberg_db/iceberg_ctas_table/'
) AS
SELECT * FROM "iceberg_db"."hive_table" limit 20
---
SELECT * FROM "iceberg_db"."iceberg_ctas_table" limit 20
```

Per ulteriori informazioni su CTAS, consulta la documentazione CTAS di Athena.

Inserimento, aggiornamento ed eliminazione dei dati

Athena supporta diversi modi di scrivere dati su una tabella Iceberg utilizzando le istruzioni`INSERT INTO`, `UPDATEMERGE INTO`, e M. `DELETE FRO`

Nota: `UPDATEMERGE INTO`, e `DELETE FROM` utilizzate l' merge-on-read approccio con le eliminazioni posizionali. L' copy-on-write approccio non è attualmente supportato in Athena SQL.

Ad esempio, l'istruzione seguente utilizza `INSERT INTO` per aggiungere dati a una tabella Iceberg:

```
INSERT INTO "iceberg_db"."ice_table" VALUES (
    'red', '222022-07-19T03:47:29', 'PersonNew', 178, 'Tuna', now()
)

SELECT * FROM "iceberg_db"."ice_table"
where color = 'red' limit 10;
```

Output di esempio:

Per ulteriori informazioni, consulta la documentazione di Athena.

Interrogazione delle tabelle Iceberg

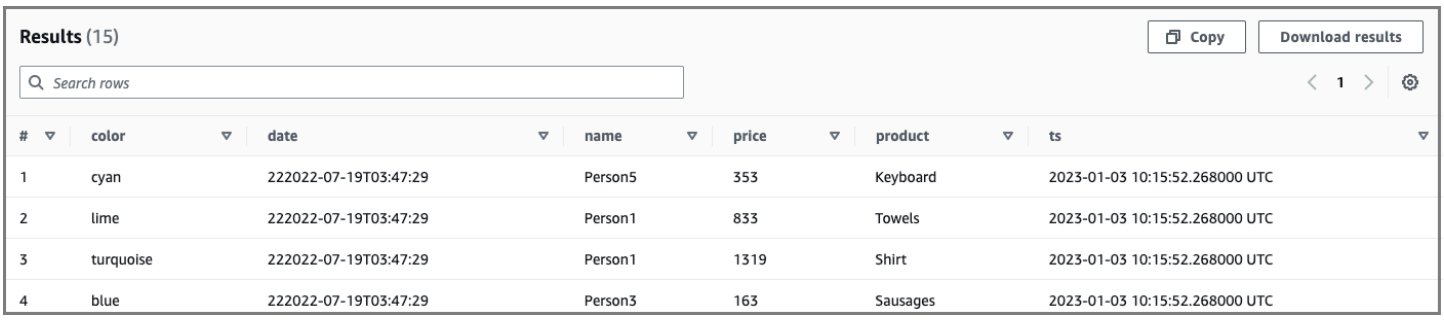
È possibile eseguire normali query SQL sulle tabelle Iceberg utilizzando Athena SQL, come illustrato nell'esempio precedente.

Oltre alle consuete interrogazioni, Athena supporta anche le interrogazioni sui viaggi nel tempo per le tabelle Iceberg. Come discusso in precedenza, è possibile modificare i record esistenti mediante aggiornamenti o eliminazioni in una tabella Iceberg, quindi è comodo utilizzare le query sui viaggi nel tempo per esaminare le versioni precedenti della tabella sulla base di un timestamp o di un ID istantaneo.

Ad esempio, l'istruzione seguente aggiorna un valore di colore per`Person5`, quindi visualizza un valore precedente del 4 gennaio 2023:

```
UPDATE ice_table SET color='new_color' WHERE name='Person5'

SELECT * FROM "iceberg_db"."ice_table" FOR TIMESTAMP AS OF TIMESTAMP '2023-01-04
 12:00:00 UTC'
```

Output di esempio:



Results (15) — Copy — Download results

Search rows — 1

#	color	date	name	price	product	ts
1	cyan	222022-07-19T03:47:29	Person5	353	Keyboard	2023-01-03 10:15:52.268000 UTC
2	lime	222022-07-19T03:47:29	Person1	833	Towels	2023-01-03 10:15:52.268000 UTC
3	turquoise	222022-07-19T03:47:29	Person1	1319	Shirt	2023-01-03 10:15:52.268000 UTC
4	blue	222022-07-19T03:47:29	Person3	163	Sausages	2023-01-03 10:15:52.268000 UTC

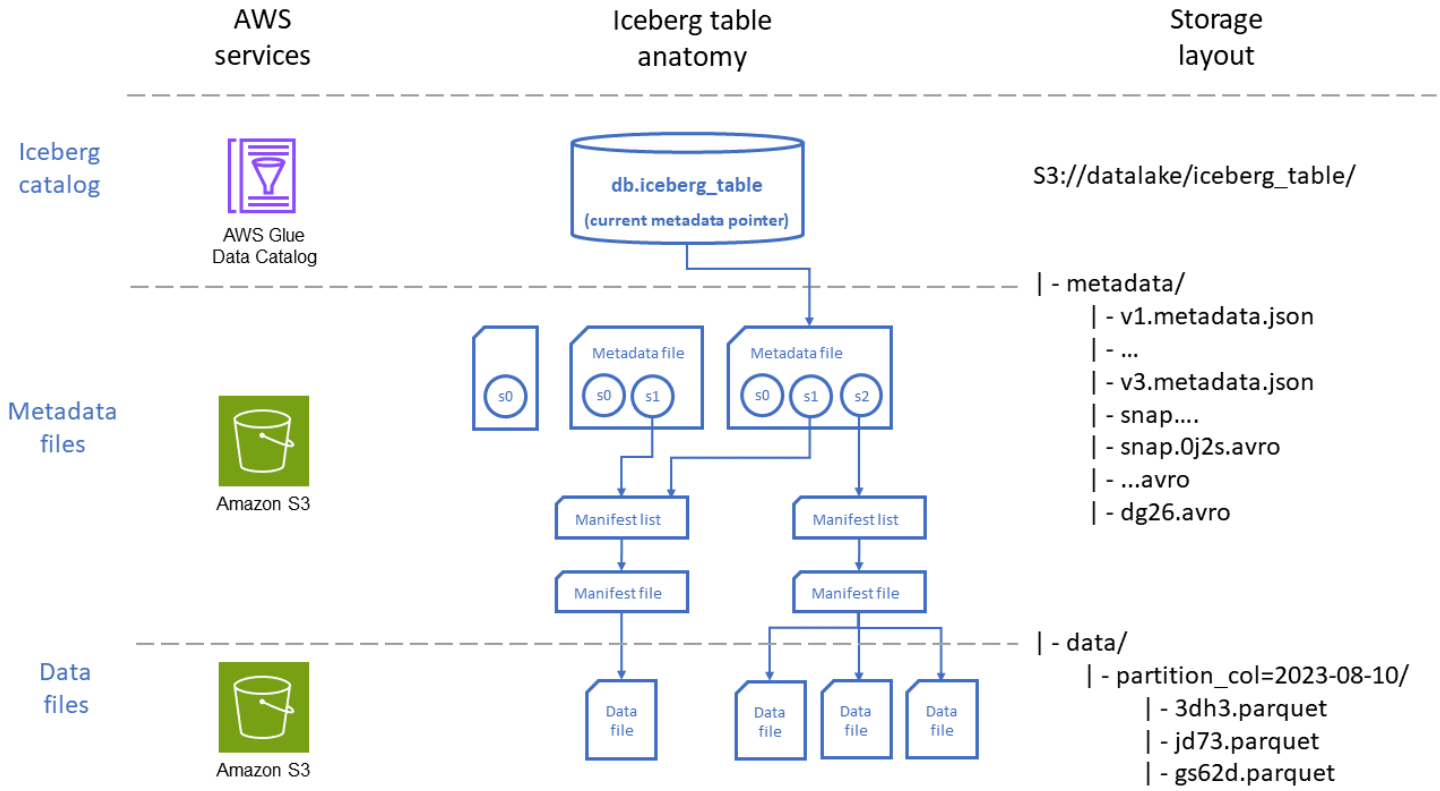
Per la sintassi e altri esempi di domande sui viaggi nel tempo, consulta la documentazione di Athena.

Anatomia del tavolo Iceberg

Ora che abbiamo spiegato i passaggi di base per lavorare con i tavoli Iceberg, approfondiamo i dettagli complessi e il design di un tavolo Iceberg.

Per abilitare le funzionalità descritte in precedenza in questa guida, Iceberg è progettato con livelli gerarchici di dati e file di metadati. Questi livelli gestiscono i metadati in modo intelligente per ottimizzare la pianificazione e l'esecuzione delle query.

Il diagramma seguente illustra l'organizzazione di una tabella Iceberg attraverso due prospettive: quella AWS servizi utilizzata per archiviare la tabella e il posizionamento dei file in Amazon S3.



Come mostrato nel diagramma, una tabella Iceberg è composta da tre livelli principali:

- Catalogo Iceberg: AWS Glue Data Catalog si integra nativamente con Iceberg ed è, per la maggior parte dei casi d'uso, l'opzione migliore per i carichi di lavoro su cui vengono eseguiti. AWS I servizi che interagiscono con le tabelle Iceberg (ad esempio Athena) utilizzano il catalogo per trovare la versione istantanea corrente della tabella, per leggere o scrivere dati.
- Livello di metadati: i file di metadati, vale a dire i file manifest e i file di elenco manifest, tengono traccia di informazioni come lo schema delle tabelle, la strategia di partizione e la posizione dei file di dati, nonché le statistiche a livello di colonna come gli intervalli minimi e massimi per i record archiviati in ogni file di dati. Questi file di metadati sono archiviati in Amazon S3 all'interno del percorso della tabella.
  - I file manifesto contengono un record per ogni file di dati, inclusi posizione, formato, dimensione, checksum e altre informazioni pertinenti.
  - Gli elenchi manifest forniscono un indice dei file manifest. Man mano che il numero di file manifest in una tabella aumenta, la suddivisione di tali informazioni in sottosezioni più piccole aiuta a ridurre il numero di file manifest che devono essere scansionati mediante query.

  - I file di metadati contengono informazioni sull'intera tabella Iceberg, inclusi gli elenchi dei manifest, gli schemi, i metadati delle partizioni, i file di istantanea e altri file utilizzati per gestire i metadati della tabella.
- Livello dati: questo livello contiene i file che contengono i record di dati su cui verranno eseguite le query. Questi file possono essere archiviati in diversi formati, tra cui Apache Parquet, ApacheAvro e Apache ORC.
  - I file di dati contengono i record di dati per una tabella.
  - I file di eliminazione codifica le operazioni di eliminazione e aggiornamento a livello di riga in una tabella Iceberg. Iceberg ha due tipi di eliminazione dei file, come descritto nella documentazione di Iceberg. Questi file vengono creati mediante operazioni che utilizzano la merge-on-read modalità.

Lavorare con Apache Iceberg in Amazon EMR

Amazon EMR fornisce elaborazione dati su scala petabyte, analisi interattive e apprendimento automatico nel cloud utilizzando framework open source come Apache Spark, Apache Hive, Flink e Trino.

> **Note**
>
> Questa guida utilizza Apache Spark per esempi.

Amazon EMR supporta diverse opzioni di distribuzione: Amazon EMR su Amazon EC2, Amazon EMR su Amazon EKS, Amazon EMR Serverless e Amazon EMR su. AWS Outposts Per scegliere un'opzione di distribuzione per il tuo carico di lavoro, consulta le domande frequenti su Amazon EMR.

Compatibilità tra versioni e funzionalità

La versione 6.5.0 e successive di Amazon EMR supportano Apache Iceberg in modo nativo. Per un elenco delle versioni di Iceberg supportate per ogni versione di Amazon EMR, consulta la cronologia delle versioni di Iceberg nella documentazione di Amazon EMR. Consulta anche le considerazioni e le limitazioni relative all'utilizzo di Iceberg su Amazon EMR per vedere quali funzionalità di Iceberg sono supportate in Amazon EMR su diversi framework.

Ti consigliamo di utilizzare la versione più recente di Amazon EMR per sfruttare l'ultima versione supportata di Iceberg. Gli esempi di codice e le configurazioni in questa sezione presuppongono che tu stia utilizzando la release emr-6.9.0 di Amazon EMR.

Creazione di un cluster Amazon EMR con Iceberg

Per creare un cluster Amazon EMR su Amazon EC2 con Iceberg installato, segui le istruzioni nella documentazione di Amazon EMR.

In particolare, il cluster deve essere configurato con la seguente classificazione:

```
[{
    "Classification": "iceberg-defaults",
    "Properties": {
        "iceberg.enabled": "true"
    }
```

```
}]
```

Puoi anche scegliere di utilizzare Amazon EMR Serverless o Amazon EMR su Amazon EKS come opzioni di distribuzione per i tuoi carichi di lavoro Iceberg, a partire da Amazon EMR 6.6.0.

Sviluppo di applicazioni Iceberg in Amazon EMR

Per sviluppare il codice Spark per le tue applicazioni Iceberg, puoi utilizzare Amazon EMR Studio, un ambiente di sviluppo integrato (IDE) basato sul web per notebook Jupyter completamente gestiti eseguiti su cluster Amazon EMR.

Utilizzo dei notebook Amazon EMR Studio

Puoi sviluppare in modo interattivo applicazioni Spark nei notebook Amazon EMR Studio Workspace e connetterli ai tuoi cluster Amazon EMR su cluster Amazon EC2 o Amazon EMR su endpoint gestiti Amazon EKS. Consulta AWS servizio la documentazione per istruzioni sulla configurazione di EMR Studio per Amazon EMR su Amazon EC2 e Amazon EMR su Amazon EKS.

Per utilizzare Iceberg in EMR Studio, segui questi passaggi:

1. Avvia un cluster Amazon EMR con Iceberg abilitato, come indicato in Utilizzare un cluster con Iceberg installato.
2. Configura uno studio EMR. Per istruzioni, consulta Configurare Amazon EMR Studio.
3. Apri un notebook EMR Studio Workspace ed esegui il codice seguente come prima cella del notebook per configurare la sessione Spark per l'utilizzo di Iceberg:

```
%%configure -f
{
    "conf": {
        "spark.sql.catalog.<catalog_name>": "org.apache.iceberg.spark.SparkCatalog",
        "spark.sql.catalog.<catalog_name>.warehouse": "s3://YOUR-BUCKET-NAME/YOUR-
FOLDER-NAME/",
        "spark.sql.catalog.<catalog_name>.catalog-impl":
 "org.apache.iceberg.aws.glue.GlueCatalog",
        "spark.sql.catalog.<catalog_name>.io-impl":
 "org.apache.iceberg.aws.s3.S3FileIO",
        "spark.sql.extensions":
 "org.apache.iceberg.spark.extensions.IcebergSparkSessionExtensions"
    }
}
```

dove:

- `<catalog_name>`è il nome del catalogo della sessione Iceberg Spark. Sostituiscilo con il nome del tuo catalogo e ricorda di cambiare i riferimenti in tutte le configurazioni associate a questo catalogo. Nel codice, dovresti quindi fare riferimento alle tabelle Iceberg con il nome completo della tabella, incluso il nome del catalogo della sessione Spark, come segue:

```
<catalog_name>.<database_name>.<table_name>
```

- `<catalog_name>.warehouse`indica il percorso Amazon S3 in cui desideri archiviare dati e metadati.
- Per rendere il catalogo un AWS Glue Data Catalog, imposta su`<catalog_name>.catalog-impl .org.apache.iceberg.aws.glue.GlueCatalog` Questa chiave è necessaria per indicare una classe di implementazione per qualsiasi implementazione del catalogo personalizzato. La sezione sulle migliori pratiche generali riportata più avanti in questa guida descrive i diversi cataloghi supportati da Iceberg.
- Usalo `org.apache.iceberg.aws.s3.S3FileIO` come per sfruttare il `<catalog_name>.io-impl` caricamento multiparte di Amazon S3 per un elevato parallelismo.

4. Ora puoi iniziare a sviluppare in modo interattivo la tua applicazione Spark per Iceberg nel notebook, come faresti per qualsiasi altra applicazione Spark.

Per ulteriori informazioni sulla configurazione di Spark per Apache Iceberg utilizzando Amazon EMR Studio, consulta il post del blog Crea un data lake in evoluzione, conforme ad ACID e ad alte prestazioni usando Apache Iceberg su Amazon EMR.

Esecuzione di lavori Iceberg in Amazon EMR

Dopo aver sviluppato il codice dell'applicazione Spark per il tuo carico di lavoro Iceberg, puoi eseguirlo su qualsiasi opzione di distribuzione Amazon EMR che supporti Iceberg (consulta le domande frequenti su Amazon EMR).

Come per gli altri job Spark, puoi inviare lavori a un Amazon EMR su un cluster Amazon EC2 aggiungendo passaggi o inviando in modo interattivo i lavori Spark al nodo principale. Per eseguire un job Spark, consulta le seguenti pagine di documentazione di Amazon EMR:

- Per una panoramica delle diverse opzioni di invio del lavoro a un cluster Amazon EMR su Amazon EC2 e istruzioni dettagliate per ciascuna opzione, consulta Invia lavoro a un cluster.

- Per Amazon EMR su Amazon EKS, consulta Running Spark jobs with. StartJobRun
- Per Amazon EMR Serverless, consulta Running jobs.

Le seguenti sezioni forniscono un esempio per ogni opzione di implementazione di Amazon EMR.

Amazon EMR su Amazon EC2

Puoi utilizzare questi passaggi per inviare il job Iceberg Spark:

1. Crea il file `emr_step_iceberg.json` con il seguente contenuto sulla tua workstation:

```
[{
    "Name": "iceberg-test-job",
    "Type": "spark",
    "ActionOnFailure": "CONTINUE",
    "Args": [
        "--deploy-mode",
        "client",
        "--conf",

  "spark.sql.extensions=org.apache.iceberg.spark.extensions.IcebergSparkSessionExtensions",
        "--conf",
        "spark.sql.catalog.<catalog_name>=org.apache.iceberg.spark.SparkCatalog",
        "--conf",
        "spark.sql.catalog.<catalog_name>.catalog-
impl=org.apache.iceberg.aws.glue.GlueCatalog",
        "--conf",
        "spark.sql.catalog.<catalog_name>.warehouse=s3://YOUR-BUCKET-NAME/YOUR-
FOLDER-NAME/",
        "--conf",
        "spark.sql.catalog.<catalog_name>.io-
impl=org.apache.iceberg.aws.s3.S3FileIO",
        "s3://YOUR-BUCKET-NAME/code/iceberg-job.py"
    ]
}]
```

2. Modifica il file di configurazione per il tuo specifico job Spark personalizzando le opzioni di configurazione di Iceberg evidenziate in grassetto.
3. Invia il passaggio utilizzando (). AWS Command Line Interface AWS CLI Esegui il comando nella directory in cui si trova il `emr_step_iceberg.json` file.

```
aws emr add-steps --cluster-id <cluster_id> --steps file://emr_step_iceberg.json
```

Amazon EMR Serverless

Per inviare un job Iceberg Spark ad Amazon EMR Serverless utilizzando: AWS CLI

1. Crea il file `emr_serverless_iceberg.json` con il seguente contenuto sulla tua workstation:

```
{
    "applicationId": "<APPLICATION_ID>",
    "executionRoleArn": "<ROLE_ARN>",
    "jobDriver": {
        "sparkSubmit": {
            "entryPoint": "s3://YOUR-BUCKET-NAME/code/iceberg-job.py",
            "entryPointArguments": [],
            "sparkSubmitParameters": "--jars /usr/share/aws/iceberg/lib/iceberg-
spark3-runtime.jar"
        }
    },
    "configurationOverrides": {
        "applicationConfiguration": [{
            "classification": "spark-defaults",
            "properties": {
                "spark.sql.extensions":
 "org.apache.iceberg.spark.extensions.IcebergSparkSessionExtensions",
                "spark.sql.catalog.<catalog_name>":
 "org.apache.iceberg.spark.SparkCatalog",
                "spark.sql.catalog.<catalog_name>.catalog-impl":
 "org.apache.iceberg.aws.glue.GlueCatalog",
                "spark.sql.catalog.<catalog_name>.warehouse": "s3://YOUR-BUCKET-NAME/
YOUR-FOLDER-NAME/",
                "spark.sql.catalog.<catalog_name>.io-impl":
 "org.apache.iceberg.aws.s3.S3FileIO",
                "spark.jars":"/usr/share/aws/iceberg/lib/iceberg-spark3-runtime.jar",

 "spark.hadoop.hive.metastore.client.factory.class":"com.amazonaws.glue.catalog.metastore.AWS
            }
        }],
        "monitoringConfiguration": {
            "s3MonitoringConfiguration": {
                "logUri": "s3://YOUR-BUCKET-NAME/emr-serverless/logs/"
```

```
            }
        }
    }
}
```

2. Modifica il file di configurazione per il tuo specifico job Spark personalizzando le opzioni di configurazione di Iceberg evidenziate in grassetto.
3. Invia il lavoro utilizzando il. AWS CLI Esegui il comando nella directory in cui si trova il `emr_serverless_iceberg.json` file:

```
aws emr-serverless start-job-run --cli-input-json file://emr_serverless_iceberg.json
```

Per inviare un job Iceberg Spark ad Amazon EMR Serverless utilizzando la console EMR Studio:

1. Segui le istruzioni nella documentazione di Amazon EMR Serverless.
2. Per la configurazione Job, usa la configurazione Iceberg per Spark fornita per AWS CLI e personalizza i campi evidenziati per Iceberg. Per istruzioni dettagliate, consulta Usare Apache Iceberg con EMR Serverless nella documentazione di Amazon EMR.

Amazon EMR su Amazon EKS

Per inviare un lavoro Iceberg Spark ad Amazon EMR su Amazon EKS utilizzando: AWS CLI

1. Crea il file `emr_eks_iceberg.json` con il seguente contenuto sulla tua workstation:

```
{
    "name": "iceberg-test-job",
    "virtualClusterId": "<VIRTUAL_CLUSTER_ID>",
    "executionRoleArn": "<ROLE_ARN>",
    "releaseLabel": "emr-6.9.0-latest",
    "jobDriver": {
        "sparkSubmitJobDriver": {
            "entryPoint": "s3://YOUR-BUCKET-NAME/code/iceberg-job.py",
            "entryPointArguments": [],
            "sparkSubmitParameters": "--jars local:///usr/share/aws/iceberg/lib/
iceberg-spark3-runtime.jar"
        }
    },
    "configurationOverrides": {
        "applicationConfiguration": [{
```

```
            "classification": "spark-defaults",
            "properties": {
                "spark.sql.extensions":
 "org.apache.iceberg.spark.extensions.IcebergSparkSessionExtensions",
                "spark.sql.catalog.<catalog_name>":
 "org.apache.iceberg.spark.SparkCatalog",
                "spark.sql.catalog.<catalog_name>.catalog-impl":
 "org.apache.iceberg.aws.glue.GlueCatalog",
                "spark.sql.catalog.<catalog_name>.warehouse": "s3://YOUR-BUCKET-NAME/
YOUR-FOLDER-NAME/",
                "spark.sql.catalog.<catalog_name>.io-impl":
 "org.apache.iceberg.aws.s3.S3FileIO",
                "spark.hadoop.hive.metastore.client.factory.class":
 "com.amazonaws.glue.catalog.metastore.AWSGlueDataCatalogHiveClientFactory"
            }
        }],
        "monitoringConfiguration": {
            "persistentAppUI": "ENABLED",
            "s3MonitoringConfiguration": {
                "logUri": "s3://YOUR-BUCKET-NAME/emr-serverless/logs/"
            }
        }
    }
}
```

2. Modifica il file di configurazione per il tuo job Spark personalizzando le opzioni di configurazione di Iceberg evidenziate in grassetto.
3. Invia il lavoro utilizzando il. AWS CLI Eseguite il comando seguente nella directory in cui si trova il `emr_eks_iceberg.json` file:

```
aws emr-containers start-job-run --cli-input-json file://emr_eks_iceberg.json
```

Per istruzioni dettagliate, consulta Usare Apache Iceberg con Amazon EMR su EKS nella documentazione di Amazon EMR su EKS.

Best practice per Amazon EMR

Questa sezione fornisce linee guida generali per l'ottimizzazione dei job Spark in Amazon EMR per ottimizzare la lettura e la scrittura dei dati nelle tabelle Iceberg. Per le best practice specifiche di Iceberg, consulta la sezione Best practice più avanti in questa guida.

- Usa la versione più recente di Amazon EMR: Amazon EMR fornisce ottimizzazioni Spark pronte all'uso con il runtime Amazon EMR Spark. AWS migliora le prestazioni del motore di runtime Spark con ogni nuova versione.
- Determina l'infrastruttura ottimale per i tuoi carichi di lavoro Spark: i carichi di lavoro Spark potrebbero richiedere diversi tipi di hardware per diverse caratteristiche di lavoro per garantire prestazioni ottimali. Amazon EMR supporta diversi tipi di istanze (ad esempio ottimizzate per elaborazione, memoria, uso generico e storage ottimizzate) per soddisfare tutti i tipi di requisiti di elaborazione. Quando esegui l'onboarding di nuovi carichi di lavoro, ti consigliamo di eseguire benchmark con tipi di istanze generali come M5 o M6g. Monitora il sistema operativo (OS) e le metriche YARN di Ganglia e Amazon CloudWatch per determinare i colli di bottiglia del sistema (CPU, memoria, storage e I/O) al picco di carico e scegli l'hardware appropriato.
- Tune`spark.sql.shuffle.partitions`: imposta la `spark.sql.shuffle.partitions` proprietà sul numero totale di core virtuali (vCore) nel cluster o su un multiplo di tale valore (in genere, da 1 a 2 volte il numero totale di vCore). Questa impostazione influisce sul parallelismo di Spark quando utilizzi il partizionamento hash e range come modalità di distribuzione della scrittura. Richiede un rimescolamento prima della scrittura per organizzare i dati, il che garantisce l'allineamento delle partizioni.
- Abilita la scalabilità gestita: per quasi tutti i casi d'uso, consigliamo di abilitare la scalabilità gestita e l'allocazione dinamica. Tuttavia, se hai un carico di lavoro con uno schema prevedibile, ti consigliamo di disabilitare la scalabilità automatica e l'allocazione dinamica. Quando la scalabilità gestita è abilitata, ti consigliamo di utilizzare le istanze Spot per ridurre i costi. Utilizza le istanze Spot per i nodi di attività anziché per i nodi principali o master. Quando utilizzi le istanze Spot, utilizza flotte di istanze con più tipi di istanze per flotta per garantire la disponibilità spot.
- Usa broadcast join quando possibile: Broadcast (mapside) join è il join ottimale, a condizione che uno dei tavoli sia sufficientemente piccolo da entrare nella memoria del nodo più piccolo (nell'ordine di MB) e che tu stia eseguendo un join equi (=). Sono supportati tutti i tipi di join ad eccezione dei join esterni completi. Un broadcast join trasmette la tabella più piccola come tabella hash su tutti i nodi di lavoro in memoria. Dopo che la tabella piccola è stata trasmessa, non è possibile modificarla. Poiché la tabella hash si trova localmente nella macchina virtuale Java (JVM), può essere facilmente unita alla tabella grande in base alla condizione di join utilizzando un hash join. I join broadcast offrono prestazioni elevate grazie al minimo sovraccarico di shuffle.
- Ottimizzazione del garbage collector: se i cicli di garbage collection (GC) sono lenti, valuta la possibilità di passare dal garbage collector parallelo predefinito a G1GC per prestazioni migliori. Per ottimizzare le prestazioni GC, puoi ottimizzare i parametri GC. Per tenere traccia delle

prestazioni di GC, puoi monitorarle utilizzando l'interfaccia utente Spark. Idealmente, il tempo GC dovrebbe essere inferiore o uguale all'1% della durata totale dell'attività.

Lavorare con Apache Iceberg in AWS Glue

AWS Glueè un servizio di integrazione dei dati senza server che semplifica la scoperta, la preparazione, lo spostamento e l'integrazione di dati provenienti da più fonti per l'analisi, l'apprendimento automatico (ML) e lo sviluppo di applicazioni. Una delle funzionalità principali di AWS Glue è la sua capacità di eseguire operazioni di estrazione, trasformazione e caricamento (ETL) in modo semplice ed economico. Questo aiuta a classificare i dati, pulirli, arricchirli e spostarli in modo affidabile tra vari archivi di dati e flussi di dati.

AWS Glue i job incapsulano script che definiscono la logica di trasformazione utilizzando un runtime Apache Spark o Python. AWS Glue i lavori possono essere eseguiti sia in modalità batch che in modalità streaming.

Quando si creano lavori Iceberg in AWS Glue, a seconda della versione di AWS Glue, è possibile utilizzare l'integrazione Iceberg nativa o una versione Iceberg personalizzata per allegare le dipendenze di Iceberg al lavoro.

Utilizzo dell'integrazione nativa di Iceberg

AWS Glue le versioni 3.0 e 4.0 supportano nativamente formati di data lake transazionali come Apache Iceberg, Apache Hudi e Linux Foundation Delta Lake per Spark. AWS Glue Questa funzionalità di integrazione semplifica i passaggi di configurazione necessari per iniziare a utilizzare questi framework in. AWS Glue

Per abilitare il supporto Iceberg per il tuo AWS Glue lavoro, imposta il lavoro: scegli la scheda Dettagli del lavoro per il tuo AWS Glue lavoro, scorri fino a Parametri del lavoro in Proprietà avanzate e imposta la chiave `--datalake-formats` e il suo valore `suiceberg`.

Se stai creando un lavoro utilizzando un taccuino, puoi configurare il parametro nella prima cella del notebook usando la `%%configure` magia seguente:

```
%%configure
{
  "--conf" : <job-specific Spark configuration discussed later>,
  "--datalake-formats" : "iceberg"
}
```

Utilizzando una versione personalizzata di Iceberg

In alcune situazioni, potresti voler mantenere il controllo sulla versione di Iceberg per il lavoro e aggiornarla secondo i tuoi ritmi. Ad esempio, l'aggiornamento a una versione successiva può sbloccare l'accesso a nuove funzionalità e miglioramenti delle prestazioni. Per utilizzare una versione specifica di Iceberg con AWS Glue, puoi utilizzare un connettore personalizzato o il tuo file JAR.

Utilizzo di un connettore personalizzato

AWS Glue supporta i connettori, che sono pacchetti di codice opzionali che facilitano l'accesso agli archivi dati in AWS Glue Studio. È possibile sottoscrivere un connettore in Marketplace AWS oppure creare un connettore personalizzato.

> **Note**
>
> Marketplace AWS offre il connettore Apache Iceberg per. AWS Glue Tuttavia, ti consigliamo di utilizzare invece un connettore personalizzato per mantenere il controllo sulle versioni di Iceberg.

Ad esempio, per creare un connettore cliente per la versione 0.13.1 di Iceberg, procedi nel seguente modo:

1. Carica i file `iceberg-spark-runtime-3.1_2.12-0.13.1.jar` e `bundle-2.17.161.jar` in un `url-connection-client-2.17.161.jar` bucket Amazon S3. Puoi scaricare questi file dai rispettivi repository Apache Maven.
2. Sulla AWS Glue Studio console, crea un connettore Spark personalizzato:
   a. Nel pannello di navigazione, scegli Connessioni dati. (Se utilizzi la navigazione precedente, scegli Connettori, Crea connettore personalizzato).
   b. Nella casella Connettori, scegli Crea connettore personalizzato.
   c. Nella pagina Crea connettore personalizzato:
      - Specificare il percorso dei file JAR in Amazon S3.
      - Inserisci un nome per il connettore.
      - Scegli Spark come tipo di connettore.
      - Per il nome della classe, specifica il nome completo della classe dell'origine dati (o il relativo alias) che usi quando carichi l'origine dati Spark con l'operatore. `format`

- (Facoltativo) Fornisci una descrizione del connettore.

3. Scegli Create connector (Crea connettore).

Quando si utilizzano i connettori AWS Glue, è necessario creare una connessione per il connettore. Una connessione contiene le proprietà necessarie per connettersi a un particolare archivio dati. È possibile utilizzare la connessione con le tue origini dati e destinazioni dati nel processo ETL. Connettori e connessioni funzionano insieme per facilitare l'accesso ai datastore.

Per creare una connessione utilizzando il connettore Iceberg personalizzato che hai creato:

1. Sulla AWS Glue Studio console, seleziona il tuo connettore Iceberg personalizzato.
2. Segui le istruzioni per fornire i dettagli, come il tuo VPC e altre configurazioni di rete richieste dal processo, quindi scegli Crea connessione.

Ora puoi usare la connessione nel tuo job ETL. AWS Glue A seconda di come crei il lavoro, ci sono diversi modi per collegare la connessione al tuo lavoro:

- Se si crea un lavoro visivo utilizzando AWS Glue Studio, è possibile selezionare la connessione dall'elenco Connessione nelle proprietà dell'origine dati — scheda Connettore.
- Se sviluppi il lavoro su un notebook, usa la `%connections` magia per impostare il nome della connessione:

```
%glue_version 3.0

%connections <name-of-the iceberg-connection>

%%configure
{
  "--conf" : "job-specific Spark configurations, to be discussed later",
  "--datalake-formats" : "iceberg"
}
```

- Se si crea il lavoro utilizzando l'editor di script, specificare la connessione nella scheda Dettagli del lavoro, in Proprietà avanzate, Connessioni di rete aggiuntive.

Per ulteriori informazioni sulle procedure di questa sezione, vedere Utilizzo di connettori e connessioni con AWS Glue Studio nella AWS Glue documentazione.

Portare i propri file JAR

Inoltre AWS Glue, puoi lavorare con Iceberg senza dover usare un connettore. Questo approccio è utile quando si desidera mantenere il controllo sulla versione di Iceberg e aggiornarla rapidamente. Per utilizzare questa opzione, carica i file JAR Iceberg richiesti in un bucket S3 a tua scelta e fai riferimento ai file del tuo job. AWS Glue Ad esempio, se stai lavorando con Iceberg 1.0.0, i file JAR richiesti sono, e. `iceberg-spark-runtime-3.0_2.12-1.0.0.jar` `url-connection-client-2.15.40.jar` `bundle-2.15.40.jar` Puoi anche dare la priorità ai file JAR aggiuntivi nel percorso della classe impostando il `--user-jars-first` parametro su for the job. `true`

Configurazioni Spark per Iceberg in AWS Glue

Questa sezione descrive le configurazioni Spark necessarie per creare un job AWS Glue ETL per un set di dati Iceberg. Puoi impostare queste configurazioni utilizzando la chiave Spark con un elenco separato da virgole di tutte le chiavi e i valori di configurazione `--conf` Spark. Puoi usare la `%%configure` magia in un notebook o nella sezione Job parameters della AWS Glue Studio console.

```
%glue_version 3.0

%connections <name-of-the iceberg-connection>

%%configure
{
  "--conf" : "spark.sql.extensions=org.apache.iceberg.spark.extensions...",
  "--datalake-formats" : "iceberg"
}
```

Configura la sessione Spark con le seguenti proprietà:

- `<catalog_name>`è il nome del catalogo della sessione Iceberg Spark. Sostituiscilo con il nome del tuo catalogo e ricorda di cambiare i riferimenti in tutte le configurazioni associate a questo catalogo. Nel codice, dovresti quindi fare riferimento alle tabelle Iceberg con il nome completo della tabella, incluso il nome del catalogo della sessione Spark, come segue:. `<catalog_name>.<database_name>.<table_name>`
- `<catalog_name>.<warehouse>`indica il percorso Amazon S3 in cui desideri archiviare dati e metadati.
- Per rendere il catalogo un AWS Glue Data Catalog, imposta su`<catalog_name>.catalog-impl`. `org.apache.iceberg.aws.glue.GlueCatalog` Questa chiave è necessaria per

indicare una classe di implementazione per qualsiasi implementazione del catalogo personalizzato. Per i cataloghi supportati da Iceberg, consulta la ??? sezione Buone pratiche generali più avanti in questa guida.

- Usalo `org.apache.iceberg.aws.s3.S3FileIO` come per sfruttare il `<catalog_name>.io-impl` caricamento multiparte di Amazon S3 per un elevato parallelismo.

Ad esempio, se hai un catalogo chiamato`glue_iceberg`, puoi configurare il tuo lavoro utilizzando più `--conf` chiavi come segue:

```
%%configure
{
  "--datalake-formats" : "iceberg",
  "--conf" :
 "spark.sql.extensions=org.apache.iceberg.spark.extensions.IcebergSparkSessionExtensions",
  "--conf" : "spark.sql.catalog.glue_iceberg=org.apache.iceberg.spark.SparkCatalog",
  "--conf" : "spark.sql.catalog.glue_iceberg.warehouse=s3://<your-warehouse-dir>=>/",
  "--conf" : " spark.sql.catalog.glue_iceberg.catalog-
impl=org.apache.iceberg.aws.glue.GlueCatalog ",
  "--conf" : " spark.sql.catalog.glue_iceberg.io-
impl=org.apache.iceberg.aws.s3.S3FileIO
}
```

In alternativa, puoi usare il codice per aggiungere le configurazioni precedenti allo script Spark come segue:

```
spark = SparkSession.builder\

 .config("spark.sql.extensions","org.apache.iceberg.spark.extensions.IcebergSparkSessionExtensi
                    .config("spark.sql.catalog.glue_iceberg",
 "org.apache.iceberg.spark.SparkCatalog")\
                    .config("spark.sql.catalog.glue_iceberg.warehouse","s3://<your-
warehouse-dir>/")\
                    .config("spark.sql.catalog.glue_iceberg.catalog-impl",
 "org.apache.iceberg.aws.glue.GlueCatalog") \
                    .config("spark.sql.catalog.glue_iceberg.io-impl",
 "org.apache.iceberg.aws.s3.S3FileIO") \
                    .getOrCreate()
```

Le migliori pratiche per le offerte di lavoro AWS Glue

Questa sezione fornisce linee guida generali per ottimizzare la lettura e la scrittura dei dati nelle tabelle Iceberg AWS Glue per ottimizzare la lettura e la scrittura dei dati nelle tabelle Iceberg. Per le best practice specifiche per Iceberg, consulta la sezione Best practice più avanti in questa guida.

- Usa l'ultima versione di AWS Glue ed esegui l'upgrade ogni volta che è possibile: le nuove versioni AWS Glue forniscono miglioramenti delle prestazioni, tempi di avvio ridotti e nuove funzionalità. Supportano anche le versioni più recenti di Spark che potrebbero essere necessarie per le ultime versioni di Iceberg. Per un elenco delle AWS Glue versioni disponibili e delle versioni di Spark supportate, consulta la documentazione.AWS Glue
- Ottimizzazione della memoria di AWS Glue lavoro: segui i consigli contenuti nel post sul AWS blog Ottimizza la gestione della memoria in AWS Glue.
- Usa AWS Glue Auto Scaling: quando abiliti Auto Scaling AWS Glue , regola automaticamente il numero di lavoratori in modo dinamico in base al AWS Glue carico di lavoro. Questo aiuta a ridurre il costo del AWS Glue lavoro durante i picchi di carico, perché riduce il numero di lavoratori quando il carico di lavoro è AWS Glue ridotto e i lavoratori sono inattivi. Per utilizzare AWS Glue Auto Scaling, è necessario specificare un numero massimo di lavoratori a cui è possibile adattare il AWS Glue lavoro. Per ulteriori informazioni, vedere Utilizzo della scalabilità automaticaAWS Glue nella AWS Glue documentazione.
- Usa connettori personalizzati o aggiungi dipendenze dalla libreria: l'integrazione AWS Glue nativa per Iceberg è la migliore per iniziare a usare Iceberg. Tuttavia, per i carichi di lavoro di produzione, consigliamo di utilizzare contenitori personalizzati o aggiungere dipendenze di libreria (come discusso in precedenza in questa guida) per avere il pieno controllo sulla versione di Iceberg. Questo approccio ti aiuta a trarre vantaggio dalle ultime funzionalità di Iceberg e dai miglioramenti delle prestazioni nei tuoi lavori. AWS Glue
- Abilita l'interfaccia utente Spark per il monitoraggio e il debug: puoi anche utilizzare l'interfaccia utente di Spark per AWS Glue ispezionare il tuo lavoro Iceberg visualizzando le diverse fasi di un lavoro Spark in un grafico aciclico diretto (DAG) e monitorando i lavori in dettaglio. L'interfaccia utente di Spark offre un modo efficace per risolvere i problemi e ottimizzare i job Iceberg. Ad esempio, è possibile identificare le fasi che presentano punti critici che presentano grosse variazioni o fuoriuscite del disco per individuare le opportunità di ottimizzazione. Per ulteriori informazioni, consulta Monitoraggio dei lavori utilizzando l'interfaccia utente web di Apache Spark nella documentazione. AWS Glue

Lavorare con le tabelle Apache Iceberg utilizzando Apache Spark

Questa sezione fornisce una panoramica sull'uso di Apache Spark per interagire con le tabelle Iceberg. Gli esempi sono codice standard che può essere eseguito su Amazon EMR o. AWS Glue

Nota: l'interfaccia principale per l'interazione con le tabelle Iceberg è SQL, quindi la maggior parte degli esempi combinerà Spark SQL con l'API. DataFrames

Creazione e scrittura di tabelle Iceberg

Puoi usare Spark SQL e Spark DataFrames per creare e aggiungere dati alle tabelle Iceberg.

Usare Spark SQL

Per scrivere un set di dati Iceberg, usa istruzioni SQL Spark standard come e. `CREATE TABLE` `INSERT INTO`

Tabelle non partizionate

Ecco un esempio di creazione di una tabella Iceberg non partizionata con Spark SQL:

```
spark.sql(f"""
    CREATE TABLE IF NOT EXISTS {CATALOG_NAME}.{DB_NAME}.{TABLE_NAME}_nopartitions (
        c_customer_sk             int,
        c_customer_id             string,
        c_first_name              string,
        c_last_name               string,
        c_birth_country           string,
        c_email_address           string)
    USING iceberg
    OPTIONS ('format-version'='2')
""")
```

Per inserire dati in una tabella non partizionata, usa un'istruzione standard: `INSERT INTO`

```
spark.sql(f"""
INSERT INTO {CATALOG_NAME}.{DB_NAME}.{TABLE_NAME}_nopartitions
```

```
SELECT c_customer_sk, c_customer_id, c_first_name, c_last_name, c_birth_country,
 c_email_address
FROM another_table
""")
```

Tabelle partizionate

Ecco un esempio di creazione di una tabella Iceberg partizionata con Spark SQL:

```
spark.sql(f"""
    CREATE TABLE IF NOT EXISTS {CATALOG_NAME}.{DB_NAME}.{TABLE_NAME}_withpartitions (
        c_customer_sk             int,
        c_customer_id             string,
        c_first_name              string,
        c_last_name               string,
        c_birth_country           string,
        c_email_address           string)
    USING iceberg
    PARTITIONED BY (c_birth_country)
    OPTIONS ('format-version'='2')
""")
```

Per inserire dati in una tabella Iceberg partizionata con Spark SQL, esegui un ordinamento globale e poi scrivi i dati:

```
spark.sql(f"""
INSERT INTO {CATALOG_NAME}.{DB_NAME}.{TABLE_NAME}_withpartitions
SELECT c_customer_sk, c_customer_id, c_first_name, c_last_name, c_birth_country,
 c_email_address
FROM another_table
ORDER BY c_birth_country
""")
```

Utilizzando l'API DataFrames

Per scrivere un set di dati Iceberg, puoi utilizzare l'`DataFrameWriterV2`API.

Per creare una tabella Iceberg e scrivere dati su di essa, usa la funzione `df.writeTo(` t). Se la tabella esiste, usa la `.append()` funzione. In caso contrario, usa `.create()`. Gli esempi seguenti usano`.createOrReplace()`, che è una variante di `.create()` che è equivalente a`CREATE OR REPLACE TABLE AS SELECT`.

Tabelle non partizionate

Per creare e popolare una tabella Iceberg non partizionata utilizzando l'API: `DataFrameWriterV2`

```
input_data.writeTo(f"{CATALOG_NAME}.{DB_NAME}.{TABLE_NAME}_nopartitions") \
    .tableProperty("format-version", "2") \
    .createOrReplace()
```

Per inserire dati in una tabella Iceberg non partizionata esistente utilizzando l'API: `DataFrameWriterV2`

```
input_data.writeTo(f"{CATALOG_NAME}.{DB_NAME}.{TABLE_NAME}_nopartitions") \
    .append()
```

Tabelle partizionate

Per creare e popolare una tabella Iceberg partizionata utilizzando l'`DataFrameWriterV2`API, puoi utilizzare un ordinamento locale per importare i dati:

```
input_data.sortWithinPartitions("c_birth_country") \
    .writeTo(f"{CATALOG_NAME}.{DB_NAME}.{TABLE_NAME}_withpartitions") \
    .tableProperty("format-version", "2") \
    .partitionedBy("c_birth_country") \
    .createOrReplace()
```

Per inserire dati in una tabella Iceberg partizionata utilizzando l'`DataFrameWriterV2`API, puoi utilizzare un ordinamento globale per inserire i dati:

```
input_data.orderBy("c_birth_country") \
    .writeTo(f"{CATALOG_NAME}.{DB_NAME}.{TABLE_NAME}_withpartitions") \
    .append()
```

Aggiornamento dei dati nelle tabelle Iceberg

L'esempio seguente mostra come aggiornare i dati in una tabella Iceberg. Questo esempio modifica tutte le righe che hanno un numero pari nella `c_customer_sk` colonna.

```
spark.sql(f"""
UPDATE {CATALOG_NAME}.{db.name}.{table.name}
SET c_email_address = 'even_row'
```

```
WHERE c_customer_sk % 2 == 0
""")
```

Questa operazione utilizza la copy-on-write strategia predefinita, quindi riscrive tutti i file di dati interessati.

Sconvolgimento dei dati nelle tabelle Iceberg

L'alterazione dei dati si riferisce all'inserimento di nuovi record di dati e all'aggiornamento dei record di dati esistenti in un'unica transazione. Per trasformare i dati in una tabella Iceberg, si utilizza l'istruzione. `SQL MERGE INTO`

L'esempio seguente inverte il contenuto della tabella} all'interno della tabella`{UPSERT_TABLE_NAME:` `{TABLE_NAME}`

```
spark.sql(f"""
    MERGE INTO {CATALOG_NAME}.{DB_NAME}.{TABLE_NAME} t
    USING {UPSERT_TABLE_NAME} s
        ON t.c_customer_id = s.c_customer_id
    WHEN MATCHED THEN UPDATE SET t.c_email_address = s.c_email_address
    WHEN NOT MATCHED THEN INSERT *
""")
```

- Se un record cliente presente in esiste `{UPSERT_TABLE_NAME}` già in `{TABLE_NAME}` con lo stesso`c_customer_id`, il valore del `{UPSERT_TABLE_NAME}` record sostituisce il `c_email_address` valore esistente (operazione di aggiornamento).
- Se un record del cliente presente in `{UPSERT_TABLE_NAME}` non esiste in`{TABLE_NAME}`, il `{UPSERT_TABLE_NAME}` record viene aggiunto a `{TABLE_NAME}` (operazione di inserimento).

Eliminazione dei dati nelle tabelle Iceberg

Per eliminare i dati da una tabella Iceberg, utilizzate l'`DELETE FROM`espressione e specificate un filtro che corrisponda alle righe da eliminare.

```
spark.sql(f"""
DELETE FROM {CATALOG_NAME}.{db.name}.{table.name}
WHERE c_customer_sk % 2 != 0
""")
```

Se il filtro corrisponde a un'intera partizione, Iceberg elimina solo i metadati e lascia i file di dati al loro posto. Altrimenti, riscrive solo i file di dati interessati.

Il metodo delete prende i file di dati interessati dalla `WHERE` clausola e ne crea una copia senza i record eliminati. Quindi crea una nuova istantanea della tabella che punta ai nuovi file di dati. Pertanto, i record eliminati sono ancora presenti nelle istantanee precedenti della tabella. Ad esempio, se recuperi l'istantanea precedente della tabella, vedrai i dati che hai appena eliminato. Per informazioni sulla rimozione di vecchie istantanee non necessarie con i relativi file di dati per scopi di pulizia, consulta la sezione Manutenzione dei file utilizzando la compattazione più avanti in questa guida.

Lettura dei dati

Puoi leggere lo stato più recente delle tue tabelle Iceberg in Spark sia con Spark SQL che. DataFrames

Esempio di utilizzo di Spark SQL:

```
spark.sql(f"""
SELECT * FROM {CATALOG_NAME}.{db.name}.{table.name} LIMIT 5
""")
```

Esempio di utilizzo dell' DataFrames API:

```
df = spark.table(f"{CATALOG_NAME}.{DB_NAME}.{TABLE_NAME}").limit(5)
```

Utilizzo del viaggio nel tempo

Ogni operazione di scrittura (inserimento, aggiornamento, annullamento, eliminazione) in una tabella Iceberg crea una nuova istantanea. È quindi possibile utilizzare queste istantanee per viaggiare nel tempo, per tornare indietro nel tempo e controllare lo stato di una tabella nel passato.

Per informazioni su come recuperare la cronologia delle istantanee per le tabelle utilizzando `snapshot-id` e temporizzando i valori, consultate la sezione Accesso ai metadati più avanti in questa guida.

La seguente query sui viaggi nel tempo mostra lo stato di una tabella in base a uno specifico. `snapshot-id`

Usando Spark SQL:

```
spark.sql(f"""
SELECT * FROM {CATALOG_NAME}.{DB_NAME}.{TABLE_NAME} VERSION AS OF {snapshot_id}
""")
```

Utilizzo dell' DataFrames API:

```
df_1st_snapshot_id = spark.read.option("snapshot-id", snapshot_id) \
     .format("iceberg") \
     .load(f"{CATALOG_NAME}.{DB_NAME}.{TABLE_NAME}") \
     .limit(5)
```

La seguente query sul viaggio nel tempo mostra lo stato di una tabella in base all'ultima istantanea creata prima di un timestamp specifico, in millisecondi (). `as-of-timestamp`

Usando Spark SQL:

```
spark.sql(f"""
SELECT * FROM dev.{db.name}.{table.name} TIMESTAMP AS OF '{snapshot_ts}'
""")
```

Utilizzo dell' DataFrames API:

```
df_1st_snapshot_ts = spark.read.option("as-of-timestamp", snapshot_ts) \
                          .format("iceberg") \
                          .load(f"dev.{DB_NAME}.{TABLE_NAME}") \
                          .limit(5)
```

Utilizzo di query incrementali

È inoltre possibile utilizzare le istantanee Iceberg per leggere i dati aggiunti in modo incrementale.

Nota: attualmente, questa operazione supporta la lettura di dati da istantanee. `append` Non supporta il recupero di dati da operazioni come `replace`, `overwrite`, o. `delete` Inoltre, le operazioni di lettura incrementali non sono supportate nella sintassi SQL di Spark.

L'esempio seguente recupera tutti i record aggiunti a una tabella Iceberg compresi tra l'istantanea `start-snapshot-id` (esclusiva) e (inclusa). `end-snapshot-id`

```
df_incremental = (spark.read.format("iceberg")
    .option("start-snapshot-id", snapshot_id_start)
    .option("end-snapshot-id", snapshot_id_end)
    .load(f"glue_catalog.{DB_NAME}.{TABLE_NAME}")
)
```

Accesso ai metadati

Iceberg fornisce l'accesso ai propri metadati tramite SQL. È possibile accedere ai metadati per ogni tabella (`<table_name>`) interrogando il namespace. `<table_name>.<metadata_table>` Per un elenco completo delle tabelle di metadati, consulta Ispezione delle tabelle nella documentazione di Iceberg.

L'esempio seguente mostra come accedere alla tabella dei metadati della cronologia di Iceberg, che mostra la cronologia dei commit (modifiche) per una tabella Iceberg.

Utilizzando Spark SQL (con la `%%sql` magia) da un notebook Amazon EMR Studio:

```
Spark.sql(f“””
SELECT * FROM {CATALOG_NAME}.{DB_NAME}.{TABLE_NAME}.history LIMIT 5
""")
```

Utilizzo dell'API: DataFrames

```
spark.read.format("iceberg").load("{CATALOG_NAME}.{DB_NAME}.
{TABLE_NAME}.history").show(5,False)
```

Output di esempio:

Type: Table Pie Scatter Line Area Bar

made_current_at	snapshot_id	parent_id	is_current_ancestor
2023-01-09 02:50:17.547000+00:00	7501027970051178613	6598755163776233735	True
2023-01-12 05:39:29.567000+00:00	7069175828427777019	7501027970051178613	True
2023-01-12 05:39:58.807000+00:00	5173022175861138222	7069175828427777019	True
2023-01-12 05:40:18.499000+00:00	3703414997660223390	5173022175861138222	True
2023-01-12 05:40:41.827000+00:00	3807904412292252460	3703414997660223390	True

Utilizzo delle tabelle Apache Iceberg utilizzando Amazon Athena SQL

Amazon Athena fornisce supporto integrato per Apache Iceberg e non richiede passaggi o configurazioni aggiuntivi. Questa sezione fornisce una panoramica dettagliata delle funzionalità supportate e linee guida di alto livello per l'utilizzo di Athena per interagire con le tabelle Iceberg.

Compatibilità tra versioni e funzionalità

> **Note**
>
> Le sezioni seguenti presuppongono che tu stia utilizzando il motore Athena versione 3.

Supporto alle specifiche della tabella Iceberg

La specifica della tabella Apache Iceberg specifica come dovrebbero comportarsi le tabelle Iceberg. Athena supporta il formato di tabella versione 2, quindi qualsiasi tabella Iceberg creata con la console, la CLI o l'SDK utilizza intrinsecamente quella versione.

Se utilizzi una tabella Iceberg creata con un altro motore, come Apache Spark su Amazon EMR oppure AWS Glue, assicurati di impostare la versione del formato della tabella utilizzando le proprietà della tabella. Come riferimento, consulta la sezione Creazione e scrittura di tabelle Iceberg precedente in questa guida.

Supporto per le funzionalità Iceberg

Puoi usare Athena per leggere e scrivere sulle tabelle Iceberg. Quando si modificano i dati utilizzando le `DELETE FROM` istruzioni`UPDATE`, e`MERGE INTO`,, Athena supporta solo la merge-on-read modalità. Questa proprietà non può essere modificata. Per aggiornare o eliminare i dati con copy-on-write, devi utilizzare altri motori come Apache Spark su Amazon EMR o. AWS Glue La tabella seguente riassume il supporto delle funzionalità Iceberg in Athena.

		Supporto DDL		supporto DML		AWS Lake Formation per motivi di sicurezza (opzionale)
	Formato della tabella	Create table (Crea tabella)	Evoluzione dello schema	Lettura dei dati	Scrittura di dati	Controllo dell'accesso a righe/colonne
Amazon Athena	Versione 2	✓	✓	✓	X C opy-on-write	✓
					✓ M erge-on-read	✓

> **Note**
>
> Athena non supporta le interrogazioni incrementali.

Lavorare con le tabelle Iceberg

Per iniziare rapidamente a usare Iceberg in Athena, consulta la sezione Guida introduttiva alle tabelle Iceberg in Athena SQL all'inizio di questa guida.

La tabella seguente elenca le limitazioni e i consigli.

Scenario	Limitazione	Raccomandazione
Tabella: generazione DDL	Le tabelle Iceberg create con altri motori possono avere proprietà che non	Usa l'istruzione equivalente nel motore che ha creato la

Scenario	Limitazione	Raccomandazione
	sono esposte in Athena. Per queste tabelle, non è possibile generare il DDL.	tabella (ad esempio, l'`SHOW CREATE TABLE` istruzione per Spark).
Prefissi Amazon S3 casuali in oggetti scritti su una tabella Iceberg	Per impostazione predefinita, le tabelle Iceberg create con Athena hanno `write.object-storage.enabled` la proprietà abilitata.	Per disabilitare questo comportamento e ottenere il pieno controllo sulle proprietà della tabella Iceberg, crea una tabella Iceberg con un altro motore come Spark su Amazon EMR o. AWS Glue
Query incrementali	Attualmente non è supportato in Athena.	Per utilizzare query incrementali per abilitare pipeline di inserimento incrementali di dati, usa Spark su Amazon EMR o. AWS Glue

Migrazione di tabelle esistenti su Iceberg

Per migrare l'attuale Athena AWS Glue o le tabelle (note anche come tabelle Hive) nel formato Iceberg, puoi utilizzare la migrazione dei dati sul posto o completa:

- La migrazione sul posto è il processo di generazione dei file di metadati di Iceberg su file di dati esistenti.
- La migrazione completa dei dati crea il livello di metadati Iceberg e riscrive anche i file di dati esistenti dalla tabella originale alla nuova tabella Iceberg.

Le sezioni seguenti forniscono una panoramica delle API disponibili per la migrazione delle tabelle e linee guida per la scelta di una strategia di migrazione. Per ulteriori informazioni su queste due strategie, consulta la sezione Table Migration nella documentazione di Iceberg.

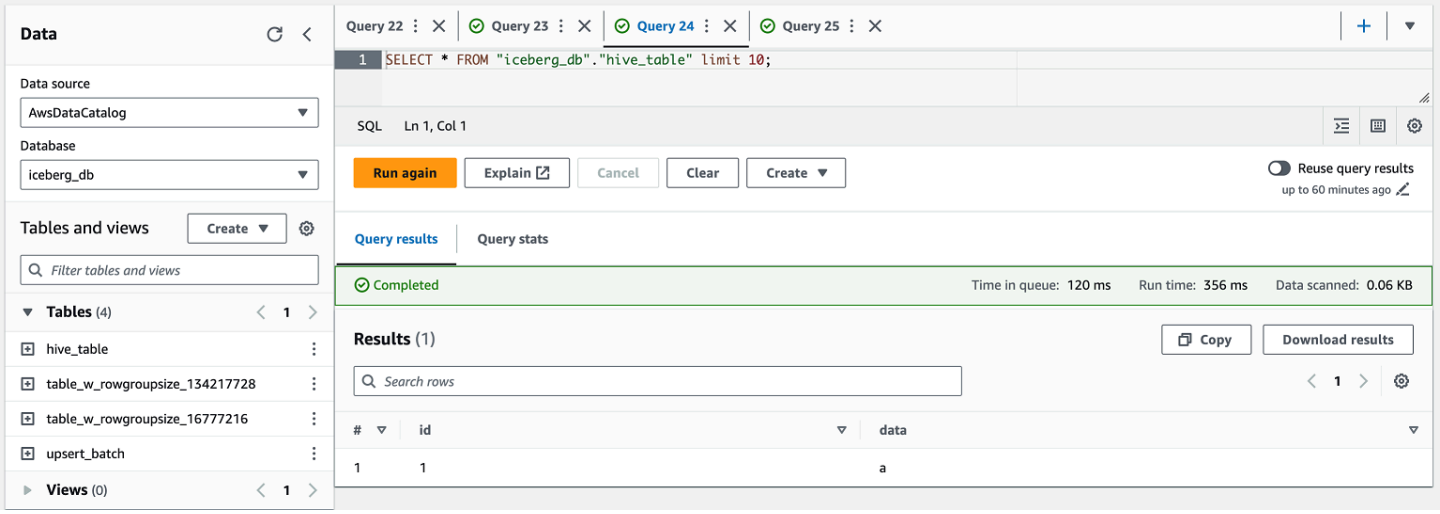
Migrazione sul posto

La migrazione sul posto elimina la necessità di riscrivere tutti i file di dati. I file di metadati Iceberg vengono invece generati e collegati ai file di dati esistenti. Iceberg offre tre opzioni per implementare la migrazione sul posto:

- Utilizzando la `snapshot` procedura, come spiegato nelle sezioni Snapshot Table e Procedura Spark: snapshot nella documentazione di Iceberg.
- Utilizzo della `add_files` procedura, come spiegato nelle sezioni Aggiungi file e Procedura Spark: add_files nella documentazione di Iceberg.
- Utilizzando la `migrate` procedura, come spiegato nelle sezioni Migrate Table e Procedura Spark: Migrate nella documentazione di Iceberg.

Attualmente, la procedura di migrazione non funziona direttamente con il metastore Hive, ma AWS Glue Data Catalog solo con il metastore Hive. Se hai l'esigenza di utilizzare la `migrate` procedura anziché `snapshot` oppure`add_files`, puoi utilizzare un cluster Amazon EMR temporaneo con il metastore Hive (HMS). Questo approccio richiede la versione 1.2 o successiva di Iceberg.

Supponiamo che tu voglia creare la seguente tabella Hive:



Puoi creare questa tabella Hive eseguendo questo codice nella console Athena:

```
CREATE EXTERNAL TABLE 'hive_table'(
  'id' bigint,
  'data' string)
USING parquet
```

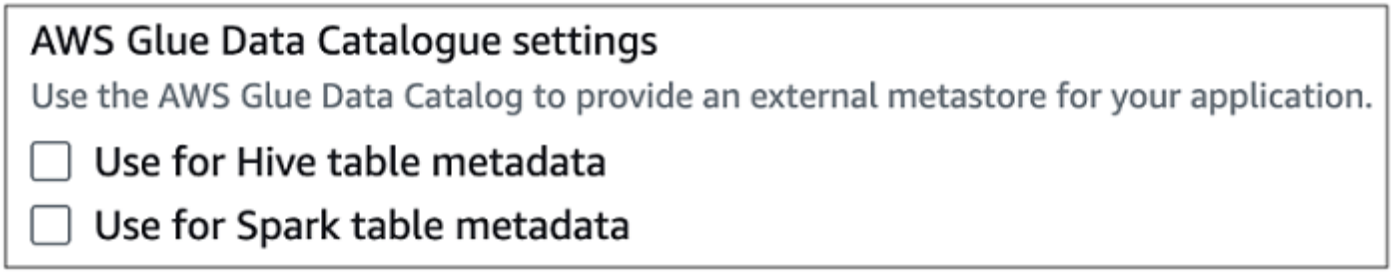
```
LOCATION
  's3://datalake-xxxx/aws_workshop/iceberg_db/hive_table'

INSERT INTO iceberg_db.hive_table VALUES (1, 'a')
```

Se la tua tabella Hive è partizionata, includi l'istruzione partition e aggiungi le partizioni in base ai requisiti di Hive.

```
ALTER TABLE default.placeholder_table_for_migration ADD
    PARTITION (date = '2023-10-10')
```

Fasi:

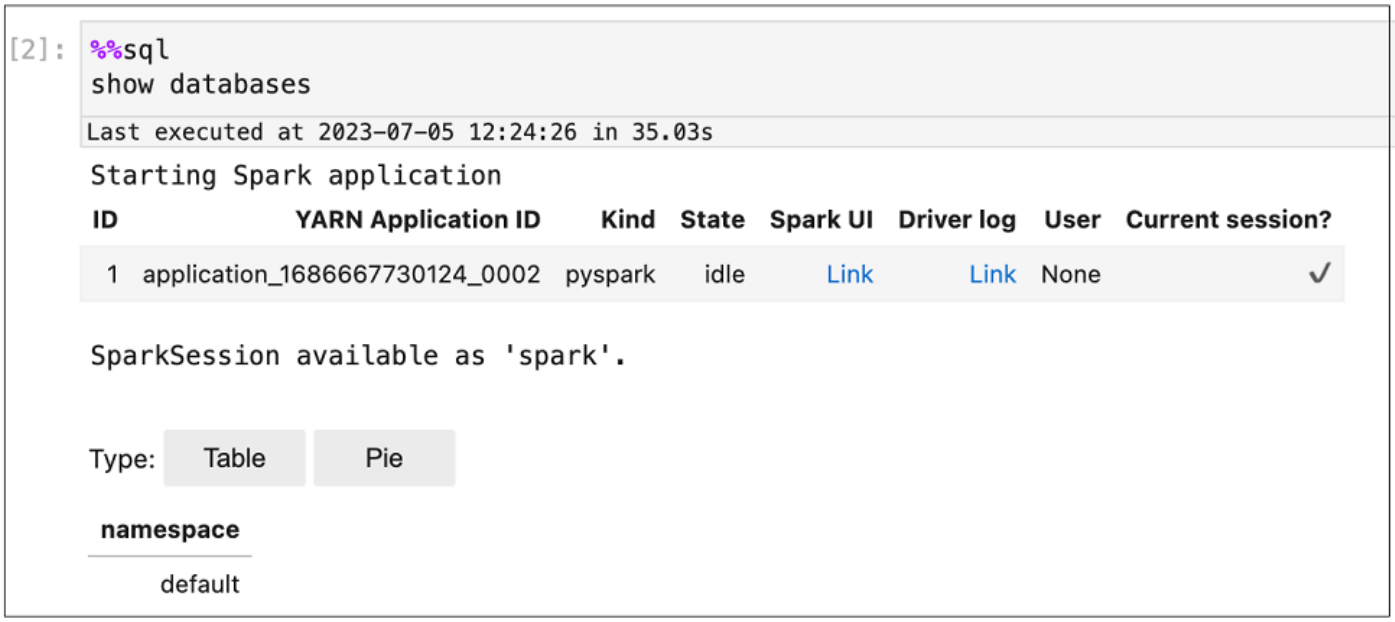
1. Crea un cluster Amazon EMR senza abilitare l' AWS Glue Data Catalog integrazione, ovvero non selezionare le caselle di controllo per i metadati delle tabelle Hive o Spark. Questo perché per questa soluzione alternativa utilizzerai il metastore Hive (HMS) nativo disponibile nel cluster.



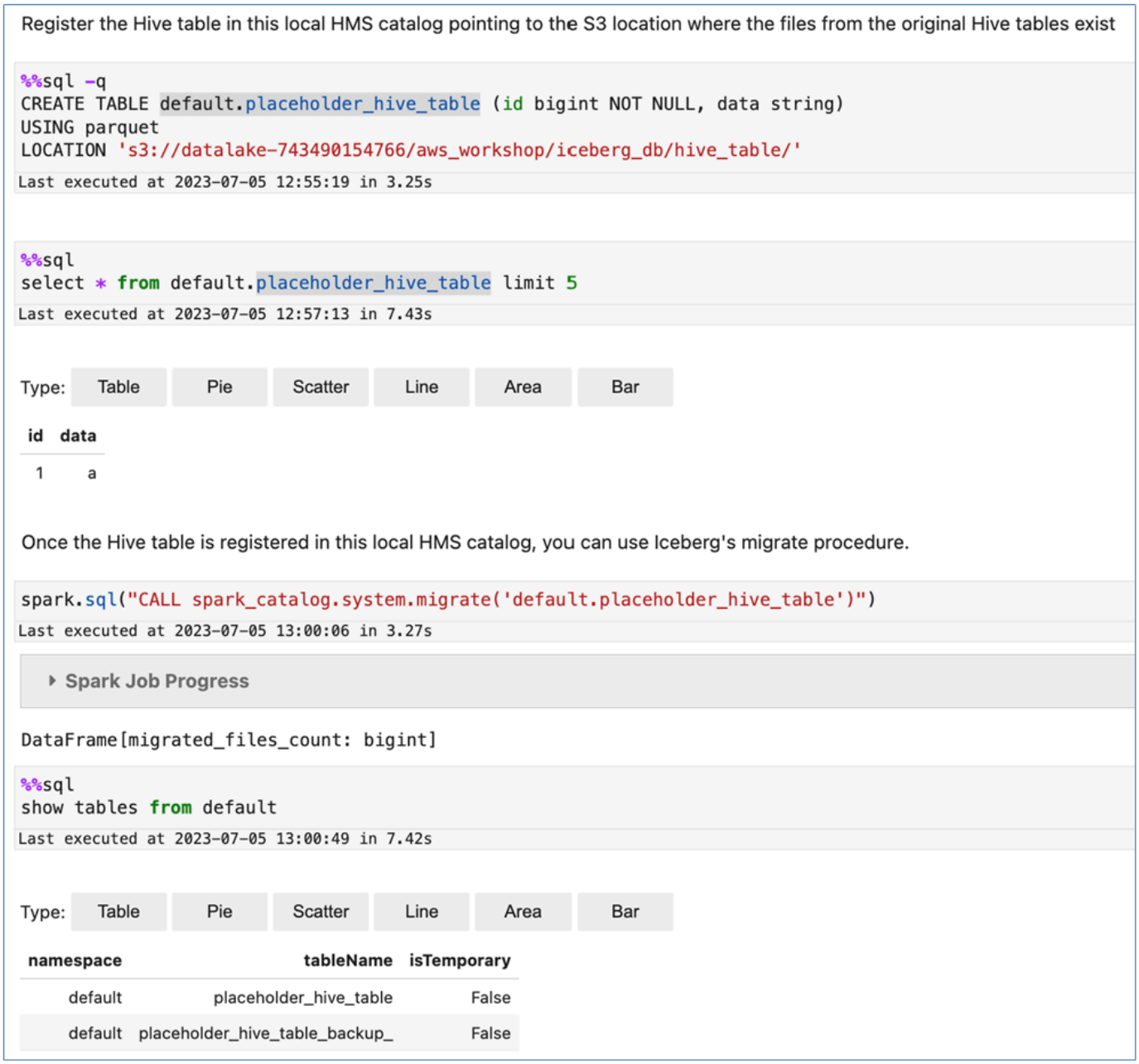
2. Configura la sessione Spark per utilizzare l'implementazione del catalogo Iceberg Hive.

```
"spark.sql.extensions":"org.apache.iceberg.spark.extensions.IcebergSparkSessionExtensions",
"spark.sql.catalog.spark_catalog": "org.apache.iceberg.spark.SparkSessionCatalog",
       "spark.sql.catalog.spark_catalog.type": "hive",
```

3. Verifica che il tuo cluster Amazon EMR non sia connesso a AWS Glue Data Catalog `show databases` `show tables`

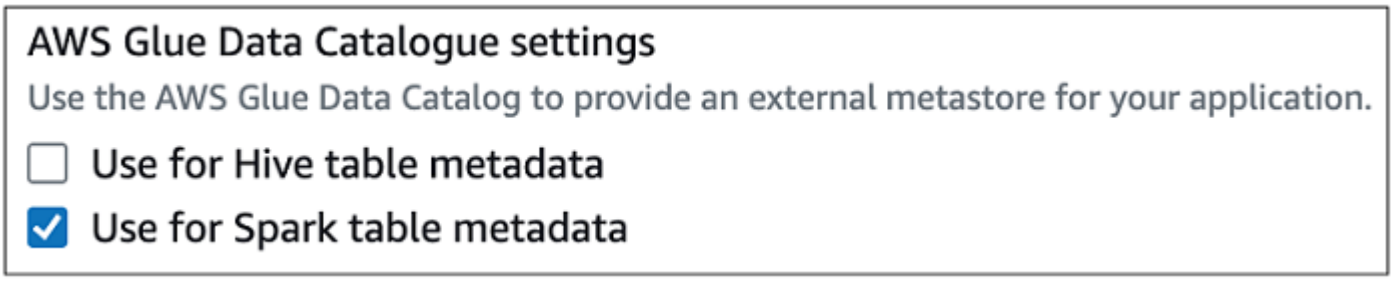


4. Registra la tabella Hive nel metastore Hive del tuo cluster Amazon EMR, quindi utilizza la procedura Iceberg. `migrate`



Questa procedura crea i file di metadati Iceberg nella stessa posizione della tabella Hive.

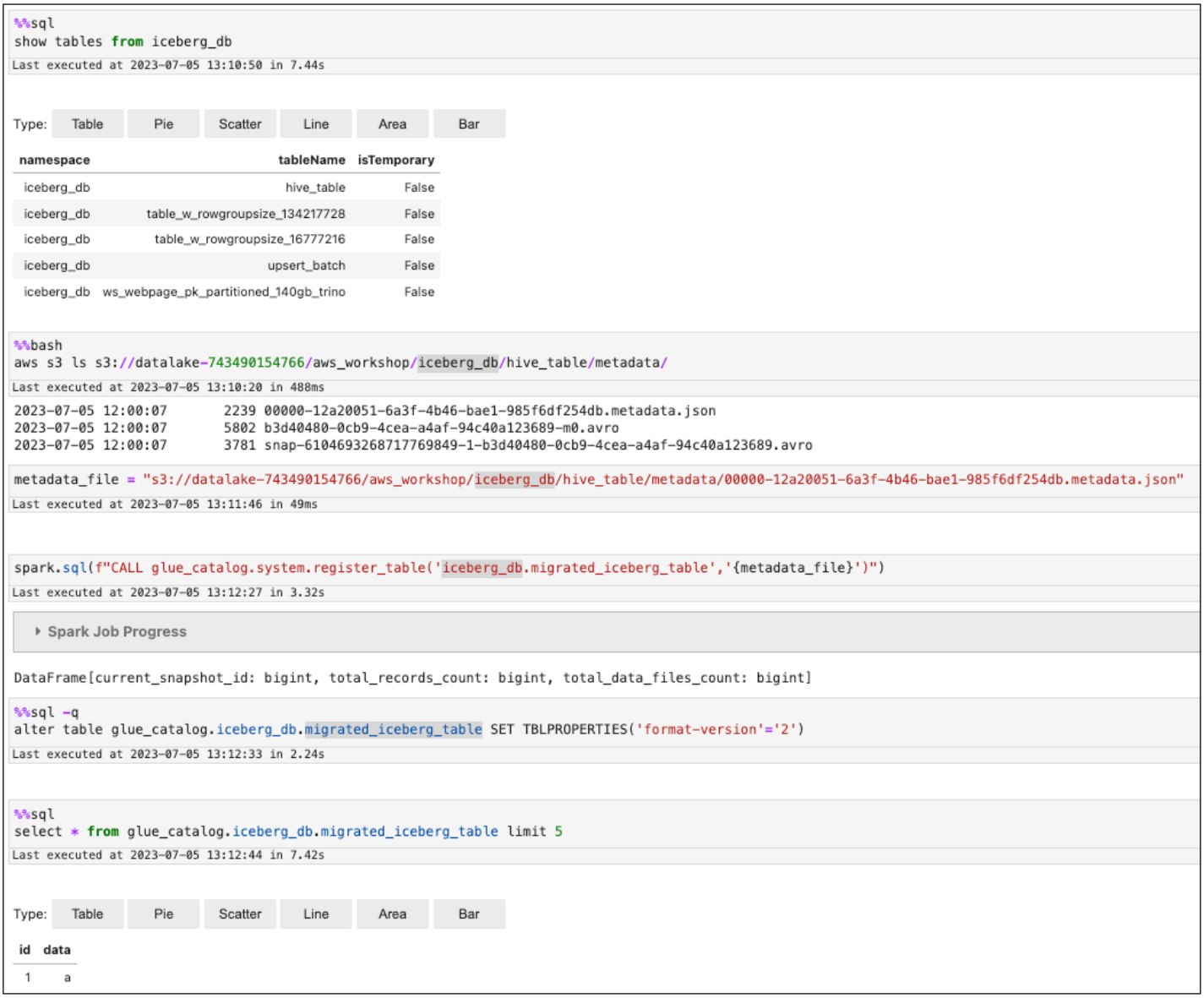
5. Registra la tabella Iceberg migrata in. AWS Glue Data Catalog
6. Torna a un cluster Amazon EMR con l' AWS Glue Data Catalog integrazione abilitata.



7. Usa la seguente configurazione di Iceberg nella sessione Spark.

```
"spark.sql.extensions":"org.apache.iceberg.spark.extensions.IcebergSparkSessionExtensions",
       "spark.sql.catalog.glue_catalog": "org.apache.iceberg.spark.SparkCatalog",
       "spark.sql.catalog.glue_catalog.warehouse": "s3://datalake-xxxx/
aws_workshop",
       "spark.sql.catalog.glue_catalog.catalog-impl":
 "org.apache.iceberg.aws.glue.GlueCatalog",
       "spark.sql.catalog.glue_catalog.io-impl":
 "org.apache.iceberg.aws.s3.S3FileIO",
```

Ora puoi interrogare questa tabella da Amazon EMR o AWS Glue Athena.



Migrazione completa dei dati

La migrazione completa dei dati ricrea i file di dati e i metadati. Questo approccio richiede più tempo e risorse di elaborazione aggiuntive rispetto alla migrazione sul posto. Tuttavia, questa opzione aiuta a migliorare la qualità delle tabelle: è possibile convalidare i dati, apportare modifiche allo schema e alle partizioni, ripristinare i dati e così via. Per implementare la migrazione completa dei dati, utilizza una delle seguenti opzioni:

- Usa l'istruzione `CREATE TABLE ... AS SELECT` (CTAS) in Spark on Amazon EMR o Athena. AWS Glue Puoi impostare le specifiche della partizione e le proprietà della tabella per la nuova tabella Iceberg utilizzando le clausole and. `PARTITIONED BY` `TBLPROPERTIES` È possibile

ottimizzare lo schema e il partizionamento per la nuova tabella in base alle proprie esigenze invece di ereditarli semplicemente dalla tabella di origine.

- Leggi dalla tabella di origine e scrivi i dati come nuova tabella Iceberg utilizzando Spark su Amazon EMR oppure AWS Glue (vedi Creazione di una tabella nella documentazione di Iceberg).

Scelta di una strategia di migrazione

Per scegliere la strategia di migrazione migliore, considera le domande nella tabella seguente.

Domanda	Raccomandazione
Qual è il formato del file di dati (ad esempio, CSV o Apache Parquet)?	• Prendi in considerazione la migrazione sul posto se il formato di file tabellare è Parquet, ORC o Avro. • Per altri formati come CSV, JSON e così via, utilizza la migrazione completa dei dati.
Vuoi aggiornare o consolidare lo schema della tabella?	• Se desideri far evolvere lo schema delle tabelle utilizzando le funzionalità native di Iceberg, prendi in considerazione la migrazione sul posto. Ad esempio, puoi rinominare le colonne dopo la migrazione. (Lo schema può essere modificato nel livello di metadati Iceberg.) • Se desideri eliminare intere colonne dai file di dati, ti consigliamo di utilizzare la migrazione completa dei dati.
La tabella trarrebbe vantaggio dalla modifica della strategia di partizione?	• Se l'approccio di partizionamento di Iceberg soddisfa i tuoi requisiti (ad esempio, i nuovi dati vengono archiviati utilizzando il nuovo layout delle partizioni mentre le partizion i esistenti rimangono invariate), prendi in considerazione la migrazione sul posto.

Domanda	Raccomandazione
	• Se desideri utilizzare partizioni nascoste nella tua tabella, prendi in considerazione la migrazione completa dei dati. Per ulteriori informazioni sulle partizioni nascoste, consulta la sezione Best practice.
La tabella trarrebbe vantaggio dall'aggiunta o dalla modifica della strategia di ordinamento?	• L'aggiunta o la modifica dell'ordinamento dei dati richiede la riscrittura del set di dati. In questo caso, valuta la possibilità di utilizzare la migrazione completa dei dati.
	• Per le tabelle di grandi dimensioni in cui riscrivere tutte le partizioni delle tabelle è proibitivo, prendi in considerazione l'utilizz o della migrazione sul posto ed esegui la compattazione (con l'ordinamento abilitato) per le partizioni a cui si accede più frequente mente.
La tabella contiene molti file di piccole dimensioni?	• L'unione di file di piccole dimensioni in file più grandi richiede la riscrittura del set di dati. In questo caso, valuta la possibilità di utilizzare la migrazione completa dei dati.
	• Per le tabelle di grandi dimensioni in cui riscrivere tutte le partizioni delle tabelle è proibitivo, prendi in considerazione l'utilizz o della migrazione sul posto ed esegui la compattazione (con l'ordinamento abilitato) per le partizioni a cui si accede più frequente mente.

Le migliori pratiche per ottimizzare i carichi di lavoro di Apache Iceberg

Iceberg è un formato di tabella progettato per semplificare la gestione dei data lake e migliorare le prestazioni dei carichi di lavoro. Casi d'uso diversi potrebbero dare priorità a diversi aspetti come costi, prestazioni di lettura, prestazioni di scrittura o conservazione dei dati, quindi Iceberg offre opzioni di configurazione per gestire questi compromessi. Questa sezione fornisce approfondimenti per ottimizzare e perfezionare i carichi di lavoro Iceberg per soddisfare le vostre esigenze.

Argomenti

Best practice generali

Indipendentemente dal caso d'uso, quando utilizzi Apache Iceberg on AWS, ti consigliamo di seguire queste best practice generali.

- Usa il formato Iceberg versione 2.

  Athena utilizza il formato Iceberg versione 2 per impostazione predefinita.

  Quando usi Spark su Amazon EMR AWS Glue o per creare tabelle Iceberg, specifica la versione del formato come descritto nella documentazione di Iceberg.
- Utilizzali AWS Glue Data Catalog come catalogo di dati.

  Athena utilizza il per impostazione AWS Glue Data Catalog predefinita.

  Quando usi Spark su Amazon EMR AWS Glue o lavori con Iceberg, aggiungi la seguente configurazione alla tua sessione Spark per usare AWS Glue Data Catalog. Per ulteriori

informazioni, consulta la sezione Configurazioni Spark per Iceberg in AWS Glue all'inizio di questa guida.

```
"spark.sql.catalog.<your_catalog_name>.catalog-impl":
 "org.apache.iceberg.aws.glue.GlueCatalog"
```

- Usa AWS Glue Data Catalog come gestore di blocchi.

  Athena utilizza di default il gestore di blocchi AWS Glue Data Catalog as per le tabelle Iceberg.

  Quando usi Spark su Amazon EMR AWS Glue o lavori con Iceberg, assicurati di configurare la configurazione della sessione Spark per AWS Glue Data Catalog utilizzarla come gestore dei blocchi. Per ulteriori informazioni, consulta Optimistic Locking nella documentazione di Iceberg.
- Usa la compressione Zstandard (ZSTD).

  Il codec di compressione predefinito di Iceberg è gzip, che può essere modificato utilizzando la proprietà table. `write.<file_type>.compression-codec` Athena utilizza già ZSTD come codec di compressione predefinito per le tabelle Iceberg.

  In generale, consigliamo di utilizzare il codec di compressione ZSTD perché raggiunge un equilibrio tra GZIP e Snappy e offre buone prestazioni di lettura/scrittura senza compromettere il rapporto di compressione. Inoltre, i livelli di compressione possono essere regolati in base alle proprie esigenze. Per ulteriori informazioni, consulta i livelli di compressione ZSTD in Athena nella documentazione di Athena.

  Snappy potrebbe fornire le migliori prestazioni complessive di lettura e scrittura, ma ha un rapporto di compressione inferiore rispetto a GZIP e ZSTD. Se dai priorità alle prestazioni, anche se ciò significa archiviare volumi di dati più grandi in Amazon S3, Snappy potrebbe essere la scelta ottimale.

Ottimizzazione delle prestazioni di lettura

Questa sezione illustra le proprietà delle tabelle che è possibile regolare per ottimizzare le prestazioni di lettura, indipendentemente dal motore.

Partizionamento

Come per le tabelle Hive, Iceberg utilizza le partizioni come livello principale di indicizzazione per evitare la lettura di file di metadati e file di dati non necessari. Le statistiche sulle colonne vengono

prese in considerazione anche come livello secondario di indicizzazione per migliorare ulteriormente la pianificazione delle query, il che porta a una riduzione dei tempi complessivi di esecuzione.

Come partizionare i dati

Per ridurre la quantità di dati analizzati durante l'interrogazione delle tabelle Iceberg, scegli una strategia di partizione bilanciata in linea con i modelli di lettura previsti:

- Identifica le colonne utilizzate di frequente nelle query. Questi sono i candidati ideali per il partizionamento. Ad esempio, se in genere si interrogano i dati di un determinato giorno, un esempio naturale di colonna di partizione sarebbe una colonna di date.
- Scegliete una colonna di partizione a bassa cardinalità per evitare di creare un numero eccessivo di partizioni. Troppe partizioni possono aumentare il numero di file nella tabella, con un impatto negativo sulle prestazioni delle query. Come regola generale, per «troppe partizioni» si può definire uno scenario in cui la dimensione dei dati nella maggior parte delle partizioni è inferiore a 2-5 volte il valore impostato da. `target-file-size-bytes`

> **Note**
>
> Se in genere esegui una query utilizzando filtri su una colonna ad alta cardinalità (ad esempio, una `id` colonna che può avere migliaia di valori), utilizza la funzionalità di partizionamento nascosto di Iceberg con trasformazioni di bucket, come spiegato nella sezione successiva.

Usa il partizionamento nascosto

Se le tue query generalmente filtrano in base a una derivata di una colonna di tabella, utilizza partizioni nascoste invece di creare esplicitamente nuove colonne da utilizzare come partizioni. Per ulteriori informazioni su questa funzionalità, consulta la documentazione di Iceberg.

Ad esempio, in un set di dati con una colonna timestamp (ad esempio,`2023-01-01 09:00:00`), invece di creare una nuova colonna con la data analizzata (ad esempio,`2023-01-01`), utilizzate le trasformazioni delle partizioni per estrarre la parte della data dal timestamp e creare queste partizioni al volo.

I casi d'uso più comuni per il partizionamento nascosto sono:

- Partizionamento in base alla data o all'ora, quando i dati hanno una colonna con timestamp. Iceberg offre più trasformazioni per estrarre le parti relative alla data o all'ora di un timestamp.
- Partizionamento su una funzione hash di una colonna, quando la colonna di partizionamento ha una cardinalità elevata e comporterebbe troppe partizioni. La trasformazione bucket di Iceberg raggruppa più valori di partizione in un numero inferiore di partizioni nascoste (bucket) utilizzando funzioni hash sulla colonna di partizionamento.

Vedi le trasformazioni delle partizioni nella documentazione di Iceberg per una panoramica di tutte le trasformazioni di partizione disponibili.

Le colonne utilizzate per il partizionamento nascosto possono diventare parte dei predicati di query attraverso l'uso di normali funzioni SQL come e. `year()` `month()` I predicati possono anche essere combinati con operatori come e. `BETWEEN` `AND`

> **Note**
>
> Iceberg non può eseguire l'eliminazione delle partizioni per funzioni che producono un tipo di dati diverso, ad esempio. `substring(event_time, 1, 10) = '2022-01-01'`

Usa l'evoluzione delle partizioni

Usa l'evoluzione delle partizioni di Iceberg quando la strategia di partizione esistente non è ottimale. Ad esempio, se scegli partizioni orarie che si rivelano troppo piccole (solo pochi megabyte ciascuna), valuta la possibilità di passare a partizioni giornaliere o mensili.

È possibile utilizzare questo approccio quando inizialmente non è chiara la migliore strategia di partizione per una tabella e si desidera perfezionare la strategia di partizionamento man mano che si ottengono maggiori informazioni. Un altro uso efficace dell'evoluzione delle partizioni è quando i volumi di dati cambiano e l'attuale strategia di partizionamento diventa meno efficace nel tempo.

Per istruzioni su come far evolvere le partizioni, consulta le estensioni SQL ALTER TABLE nella documentazione di Iceberg.

Ottimizzazione delle dimensioni dei file

L'ottimizzazione delle prestazioni delle query implica la riduzione al minimo del numero di file di piccole dimensioni nelle tabelle. Per ottenere buone prestazioni di interrogazione, in genere consigliamo di conservare file Parquet e ORC di dimensioni superiori a 100 MB.

Le dimensioni dei file influiscono anche sulla pianificazione delle query per le tabelle Iceberg. All'aumentare del numero di file in una tabella, aumenta anche la dimensione dei file di metadati. File di metadati più grandi possono rallentare la pianificazione delle query. Pertanto, quando le dimensioni della tabella aumentano, aumentate le dimensioni del file per ridurre l'espansione esponenziale dei metadati.

Utilizza le seguenti best practice per creare file di dimensioni adeguate nelle tabelle Iceberg.

Imposta la dimensione del file di destinazione e del gruppo di righe

Iceberg offre i seguenti parametri di configurazione chiave per ottimizzare il layout del file di dati. Si consiglia di utilizzare questi parametri per impostare la dimensione del file di destinazione e il gruppo di righe o la dimensione del strike.

Parameter	Valore predefinito	Commento
`write.target-file-size-bytes`	512 MB	Questo parametro specifica la dimensione massima del file che Iceberg creerà. Tuttavia, alcuni file potrebbero essere scritti con una dimensione inferiore a questo limite.
`write.parquet.row-group-size-bytes`	128 MB	Sia Parquet che ORC archiviano i dati in blocchi in modo che i motori possano evitare di leggere l'intero file per alcune operazioni.
`write.orc.stripe-size-bytes`	64 MB	
`write.distribution-mode`	Nessuno, per Iceberg versione 1.1 e precedenti Hash, a partire dalla versione 1.2 di Iceberg	Iceberg richiede a Spark di ordinare i dati tra le sue attività prima di scriverli sullo storage.

- In base alla dimensione prevista della tabella, segui queste linee guida generali:

  - Tabelle di piccole dimensioni (fino a pochi gigabyte): riducono la dimensione del file di destinazione a 128 MB. Riduci anche la dimensione del gruppo di righe o della banda (ad esempio, a 8 o 16 MB).
  - Tabelle di medie e grandi dimensioni (da pochi gigabyte a centinaia di gigabyte): i valori predefiniti sono un buon punto di partenza per queste tabelle. Se le tue query sono molto selettive, modifica la dimensione del gruppo di righe o della banda (ad esempio, a 16 MB).
  - Tabelle molto grandi (centinaia di gigabyte o terabyte): aumenta la dimensione del file di destinazione a 1024 MB o più e valuta la possibilità di aumentare la dimensione del gruppo di righe o dello stripe se le tue query in genere richiamano set di dati di grandi dimensioni.

- Per garantire che le applicazioni Spark che scrivono su tabelle Iceberg creino file di dimensioni adeguate, imposta la proprietà su o. `write.distribution-mode` `hash` `range` Per una spiegazione dettagliata della differenza tra queste modalità, consulta Writing Distribution Modes nella documentazione di Iceberg.

Queste sono linee guida generali. Ti consigliamo di eseguire dei test per identificare i valori più adatti alle tue tabelle e ai tuoi carichi di lavoro specifici.

Esegui una compattazione regolare

Le configurazioni nella tabella precedente impostano la dimensione massima dei file che le attività di scrittura possono creare, ma non garantiscono che i file abbiano tale dimensione. Per garantire le dimensioni corrette dei file, esegui regolarmente la compattazione per combinare file di piccole dimensioni in file più grandi. Per una guida dettagliata sulla compattazione in esecuzione, consulta Iceberg compaction più avanti in questa guida.

Ottimizza le statistiche delle colonne

Iceberg utilizza le statistiche sulle colonne per eseguire l'eliminazione dei file, che migliora le prestazioni delle query riducendo la quantità di dati analizzati dalle query. Per trarre vantaggio dalle statistiche sulle colonne, assicurati che Iceberg raccolga le statistiche per tutte le colonne che vengono utilizzate frequentemente nei filtri di query.

Per impostazione predefinita, Iceberg raccoglie le statistiche solo per le prime 100 colonne di ogni tabella, come definito dalla proprietà table. `write.metadata.metrics.max-inferred-column-defaults` Se la tua tabella ha più di 100 colonne e le tue query fanno spesso riferimento a colonne al di fuori delle prime 100 colonne (ad esempio, potresti avere delle query che filtrano sulla

colonna 132), assicurati che Iceberg raccolga statistiche su quelle colonne. Esistono due opzioni per raggiungere questo obiettivo:

- Quando crei la tabella Iceberg, riordina le colonne in modo che le colonne necessarie per le query rientrino nell'intervallo di colonne impostato da `write.metadata.metrics.max-inferred-column-defaults` (l'impostazione predefinita è 100).

  Nota: se non hai bisogno di statistiche su 100 colonne, puoi regolare la `write.metadata.metrics.max-inferred-column-defaults` configurazione in base al valore desiderato (ad esempio 20) e riordinare le colonne in modo che le colonne necessarie per leggere e scrivere le query rientrino nelle prime 20 colonne sul lato sinistro del set di dati.
- Se utilizzi solo poche colonne nei filtri di query, puoi disabilitare la proprietà complessiva per la raccolta delle metriche e scegliere selettivamente singole colonne per le quali raccogliere statistiche, come mostrato in questo esempio:

```
.tableProperty("write.metadata.metrics.default", "none")
.tableProperty("write.metadata.metrics.column.my_col_a", "full")
.tableProperty("write.metadata.metrics.column.my_col_b", "full")
```

Nota: le statistiche sulle colonne sono più efficaci quando i dati vengono ordinati in base a tali colonne. Per ulteriori informazioni, consulta la sezione Impostare l'ordinamento più avanti in questa guida.

Scegli la giusta strategia di aggiornamento

Utilizzate una copy-on-write strategia per ottimizzare le prestazioni di lettura, quando le operazioni di scrittura più lente sono accettabili per il vostro caso d'uso. Questa è la strategia predefinita utilizzata da Iceberg.

C opy-on-write offre prestazioni di lettura migliori, poiché i file vengono scritti direttamente nell'archivio in modo ottimizzato per la lettura. Tuttavia, rispetto a merge-on-read, ogni operazione di scrittura richiede più tempo e consuma più risorse di elaborazione. Ciò rappresenta un classico compromesso tra latenza di lettura e scrittura. In genere, copy-on-write è ideale per i casi d'uso in cui la maggior parte degli aggiornamenti è collocata nelle stesse partizioni di tabella (ad esempio, per carichi batch giornalieri).

opy-on-write Le configurazioni C (`write.update.mode`,`write.delete.mode`, e`write.merge.mode`) possono essere impostate a livello di tabella o indipendentemente dal lato dell'applicazione.

Usa la compressione ZSTD

È possibile modificare il codec di compressione utilizzato da Iceberg utilizzando la proprietà table. `write.<file_type>.compression-codec` Si consiglia di utilizzare il codec di compressione ZSTD per migliorare le prestazioni complessive sulle tabelle.

Per impostazione predefinita, le versioni 1.3 e precedenti di Iceberg utilizzano la compressione GZIP, che offre prestazioni di lettura/scrittura più lente rispetto a ZSTD.

Nota: alcuni motori potrebbero utilizzare valori predefiniti diversi. Questo è il caso delle tabelle Iceberg create con Athena o Amazon EMR versione 7.x.

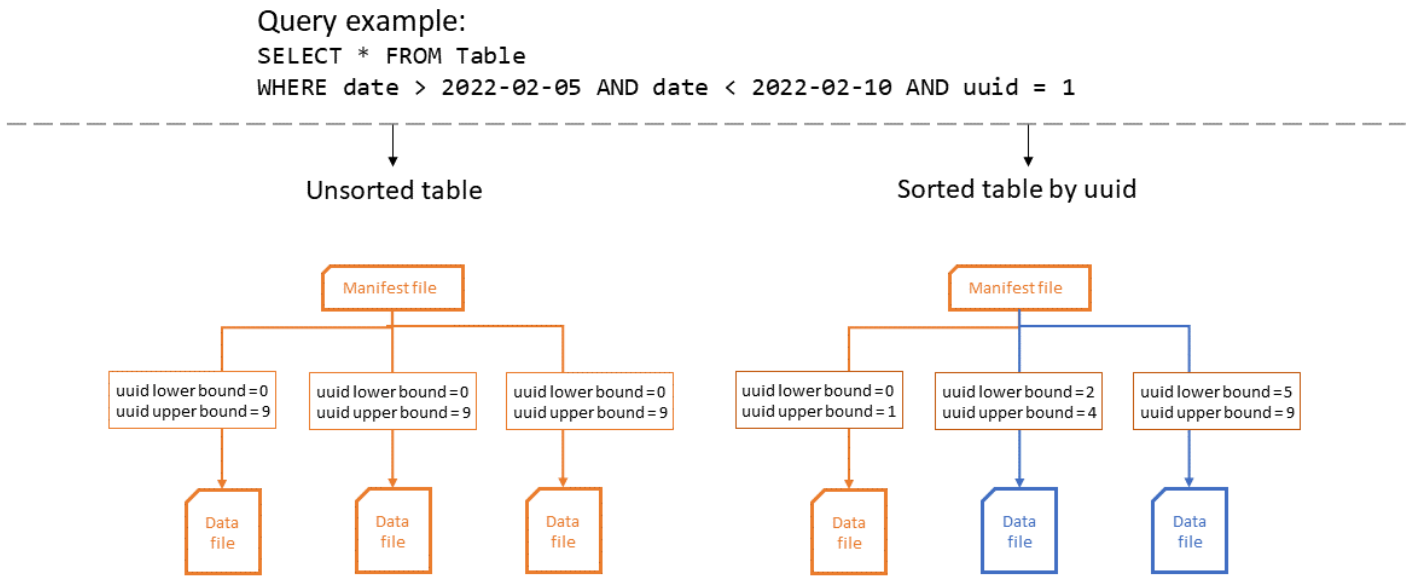
Imposta l'ordinamento

Per migliorare le prestazioni di lettura sulle tabelle Iceberg, ti consigliamo di ordinare la tabella in base a una o più colonne che vengono utilizzate frequentemente nei filtri di query. L'ordinamento, combinato con le statistiche sulle colonne di Iceberg, può rendere l'eliminazione dei file notevolmente più efficiente, il che si traduce in operazioni di lettura più rapide. L'ordinamento riduce anche il numero di richieste Amazon S3 per le query che utilizzano le colonne di ordinamento nei filtri di query.

Puoi impostare un ordinamento gerarchico a livello di tabella eseguendo un'istruzione DDL (Data Definition Language) con Spark. Per le opzioni disponibili, consulta la documentazione di Iceberg. Dopo aver impostato l'ordinamento, gli autori applicheranno questo ordinamento alle successive operazioni di scrittura dei dati nella tabella Iceberg.

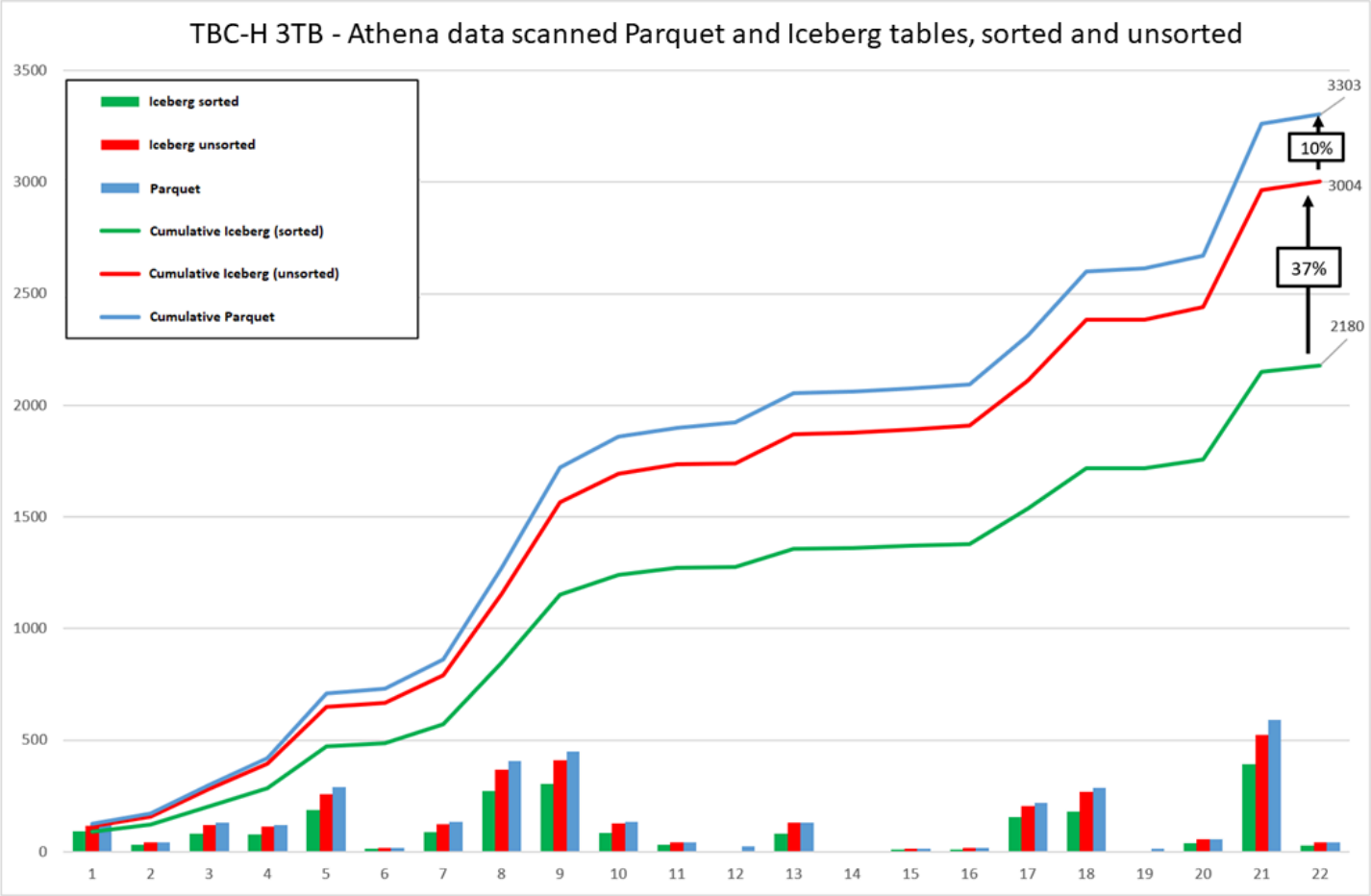
Ad esempio, nelle tabelle partizionate per date (`yyyy-mm-dd`) in base alle quali filtra la maggior parte delle query`uuid`, puoi usare l'opzione DDL `Write Distributed By Partition Locally Ordered` per assicurarti che Spark scriva file con intervalli non sovrapposti.

Il diagramma seguente illustra come l'efficienza delle statistiche delle colonne migliora quando le tabelle vengono ordinate. Nell'esempio, la tabella ordinata deve aprire solo un singolo file e sfruttare al massimo la partizione e il file di Iceberg. Nella tabella non ordinata, tutto `uuid` può potenzialmente esistere in qualsiasi file di dati, quindi la query deve aprire tutti i file di dati.

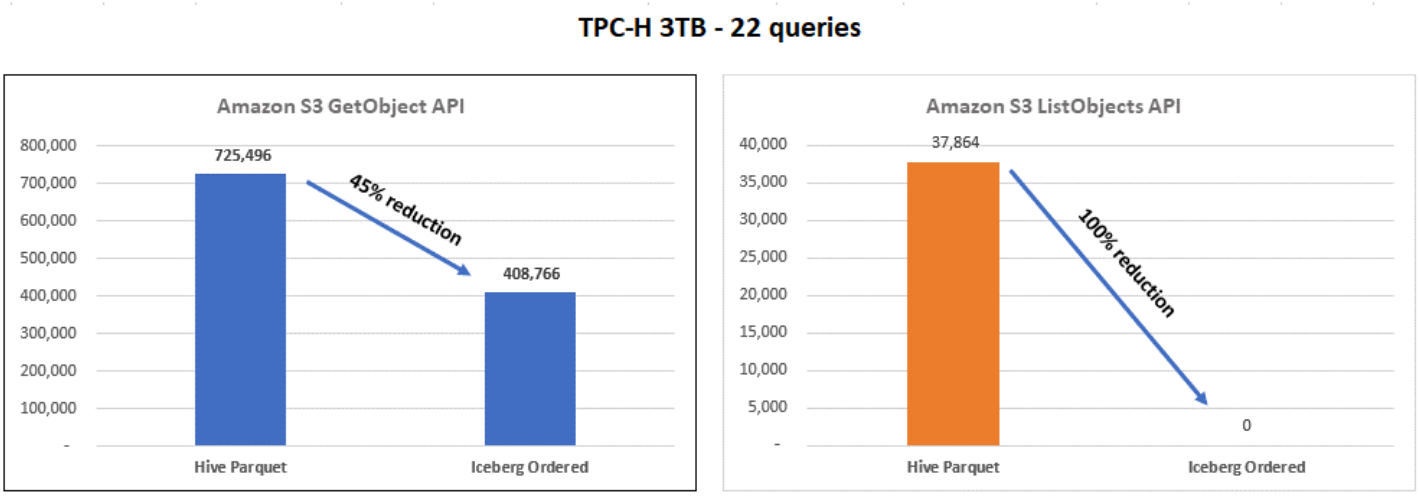


La modifica dell'ordinamento non influisce sui file di dati esistenti. Puoi usare la compattazione Iceberg per applicare l'ordinamento su questi file.

L'utilizzo delle tabelle ordinate Iceberg può ridurre i costi del carico di lavoro, come illustrato nel grafico seguente.



Questi grafici riassumono i risultati dell'esecuzione del benchmark TPC-H per le tabelle Hive (Parquet) rispetto alle tabelle ordinate Iceberg. Tuttavia, i risultati potrebbero essere diversi per altri set di dati o carichi di lavoro.



Ottimizzazione delle prestazioni di scrittura

Questa sezione illustra le proprietà delle tabelle che è possibile regolare per ottimizzare le prestazioni di scrittura sulle tabelle Iceberg, indipendentemente dal motore.

Imposta la modalità di distribuzione della tabella

Iceberg offre diverse modalità di distribuzione di scrittura che definiscono come i dati di scrittura vengono distribuiti tra le attività Spark. Per una panoramica delle modalità disponibili, consulta Writing Distribution Modes nella documentazione di Iceberg.

Per i casi d'uso che danno priorità alla velocità di scrittura, specialmente nei carichi di lavoro di streaming, imposta su. `write.distribution-mode` `none` Ciò garantisce che Iceberg non richieda un ulteriore shuffling di Spark e che i dati vengano scritti non appena diventano disponibili nelle attività Spark. Questa modalità è particolarmente adatta per le applicazioni Spark Structured Streaming.

> **Note**
>
> L'impostazione della modalità di distribuzione di scrittura `none` tende a produrre numerosi file di piccole dimensioni, il che riduce le prestazioni di lettura. Si consiglia una compattazione regolare per consolidare questi file di piccole dimensioni in file di dimensioni adeguate per le prestazioni delle query.

Scegliete la giusta strategia di aggiornamento

Utilizzate una merge-on-read strategia per ottimizzare le prestazioni di scrittura, quando le operazioni di lettura più lente sui dati più recenti sono accettabili per il vostro caso d'uso.

Quando si utilizza merge-on-read, Iceberg scrive gli aggiornamenti e le eliminazioni nell'archivio come piccoli file separati. Quando la tabella viene letta, il lettore deve unire queste modifiche ai file di base per restituire la visualizzazione più recente dei dati. Ciò comporta una riduzione delle prestazioni per le operazioni di lettura, ma accelera la scrittura di aggiornamenti ed eliminazioni. In genere, merge-on-read è ideale per lo streaming di carichi di lavoro con aggiornamenti o lavori con pochi aggiornamenti distribuiti su molte partizioni di tabella.

È possibile impostare merge-on-read le configurazioni (`write.update.modewrite.delete.mode`, e`write.merge.mode`) a livello di tabella o indipendentemente dal lato dell'applicazione.

L'utilizzo merge-on-read richiede una compattazione regolare per evitare che le prestazioni di lettura peggiorino nel tempo. La compattazione riconcilia gli aggiornamenti e le eliminazioni con i file di dati esistenti per creare un nuovo set di file di dati, eliminando in tal modo la riduzione delle prestazioni dovuta alla lettura. Per impostazione predefinita, la compattazione di Iceberg non unisce i file eliminati a meno che non si modifichi l'impostazione predefinita della `delete-file-threshold` proprietà con un valore inferiore (vedi la documentazione di Iceberg). Per saperne di più sulla compattazione, consulta la sezione Compattazione di Iceberg più avanti in questa guida.

Scegli il formato di file giusto

Iceberg supporta la scrittura di dati nei formati Parquet, ORC e Avro. Parquet è il formato predefinito. Parquet e ORC sono formati colonnari che offrono prestazioni di lettura superiori ma sono generalmente più lenti da scrivere. Questo rappresenta il tipico compromesso tra prestazioni di lettura e scrittura.

Se la velocità di scrittura è importante per il tuo caso d'uso, ad esempio nei carichi di lavoro in streaming, prendi in considerazione la scrittura in formato Avro impostando `Avro` nelle `write-format` opzioni dello scrittore. Poiché Avro è un formato basato su righe, offre tempi di scrittura più rapidi a scapito di prestazioni di lettura più lente.

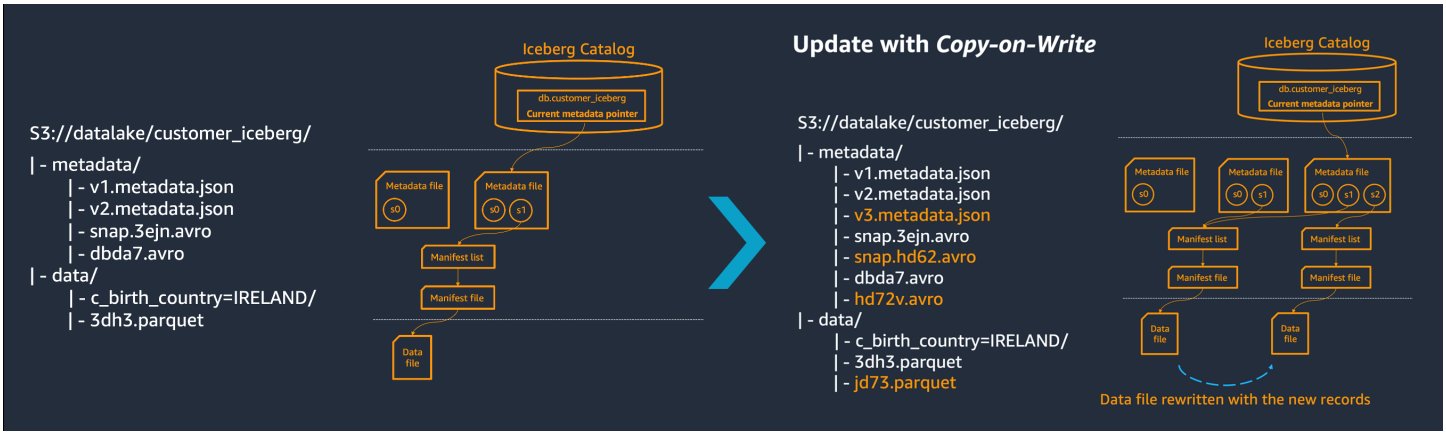
Per migliorare le prestazioni di lettura, esegui una compattazione regolare per unire e trasformare piccoli file Avro in file Parquet più grandi. L'esito del processo di compattazione è determinato dall'impostazione della tabella. `write.format.default` Il formato predefinito per Iceberg è Parquet, quindi se si scrive in Avro e poi si esegue la compattazione, Iceberg trasformerà i file Avro in file Parquet. Ecco un esempio:

```
spark.sql(f"""
    CREATE TABLE IF NOT EXISTS glue_catalog.{DB_NAME}.{TABLE_NAME} (
        Col_1 float,
        <<<…other columns…>>
        ts timestamp)
    USING iceberg
    PARTITIONED BY (days(ts))
    OPTIONS (
      'format-version'='2',
      write.format.default'=parquet)
```

```
""")

query = df \
    .writeStream \
    .format("iceberg") \
    .option("write-format", "avro") \
    .outputMode("append") \
    .trigger(processingTime='60 seconds') \
    .option("path", f"glue_catalog.{DB_NAME}.{TABLE_NAME}") \
    .option("checkpointLocation", f"s3://{BUCKET_NAME}/checkpoints/iceberg/")

    .start()
```

Ottimizzazione dello storage

L'aggiornamento o l'eliminazione dei dati in una tabella Iceberg aumenta il numero di copie dei dati, come illustrato nel diagramma seguente. Lo stesso vale per l'esecuzione della compattazione: aumenta il numero di copie dei dati in Amazon S3. Questo perché Iceberg considera immutabili i file alla base di tutte le tabelle.



Segui le best practice riportate in questa sezione per gestire i costi di archiviazione.

Abilita S3 Intelligent-Tiering

Usa la classe di storage Amazon S3 Intelligent-Tiering per spostare automaticamente i dati al livello di accesso più conveniente quando i modelli di accesso cambiano. Questa opzione non ha alcun sovraccarico operativo o impatto sulle prestazioni.

Nota: non utilizzare i livelli opzionali (come Archive Access e Deep Archive Access) nelle tabelle S3 Intelligent-Tiering with Iceberg. Per archiviare i dati, consulta le linee guida nella sezione successiva.

Puoi anche utilizzare le regole del ciclo di vita di Amazon S3 per impostare regole personalizzate per lo spostamento di oggetti in un'altra classe di storage Amazon S3, come S3 Standard-IA o S3 One Zone-IA (consulta Transizioni supportate e vincoli correlati nella documentazione di Amazon S3).

Archivia o elimina istantanee storiche

Per ogni transazione confermata (inserimento, aggiornamento, unione, compattazione) su una tabella Iceberg, viene creata una nuova versione o istantanea della tabella. Nel tempo, il numero di versioni e il numero di file di metadati in Amazon S3 si accumulano.

La conservazione delle istantanee di una tabella è necessaria per funzionalità come l'isolamento delle istantanee, il rollback delle tabelle e le query sui viaggi nel tempo. Tuttavia, i costi di storage aumentano con il numero di versioni conservate.

La tabella seguente descrive i modelli di progettazione che è possibile implementare per gestire i costi in base ai requisiti di conservazione dei dati.

Modello di progettazione	Soluzione	Casi d'uso
Eliminare vecchie istantanee	• Utilizzate l'istruzione VACUUM in Athena per rimuovere le vecchie istantanee. Questa operazione non comporta alcun costo di calcolo. • In alternativa, puoi usare Spark su Amazon EMR AWS Glue o rimuovere le istantanee. Per ulteriori informazioni, consulta expire_snapshots nella documentazione di Iceberg.	Questo approccio elimina le istantanee che non sono più necessarie per ridurre i costi di storage. È possibile configurare quante istantane e conservare o per quanto tempo, in base ai requisiti di conservazione dei dati. Questa opzione esegue un'eliminazione definitiva delle istantanee. Non è possibile tornare indietro o viaggiare nel tempo verso istantanee scadute.
Imposta politiche di conservazione per istantanee specifiche	1. Usa i tag per contrassegnare istantanee specifiche e definire una politica di conservazione in Iceberg.	Questo modello è utile per la conformità ai requisiti aziendali o legali che richiedono di mostrare lo stato di una tabella

Modello di progettazione	Soluzione	Casi d'uso
	Per ulteriori informazioni, consulta Tag storici nella documentazione di Iceberg. Ad esempio, puoi conservare uno snapshot al mese per un anno utilizzando la seguente istruzione SQL in Spark su Amazon EMR: `ALTER TABLE glue_catalog.db.table CREATE TAG 'EOM-01' AS OF VERSION 30 RETAIN 365 DAYS` 2. Usa Spark su Amazon EMR AWS Glue o per rimuovere gli snapshot intermedi rimanenti senza tag.	in un determinato momento del passato. Inserendo politiche di conservazione su istantanee con tag specifici , è possibile rimuovere altre istantanee (senza tag) che sono state create. In questo modo, è possibile soddisfare i requisiti di conservazione dei dati senza conservare ogni singola istantanea creata.

Modello di progettazione	Soluzione	Casi d'uso
Archivia vecchie istantanee	1. Usa i tag Amazon S3 per contrassegnare gli oggetti con Spark. (I tag Amazon S3 sono diversi dai tag Iceberg; per ulteriori informazioni, consulta la documentazione di Iceberg.) Per esempio: `spark.sql.catalog.my_catalog.s3.delete-enabled=false and \ spark.sql.catalog.my_catalog.s3.delete.tags.my_key=to_archive` 2. Usa Spark su Amazon EMR AWS Glue o per rimuovere istantanee. Quando si utilizzano le impostazioni dell'esempio, questa procedura etichetta gli oggetti e li scollega dai metadati della tabella Iceberg anziché eliminarli da Amazon S3. 3. Utilizza le regole del ciclo di vita di S3 per trasferire gli oggetti etichettati come `to_archive` verso una delle classi di storage S3 Glacier. 4. Per interrogare i dati archiviati:	Questo modello consente di conservare tutte le versioni e le istantanee delle tabelle a un costo inferiore. Non è possibile viaggiare nel tempo o tornare alle istantanee archiviate senza prima ripristinare tali versioni come nuove tabelle. Ciò è generalmente accettabile per scopi di controllo. È possibile combinare questo approccio con il modello di progettazione precedent e, impostando politiche di conservazione per istantanee specifiche.

Modello di progettazione	Soluzione	Casi d'uso
	• Ripristina gli oggetti archiviati. • Usa la procedura register_table in Iceberg per registrare l'istantanea come tabella nel catalogo. Per istruzioni dettagliate, consulta il post AWS sul blog Migliorare l'efficienza operativa delle tabelle Apache Iceberg create sui data lake Amazon S3.	

Eliminare i file orfani

In alcune situazioni, le applicazioni Iceberg possono fallire prima di eseguire le transazioni. Questo lascia i file di dati in Amazon S3. Poiché non è stato eseguito alcun commit, questi file non verranno associati a nessuna tabella, quindi potrebbe essere necessario pulirli in modo asincrono.

Per gestire queste eliminazioni, puoi utilizzare l'istruzione VACUUM in Amazon Athena. Questa istruzione rimuove le istantanee ed elimina anche i file orfani. Questo è molto conveniente, perché Athena non addebita il costo di calcolo di questa operazione. Inoltre, non è necessario pianificare alcuna operazione aggiuntiva quando si utilizza l'istruzione. `VACUUM`

In alternativa, puoi usare Spark su Amazon EMR AWS Glue o eseguire `remove_orphan_files` la procedura. Questa operazione ha un costo di calcolo e deve essere pianificata in modo indipendente. Per ulteriori informazioni, consulta la documentazione di Iceberg.

Manutenzione delle tabelle mediante compattazione

Iceberg include funzionalità che consentono di eseguire operazioni di manutenzione delle tabelle dopo avervi scritto i dati. Alcune operazioni di manutenzione si concentrano sulla semplificazione dei file di metadati, mentre altre migliorano il modo in cui i dati sono raggruppati nei file in modo che i motori di query possano localizzare in modo efficiente le informazioni necessarie per rispondere alle richieste degli utenti. Questa sezione si concentra sulle ottimizzazioni relative alla compattazione.

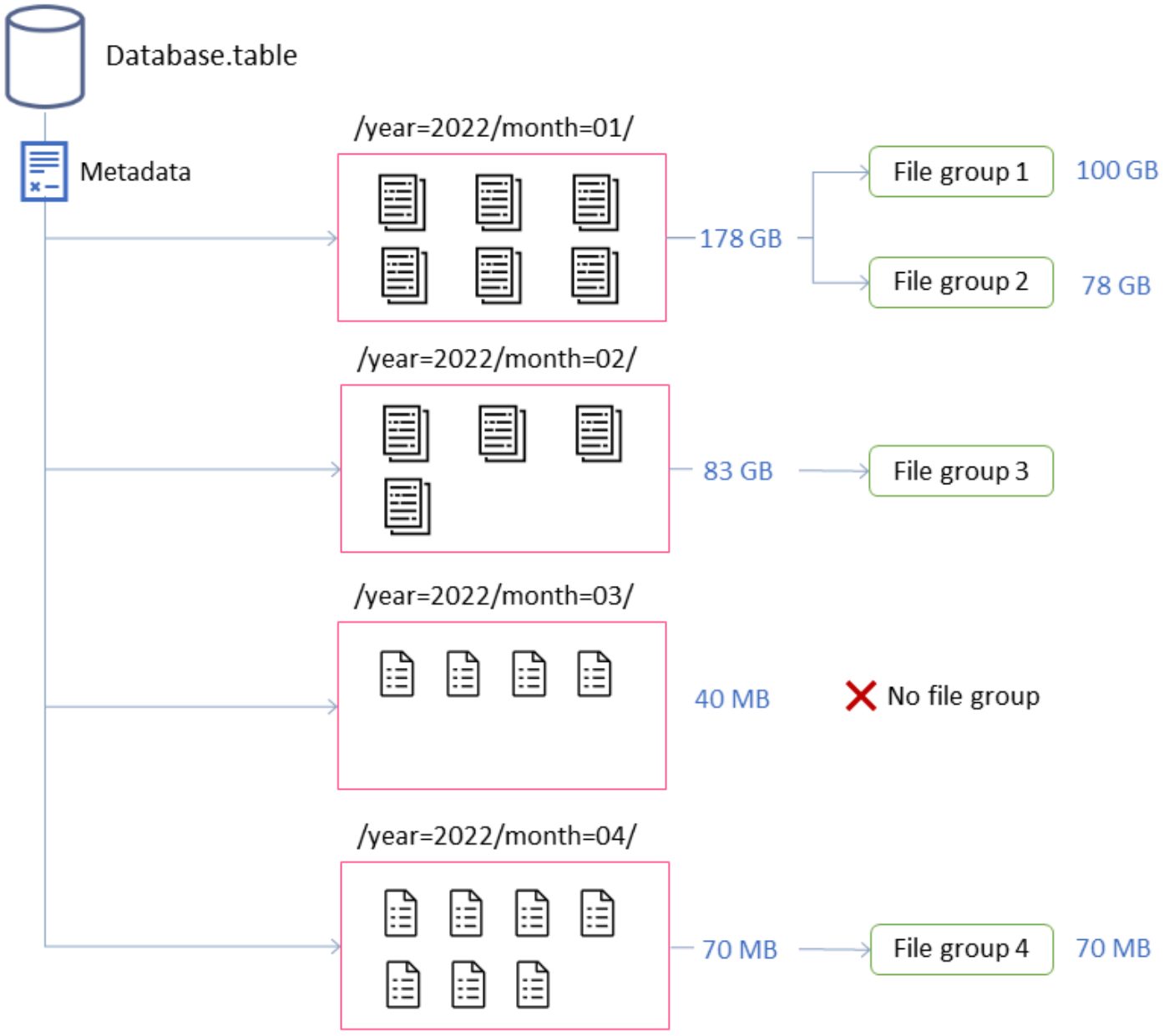
Compattazione degli iceberg

In Iceberg, è possibile utilizzare la compattazione per eseguire quattro attività:

- Combinazione di file di piccole dimensioni in file più grandi che generalmente superano i 100 MB. Questa tecnica è nota come imballaggio in contenitori.
- Unione di file eliminati con file di dati. I file di eliminazione vengono generati da aggiornamenti o eliminazioni che utilizzano questo approccio. merge-on-read
- (Ri) ordinamento dei dati in base ai modelli di interrogazione. I dati possono essere scritti senza alcun criterio di ordinamento o con un ordinamento adatto per scritture e aggiornamenti.
- Raggruppamento dei dati utilizzando curve di riempimento degli spazi per ottimizzare modelli di query distinti, in particolare l'ordinamento con ordine z.

Su AWS, puoi eseguire operazioni di compattazione e manutenzione delle tabelle per Iceberg tramite Amazon Athena o utilizzando Spark in Amazon EMR o. AWS Glue

Quando esegui la compattazione utilizzando la procedura rewrite_data_files, puoi regolare diverse manopole per controllare il comportamento di compattazione. Il diagramma seguente mostra il comportamento predefinito del bin packing. Comprendere la compattazione dell'imballaggio in contenitori è fondamentale per comprendere le implementazioni dell'ordinamento gerarchico e dell'ordinamento con ordine Z, poiché sono estensioni dell'interfaccia di imballaggio dei contenitori e funzionano in modo simile. La differenza principale è il passaggio aggiuntivo necessario per l'ordinamento o il raggruppamento dei dati.



In questo esempio, la tabella Iceberg è composta da quattro partizioni. Ogni partizione ha dimensioni e numero di file diversi. Se avvii un'applicazione Spark per eseguire la compattazione, l'applicazione crea un totale di quattro gruppi di file da elaborare. Un gruppo di file è un'astrazione di Iceberg che rappresenta una raccolta di file che verranno elaborati da un singolo lavoro Spark. Cioè, l'applicazione Spark che esegue la compattazione creerà quattro job Spark per elaborare i dati.

Ottimizzazione del comportamento di compattazione

Le seguenti proprietà chiave controllano il modo in cui i file di dati vengono selezionati per la compattazione:

- MAX_FILE_GROUP_SIZE_BYTES imposta il limite di dati per un singolo gruppo di file (job Spark) a 100 GB per impostazione predefinita. Questa proprietà è particolarmente importante per le tabelle senza partizioni o le tabelle con partizioni che si estendono su centinaia di gigabyte. Impostando questo limite, è possibile suddividere le operazioni per pianificare il lavoro e fare progressi, evitando al contempo l'esaurimento delle risorse del cluster.

  Nota: ogni gruppo di file viene ordinato separatamente. Pertanto, se si desidera eseguire un ordinamento a livello di partizione, è necessario modificare questo limite in modo che corrisponda alla dimensione della partizione.
- Il valore predefinito di MIN_FILE_SIZE_BYTES o MIN_FILE_SIZE_DEFAULT_RATIO è il 75 percento della dimensione del file di destinazione impostata a livello di tabella. Ad esempio, se una tabella ha una dimensione di destinazione di 512 MB, qualsiasi file inferiore a 384 MB viene incluso nel set di file che verrà compattato.
- Il valore predefinito di MAX_FILE_SIZE_BYTES o MAX_FILE_SIZE_DEFAULT_RATIO è pari al 180 percento della dimensione del file di destinazione. Analogamente alle due proprietà che impostano le dimensioni minime dei file, queste proprietà vengono utilizzate per identificare i file candidati per il lavoro di compattazione.
- MIN_INPUT_FILES specifica il numero minimo di file da compattare se la dimensione della partizione della tabella è inferiore alla dimensione del file di destinazione. Il valore di questa proprietà viene utilizzato per determinare se vale la pena compattare i file in base al numero di file (il valore predefinito è 5).
- DELETE_FILE_THRESHOLD specifica il numero minimo di operazioni di eliminazione per un file prima che venga incluso nella compattazione. A meno che non venga specificato diversamente, la compattazione non combina i file di eliminazione con i file di dati. Per abilitare questa funzionalità, è necessario impostare un valore di soglia utilizzando questa proprietà. Questa soglia è specifica per i singoli file di dati, quindi se la impostate su 3, un file di dati verrà riscritto solo se vi sono tre o più file di eliminazione che vi fanno riferimento.

Queste proprietà forniscono informazioni sulla formazione dei gruppi di file nel diagramma precedente.

Ad esempio, la partizione etichettata `month=01` include due gruppi di file perché supera il limite di dimensione massima di 100 GB. Al contrario, la `month=02` partizione contiene un singolo gruppo di file perché è inferiore a 100 GB. La `month=03` partizione non soddisfa il requisito minimo predefinito di cinque file di input. Di conseguenza, non verrà compattata. Infine, sebbene la `month=04` partizione non contenga dati sufficienti per formare un singolo file della dimensione desiderata, i file verranno compattati perché la partizione include più di cinque file di piccole dimensioni.

Puoi impostare questi parametri per Spark in esecuzione su Amazon AWS Glue EMR o. Per Amazon Athena, puoi gestire proprietà simili utilizzando le proprietà della tabella che iniziano con il prefisso`optimize_`).

Esecuzione della compattazione con Spark su Amazon EMR oppure AWS Glue

Questa sezione descrive come dimensionare correttamente un cluster Spark per eseguire l'utilità di compattazione di Iceberg. L'esempio seguente utilizza Amazon EMR Serverless, ma è possibile utilizzare la stessa metodologia in Amazon EMR su Amazon EC2 o Amazon EKS o in. AWS Glue

Puoi sfruttare la correlazione tra i gruppi di file e i job Spark per pianificare le risorse del cluster. Per elaborare i gruppi di file in sequenza, considerando la dimensione massima di 100 GB per gruppo di file, puoi impostare le seguenti proprietà Spark:

- `spark.dynamicAllocation.enabled` = `FALSE`
- `spark.executor.memory` = `20 GB`
- `spark.executor.instances` = `5`

Se si desidera accelerare la compattazione, è possibile ridimensionare orizzontalmente aumentando il numero di gruppi di file compattati in parallelo. Puoi anche scalare Amazon EMR utilizzando la scalabilità manuale o dinamica.

- Ridimensionamento manuale (ad esempio, di un fattore 4)
  - `MAX_CONCURRENT_FILE_GROUP_REWRITES`= `4` (il nostro fattore)
  - `spark.executor.instances`= `5` (valore usato nell'esempio) x `4` (il nostro fattore) = `20`
  - `spark.dynamicAllocation.enabled` = `FALSE`
- Ridimensionamento dinamico
  - `spark.dynamicAllocation.enabled`= `TRUE` (impostazione predefinita, nessuna azione richiesta)

- MAX_CONCURRENT_FILE_GROUP_REWRITES = `N` (allinea questo valore `aspark.dynamicAllocation.maxExecutors`, che è 100 per impostazione predefinita; in base alle configurazioni dell'esecutore nell'esempio, puoi impostarlo su 20) `N`

Queste sono linee guida per aiutare a dimensionare il cluster. Tuttavia, dovresti anche monitorare le prestazioni dei tuoi job Spark per trovare le impostazioni migliori per i tuoi carichi di lavoro.

Esecuzione della compattazione con Amazon Athena

Athena offre un'implementazione dell'utilità di compattazione di Iceberg come funzionalità gestita tramite l'istruzione OPTIMIZE. È possibile utilizzare questa istruzione per eseguire la compattazione senza dover valutare l'infrastruttura.

Questa istruzione raggruppa file di piccole dimensioni in file più grandi utilizzando l'algoritmo bin packing e unisce i file delete con i file di dati esistenti. Per raggruppare i dati utilizzando l'ordinamento gerarchico o l'ordinamento con ordine z, usa Spark su Amazon EMR o. AWS Glue

Puoi modificare il comportamento predefinito dell'`OPTIMIZE`istruzione al momento della creazione della tabella inserendo le proprietà della tabella nell'istruzione o dopo la creazione della tabella utilizzando l'`CREATE TABLE`istruzione. `ALTER TABLE` Per i valori predefiniti, consulta la documentazione di Athena.

Consigli per eseguire la compattazione

Caso d'uso	Raccomandazione
Esecuzione della compattazione dell'imballaggio in contenitori in base a una pianificazione	• Usa l'`OPTIMIZE`istruzione in Athena se non sai quanti file di piccole dimensioni contiene la tabella. Il modello di prezzo di Athena si basa sui dati scansionati, quindi se non ci sono file da compattare, non ci sono costi associati a queste operazioni. Per evitare che si verifichino dei timeout sulle tabelle Athena, esegui su base base. `OPTIMIZE` per-table-partition

Caso d'uso	Raccomandazione
	• Usa Amazon EMR o AWS Glue con scalabilità dinamica quando prevedi di compattare grandi volumi di file di piccole dimensioni.
Esecuzione della compattazione dei contenitori in base agli eventi	• Usa Amazon EMR o AWS Glue con scalabilità dinamica quando prevedi di compattare grandi volumi di file di piccole dimensioni.
Esecuzione della compattazione per ordinare i dati	• Usa Amazon EMR oppure AWS Glue, perché l'ordinamento è un'operazione costosa e potrebbe dover trasferire i dati su disco.
Esecuzione della compattazione per raggruppare i dati utilizzando l'ordinamento z-order	• Usa Amazon EMR oppure AWS Glue, poiché l'ordinamento z-order è un'operazione molto costosa e potrebbe dover trasferire i dati su disco.
Esecuzione della compattazione su partizioni che potrebbero essere aggiornate da altre applicazioni a causa dell'arrivo tardivo dei dati	• Usa Amazon EMR o. AWS Glue Abilita la proprietà Iceberg PARTIAL_PROGRESS_ENABLED. Quando utilizzate questa opzione, Iceberg divide l'output della compattazione in più commit. In caso di collisione (ovvero se il file di dati viene aggiornato mentre è in corso la compattazione), questa impostazione riduce il costo di un nuovo tentativo limitandolo al commit che include il file interessato. In caso contrario, potrebbe essere necessario ricompattare tutti i file.

Utilizzo dei carichi di lavoro Iceberg in Amazon S3

Questa sezione illustra le proprietà di Iceberg che puoi utilizzare per ottimizzare l'interazione di Iceberg con Amazon S3.

Impedisci il partizionamento a caldo (errori HTTP 503)

Alcune applicazioni data lake eseguite su Amazon S3 gestiscono milioni o miliardi di oggetti ed elaborano petabyte di dati. Ciò può portare a prefissi che ricevono un volume di traffico elevato, che in genere vengono rilevati tramite errori HTTP 503 (servizio non disponibile). Per evitare questo problema, utilizzate le seguenti proprietà Iceberg:

- Impostato su `hash` o `range` in modo che Iceberg `write.distribution-mode` scriva file di grandi dimensioni, il che si traduce in un minor numero di richieste Amazon S3. Questa è la configurazione preferita e dovrebbe risolvere la maggior parte dei casi.
- Se continui a riscontrare 503 errori a causa di un enorme volume di dati nei tuoi carichi di lavoro, puoi `write.object-storage.enabled` impostare `true` in Iceberg. Questo indica a Iceberg di eseguire l'hash dei nomi degli oggetti e di distribuire il carico su più prefissi Amazon S3 randomizzati.

Per ulteriori informazioni su queste proprietà, consulta Write properties nella documentazione di Iceberg.

Usa le operazioni di manutenzione di Iceberg per rilasciare i dati non utilizzati

Per gestire le tabelle Iceberg, puoi utilizzare l'API di base Iceberg, i client Iceberg (come Spark) o servizi gestiti come Amazon Athena. Per eliminare file vecchi o inutilizzati da Amazon S3, ti consigliamo di utilizzare solo le API native di Iceberg per rimuovere istantanee, rimuoverevecchi file di metadati ed eliminare file orfani.

L'uso delle API di Amazon S3 tramite Boto3, l'SDK Amazon S3 o AWS Command Line Interface (AWS CLI) o l'utilizzo di qualsiasi altro metodo diverso da Iceberg per sovrascrivere o rimuovere i file Amazon S3 per una tabella Iceberg causa il danneggiamento delle tabelle e gli errori delle query.

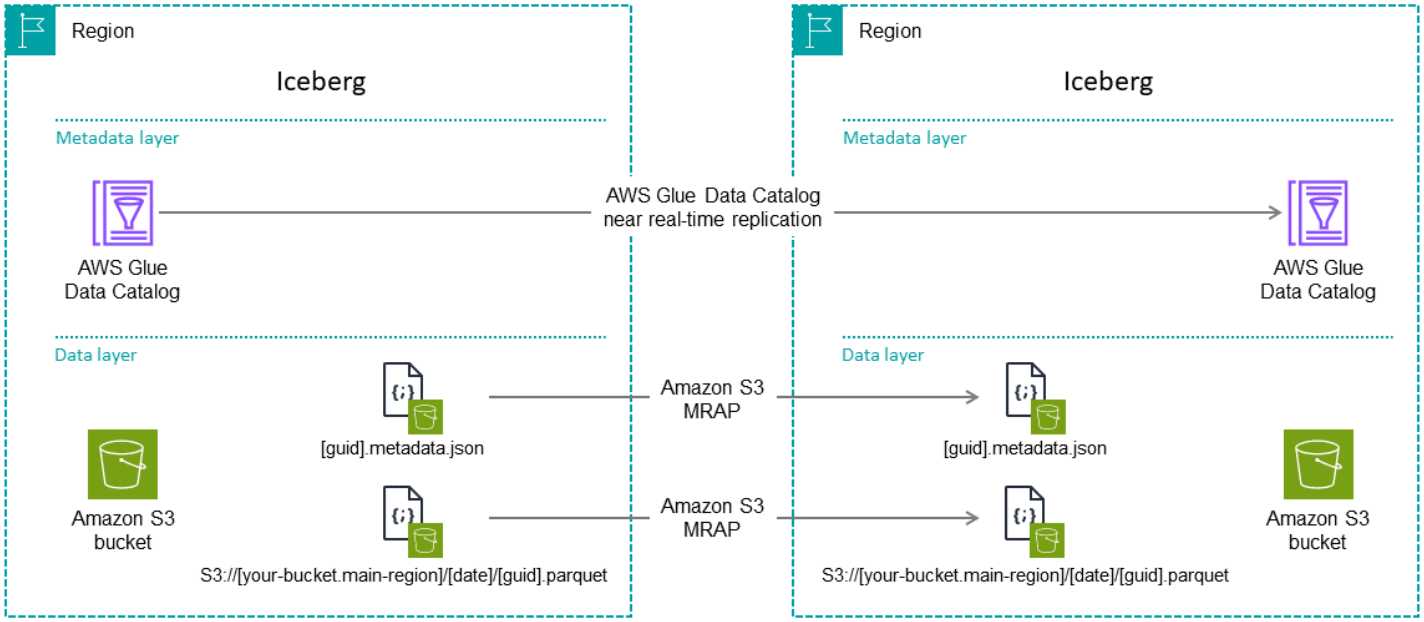
Replica i dati tra Regioni AWS

Quando memorizzi le tabelle Iceberg in Amazon S3, puoi utilizzare le funzionalità integrate in Amazon S3, come Cross-Region Replication (CRR) e Multi-Region Access Points (MRAP), per replicare i dati su più regioni AWS. MRAP fornisce un endpoint globale per consentire alle applicazioni di accedere ai bucket S3 dislocati in più bucket. Regioni AWS Iceberg non supporta percorsi relativi, ma puoi usare MRAP per eseguire operazioni su Amazon S3 mappando i bucket ai punti di accesso. MRAP si

integra inoltre perfettamente con il processo di replica interregionale di Amazon S3, che introduce un ritardo fino a 15 minuti. È necessario replicare sia i file di dati che i file di metadati.

> **Important**
>
> Attualmente, l'integrazione di Iceberg con MRAP funziona solo con Apache Spark. Se devi eseguire il failover sul sistema secondario Regione AWS, devi pianificare il reindirizzamento delle query degli utenti a un ambiente SQL Spark (come Amazon EMR) nella regione di failover.

Le funzionalità CRR e MRAP consentono di creare una soluzione di replica interregionale per le tabelle Iceberg, come illustrato nel diagramma seguente.



Per configurare questa architettura di replica interregionale:

1. Creare tabelle utilizzando la posizione MRAP. Ciò garantisce che i file di metadati Iceberg puntino alla posizione MRAP anziché alla posizione fisica del bucket.
2. Replica i file Iceberg utilizzando Amazon S3 MRAP. MRAP supporta la replica dei dati con un accordo sul livello di servizio (SLA) di 15 minuti. Iceberg impedisce alle operazioni di lettura di introdurre incongruenze durante la replica.
3. Rendi disponibili le tabelle AWS Glue Data Catalog nella regione secondaria. Puoi scegliere tra due opzioni:

- Imposta una pipeline per replicare i metadati della tabella Iceberg utilizzando la replica. AWS Glue Data Catalog Questa utilità è disponibile nell'archivio di GitHub replica Glue Catalog e Lake Formation Permissions. Questo meccanismo basato sugli eventi replica le tabelle nella regione di destinazione in base ai registri degli eventi.
- Registra le tabelle nella regione secondaria quando devi eseguire il failover. Per questa opzione, è possibile utilizzare l'utilità precedente o la procedura Iceberg register_table e indirizzarla al file più recente. `metadata.json`

Monitoraggio dei carichi di lavoro di Apache Iceberg

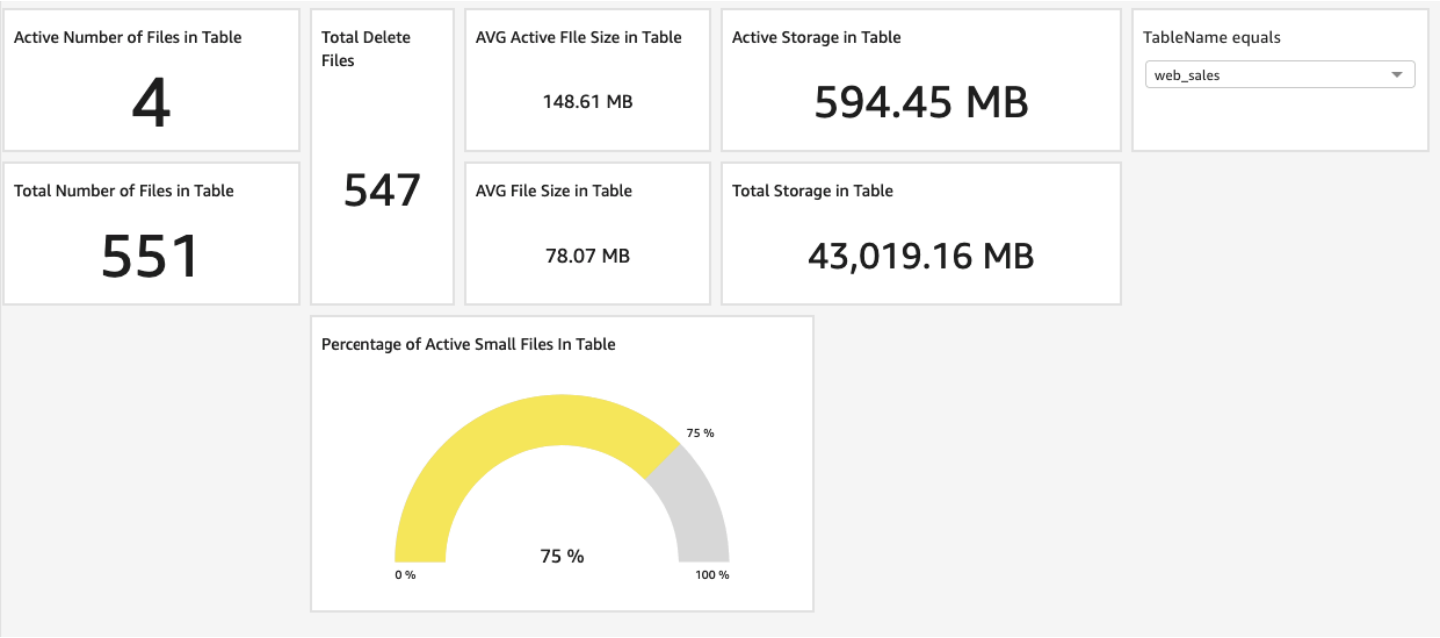
Per monitorare i carichi di lavoro di Iceberg, hai due opzioni: analizzare le tabelle di metadati o utilizzare i reporter di metriche. I reporter di metriche sono stati introdotti nella versione 1.2 di Iceberg e sono disponibili solo per i cataloghi REST e JDBC.

Se lo utilizzi AWS Glue Data Catalog, puoi ottenere informazioni sullo stato delle tue tabelle Iceberg impostando il monitoraggio in aggiunta alle tabelle di metadati esposte da Iceberg.

Il monitoraggio è fondamentale per la gestione delle prestazioni e la risoluzione dei problemi. Ad esempio, quando una partizione in una tabella Iceberg raggiunge una certa percentuale di file di piccole dimensioni, il carico di lavoro può avviare un processo di compattazione per consolidare i file in file più grandi. In questo modo si evita che le query rallentino oltre un livello accettabile.

Monitoraggio a livello di tabella

La schermata seguente mostra una dashboard di monitoraggio delle tabelle creata in Amazon QuickSight. Questa dashboard interroga le tabelle di metadati di Iceberg utilizzando Spark SQL e acquisisce metriche dettagliate come il numero di file attivi e lo spazio di archiviazione totale. Queste informazioni vengono quindi archiviate in tabelle per scopi operativi. AWS Glue Infine, viene creata una QuickSight dashboard, come illustrato nella figura seguente, utilizzando Amazon Athena. Queste informazioni ti aiutano a identificare e risolvere problemi specifici nei tuoi sistemi.



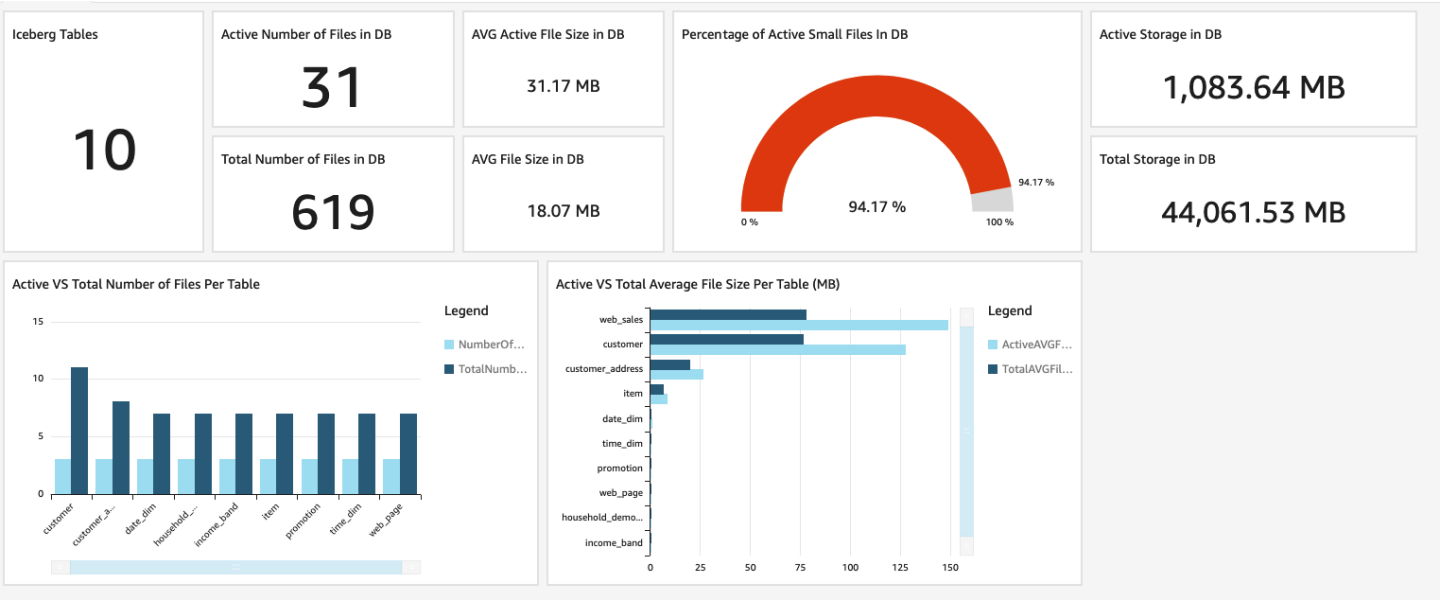
La QuickSight dashboard di esempio raccoglie i seguenti indicatori chiave di prestazione (KPI) per una tabella Iceberg:

KPI	Descrizione	Query
Numero di file	Il numero di file nella tabella Iceberg (per tutte le istantanee)	`select count(*) from <catalog.database.table_name>.all_files`
Numero di file attivi	Il numero di file attivi nell'ultima istantanea della tabella Iceberg	`select count(*) from <catalog.database.table_name>.files`
Dimensione media del file	La dimensione media dei file, in megabyte, per tutti i file della tabella Iceberg	`select avg(file_size_in_bytes)/1000000 from <catalog.database.table_name>.all_files`
Dimensione media dei file attivi	La dimensione media dei file, in megabyte, per i file attivi nella tabella Iceberg	`select avg(file_size_in_bytes)/1000000 from <catalog.database.table_name>.files`
Percentuale di file di piccole dimensioni	La percentuale di file attivi di dimensioni inferiori a 100 MB	`select cast(sum( case when file_size_in_bytes < 100000000 then 1 else 0 end)*100/count(*) as decimal(10,2)) from <catalog.database.table_name>.files`
Dimensione totale dello spazio di archiviazione	La dimensione totale di tutti i file nella tabella, esclusi i file orfani e le versioni degli	`select sum(file_size_in_bytes)/1000000`

KPI	Descrizione	Query
	oggetti Amazon S3 (se abilitate)	`from <catalog.database.table_name>.all_files`
Dimensione totale dello storage attivo	La dimensione totale di tutti i file nelle istantanee correnti di una determinata tabella	`select sum(file_size_in_bytes)/1000000 from <catalog.database.table_name>.files`

Monitoraggio a livello di database

L'esempio seguente mostra un dashboard di monitoraggio creato nel QuickSight per fornire una panoramica dei KPI a livello di database per una raccolta di tabelle Iceberg.



Questa dashboard raccoglie i seguenti KPI:

KPI	Descrizione	Query
Numero di file	Il numero di file nel database Iceberg (per tutte le istantanee)	Questa dashboard utilizza le query a livello di tabella fornite nella sezione precedente e consolida i risultati.

KPI	Descrizione	Query
Numero di file attivi	Il numero di file attivi nel database Iceberg (basato sulle ultime istantanee delle tabelle Iceberg)	
Dimensione media del file	La dimensione media dei file, in megabyte, per tutti i file del database Iceberg	
Dimensione media dei file attivi	La dimensione media dei file, in megabyte, per tutti i file attivi nel database Iceberg	
Percentuale di file di piccole dimensioni	La percentuale di file attivi di dimensioni inferiori a 100 MB nel database Iceberg	
Dimensione totale dello spazio di archiviazione	La dimensione totale di tutti i file nel database, esclusi i file orfani e le versioni degli oggetti Amazon S3 (se abilitate)	
Dimensione totale dello storage attivo	La dimensione totale di tutti i file nelle istantanee correnti di tutte le tabelle del database	

Manutenzione preventiva

Impostando le funzionalità di monitoraggio illustrate nelle sezioni precedenti, è possibile affrontare la manutenzione delle tabelle da un punto di vista preventivo anziché reattivo. Ad esempio, puoi utilizzare le metriche a livello di tabella e a livello di database per pianificare azioni come le seguenti:

- Utilizzate la compattazione con imballaggio in contenitori per raggruppare file di piccole dimensioni quando una tabella raggiunge N file di piccole dimensioni.

- Usa la compressione tramite bin packing per unire i file eliminati quando una tabella raggiunge N, elimina i file in una determinata partizione.
- Rimuovi i file di piccole dimensioni che erano già compattati rimuovendo le istantanee quando lo spazio di archiviazione totale è X volte superiore allo storage attivo.

Governance e controllo degli accessi per Apache Iceberg on AWS

Apache Iceberg si integra con AWS Lake Formation per semplificare la governance dei dati. Questa integrazione consente agli amministratori del data lake di assegnare autorizzazioni di accesso a livello di cella alle tabelle Iceberg. Per un esempio di interrogazione delle tabelle Iceberg utilizzando Amazon Athena AWS Lake Formation e, consulta AWS il post del blog Interagisci con le tabelle Apache Iceberg utilizzando Amazon Athena e utilizzando autorizzazioni granulari tra account. AWS Lake Formation

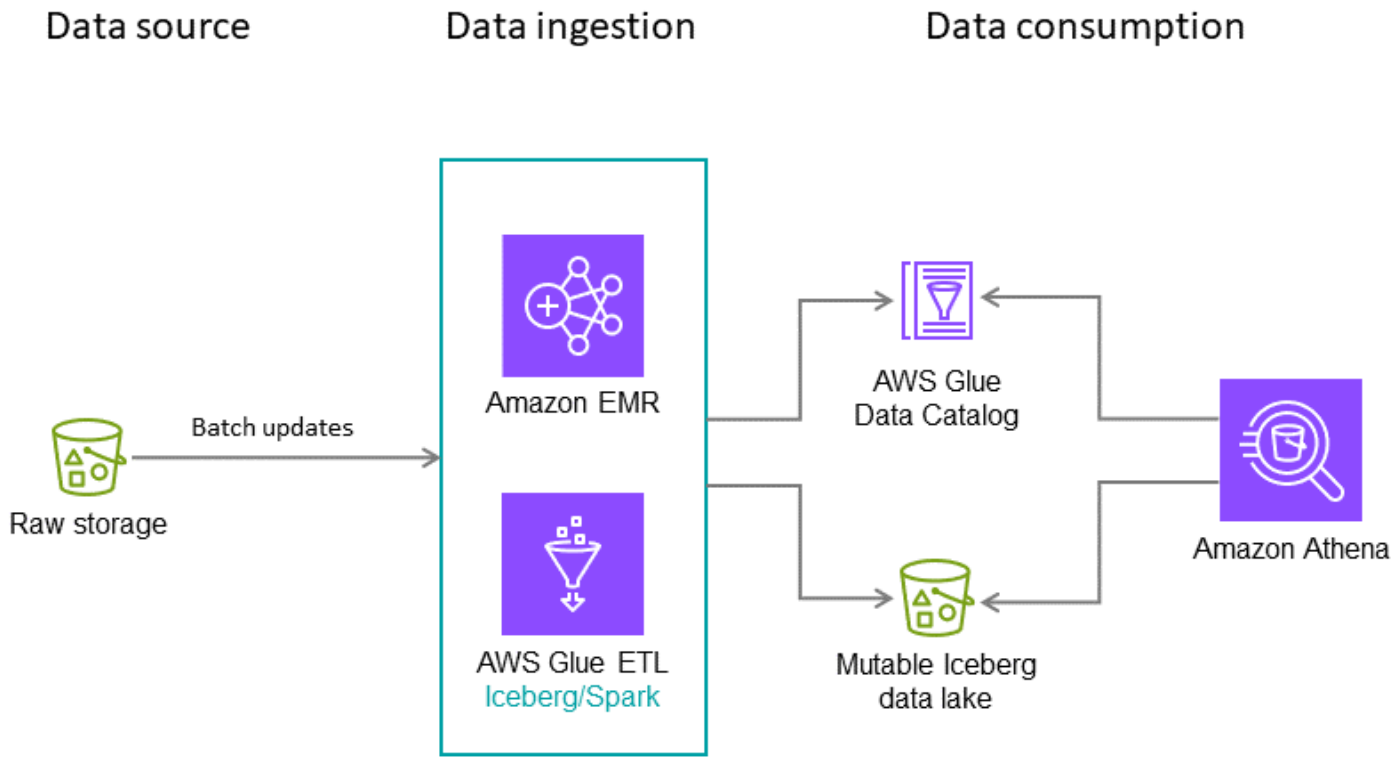
Architetture di riferimento per Apache Iceberg su AWS

Questa sezione fornisce esempi di come applicare le migliori pratiche in diversi casi d'uso, come l'ingestione in batch e un data lake che combina l'ingestione di dati in batch e in streaming.

Inserimento notturno in batch

In questo caso d'uso ipotetico, supponiamo che la tabella Iceberg inserisca le transazioni con carta di credito su base notturna. Ogni batch contiene solo aggiornamenti incrementali, che devono essere uniti nella tabella di destinazione. Più volte all'anno, vengono ricevuti dati storici completi. Per questo scenario, consigliamo l'architettura e le configurazioni seguenti.

Nota: questo è solo un esempio. La configurazione ottimale dipende dai dati e dai requisiti.



Raccomandazioni:

- Dimensione del file: 128 MB, poiché le attività di Apache Spark elaborano i dati in blocchi da 128 MB.
- Tipo di scrittura:. copy-on-write Come dettagliato in precedenza in questa guida, questo approccio aiuta a garantire che i dati vengano scritti in modo ottimizzato per la lettura.

- Variabili di partizione: anno/mese/giorno. Nel nostro caso d'uso ipotetico, interroghiamo i dati recenti più frequentemente, anche se occasionalmente eseguiamo scansioni complete delle tabelle per gli ultimi due anni di dati. L'obiettivo del partizionamento è quello di velocizzare le operazioni di lettura in base ai requisiti del caso d'uso.
- Ordinamento: timestamp
- Catalogo dati: AWS Glue Data Catalog

Data lake che combina ingestione in batch e quasi in tempo reale

Puoi fornire un data lake su Amazon S3 che condivide dati in batch e in streaming tra account e regioni. Per un diagramma di architettura e dettagli, consulta il post del AWS blog Crea un data lake transazionale usando Apache Iceberg e le condivisioni di dati tra account utilizzando Amazon Athena. AWS Glue AWS Lake Formation

Risorse

- Utilizzo del framework Iceberg in AWS Glue (documentazione)AWS Glue
- Iceberg (documentazione Amazon EMR)
- Utilizzo delle tabelle Apache Iceberg (documentazione Amazon Athena)
- Documentazione Amazon S3
- QuickSight Documentazione Amazon
- Replica dei permessi di Glue Catalog e Lake Formation (repository) GitHub
- Documentazione di Apache Iceberg
- Documentazione di Apache Spark

Collaboratori

Le seguenti persone hanno scritto, AWS coautore e recensito questa guida.

Collaboratori

- Carlos Rodrigues, architetto delle soluzioni, Big Data
- Imtiaz (Taz) Sayed, architetto di soluzioni, responsabile tecnico, analisi
- Shana Schipers, architetto di soluzioni, Big Data
- Prashant Singh, ingegnere di sviluppo software, Amazon EMR
- Stefano Sandona, architetto di soluzioni, Big Data
- Arun A K, architetto di soluzioni, Big Data ed ETL
- Francisco Morillo, architetto di soluzioni, streaming
- Suthan Phillips, architetto di analisi, Amazon EMR
- Sercan Karaoglu, architetto di soluzioni
- Yonatan Dolan, specialista in analisi
- Guy Bachar, architetto di soluzioni
- Sofia Zilberman, architetto di soluzioni, streaming
- Ismail Makhlouf, architetto di soluzioni, analisi
- Dan Stair, architetto specializzato in soluzioni
- Sakti Mishra, architetto delle soluzioni

Revisori

- Rick Sears, direttore generale, Amazon EMR
- Linda OConnor, specialista di Amazon EMR
- Ian Meyers, direttore, Amazon EMR
- Vinita Ananth, direttrice della gestione dei prodotti Amazon EMR
- Jason Berkowitz, responsabile di prodotto, AWS Lake Formation
- Mahesh Mishra, responsabile di prodotto, Amazon Redshift
- Vladimir Zlatkin, responsabile, architettura delle soluzioni, Big Data
- Karthik Prabhakar, architetto di analisi, Amazon EMR

- Jack Ye, ingegnere di sviluppo software, Amazon EMR
- Vijay Jain, responsabile di prodotto
- Anupriti Warade, responsabile di prodotto, Amazon S3
- Molly Brown, direttore generale, AWS Lake Formation
- Ajit Tandale, architetto di soluzioni, Data
- Gwen Chen, responsabile marketing del prodotto

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	—	30 aprile 2024

AWS Glossario delle linee guida prescrittive

I seguenti sono termini comunemente usati nelle strategie, nelle guide e nei modelli forniti da AWS Prescriptive Guidance. Per suggerire voci, utilizza il link Fornisci feedback alla fine del glossario.

Numeri

7 R

Sette strategie di migrazione comuni per trasferire le applicazioni sul cloud. Queste strategie si basano sulle 5 R identificate da Gartner nel 2011 e sono le seguenti:

- Rifattorizzare/riprogettare: trasferisci un'applicazione e modifica la sua architettura sfruttando appieno le funzionalità native del cloud per migliorare l'agilità, le prestazioni e la scalabilità. Ciò comporta in genere la portabilità del sistema operativo e del database. Esempio: migra il tuo database Oracle locale all'edizione compatibile con Amazon Aurora PostgreSQL.
- Ridefinire la piattaforma (lift and reshape): trasferisci un'applicazione nel cloud e introduci un certo livello di ottimizzazione per sfruttare le funzionalità del cloud. Esempio: migra il tuo database Oracle locale ad Amazon Relational Database Service (Amazon RDS) per Oracle in. Cloud AWS
- Riacquistare (drop and shop): passa a un prodotto diverso, in genere effettuando la transizione da una licenza tradizionale a un modello SaaS. Esempio: migra il tuo sistema di gestione delle relazioni con i clienti (CRM) su Salesforce.com.
- Eseguire il rehosting (lift and shift): trasferisci un'applicazione sul cloud senza apportare modifiche per sfruttare le funzionalità del cloud. Esempio: migra il tuo database Oracle locale a Oracle su un'istanza EC2 in. Cloud AWS
- Trasferire (eseguire il rehosting a livello hypervisor): trasferisci l'infrastruttura sul cloud senza acquistare nuovo hardware, riscrivere le applicazioni o modificare le operazioni esistenti. Esegui la migrazione dei server da una piattaforma locale a un servizio cloud per la stessa piattaforma. Esempio: migra un'applicazione suMicrosoft Hyper-V. AWS
- Riesaminare (mantenere): mantieni le applicazioni nell'ambiente di origine. Queste potrebbero includere applicazioni che richiedono una rifattorizzazione significativa che desideri rimandare a un momento successivo e applicazioni legacy che desideri mantenere, perché non vi è alcuna giustificazione aziendale per effettuarne la migrazione.
- Ritirare: disattiva o rimuovi le applicazioni che non sono più necessarie nell'ambiente di origine.

A

ABAC

Vedi controllo degli accessi basato sugli attributi.

servizi astratti

Vedi servizi gestiti.

ACIDO

Vedi atomicità, consistenza, isolamento, durata.

migrazione attiva-attiva

Un metodo di migrazione del database in cui i database di origine e di destinazione vengono mantenuti sincronizzati (utilizzando uno strumento di replica bidirezionale o operazioni di doppia scrittura) ed entrambi i database gestiscono le transazioni provenienti dalle applicazioni di connessione durante la migrazione. Questo metodo supporta la migrazione in piccoli batch controllati anziché richiedere una conversione una tantum. È più flessibile ma richiede più lavoro rispetto alla migrazione attiva-passiva.

migrazione attiva-passiva

Un metodo di migrazione di database in cui i database di origine e di destinazione vengono mantenuti sincronizzati, ma solo il database di origine gestisce le transazioni provenienti dalle applicazioni di connessione mentre i dati vengono replicati nel database di destinazione. Il database di destinazione non accetta alcuna transazione durante la migrazione.

funzione aggregata

Una funzione SQL che opera su un gruppo di righe e calcola un singolo valore restituito per il gruppo. Esempi di funzioni aggregate includono `SUM` e. `MAX`

Intelligenza artificiale

Vedi intelligenza artificiale.

AIOps

Guarda le operazioni di intelligenza artificiale.

anonimizzazione

Il processo di eliminazione permanente delle informazioni personali in un set di dati. L'anonimizzazione può aiutare a proteggere la privacy personale. I dati anonimi non sono più considerati dati personali.

anti-modello

Una soluzione utilizzata frequentemente per un problema ricorrente in cui la soluzione è controproducente, inefficace o meno efficace di un'alternativa.

controllo delle applicazioni

Un approccio alla sicurezza che consente l'uso solo di applicazioni approvate per proteggere un sistema dal malware.

portfolio di applicazioni

Una raccolta di informazioni dettagliate su ogni applicazione utilizzata da un'organizzazione, compresi i costi di creazione e manutenzione dell'applicazione e il relativo valore aziendale. Queste informazioni sono fondamentali per il processo di scoperta e analisi del portfolio e aiutano a identificare e ad assegnare la priorità alle applicazioni da migrare, modernizzare e ottimizzare.

intelligenza artificiale (IA)

Il campo dell'informatica dedicato all'uso delle tecnologie informatiche per svolgere funzioni cognitive tipicamente associate agli esseri umani, come l'apprendimento, la risoluzione di problemi e il riconoscimento di schemi. Per ulteriori informazioni, consulta la sezione Che cos'è l'intelligenza artificiale?

operazioni di intelligenza artificiale (AIOps)

Il processo di utilizzo delle tecniche di machine learning per risolvere problemi operativi, ridurre gli incidenti operativi e l'intervento umano e aumentare la qualità del servizio. Per ulteriori informazioni su come viene utilizzato AIOps nella strategia di migrazione AWS , consulta la guida all'integrazione delle operazioni.

crittografia asimmetrica

Un algoritmo di crittografia che utilizza una coppia di chiavi, una chiave pubblica per la crittografia e una chiave privata per la decrittografia. Puoi condividere la chiave pubblica perché non viene utilizzata per la decrittografia, ma l'accesso alla chiave privata deve essere altamente limitato.

atomicità, consistenza, isolamento, durabilità (ACID)

Un insieme di proprietà del software che garantiscono la validità dei dati e l'affidabilità operativa di un database, anche in caso di errori, interruzioni di corrente o altri problemi.

Controllo degli accessi basato su attributi (ABAC)

La pratica di creare autorizzazioni dettagliate basate su attributi utente, come reparto, ruolo professionale e nome del team. Per ulteriori informazioni, consulta ABAC for AWS nella documentazione AWS Identity and Access Management (IAM).

fonte di dati autorevole

Una posizione in cui è archiviata la versione principale dei dati, considerata la fonte di informazioni più affidabile. È possibile copiare i dati dalla fonte di dati autorevole in altre posizioni allo scopo di elaborarli o modificarli, ad esempio anonimizzandoli, oscurandoli o pseudonimizzandoli.

Zona di disponibilità

Una posizione distinta all'interno di un edificio Regione AWS che è isolata dai guasti in altre zone di disponibilità e offre una connettività di rete economica e a bassa latenza verso altre zone di disponibilità nella stessa regione.

AWS Cloud Adoption Framework (CAF)AWS

Un framework di linee guida e best practice AWS per aiutare le organizzazioni a sviluppare un piano efficiente ed efficace per passare con successo al cloud. AWS CAF organizza le linee guida in sei aree di interesse chiamate prospettive: business, persone, governance, piattaforma, sicurezza e operazioni. Le prospettive relative ad azienda, persone e governance si concentrano sulle competenze e sui processi aziendali; le prospettive relative alla piattaforma, alla sicurezza e alle operazioni si concentrano sulle competenze e sui processi tecnici. Ad esempio, la prospettiva relativa alle persone si rivolge alle parti interessate che gestiscono le risorse umane (HR), le funzioni del personale e la gestione del personale. In questa prospettiva, AWS CAF fornisce linee guida per lo sviluppo delle persone, la formazione e le comunicazioni per aiutare a preparare l'organizzazione all'adozione del cloud di successo. Per ulteriori informazioni, consulta il sito web di AWS CAF e il white paper AWS CAF.

AWS Workload Qualification Framework (WQF)AWS

Uno strumento che valuta i carichi di lavoro di migrazione dei database, consiglia strategie di migrazione e fornisce stime del lavoro. AWS WQF è incluso in (). AWS Schema Conversion Tool AWS SCT Analizza gli schemi di database e gli oggetti di codice, il codice dell'applicazione, le dipendenze e le caratteristiche delle prestazioni e fornisce report di valutazione.

B

bot difettoso

Un bot che ha lo scopo di disturbare o causare danni a individui o organizzazioni.

BCP

Vedi la pianificazione della continuità operativa.

grafico comportamentale

Una vista unificata, interattiva dei comportamenti delle risorse e delle interazioni nel tempo. Puoi utilizzare un grafico comportamentale con Amazon Detective per esaminare tentativi di accesso non riusciti, chiamate API sospette e azioni simili. Per ulteriori informazioni, consulta Dati in un grafico comportamentale nella documentazione di Detective.

sistema big-endian

Un sistema che memorizza per primo il byte più importante. Vedi anche endianness.

Classificazione binaria

Un processo che prevede un risultato binario (una delle due classi possibili). Ad esempio, il modello di machine learning potrebbe dover prevedere problemi come "Questa e-mail è spam o non è spam?" o "Questo prodotto è un libro o un'auto?"

filtro Bloom

Una struttura di dati probabilistica ed efficiente in termini di memoria che viene utilizzata per verificare se un elemento fa parte di un set.

distribuzioni blu/verdi

Una strategia di implementazione in cui si creano due ambienti separati ma identici. La versione corrente dell'applicazione viene eseguita in un ambiente (blu) e la nuova versione dell'applicazione nell'altro ambiente (verde). Questa strategia consente di ripristinare rapidamente il sistema con un impatto minimo.

bot

Un'applicazione software che esegue attività automatizzate su Internet e simula l'attività o l'interazione umana. Alcuni bot sono utili o utili, come i web crawler che indicizzano le informazioni su Internet. Alcuni altri bot, noti come bot dannosi, hanno lo scopo di disturbare o causare danni a individui o organizzazioni.

botnet

Reti di bot infettate da malware e controllate da un'unica parte, nota come bot herder o bot operator. Le botnet sono il meccanismo più noto per scalare i bot e il loro impatto.

ramo

Un'area contenuta di un repository di codice. Il primo ramo creato in un repository è il ramo principale. È possibile creare un nuovo ramo a partire da un ramo esistente e quindi sviluppare funzionalità o correggere bug al suo interno. Un ramo creato per sviluppare una funzionalità viene comunemente detto ramo di funzionalità. Quando la funzionalità è pronta per il rilascio, il ramo di funzionalità viene ricongiunto al ramo principale. Per ulteriori informazioni, consulta Informazioni sulle filiali (documentazione). GitHub

accesso break-glass

In circostanze eccezionali e tramite una procedura approvata, un mezzo rapido per consentire a un utente di accedere a un sito a Account AWS cui in genere non dispone delle autorizzazioni necessarie. Per ulteriori informazioni, vedere l'indicatore Implementate break-glass procedures nella guida Well-Architected AWS .

strategia brownfield

L'infrastruttura esistente nell'ambiente. Quando si adotta una strategia brownfield per un'architettura di sistema, si progetta l'architettura in base ai vincoli dei sistemi e dell'infrastruttura attuali. Per l'espansione dell'infrastruttura esistente, è possibile combinare strategie brownfield e greenfield.

cache del buffer

L'area di memoria in cui sono archiviati i dati a cui si accede con maggiore frequenza.

capacità di business

Azioni intraprese da un'azienda per generare valore (ad esempio vendite, assistenza clienti o marketing). Le architetture dei microservizi e le decisioni di sviluppo possono essere guidate dalle capacità aziendali. Per ulteriori informazioni, consulta la sezione Organizzazione in base alle funzionalità aziendali del whitepaper Esecuzione di microservizi containerizzati su AWS.

pianificazione della continuità operativa (BCP)

Un piano che affronta il potenziale impatto di un evento che comporta l'interruzione dell'attività, come una migrazione su larga scala, sulle operazioni e consente a un'azienda di riprendere rapidamente le operazioni.

C

CAF

Vedi AWS Cloud Adoption Framework.

implementazione canaria

Il rilascio lento e incrementale di una versione agli utenti finali. Quando sei sicuro, distribuisci la nuova versione e sostituisci la versione corrente nella sua interezza.

CoE

Vedi Cloud Center of Excellence.

CDC

Vedi Change Data Capture.

Change Data Capture (CDC)

Il processo di tracciamento delle modifiche a un'origine dati, ad esempio una tabella di database, e di registrazione dei metadati relativi alla modifica. È possibile utilizzare CDC per vari scopi, ad esempio il controllo o la replica delle modifiche in un sistema di destinazione per mantenere la sincronizzazione.

ingegneria del caos

Introduzione intenzionale di guasti o eventi dirompenti per testare la resilienza di un sistema. Puoi usare AWS Fault Injection Service (AWS FIS) per eseguire esperimenti che stressano i tuoi AWS carichi di lavoro e valutarne la risposta.

CI/CD

Vedi integrazione continua e distribuzione continua.

classificazione

Un processo di categorizzazione che aiuta a generare previsioni. I modelli di ML per problemi di classificazione prevedono un valore discreto. I valori discreti sono sempre distinti l'uno dall'altro. Ad esempio, un modello potrebbe dover valutare se in un'immagine è presente o meno un'auto.

crittografia lato client

Crittografia dei dati a livello locale, prima che il destinatario li AWS servizio riceva.

centro di eccellenza del cloud (CCoE)

Un team multidisciplinare che guida le iniziative di adozione del cloud in tutta l'organizzazione, tra cui lo sviluppo di best practice per il cloud, la mobilitazione delle risorse, la definizione delle tempistiche di migrazione e la guida dell'organizzazione attraverso trasformazioni su larga scala. Per ulteriori informazioni, consulta i post di CCoE sull' Cloud AWS Enterprise Strategy Blog.

cloud computing

La tecnologia cloud generalmente utilizzata per l'archiviazione remota di dati e la gestione dei dispositivi IoT. Il cloud computing è generalmente collegato alla tecnologia di edge computing.

modello operativo cloud

In un'organizzazione IT, il modello operativo utilizzato per creare, maturare e ottimizzare uno o più ambienti cloud. Per ulteriori informazioni, consulta Building your Cloud Operating Model.

fasi di adozione del cloud

Le quattro fasi che le organizzazioni in genere attraversano quando migrano verso Cloud AWS:

- Progetto: esecuzione di alcuni progetti relativi al cloud per scopi di dimostrazione e apprendimento
- Fondamento: effettuare investimenti fondamentali per dimensionare l'adozione del cloud (ad esempio, creazione di una zona di destinazione, definizione di un CCoE, definizione di un modello operativo)
- Migrazione: migrazione di singole applicazioni
- Reinvenzione: ottimizzazione di prodotti e servizi e innovazione nel cloud

Queste fasi sono state definite da Stephen Orban nel post del blog The Journey Toward Cloud-First & the Stages of Adoption on the Enterprise Strategy. Cloud AWS Per informazioni su come si relazionano alla strategia di AWS migrazione, consulta la guida alla preparazione alla migrazione.

CMDB

Vedi database di gestione della configurazione.

repository di codice

Una posizione in cui il codice di origine e altri asset, come documentazione, esempi e script, vengono archiviati e aggiornati attraverso processi di controllo delle versioni. Gli archivi cloud più comuni includono GitHub o AWS CodeCommit. Ogni versione del codice è denominata ramo. In una struttura a microservizi, ogni repository è dedicato a una singola funzionalità. Una singola pipeline CI/CD può utilizzare più repository.

cache fredda

Una cache del buffer vuota, non ben popolata o contenente dati obsoleti o irrilevanti. Ciò influisce sulle prestazioni perché l'istanza di database deve leggere dalla memoria o dal disco principale, il che richiede più tempo rispetto alla lettura dalla cache del buffer.

dati freddi

Dati a cui si accede raramente e che in genere sono storici. Quando si eseguono interrogazioni di questo tipo di dati, le interrogazioni lente sono in genere accettabili. Lo spostamento di questi dati su livelli o classi di storage meno costosi e con prestazioni inferiori può ridurre i costi.

visione artificiale (CV)

Un campo dell'intelligenza artificiale che utilizza l'apprendimento automatico per analizzare ed estrarre informazioni da formati visivi come immagini e video digitali. Ad esempio, AWS Panorama offre dispositivi che aggiungono CV alle reti di telecamere locali e Amazon SageMaker fornisce algoritmi di elaborazione delle immagini per CV.

deriva della configurazione

Per un carico di lavoro, una modifica della configurazione rispetto allo stato previsto. Potrebbe causare la non conformità del carico di lavoro e in genere è graduale e involontaria.

database di gestione della configurazione (CMDB)

Un repository che archivia e gestisce le informazioni su un database e il relativo ambiente IT, inclusi i componenti hardware e software e le relative configurazioni. In genere si utilizzano i dati di un CMDB nella fase di individuazione e analisi del portafoglio della migrazione.

Pacchetto di conformità

Una raccolta di AWS Config regole e azioni correttive che puoi assemblare per personalizzare i controlli di conformità e sicurezza. È possibile distribuire un pacchetto di conformità come singola entità in una regione Account AWS and o all'interno di un'organizzazione utilizzando un modello YAML. Per ulteriori informazioni, consulta i Conformance Pack nella documentazione. AWS Config

integrazione e distribuzione continua (continuous integration and continuous delivery, CI/CD)

Il processo di automazione delle fasi di origine, creazione, test, gestione temporanea e produzione del processo di rilascio del software. Il processo CI/CD è comunemente descritto come una pipeline. CI/CD può aiutare ad automatizzare i processi, migliorare la produttività, migliorare

la qualità del codice e velocizzare le distribuzioni. Per ulteriori informazioni, consulta Vantaggi della distribuzione continua. CD può anche significare continuous deployment (implementazione continua). Per ulteriori informazioni, consulta Distribuzione continua e implementazione continua a confronto.

CV

Vedi visione artificiale.

D

dati a riposo

Dati stazionari nella rete, ad esempio i dati archiviati.

classificazione dei dati

Un processo per identificare e classificare i dati nella rete in base alla loro criticità e sensibilità. È un componente fondamentale di qualsiasi strategia di gestione dei rischi di sicurezza informatica perché consente di determinare i controlli di protezione e conservazione appropriati per i dati. La classificazione dei dati è un componente del pilastro della sicurezza nel AWS Well-Architected Framework. Per ulteriori informazioni, consulta Classificazione dei dati.

deriva dei dati

Una variazione significativa tra i dati di produzione e i dati utilizzati per addestrare un modello di machine learning o una modifica significativa dei dati di input nel tempo. La deriva dei dati può ridurre la qualità, l'accuratezza e l'equità complessive nelle previsioni dei modelli ML.

dati in transito

Dati che si spostano attivamente attraverso la rete, ad esempio tra le risorse di rete.

rete di dati

Un framework architettonico che fornisce la proprietà distribuita e decentralizzata dei dati con gestione e governance centralizzate.

riduzione al minimo dei dati

Il principio della raccolta e del trattamento dei soli dati strettamente necessari. Praticare la riduzione al minimo dei dati in the Cloud AWS può ridurre i rischi per la privacy, i costi e l'impronta di carbonio delle analisi.

perimetro dei dati

Una serie di barriere preventive nell' AWS ambiente che aiutano a garantire che solo le identità attendibili accedano alle risorse attendibili delle reti previste. Per ulteriori informazioni, consulta Building a data perimeter on. AWS

pre-elaborazione dei dati

Trasformare i dati grezzi in un formato che possa essere facilmente analizzato dal modello di ML. La pre-elaborazione dei dati può comportare la rimozione di determinate colonne o righe e l'eliminazione di valori mancanti, incoerenti o duplicati.

provenienza dei dati

Il processo di tracciamento dell'origine e della cronologia dei dati durante il loro ciclo di vita, ad esempio il modo in cui i dati sono stati generati, trasmessi e archiviati.

soggetto dei dati

Un individuo i cui dati vengono raccolti ed elaborati.

data warehouse

Un sistema di gestione dei dati che supporta la business intelligence, come l'analisi. I data warehouse contengono in genere grandi quantità di dati storici e vengono generalmente utilizzati per interrogazioni e analisi.

linguaggio di definizione del database (DDL)

Istruzioni o comandi per creare o modificare la struttura di tabelle e oggetti in un database.

linguaggio di manipolazione del database (DML)

Istruzioni o comandi per modificare (inserire, aggiornare ed eliminare) informazioni in un database.

DDL

Vedi linguaggio di definizione del database.

deep ensemble

Combinare più modelli di deep learning per la previsione. È possibile utilizzare i deep ensemble per ottenere una previsione più accurata o per stimare l'incertezza nelle previsioni.

deep learning

Un sottocampo del ML che utilizza più livelli di reti neurali artificiali per identificare la mappatura tra i dati di input e le variabili target di interesse.

defense-in-depth

Un approccio alla sicurezza delle informazioni in cui una serie di meccanismi e controlli di sicurezza sono accuratamente stratificati su una rete di computer per proteggere la riservatezza, l'integrità e la disponibilità della rete e dei dati al suo interno. Quando si adotta questa strategia AWS, si aggiungono più controlli a diversi livelli della AWS Organizations struttura per proteggere le risorse. Ad esempio, un defense-in-depth approccio potrebbe combinare l'autenticazione a più fattori, la segmentazione della rete e la crittografia.

amministratore delegato

In AWS Organizations, un servizio compatibile può registrare un account AWS membro per amministrare gli account dell'organizzazione e gestire le autorizzazioni per quel servizio. Questo account è denominato amministratore delegato per quel servizio specifico. Per ulteriori informazioni e un elenco di servizi compatibili, consulta Servizi che funzionano con AWS Organizations nella documentazione di AWS Organizations .

implementazione

Il processo di creazione di un'applicazione, di nuove funzionalità o di correzioni di codice disponibili nell'ambiente di destinazione. L'implementazione prevede l'applicazione di modifiche in una base di codice, seguita dalla creazione e dall'esecuzione di tale base di codice negli ambienti applicativi.

Ambiente di sviluppo

Vedi ambiente.

controllo di rilevamento

Un controllo di sicurezza progettato per rilevare, registrare e avvisare dopo che si è verificato un evento. Questi controlli rappresentano una seconda linea di difesa e avvisano l'utente in caso di eventi di sicurezza che aggirano i controlli preventivi in vigore. Per ulteriori informazioni, consulta Controlli di rilevamento in Implementazione dei controlli di sicurezza in AWS.

mappatura del flusso di valore dello sviluppo (DVSM)

Un processo utilizzato per identificare e dare priorità ai vincoli che influiscono negativamente sulla velocità e sulla qualità nel ciclo di vita dello sviluppo del software. DVSM estende il processo di

mappatura del flusso di valore originariamente progettato per pratiche di produzione snella. Si concentra sulle fasi e sui team necessari per creare e trasferire valore attraverso il processo di sviluppo del software.

gemello digitale

Una rappresentazione virtuale di un sistema reale, ad esempio un edificio, una fabbrica, un'attrezzatura industriale o una linea di produzione. I gemelli digitali supportano la manutenzione predittiva, il monitoraggio remoto e l'ottimizzazione della produzione.

tabella delle dimensioni

In uno schema a stella, una tabella più piccola che contiene gli attributi dei dati quantitativi in una tabella dei fatti. Gli attributi della tabella delle dimensioni sono in genere campi di testo o numeri discreti che si comportano come testo. Questi attributi vengono comunemente utilizzati per il vincolo delle query, il filtraggio e l'etichettatura dei set di risultati.

disastro

Un evento che impedisce a un carico di lavoro o a un sistema di raggiungere gli obiettivi aziendali nella sua sede principale di implementazione. Questi eventi possono essere disastri naturali, guasti tecnici o il risultato di azioni umane, come errori di configurazione involontari o attacchi di malware.

disaster recovery (DR)

La strategia e il processo utilizzati per ridurre al minimo i tempi di inattività e la perdita di dati causati da un disastro. Per ulteriori informazioni, consulta Disaster Recovery of Workloads su AWS: Recovery in the Cloud in the AWS Well-Architected Framework.

DML

Vedi linguaggio di manipolazione del database.

progettazione basata sul dominio

Un approccio allo sviluppo di un sistema software complesso collegandone i componenti a domini in evoluzione, o obiettivi aziendali principali, perseguiti da ciascun componente. Questo concetto è stato introdotto da Eric Evans nel suo libro,Domain-Driven Design: Tackling Complexity in the Heart of Software (Boston: Addison-Wesley Professional, 2003). Per informazioni su come utilizzare la progettazione basata sul dominio con il modello del fico strangolatore (Strangler Fig), consulta la sezione Modernizzazione incrementale dei servizi Web Microsoft ASP.NET (ASMX) legacy utilizzando container e il Gateway Amazon API.

DOTT.

Vedi disaster recovery.

rilevamento della deriva

Tracciamento delle deviazioni da una configurazione di base. Ad esempio, puoi utilizzarlo AWS CloudFormation per rilevare la deriva nelle risorse di sistema oppure puoi usarlo AWS Control Tower per rilevare cambiamenti nella tua landing zone che potrebbero influire sulla conformità ai requisiti di governance.

DVSM

Vedi la mappatura del flusso di valore dello sviluppo.

E

EDA

Vedi analisi esplorativa dei dati.

edge computing

La tecnologia che aumenta la potenza di calcolo per i dispositivi intelligenti all'edge di una rete IoT. Rispetto al cloud computing, l'edge computing può ridurre la latenza di comunicazione e migliorare i tempi di risposta.

crittografia

Un processo di elaborazione che trasforma i dati in chiaro, leggibili dall'uomo, in testo cifrato.

chiave crittografica

Una stringa crittografica di bit randomizzati generata da un algoritmo di crittografia. Le chiavi possono variare di lunghezza e ogni chiave è progettata per essere imprevedibile e univoca.

endianità

L'ordine in cui i byte vengono archiviati nella memoria del computer. I sistemi big-endian memorizzano per primo il byte più importante. I sistemi little-endian memorizzano per primo il byte meno importante.

endpoint

Vedi service endpoint.

servizio endpoint

Un servizio che puoi ospitare in un cloud privato virtuale (VPC) da condividere con altri utenti. Puoi creare un servizio endpoint con AWS PrivateLink e concedere autorizzazioni ad altri Account AWS o a AWS Identity and Access Management (IAM) principali. Questi account o principali possono connettersi al servizio endpoint in privato creando endpoint VPC di interfaccia. Per ulteriori informazioni, consulta Creazione di un servizio endpoint nella documentazione di Amazon Virtual Private Cloud (Amazon VPC).

pianificazione delle risorse aziendali (ERP)

Un sistema che automatizza e gestisce i processi aziendali chiave (come contabilità, MES e gestione dei progetti) per un'azienda.

crittografia envelope

Il processo di crittografia di una chiave di crittografia con un'altra chiave di crittografia. Per ulteriori informazioni, vedete Envelope encryption nella documentazione AWS Key Management Service (AWS KMS).

ambiente

Un'istanza di un'applicazione in esecuzione. Di seguito sono riportati i tipi di ambiente più comuni nel cloud computing:

- ambiente di sviluppo: un'istanza di un'applicazione in esecuzione disponibile solo per il team principale responsabile della manutenzione dell'applicazione. Gli ambienti di sviluppo vengono utilizzati per testare le modifiche prima di promuoverle negli ambienti superiori. Questo tipo di ambiente viene talvolta definito ambiente di test.
- ambienti inferiori: tutti gli ambienti di sviluppo di un'applicazione, ad esempio quelli utilizzati per le build e i test iniziali.
- ambiente di produzione: un'istanza di un'applicazione in esecuzione a cui gli utenti finali possono accedere. In una pipeline CI/CD, l'ambiente di produzione è l'ultimo ambiente di implementazione.
- ambienti superiori: tutti gli ambienti a cui possono accedere utenti diversi dal team di sviluppo principale. Si può trattare di un ambiente di produzione, ambienti di preproduzione e ambienti per i test di accettazione da parte degli utenti.

epica

Nelle metodologie agili, categorie funzionali che aiutano a organizzare e dare priorità al lavoro. Le epiche forniscono una descrizione di alto livello dei requisiti e delle attività di implementazione.

Ad esempio, le epopee della sicurezza AWS CAF includono la gestione delle identità e degli accessi, i controlli investigativi, la sicurezza dell'infrastruttura, la protezione dei dati e la risposta agli incidenti. Per ulteriori informazioni sulle epiche, consulta la strategia di migrazione AWS , consulta la guida all'implementazione del programma.

ERP

Vedi la pianificazione delle risorse aziendali.

analisi esplorativa dei dati (EDA)

Il processo di analisi di un set di dati per comprenderne le caratteristiche principali. Si raccolgono o si aggregano dati e quindi si eseguono indagini iniziali per trovare modelli, rilevare anomalie e verificare ipotesi. L'EDA viene eseguita calcolando statistiche di riepilogo e creando visualizzazioni di dati.

F

tabella dei fatti

Il tavolo centrale in uno schema a stella. Memorizza dati quantitativi sulle operazioni aziendali. In genere, una tabella dei fatti contiene due tipi di colonne: quelle che contengono misure e quelle che contengono una chiave esterna per una tabella di dimensioni.

fallire velocemente

Una filosofia che utilizza test frequenti e incrementali per ridurre il ciclo di vita dello sviluppo. È una parte fondamentale di un approccio agile.

limite di isolamento dei guasti

Nel Cloud AWS, un limite come una zona di disponibilità Regione AWS, un piano di controllo o un piano dati che limita l'effetto di un errore e aiuta a migliorare la resilienza dei carichi di lavoro. Per ulteriori informazioni, consulta AWS Fault Isolation Boundaries.

ramo di funzionalità

Vedi filiale.

caratteristiche

I dati di input che usi per fare una previsione. Ad esempio, in un contesto di produzione, le caratteristiche potrebbero essere immagini acquisite periodicamente dalla linea di produzione.

importanza delle caratteristiche

Quanto è importante una caratteristica per le previsioni di un modello. Di solito viene espresso come punteggio numerico che può essere calcolato con varie tecniche, come Shapley Additive Explanations (SHAP) e gradienti integrati. Per ulteriori informazioni, vedere Interpretabilità del modello di machine learning con:AWS.

trasformazione delle funzionalità

Per ottimizzare i dati per il processo di machine learning, incluso l'arricchimento dei dati con fonti aggiuntive, il dimensionamento dei valori o l'estrazione di più set di informazioni da un singolo campo di dati. Ciò consente al modello di ML di trarre vantaggio dai dati. Ad esempio, se suddividi la data "2021-05-27 00:15:37" in "2021", "maggio", "giovedì" e "15", puoi aiutare l'algoritmo di apprendimento ad apprendere modelli sfumati associati a diversi componenti dei dati.

FGAC

Vedi il controllo granulare degli accessi.

controllo granulare degli accessi (FGAC)

L'uso di più condizioni per consentire o rifiutare una richiesta di accesso.

migrazione flash-cut

Un metodo di migrazione del database che utilizza la replica continua dei dati tramite l'acquisizione dei dati delle modifiche per migrare i dati nel più breve tempo possibile, anziché utilizzare un approccio graduale. L'obiettivo è ridurre al minimo i tempi di inattività.

G

blocco geografico

Vedi restrizioni geografiche.

limitazioni geografiche (blocco geografico)

In Amazon CloudFront, un'opzione per impedire agli utenti di determinati paesi di accedere alle distribuzioni di contenuti. Puoi utilizzare un elenco consentito o un elenco di blocco per specificare i paesi approvati e vietati. Per ulteriori informazioni, consulta Limitare la distribuzione geografica dei contenuti nella CloudFront documentazione.

Flusso di lavoro di GitFlow

Un approccio in cui gli ambienti inferiori e superiori utilizzano rami diversi in un repository di codice di origine. Il flusso di lavoro Gitflow è considerato obsoleto e il flusso di lavoro basato su trunk è l'approccio moderno e preferito.

strategia greenfield

L'assenza di infrastrutture esistenti in un nuovo ambiente. Quando si adotta una strategia greenfield per un'architettura di sistema, è possibile selezionare tutte le nuove tecnologie senza il vincolo della compatibilità con l'infrastruttura esistente, nota anche come brownfield. Per l'espansione dell'infrastruttura esistente, è possibile combinare strategie brownfield e greenfield.

guardrail

Una regola di livello elevato che consente di governare risorse, policy e conformità tra le unità organizzative (OU). I guardrail preventivi applicano le policy per garantire l'allineamento agli standard di conformità. Vengono implementati utilizzando le policy di controllo dei servizi e i limiti delle autorizzazioni IAM. I guardrail di rilevamento rilevano le violazioni delle policy e i problemi di conformità e generano avvisi per porvi rimedio. Sono implementati utilizzando Amazon AWS Config AWS Security Hub GuardDuty AWS Trusted Advisor, Amazon Inspector e controlli personalizzati AWS Lambda .

H

AH

Vedi disponibilità elevata.

migrazione di database eterogenea

Migrazione del database di origine in un database di destinazione che utilizza un motore di database diverso (ad esempio, da Oracle ad Amazon Aurora). La migrazione eterogenea fa in genere parte di uno sforzo di riprogettazione e la conversione dello schema può essere un'attività complessa. AWS offre AWS SCT che aiuta con le conversioni dello schema.

alta disponibilità (HA)

La capacità di un carico di lavoro di funzionare in modo continuo, senza intervento, in caso di sfide o disastri. I sistemi HA sono progettati per il failover automatico, fornire costantemente prestazioni di alta qualità e gestire carichi e guasti diversi con un impatto minimo sulle prestazioni.

modernizzazione storica

Un approccio utilizzato per modernizzare e aggiornare i sistemi di tecnologia operativa (OT) per soddisfare meglio le esigenze dell'industria manifatturiera. Uno storico è un tipo di database utilizzato per raccogliere e archiviare dati da varie fonti in una fabbrica.

migrazione di database omogenea

Migrazione del database di origine in un database di destinazione che condivide lo stesso motore di database (ad esempio, da Microsoft SQL Server ad Amazon RDS per SQL Server). La migrazione omogenea fa in genere parte di un'operazione di rehosting o ridefinizione della piattaforma. Per migrare lo schema è possibile utilizzare le utilità native del database.

dati caldi

Dati a cui si accede frequentemente, ad esempio dati in tempo reale o dati di traduzione recenti. Questi dati richiedono in genere un livello o una classe di storage ad alte prestazioni per fornire risposte rapide alle query.

hotfix

Una soluzione urgente per un problema critico in un ambiente di produzione. A causa della sua urgenza, un hotfix viene in genere creato al di fuori del tipico DevOps flusso di lavoro di rilascio.

periodo di hypercare

Subito dopo la conversione, il periodo di tempo in cui un team di migrazione gestisce e monitora le applicazioni migrate nel cloud per risolvere eventuali problemi. In genere, questo periodo dura da 1 a 4 giorni. Al termine del periodo di hypercare, il team addetto alla migrazione in genere trasferisce la responsabilità delle applicazioni al team addetto alle operazioni cloud.

I

IaC

Considera l'infrastruttura come codice.

Policy basata su identità

Una policy associata a uno o più principi IAM che definisce le relative autorizzazioni all'interno dell' Cloud AWS ambiente.

applicazione inattiva

Un'applicazione che prevede un uso di CPU e memoria medio compreso tra il 5% e il 20% in un periodo di 90 giorni. In un progetto di migrazione, è normale ritirare queste applicazioni o mantenerle on-premise.

IIoT

Vedi Industrial Internet of Things.

infrastruttura immutabile

Un modello che implementa una nuova infrastruttura per i carichi di lavoro di produzione anziché aggiornare, applicare patch o modificare l'infrastruttura esistente. Le infrastrutture immutabili sono intrinsecamente più coerenti, affidabili e prevedibili delle infrastrutture mutabili. Per ulteriori informazioni, consulta la best practice Deploy using immutable infrastructure in Well-Architected AWS Framework.

VPC in ingresso (ingress)

In un'architettura AWS multi-account, un VPC che accetta, ispeziona e indirizza le connessioni di rete dall'esterno di un'applicazione. Nel documento Architettura di riferimento per la sicurezza di AWS si consiglia di configurare l'account di rete con VPC in entrata, in uscita e di ispezione per proteggere l'interfaccia bidirezionale tra l'applicazione e Internet in generale.

migrazione incrementale

Una strategia di conversione in cui si esegue la migrazione dell'applicazione in piccole parti anziché eseguire una conversione singola e completa. Ad esempio, inizialmente potresti spostare solo alcuni microservizi o utenti nel nuovo sistema. Dopo aver verificato che tutto funzioni correttamente, puoi spostare in modo incrementale microservizi o utenti aggiuntivi fino alla disattivazione del sistema legacy. Questa strategia riduce i rischi associati alle migrazioni di grandi dimensioni.

Industria 4.0

Un termine introdotto da Klaus Schwab nel 2016 per riferirsi alla modernizzazione dei processi di produzione attraverso progressi in termini di connettività, dati in tempo reale, automazione, analisi e AI/ML.

infrastruttura

Tutte le risorse e gli asset contenuti nell'ambiente di un'applicazione.

infrastruttura come codice (IaC)

Il processo di provisioning e gestione dell'infrastruttura di un'applicazione tramite un insieme di file di configurazione. Il processo IaC è progettato per aiutarti a centralizzare la gestione dell'infrastruttura, a standardizzare le risorse e a dimensionare rapidamente, in modo che i nuovi ambienti siano ripetibili, affidabili e coerenti.

Internet delle cose industriale (IIoT)

L'uso di sensori e dispositivi connessi a Internet nei settori industriali, come quello manifatturiero, energetico, automobilistico, sanitario, delle scienze della vita e dell'agricoltura. Per ulteriori informazioni, consulta Creazione di una strategia di trasformazione digitale dell'Internet delle cose industriale (IIoT).

VPC di ispezione

In un'architettura AWS multi-account, un VPC centralizzato che gestisce le ispezioni del traffico di rete tra VPC (uguali o diversi Regioni AWS), Internet e reti locali. Nel documento Architettura di riferimento per la sicurezza di AWS si consiglia di configurare l'account di rete con VPC in entrata, in uscita e di ispezione per proteggere l'interfaccia bidirezionale tra l'applicazione e Internet in generale.

Internet of Things (IoT)

La rete di oggetti fisici connessi con sensori o processori incorporati che comunicano con altri dispositivi e sistemi tramite Internet o una rete di comunicazione locale. Per ulteriori informazioni, consulta Cos'è l'IoT?

interpretabilità

Una caratteristica di un modello di machine learning che descrive il grado in cui un essere umano è in grado di comprendere in che modo le previsioni del modello dipendono dai suoi input. Per ulteriori informazioni, consulta la sezione Interpretabilità dei modelli di machine learning con AWS.

IoT

Vedi Internet of Things.

libreria di informazioni IT (ITIL)

Una serie di best practice per offrire servizi IT e allinearli ai requisiti aziendali. ITIL fornisce le basi per ITSM.

gestione dei servizi IT (ITSM)

Attività associate alla progettazione, implementazione, gestione e supporto dei servizi IT per un'organizzazione. Per informazioni sull'integrazione delle operazioni cloud con gli strumenti ITSM, consulta la guida all'integrazione delle operazioni.

ITIL

Vedi la libreria di informazioni IT.

ITSM

Vedi Gestione dei servizi IT.

L

controllo degli accessi basato su etichette (LBAC)

Un'implementazione del controllo di accesso obbligatorio (MAC) in cui agli utenti e ai dati stessi viene assegnato esplicitamente un valore di etichetta di sicurezza. L'intersezione tra l'etichetta di sicurezza utente e l'etichetta di sicurezza dei dati determina quali righe e colonne possono essere visualizzate dall'utente.

zona di destinazione

Una landing zone è un AWS ambiente multi-account ben progettato, scalabile e sicuro. Questo è un punto di partenza dal quale le organizzazioni possono avviare e distribuire rapidamente carichi di lavoro e applicazioni con fiducia nel loro ambiente di sicurezza e infrastruttura. Per ulteriori informazioni sulle zone di destinazione, consulta la sezione Configurazione di un ambiente AWS multi-account sicuro e scalabile.

migrazione su larga scala

Una migrazione di 300 o più server.

BIANCO

Vedi controllo degli accessi basato su etichette.

Privilegio minimo

La best practice di sicurezza per la concessione delle autorizzazioni minime richieste per eseguire un'attività. Per ulteriori informazioni, consulta Applicazione delle autorizzazioni del privilegio minimo nella documentazione di IAM.

eseguire il rehosting (lift and shift)

Vedi 7 R.

sistema little-endian

Un sistema che memorizza per primo il byte meno importante. Vedi anche endianità.

ambienti inferiori

Vedi ambiente.

M

machine learning (ML)

Un tipo di intelligenza artificiale che utilizza algoritmi e tecniche per il riconoscimento e l'apprendimento di schemi. Il machine learning analizza e apprende dai dati registrati, come i dati dell'Internet delle cose (IoT), per generare un modello statistico basato su modelli. Per ulteriori informazioni, consulta la sezione Machine learning.

ramo principale

Vedi filiale.

malware

Software progettato per compromettere la sicurezza o la privacy del computer. Il malware potrebbe interrompere i sistemi informatici, divulgare informazioni sensibili o ottenere accessi non autorizzati. Esempi di malware includono virus, worm, ransomware, trojan horse, spyware e keylogger.

servizi gestiti

AWS servizi per cui AWS gestisce il livello di infrastruttura, il sistema operativo e le piattaforme e si accede agli endpoint per archiviare e recuperare i dati. Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3) e Amazon DynamoDB sono esempi di servizi gestiti. Questi sono noti anche come servizi astratti.

sistema di esecuzione della produzione (MES)

Un sistema software per tracciare, monitorare, documentare e controllare i processi di produzione che convertono le materie prime in prodotti finiti in officina.

MAP

Vedi Migration Acceleration Program.

meccanismo

Un processo completo in cui si crea uno strumento, si promuove l'adozione dello strumento e quindi si esaminano i risultati per apportare le modifiche. Un meccanismo è un ciclo che si rafforza e si migliora man mano che funziona. Per ulteriori informazioni, consulta Creazione di meccanismi nel AWS Well-Architected Framework.

account membro

Tutti gli account Account AWS diversi dall'account di gestione che fanno parte di un'organizzazione in. AWS Organizations Un account può essere membro di una sola organizzazione alla volta.

MEH

Vedi sistema di esecuzione della produzione.

Message Queuing Telemetry Transport (MQTT)

Un protocollo di comunicazione machine-to-machine (M2M) leggero, basato sul modello di pubblicazione/sottoscrizione, per dispositivi IoT con risorse limitate.

microservizio

Un piccolo servizio indipendente che comunica tramite API ben definite ed è in genere di proprietà di piccoli team autonomi. Ad esempio, un sistema assicurativo potrebbe includere microservizi che si riferiscono a funzionalità aziendali, come vendite o marketing, o sottodomini, come acquisti, reclami o analisi. I vantaggi dei microservizi includono agilità, dimensionamento flessibile, facilità di implementazione, codice riutilizzabile e resilienza. Per ulteriori informazioni, consulta Integrazione dei microservizi utilizzando servizi serverless. AWS

architettura di microservizi

Un approccio alla creazione di un'applicazione con componenti indipendenti che eseguono ogni processo applicativo come microservizio. Questi microservizi comunicano tramite un'interfaccia ben definita utilizzando API leggere. Ogni microservizio in questa architettura può essere aggiornato, distribuito e dimensionato per soddisfare la richiesta di funzioni specifiche di un'applicazione. Per ulteriori informazioni, vedere Implementazione dei microservizi su. AWS

Programma di accelerazione della migrazione (MAP)

Un AWS programma che fornisce consulenza, supporto, formazione e servizi per aiutare le organizzazioni a costruire una solida base operativa per il passaggio al cloud e per contribuire a compensare il costo iniziale delle migrazioni. MAP include una metodologia di migrazione per eseguire le migrazioni precedenti in modo metodico e un set di strumenti per automatizzare e accelerare gli scenari di migrazione comuni.

migrazione su larga scala

Il processo di trasferimento della maggior parte del portfolio di applicazioni sul cloud avviene a ondate, con più applicazioni trasferite a una velocità maggiore in ogni ondata. Questa fase utilizza le migliori pratiche e le lezioni apprese nelle fasi precedenti per implementare una fabbrica di migrazione di team, strumenti e processi per semplificare la migrazione dei carichi di lavoro attraverso l'automazione e la distribuzione agile. Questa è la terza fase della strategia di migrazione AWS.

fabbrica di migrazione

Team interfunzionali che semplificano la migrazione dei carichi di lavoro attraverso approcci automatizzati e agili. I team di Migration Factory includono in genere operazioni, analisti e proprietari aziendali, ingegneri addetti alla migrazione, sviluppatori e DevOps professionisti che lavorano nell'ambito degli sprint. Tra il 20% e il 50% di un portfolio di applicazioni aziendali è costituito da schemi ripetuti che possono essere ottimizzati con un approccio di fabbrica. Per ulteriori informazioni, consulta la discussione sulle fabbriche di migrazione e la Guida alla fabbrica di migrazione al cloud in questo set di contenuti.

metadati di migrazione

Le informazioni sull'applicazione e sul server necessarie per completare la migrazione. Ogni modello di migrazione richiede un set diverso di metadati di migrazione. Esempi di metadati di migrazione includono la sottorete, il gruppo di sicurezza e l'account di destinazione. AWS

modello di migrazione

Un'attività di migrazione ripetibile che descrive in dettaglio la strategia di migrazione, la destinazione della migrazione e l'applicazione o il servizio di migrazione utilizzati. Esempio: riorganizza la migrazione su Amazon EC2 AWS con Application Migration Service.

Valutazione del portfolio di migrazione (MPA)

Uno strumento online che fornisce informazioni per la convalida del business case per la migrazione a. Cloud AWS MPA offre una valutazione dettagliata del portfolio (dimensionamento

corretto dei server, prezzi, confronto del TCO, analisi dei costi di migrazione) e pianificazione della migrazione (analisi e raccolta dei dati delle applicazioni, raggruppamento delle applicazioni, prioritizzazione delle migrazioni e pianificazione delle ondate). Lo strumento MPA (richiede l'accesso) è disponibile gratuitamente per tutti i AWS consulenti e i consulenti dei partner APN.

valutazione della preparazione alla migrazione (MRA)

Il processo di acquisizione di informazioni sullo stato di preparazione al cloud di un'organizzazione, l'identificazione dei punti di forza e di debolezza e la creazione di un piano d'azione per colmare le lacune identificate, utilizzando il CAF. AWS Per ulteriori informazioni, consulta la guida di preparazione alla migrazione. MRA è la prima fase della strategia di migrazione AWS.

strategia di migrazione

L'approccio utilizzato per migrare un carico di lavoro verso. Cloud AWS Per ulteriori informazioni, consulta la voce 7 R in questo glossario e consulta Mobilita la tua organizzazione per accelerare le migrazioni su larga scala.

ML

Vedi machine learning.

modernizzazione

Trasformazione di un'applicazione obsoleta (legacy o monolitica) e della relativa infrastruttura in un sistema agile, elastico e altamente disponibile nel cloud per ridurre i costi, aumentare l'efficienza e sfruttare le innovazioni. Per ulteriori informazioni, vedere Strategia per la modernizzazione delle applicazioni in. Cloud AWS

valutazione della preparazione alla modernizzazione

Una valutazione che aiuta a determinare la preparazione alla modernizzazione delle applicazioni di un'organizzazione, identifica vantaggi, rischi e dipendenze e determina in che misura l'organizzazione può supportare lo stato futuro di tali applicazioni. Il risultato della valutazione è uno schema dell'architettura di destinazione, una tabella di marcia che descrive in dettaglio le fasi di sviluppo e le tappe fondamentali del processo di modernizzazione e un piano d'azione per colmare le lacune identificate. Per ulteriori informazioni, vedere Valutazione della preparazione alla modernizzazione per le applicazioni in. Cloud AWS

applicazioni monolitiche (monoliti)

Applicazioni eseguite come un unico servizio con processi strettamente collegati. Le applicazioni monolitiche presentano diversi inconvenienti. Se una funzionalità dell'applicazione registra un

picco di domanda, l'intera architettura deve essere dimensionata. L'aggiunta o il miglioramento delle funzionalità di un'applicazione monolitica diventa inoltre più complessa man mano che la base di codice cresce. Per risolvere questi problemi, puoi utilizzare un'architettura di microservizi. Per ulteriori informazioni, consulta la sezione Scomposizione dei monoliti in microservizi.

MAPPA

Vedi Migration Portfolio Assessment.

MQTT

Vedi Message Queuing Telemetry Transport.

classificazione multiclasse

Un processo che aiuta a generare previsioni per più classi (prevedendo uno o più di due risultati). Ad esempio, un modello di machine learning potrebbe chiedere "Questo prodotto è un libro, un'auto o un telefono?" oppure "Quale categoria di prodotti è più interessante per questo cliente?"

infrastruttura mutabile

Un modello che aggiorna e modifica l'infrastruttura esistente per i carichi di lavoro di produzione. Per migliorare la coerenza, l'affidabilità e la prevedibilità, il AWS Well-Architected Framework consiglia l'uso di un'infrastruttura immutabile come best practice.

O

OAC

Vedi Origin Access Control.

QUERCIA

Vedi Origin Access Identity.

OCM

Vedi gestione delle modifiche organizzative.

migrazione offline

Un metodo di migrazione in cui il carico di lavoro di origine viene eliminato durante il processo di migrazione. Questo metodo prevede tempi di inattività prolungati e viene in genere utilizzato per carichi di lavoro piccoli e non critici.

OI

Vedi l'integrazione delle operazioni.

OLA

Vedi accordo a livello operativo.

migrazione online

Un metodo di migrazione in cui il carico di lavoro di origine viene copiato sul sistema di destinazione senza essere messo offline. Le applicazioni connesse al carico di lavoro possono continuare a funzionare durante la migrazione. Questo metodo comporta tempi di inattività pari a zero o comunque minimi e viene in genere utilizzato per carichi di lavoro di produzione critici.

OPC-UA

Vedi Open Process Communications - Unified Architecture.

Comunicazioni a processo aperto - Architettura unificata (OPC-UA)

Un protocollo di comunicazione machine-to-machine (M2M) per l'automazione industriale. OPC-UA fornisce uno standard di interoperabilità con schemi di crittografia, autenticazione e autorizzazione dei dati.

accordo a livello operativo (OLA)

Un accordo che chiarisce quali sono gli impegni reciproci tra i gruppi IT funzionali, a supporto di un accordo sul livello di servizio (SLA).

revisione della prontezza operativa (ORR)

Un elenco di domande e best practice associate che aiutano a comprendere, valutare, prevenire o ridurre la portata degli incidenti e dei possibili guasti. Per ulteriori informazioni, vedere Operational Readiness Reviews (ORR) nel Well-Architected AWS Framework.

tecnologia operativa (OT)

Sistemi hardware e software che interagiscono con l'ambiente fisico per controllare le operazioni, le apparecchiature e le infrastrutture industriali. Nella produzione, l'integrazione di sistemi OT e di tecnologia dell'informazione (IT) è un obiettivo chiave per le trasformazioni dell'Industria 4.0.

integrazione delle operazioni (OI)

Il processo di modernizzazione delle operazioni nel cloud, che prevede la pianificazione, l'automazione e l'integrazione della disponibilità. Per ulteriori informazioni, consulta la guida all'integrazione delle operazioni.

trail organizzativo

Un percorso creato da noi AWS CloudTrail che registra tutti gli eventi di un'organizzazione per tutti Account AWS . AWS Organizations Questo percorso viene creato in ogni Account AWS che fa parte dell'organizzazione e tiene traccia dell'attività in ogni account. Per ulteriori informazioni, consulta Creazione di un percorso per un'organizzazione nella CloudTrail documentazione.

gestione del cambiamento organizzativo (OCM)

Un framework per la gestione di trasformazioni aziendali importanti e che comportano l'interruzione delle attività dal punto di vista delle persone, della cultura e della leadership. OCM aiuta le organizzazioni a prepararsi e passare a nuovi sistemi e strategie accelerando l'adozione del cambiamento, affrontando i problemi di transizione e promuovendo cambiamenti culturali e organizzativi. Nella strategia di AWS migrazione, questo framework si chiama accelerazione delle persone, a causa della velocità di cambiamento richiesta nei progetti di adozione del cloud. Per ulteriori informazioni, consultare la Guida OCM.

controllo dell'accesso all'origine (OAC)

In CloudFront, un'opzione avanzata per limitare l'accesso per proteggere i contenuti di Amazon Simple Storage Service (Amazon S3). OAC supporta tutti i bucket S3 in generale Regioni AWS, la crittografia lato server con AWS KMS (SSE-KMS) e le richieste dinamiche e dirette al bucket S3. `PUT` `DELETE`

identità di accesso origine (OAI)

Nel CloudFront, un'opzione per limitare l'accesso per proteggere i tuoi contenuti Amazon S3. Quando usi OAI, CloudFront crea un principale con cui Amazon S3 può autenticarsi. I principali autenticati possono accedere ai contenuti in un bucket S3 solo tramite una distribuzione specifica. CloudFront Vedi anche OAC, che fornisce un controllo degli accessi più granulare e avanzato.

O

Vedi la revisione della prontezza operativa.

- NON

Vedi la tecnologia operativa.

VPC in uscita (egress)

In un'architettura AWS multi-account, un VPC che gestisce le connessioni di rete avviate dall'interno di un'applicazione. Nel documento Architettura di riferimento per la sicurezza di AWS si consiglia di configurare l'account di rete con VPC in entrata, in uscita e di ispezione per proteggere l'interfaccia bidirezionale tra l'applicazione e Internet in generale.

P

limite delle autorizzazioni

Una policy di gestione IAM collegata ai principali IAM per impostare le autorizzazioni massime che l'utente o il ruolo possono avere. Per ulteriori informazioni, consulta Limiti delle autorizzazioni nella documentazione di IAM.

informazioni di identificazione personale (PII)

Informazioni che, se visualizzate direttamente o abbinate ad altri dati correlati, possono essere utilizzate per dedurre ragionevolmente l'identità di un individuo. Esempi di informazioni personali includono nomi, indirizzi e informazioni di contatto.

Informazioni che consentono l'identificazione personale degli utenti

Visualizza le informazioni di identificazione personale.

playbook

Una serie di passaggi predefiniti che raccolgono il lavoro associato alle migrazioni, come l'erogazione delle funzioni operative principali nel cloud. Un playbook può assumere la forma di script, runbook automatici o un riepilogo dei processi o dei passaggi necessari per gestire un ambiente modernizzato.

PLC

Vedi controllore logico programmabile.

PLM

Vedi la gestione del ciclo di vita del prodotto.

policy

Un oggetto in grado di definire le autorizzazioni (vedi politica basata sull'identità), specificare le condizioni di accesso (vedi politicabasata sulle risorse) o definire le autorizzazioni massime per tutti gli account di un'organizzazione in (vedi politica di controllo dei servizi). AWS Organizations

persistenza poliglotta

Scelta indipendente della tecnologia di archiviazione di dati di un microservizio in base ai modelli di accesso ai dati e ad altri requisiti. Se i microservizi utilizzano la stessa tecnologia di archiviazione di dati, possono incontrare problemi di implementazione o registrare prestazioni

scadenti. I microservizi vengono implementati più facilmente e ottengono prestazioni e scalabilità migliori se utilizzano l'archivio dati più adatto alle loro esigenze. Per ulteriori informazioni, consulta la sezione Abilitazione della persistenza dei dati nei microservizi.

valutazione del portfolio

Un processo di scoperta, analisi e definizione delle priorità del portfolio di applicazioni per pianificare la migrazione. Per ulteriori informazioni, consulta la pagina Valutazione della preparazione alla migrazione.

predicate

Una condizione di interrogazione che restituisce o, in genere, si trova in una clausola `WHERE` che restituisce `true` o `false`.

predicato pushdown

Una tecnica di ottimizzazione delle query del database che filtra i dati della query prima del trasferimento. Ciò riduce la quantità di dati che devono essere recuperati ed elaborati dal database relazionale e migliora le prestazioni delle query.

controllo preventivo

Un controllo di sicurezza progettato per impedire il verificarsi di un evento. Questi controlli sono la prima linea di difesa per impedire accessi non autorizzati o modifiche indesiderate alla rete. Per ulteriori informazioni, consulta Controlli preventivi in Implementazione dei controlli di sicurezza in AWS.

principale

Un'entità in AWS grado di eseguire azioni e accedere alle risorse. Questa entità è in genere un utente root per un Account AWS ruolo IAM o un utente. Per ulteriori informazioni, consulta Principali in Termini e concetti dei ruoli nella documentazione di IAM.

Privacy fin dalla progettazione

Un approccio all'ingegneria dei sistemi che tiene conto della privacy durante l'intero processo di progettazione.

zone ospitate private

Un container che contiene informazioni su come si desidera che Amazon Route 53 risponda alle query DNS per un dominio e i relativi sottodomini all'interno di uno o più VPC. Per ulteriori informazioni, consulta Utilizzo delle zone ospitate private nella documentazione di Route 53.

controllo proattivo

Un controllo di sicurezza progettato per impedire l'implementazione di risorse non conformi. Questi controlli analizzano le risorse prima del loro provisioning. Se la risorsa non è conforme al controllo, non viene fornita. Per ulteriori informazioni, consulta la guida di riferimento sui controlli nella AWS Control Tower documentazione e consulta Controlli proattivi in Implementazione dei controlli di sicurezza su. AWS

gestione del ciclo di vita del prodotto (PLM)

La gestione dei dati e dei processi di un prodotto durante l'intero ciclo di vita, dalla progettazione, sviluppo e lancio, attraverso la crescita e la maturità, fino al declino e alla rimozione.

Ambiente di produzione

Vedi ambiente.

controllore logico programmabile (PLC)

Nella produzione, un computer altamente affidabile e adattabile che monitora le macchine e automatizza i processi di produzione.

pseudonimizzazione

Il processo di sostituzione degli identificatori personali in un set di dati con valori segnaposto. La pseudonimizzazione può aiutare a proteggere la privacy personale. I dati pseudonimizzati sono ancora considerati dati personali.

pubblica/sottoscrivi (pub/sub)

Un pattern che consente comunicazioni asincrone tra microservizi per migliorare la scalabilità e la reattività. Ad esempio, in un MES basato su microservizi, un microservizio può pubblicare messaggi di eventi su un canale a cui altri microservizi possono abbonarsi. Il sistema può aggiungere nuovi microservizi senza modificare il servizio di pubblicazione.

Q

Piano di query

Una serie di passaggi, come le istruzioni, utilizzati per accedere ai dati in un sistema di database relazionale SQL.

regressione del piano di query

Quando un ottimizzatore del servizio di database sceglie un piano non ottimale rispetto a prima di una determinata modifica all'ambiente di database. Questo può essere causato da modifiche a statistiche, vincoli, impostazioni dell'ambiente, associazioni dei parametri di query e aggiornamenti al motore di database.

R

Matrice RACI

Vedi responsabile, responsabile, consultato, informato (RACI).

ransomware

Un software dannoso progettato per bloccare l'accesso a un sistema informatico o ai dati fino a quando non viene effettuato un pagamento.

Matrice RASCI

Vedi responsabile, responsabile, consultato, informato (RACI).

RCAC

Vedi controllo dell'accesso a righe e colonne.

replica di lettura

Una copia di un database utilizzata per scopi di sola lettura. È possibile indirizzare le query alla replica di lettura per ridurre il carico sul database principale.

riprogettare

Vedi 7 Rs.

obiettivo del punto di ripristino (RPO)

Il periodo di tempo massimo accettabile dall'ultimo punto di ripristino dei dati. Ciò determina quella che viene considerata una perdita di dati accettabile tra l'ultimo punto di ripristino e l'interruzione del servizio.

obiettivo del tempo di ripristino (RTO)

Il ritardo massimo accettabile tra l'interruzione del servizio e il ripristino del servizio.

rifattorizzare

Vedi 7 R.

Regione

Una raccolta di AWS risorse in un'area geografica. Ciascuna Regione AWS è isolata e indipendente dalle altre per fornire tolleranza agli errori, stabilità e resilienza. Per ulteriori informazioni, consulta Specificare cosa può utilizzare Regioni AWS il proprio account.

regressione

Una tecnica di ML che prevede un valore numerico. Ad esempio, per risolvere il problema "A che prezzo verrà venduta questa casa?" un modello di ML potrebbe utilizzare un modello di regressione lineare per prevedere il prezzo di vendita di una casa sulla base di dati noti sulla casa (ad esempio, la metratura).

riospitare

Vedi 7 R.

rilascio

In un processo di implementazione, l'atto di promuovere modifiche a un ambiente di produzione.

trasferisco

Vedi 7 Rs.

ripiattaforma

Vedi 7 Rs.

riacquisto

Vedi 7 Rs.

resilienza

La capacità di un'applicazione di resistere o ripristinare le interruzioni. L'elevata disponibilità e il disaster recovery sono considerazioni comuni quando si pianifica la resilienza in. Cloud AWSPer ulteriori informazioni, vedere Cloud AWS Resilience.

policy basata su risorse

Una policy associata a una risorsa, ad esempio un bucket Amazon S3, un endpoint o una chiave di crittografia. Questo tipo di policy specifica a quali principali è consentito l'accesso, le azioni supportate e qualsiasi altra condizione che deve essere soddisfatta.

matrice di assegnazione di responsabilità (RACI)

Una matrice che definisce i ruoli e le responsabilità di tutte le parti coinvolte nelle attività di migrazione e nelle operazioni cloud. Il nome della matrice deriva dai tipi di responsabilità definiti nella matrice: responsabile (R), responsabile (A), consultato (C) e informato (I). Il tipo di supporto (S) è facoltativo. Se includi il supporto, la matrice viene chiamata matrice RASCI e, se la escludi, viene chiamata matrice RACI.

controllo reattivo

Un controllo di sicurezza progettato per favorire la correzione di eventi avversi o deviazioni dalla baseline di sicurezza. Per ulteriori informazioni, consulta Controlli reattivi in Implementazione dei controlli di sicurezza in AWS.

retain

Vedi 7 R.

andare in pensione

Vedi 7 Rs.

rotazione

Processo di aggiornamento periodico di un segreto per rendere più difficile l'accesso alle credenziali da parte di un utente malintenzionato.

controllo dell'accesso a righe e colonne (RCAC)

L'uso di espressioni SQL di base e flessibili con regole di accesso definite. RCAC è costituito da autorizzazioni di riga e maschere di colonna.

RPO

Vedi l'obiettivo del punto di ripristino.

RTO

Vedi l'obiettivo del tempo di ripristino.

runbook

Un insieme di procedure manuali o automatizzate necessarie per eseguire un'attività specifica. In genere sono progettati per semplificare operazioni o procedure ripetitive con tassi di errore elevati.

S

SAML 2.0

Uno standard aperto utilizzato da molti provider di identità (IdPs). Questa funzionalità abilita il single sign-on (SSO) federato, in modo che gli utenti possano accedere AWS Management Console o chiamare le operazioni AWS API senza che tu debba creare un utente in IAM per tutti i membri dell'organizzazione. Per ulteriori informazioni sulla federazione basata su SAML 2.0, consulta Informazioni sulla federazione basata su SAML 2.0 nella documentazione di IAM.

SCADA

Vedi controllo di supervisione e acquisizione dati.

SCP

Vedi la politica di controllo del servizio.

Secret

In AWS Secrets Manager, informazioni riservate o riservate, come una password o le credenziali utente, archiviate in forma crittografata. È costituito dal valore segreto e dai relativi metadati. Il valore segreto può essere binario, una stringa singola o più stringhe. Per ulteriori informazioni, consulta Cosa c'è in un segreto di Secrets Manager? nella documentazione di Secrets Manager.

controllo di sicurezza

Un guardrail tecnico o amministrativo che impedisce, rileva o riduce la capacità di un autore di minacce di sfruttare una vulnerabilità di sicurezza. Esistono quattro tipi principali di controlli di sicurezza: preventivi, investigativi, reattivi e proattivi.

rafforzamento della sicurezza

Il processo di riduzione della superficie di attacco per renderla più resistente agli attacchi. Può includere azioni come la rimozione di risorse che non sono più necessarie, l'implementazione di best practice di sicurezza che prevedono la concessione del privilegio minimo o la disattivazione di funzionalità non necessarie nei file di configurazione.

sistema di gestione delle informazioni e degli eventi di sicurezza (SIEM)

Strumenti e servizi che combinano sistemi di gestione delle informazioni di sicurezza (SIM) e sistemi di gestione degli eventi di sicurezza (SEM). Un sistema SIEM raccoglie, monitora e analizza i dati da server, reti, dispositivi e altre fonti per rilevare minacce e violazioni della sicurezza e generare avvisi.

automazione della risposta alla sicurezza

Un'azione predefinita e programmata progettata per rispondere o porre rimedio automaticamente a un evento di sicurezza. Queste automazioni fungono da controlli di sicurezza investigativi o reattivi che aiutano a implementare le migliori pratiche di sicurezza. AWS Esempi di azioni di risposta automatizzate includono la modifica di un gruppo di sicurezza VPC, l'applicazione di patch a un'istanza Amazon EC2 o la rotazione delle credenziali.

Crittografia lato server

Crittografia dei dati a destinazione, da parte di chi li riceve. AWS servizio

Policy di controllo dei servizi (SCP)

Una policy che fornisce il controllo centralizzato sulle autorizzazioni per tutti gli account di un'organizzazione in AWS Organizations. Le SCP definiscono i guardrail o fissano i limiti alle azioni che un amministratore può delegare a utenti o ruoli. Puoi utilizzare le SCP come elenchi consentiti o elenchi di rifiuto, per specificare quali servizi o azioni sono consentiti o proibiti. Per ulteriori informazioni, consulta le politiche di controllo del servizio nella AWS Organizations documentazione.

endpoint del servizio

L'URL del punto di ingresso per un AWS servizio. Puoi utilizzare l'endpoint per connetterti a livello di programmazione al servizio di destinazione. Per ulteriori informazioni, consulta Endpoint del AWS servizio nei Riferimenti generali di AWS.

accordo sul livello di servizio (SLA)

Un accordo che chiarisce ciò che un team IT promette di offrire ai propri clienti, ad esempio l'operatività e le prestazioni del servizio.

indicatore del livello di servizio (SLI)

Misurazione di un aspetto prestazionale di un servizio, ad esempio il tasso di errore, la disponibilità o la velocità effettiva.

obiettivo a livello di servizio (SLO)

Una metrica target che rappresenta lo stato di un servizio, misurato da un indicatore del livello di servizio.

Modello di responsabilità condivisa

Un modello che descrive la responsabilità condivisa AWS per la sicurezza e la conformità del cloud. AWS è responsabile della sicurezza del cloud, mentre tu sei responsabile della sicurezza nel cloud. Per ulteriori informazioni, consulta Modello di responsabilità condivisa.

SIEM

Vedi il sistema di gestione delle informazioni e degli eventi sulla sicurezza.

punto di errore singolo (SPOF)

Un guasto in un singolo componente critico di un'applicazione che può disturbare il sistema.

SLAM

Vedi il contratto sul livello di servizio.

SLI

Vedi l'indicatore del livello di servizio.

LENTA

Vedi obiettivo del livello di servizio.

split-and-seed modello

Un modello per dimensionare e accelerare i progetti di modernizzazione. Man mano che vengono definite nuove funzionalità e versioni dei prodotti, il team principale si divide per creare nuovi team di prodotto. Questo aiuta a dimensionare le capacità e i servizi dell'organizzazione, migliora la produttività degli sviluppatori e supporta una rapida innovazione. Per ulteriori informazioni, vedere Approccio graduale alla modernizzazione delle applicazioni in. Cloud AWS

SPOF

Vedi punto di errore singolo.

schema a stella

Una struttura organizzativa di database che utilizza un'unica tabella dei fatti di grandi dimensioni per archiviare i dati transazionali o misurati e utilizza una o più tabelle dimensionali più piccole per memorizzare gli attributi dei dati. Questa struttura è progettata per l'uso in un data warehouse o per scopi di business intelligence.

modello del fico strangolatore

Un approccio alla modernizzazione dei sistemi monolitici mediante la riscrittura e la sostituzione incrementali delle funzionalità del sistema fino alla disattivazione del sistema legacy. Questo modello utilizza l'analogia di una pianta di fico che cresce fino a diventare un albero robusto e alla fine annienta e sostituisce il suo ospite. Il modello è stato introdotto da Martin Fowler come metodo per gestire il rischio durante la riscrittura di sistemi monolitici. Per un esempio di come applicare questo modello, consulta Modernizzazione incrementale dei servizi Web legacy di Microsoft ASP.NET (ASMX) mediante container e Gateway Amazon API.

sottorete

Un intervallo di indirizzi IP nel VPC. Una sottorete deve risiedere in una singola zona di disponibilità.

controllo di supervisione e acquisizione dati (SCADA)

Nella produzione, un sistema che utilizza hardware e software per monitorare gli asset fisici e le operazioni di produzione.

crittografia simmetrica

Un algoritmo di crittografia che utilizza la stessa chiave per crittografare e decrittografare i dati.

test sintetici

Test di un sistema in modo da simulare le interazioni degli utenti per rilevare potenziali problemi o monitorare le prestazioni. Puoi usare Amazon CloudWatch Synthetics per creare questi test.

T

tags

Coppie chiave-valore che fungono da metadati per l'organizzazione delle risorse. AWS Con i tag è possibile a gestire, identificare, organizzare, cercare e filtrare le risorse. Per ulteriori informazioni, consulta Tagging delle risorse AWS.

variabile di destinazione

Il valore che stai cercando di prevedere nel machine learning supervisionato. Questo è indicato anche come variabile di risultato. Ad esempio, in un ambiente di produzione la variabile di destinazione potrebbe essere un difetto del prodotto.

elenco di attività

Uno strumento che viene utilizzato per tenere traccia dei progressi tramite un runbook. Un elenco di attività contiene una panoramica del runbook e un elenco di attività generali da completare. Per ogni attività generale, include la quantità stimata di tempo richiesta, il proprietario e lo stato di avanzamento.

Ambiente di test

Vedi ambiente.

training

Fornire dati da cui trarre ispirazione dal modello di machine learning. I dati di training devono contenere la risposta corretta. L'algoritmo di apprendimento trova nei dati di addestramento i pattern che mappano gli attributi dei dati di input al target (la risposta che si desidera prevedere). Produce un modello di ML che acquisisce questi modelli. Puoi quindi utilizzare il modello di ML per creare previsioni su nuovi dati di cui non si conosce il target.

Transit Gateway

Un hub di transito di rete che è possibile utilizzare per collegare i VPC e le reti on-premise. Per ulteriori informazioni, consulta Cos'è un gateway di transito nella AWS Transit Gateway documentazione.

flusso di lavoro basato su trunk

Un approccio in cui gli sviluppatori creano e testano le funzionalità localmente in un ramo di funzionalità e quindi uniscono tali modifiche al ramo principale. Il ramo principale viene quindi integrato negli ambienti di sviluppo, preproduzione e produzione, in sequenza.

Accesso attendibile

Concessione delle autorizzazioni a un servizio specificato dall'utente per eseguire attività all'interno dell'organizzazione AWS Organizations e nei suoi account per conto dell'utente. Il servizio attendibile crea un ruolo collegato al servizio in ogni account, quando tale ruolo è necessario, per eseguire attività di gestione per conto dell'utente. Per ulteriori informazioni, consulta Utilizzo AWS Organizations con altri AWS servizi nella AWS Organizations documentazione.

regolazione

Modificare alcuni aspetti del processo di training per migliorare la precisione del modello di ML. Ad esempio, puoi addestrare il modello di ML generando un set di etichette, aggiungendo etichette e quindi ripetendo questi passaggi più volte con impostazioni diverse per ottimizzare il modello.

team da due pizze

Una piccola DevOps squadra che puoi sfamare con due pizze. Un team composto da due persone garantisce la migliore opportunità possibile di collaborazione nello sviluppo del software.

U

incertezza

Un concetto che si riferisce a informazioni imprecise, incomplete o sconosciute che possono minare l'affidabilità dei modelli di machine learning predittivi. Esistono due tipi di incertezza: l'incertezza epistemica, che è causata da dati limitati e incompleti, mentre l'incertezza aleatoria è causata dal rumore e dalla casualità insiti nei dati. Per ulteriori informazioni, consulta la guida Quantificazione dell'incertezza nei sistemi di deep learning.

compiti indifferenziati

Conosciuto anche come sollevamento di carichi pesanti, è un lavoro necessario per creare e far funzionare un'applicazione, ma che non apporta valore diretto all'utente finale né offre vantaggi competitivi. Esempi di attività indifferenziate includono l'approvvigionamento, la manutenzione e la pianificazione della capacità.

ambienti superiori

Vedi ambiente.

V

vacuum

Un'operazione di manutenzione del database che prevede la pulizia dopo aggiornamenti incrementali per recuperare lo spazio di archiviazione e migliorare le prestazioni.

controllo delle versioni

Processi e strumenti che tengono traccia delle modifiche, ad esempio le modifiche al codice di origine in un repository.

Peering VPC

Una connessione tra due VPC che consente di instradare il traffico tramite indirizzi IP privati. Per ulteriori informazioni, consulta Che cos'è il peering VPC? nella documentazione di Amazon VPC.

vulnerabilità

Un difetto software o hardware che compromette la sicurezza del sistema.

W

cache calda

Una cache del buffer che contiene dati correnti e pertinenti a cui si accede frequentemente. L'istanza di database può leggere dalla cache del buffer, il che richiede meno tempo rispetto alla lettura dalla memoria dal disco principale.

dati caldi

Dati a cui si accede raramente. Quando si eseguono interrogazioni di questo tipo di dati, in genere sono accettabili interrogazioni moderatamente lente.

funzione finestra

Una funzione SQL che esegue un calcolo su un gruppo di righe che si riferiscono in qualche modo al record corrente. Le funzioni della finestra sono utili per l'elaborazione di attività, come il calcolo di una media mobile o l'accesso al valore delle righe in base alla posizione relativa della riga corrente.

Carico di lavoro

Una raccolta di risorse e codice che fornisce valore aziendale, ad esempio un'applicazione rivolta ai clienti o un processo back-end.

flusso di lavoro

Gruppi funzionali in un progetto di migrazione responsabili di una serie specifica di attività. Ogni flusso di lavoro è indipendente ma supporta gli altri flussi di lavoro del progetto. Ad esempio, il flusso di lavoro del portfolio è responsabile della definizione delle priorità delle applicazioni, della pianificazione delle ondate e della raccolta dei metadati di migrazione. Il flusso di lavoro del portfolio fornisce queste risorse al flusso di lavoro di migrazione, che quindi migra i server e le applicazioni.

VERME

Vedi scrivere una volta, leggere molti.

WQF

Vedi AWS Workload Qualification Framework.

scrivi una volta, leggi molte (WORM)

Un modello di storage che scrive i dati una sola volta e ne impedisce l'eliminazione o la modifica. Gli utenti autorizzati possono leggere i dati tutte le volte che è necessario, ma non possono modificarli. Questa infrastruttura di archiviazione dei dati è considerata immutabile.

Z

exploit zero-day

Un attacco, in genere malware, che sfrutta una vulnerabilità zero-day.

vulnerabilità zero-day

Un difetto o una vulnerabilità assoluta in un sistema di produzione. Gli autori delle minacce possono utilizzare questo tipo di vulnerabilità per attaccare il sistema. Gli sviluppatori vengono spesso a conoscenza della vulnerabilità causata dall'attacco.

applicazione zombie

Un'applicazione che prevede un utilizzo CPU e memoria inferiore al 5%. In un progetto di migrazione, è normale ritirare queste applicazioni.

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.