Uso de Apache Iceberg en AWS

AWS Guía prescriptiva

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

AWS Guía prescriptiva: Uso de Apache Iceberg en AWS

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

Las marcas comerciales y la imagen comercial de Amazon no se pueden utilizar en relación con ningún producto o servicio que no sea de Amazon, de ninguna manera que pueda causar confusión entre los clientes y que menosprecie o desacredite a Amazon. Todas las demás marcas registradas que no son propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, conectados o patrocinados por Amazon.

Table of Contents

Uso de Apache Iceberg en AWS

Amazon Web Services (colaboradores)

Noviembre de 2025 (historial del documento)

Apache Iceberg es un formato de tabla de código abierto que simplifica la administración de tablas y mejora el rendimiento. AWS Los servicios de análisis como Amazon EMR AWS Glue, Amazon Athena y Amazon Redshift incluyen soporte nativo para Iceberg, de modo que puede crear fácilmente lagos de datos transaccionales sobre Amazon Simple Storage Service (Amazon S3). AWS

Además, la próxima generación de Amazon SageMaker se basa en una arquitectura abierta que unifica el acceso a los datos a través de lagos de AWS datos, almacenes de datos y fuentes federadas y de terceros. The Lakehouse es totalmente compatible con Iceberg y le brinda la flexibilidad de acceder a los datos y consultarlos in situ mediante la API REST de Iceberg.

Esta guía técnica proporciona orientación sobre cómo empezar a utilizar Iceberg en distintos ámbitos e incluye las mejores prácticas y recomendaciones para utilizar Iceberg a gran escala y Servicios de AWS, al mismo tiempo, optimizar los costes y el AWS rendimiento.

Tanto si acaba de empezar con Iceberg como si es un usuario experimentado que busca optimizar sus cargas de trabajo actuales de Iceberg AWS, esta guía ofrece información valiosa para cada etapa de su proyecto

En esta guía:

Lagos de datos modernos

Casos de uso avanzados en lagos de datos modernos

La evolución del almacenamiento de datos ha pasado de las bases de datos a los almacenes y lagos de datos, donde cada tecnología responde a requisitos empresariales y de datos únicos. Las bases de datos tradicionales se destacaban en el manejo de datos estructurados y cargas de trabajo transaccionales, pero se enfrentaban a desafíos de rendimiento a medida que aumentaban los volúmenes de datos. Los almacenes de datos surgieron para abordar los problemas de rendimiento y escalabilidad, pero al igual que las bases de datos, dependían de formatos patentados dentro de sistemas integrados verticalmente.

Los lagos de datos ofrecen una de las mejores opciones para almacenar datos en términos de costo, escalabilidad y flexibilidad. Puede utilizar un lago de datos para conservar grandes volúmenes de datos estructurados y no estructurados a un bajo coste y utilizar estos datos para distintos tipos de cargas de trabajo analíticas, desde la elaboración de informes de inteligencia empresarial hasta el procesamiento de macrodatos, la analítica en tiempo real, el aprendizaje automático y la inteligencia artificial generativa (IA), para ayudar a tomar mejores decisiones.

A pesar de estas ventajas, los lagos de datos no se diseñaron inicialmente con capacidades similares a las de las bases de datos. Un lago de datos no admite la semántica de procesamiento de atomicidad, coherencia, aislamiento y durabilidad (ACID), que podría necesitar para optimizar y administrar sus datos de manera eficaz a escala para cientos o miles de usuarios mediante el uso de muchas tecnologías diferentes. Los lagos de datos no ofrecen soporte nativo para las siguientes funciones:

- Realizar actualizaciones y eliminaciones eficientes a nivel de registros a medida que los datos cambian en su empresa
- Gestione el rendimiento de las consultas a medida que las tablas crecen hasta convertirse en millones de archivos y cientos de miles de particiones
- Garantizar la coherencia de los datos entre varios escritores y lectores simultáneos
- Prevenir la corrupción de los datos cuando las operaciones de escritura fallan a mitad de la operación
- Los esquemas de tablas evolucionan a lo largo del tiempo sin reescribir (parcialmente) los conjuntos de datos

Estos desafíos se han vuelto particularmente frecuentes en casos de uso, como la gestión de la captura de datos modificados (CDC) o los casos de uso relacionados con la privacidad, la eliminación de datos y la ingesta de datos en streaming, lo que puede dar lugar a que las tablas no sean óptimas.

Los lagos de datos que utilizan las tablas tradicionales con formato HIVE solo admiten operaciones de escritura para archivos completos. Esto hace que las actualizaciones y eliminaciones sean difíciles de implementar, además de llevar mucho tiempo y ser costosas. Además, los controles y garantías de simultaneidad que ofrecen los sistemas compatibles con ACID son necesarios para garantizar la integridad y la coherencia de los datos.

Estos desafíos dejan a los usuarios ante un dilema: elegir entre una plataforma totalmente integrada pero propia, u optar por un lago de datos autoconstruido y autónomo, independiente del proveedor, que requiera un mantenimiento y una migración constantes para aprovechar su valor potencial.

Para ayudar a superar estos desafíos, Iceberg proporciona una funcionalidad adicional similar a una base de datos que simplifica la sobrecarga de optimización y administración de los lagos de datos y, al mismo tiempo, admite el almacenamiento en sistemas rentables como Amazon S3.

Introducción a Apache Iceberg

Apache Iceberg es un formato de tabla de código abierto que proporciona funciones en las tablas de lagos de datos que anteriormente solo estaban disponibles en bases de datos o almacenes de datos. Está diseñado para ofrecer escalabilidad y rendimiento, y es ideal para administrar tablas de más de cientos de gigabytes. Algunas de las principales características de las mesas Iceberg son:

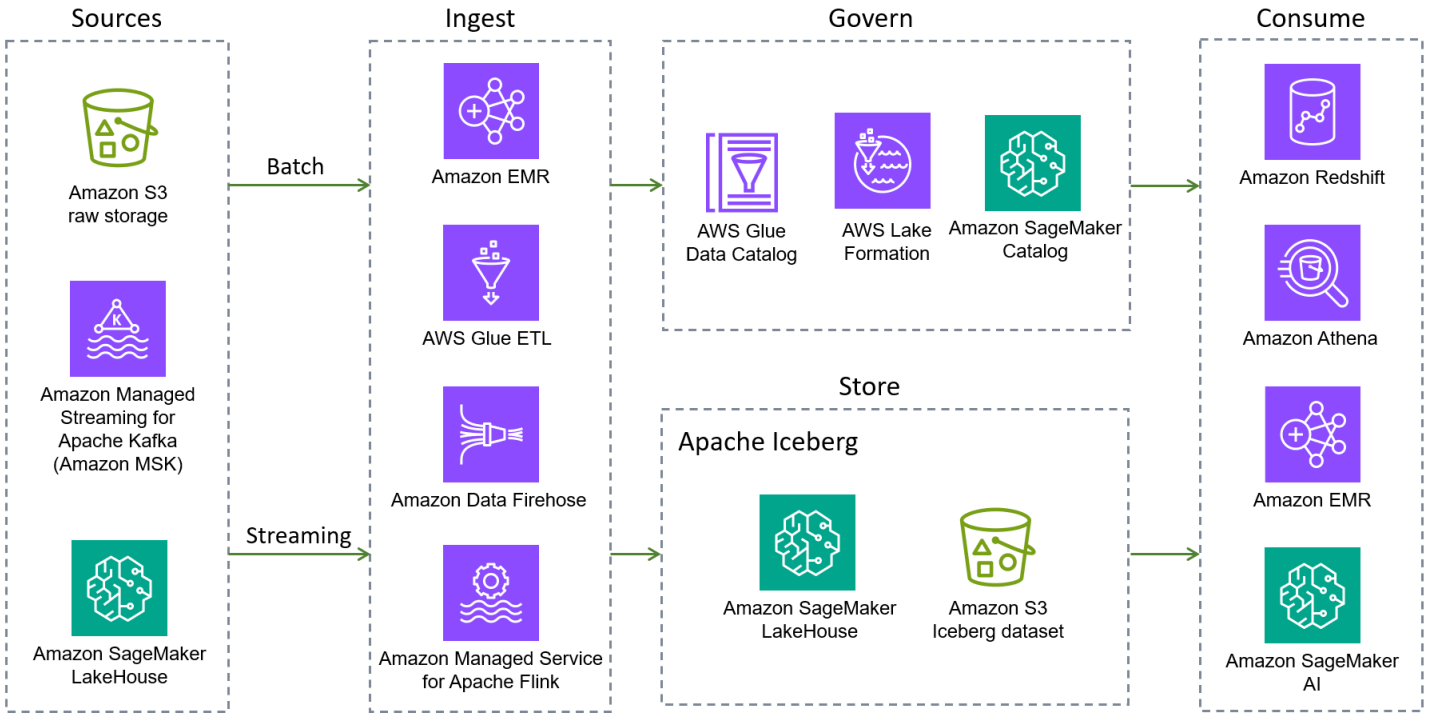
- Eliminar, actualizar y combinar.Iceberg admite comandos SQL estándar para el almacenamiento de datos para su uso con tablas de lagos de datos.
- Planificación rápida del escaneo y filtrado avanzado. Iceberg almacena metadatos, como estadísticas a nivel de partición y columna, que los motores pueden utilizar para acelerar la planificación y la ejecución de las consultas.
- Evolución completa del esquema. Iceberg permite añadir, eliminar, actualizar o cambiar el nombre de columnas sin efectos secundarios.
- Evolución de particiones. Puede actualizar el diseño de particiones de una tabla a medida que cambien el volumen de datos o los patrones de consulta. Iceberg permite cambiar las columnas en las que se divide una tabla, así como añadir columnas o eliminar columnas de las particiones compuestas.

- Particiones ocultas.Esta función evita la lectura automática de particiones innecesarias. Esto elimina la necesidad de que los usuarios entiendan los detalles de las particiones de la tabla o agreguen filtros adicionales a sus consultas.
- Reversión de versiones. Los usuarios pueden corregir rápidamente los problemas volviendo al estado anterior a la transacción.
- Viaje en el tiempo. Los usuarios pueden consultar una versión anterior específica de una tabla.
- Aislamiento serializable. Los cambios en las tablas son atómicos, por lo que los lectores nunca ven cambios parciales o no confirmados.
- Escritores simultáneos. Iceberg utiliza una simultaneidad optimista para permitir que varias transacciones tengan éxito. En caso de conflictos, uno de los redactores debe volver a intentar la transacción.
- Formatos de archivo abiertos. Iceberg admite varios formatos de archivo de código abierto, incluidos Apache Parquet, Apache Avro y Apache ORC.

En resumen, los lagos de datos que utilizan el formato Iceberg se benefician de la coherencia transaccional, la velocidad, la escala y la evolución del esquema. Para obtener más información sobre estas y otras funciones de Iceberg, consulte la documentación de Apache Iceberg.

AWS soporte para Apache Iceberg

Apache Iceberg es compatible Servicios de AWS con Amazon EMR, Amazon Athena, Amazon AWS Gluehttps://aws.amazon.com/glue/Redshift y Amazon. SageMaker El siguiente diagrama muestra una arquitectura de referencia simplificada de un lago de datos basado en Iceberg.



A continuación, se Servicios de AWS proporcionan integraciones nativas de Iceberg. Hay otras Servicios de AWS que pueden interactuar con Iceberg, ya sea de forma indirecta o empaquetando las bibliotecas de Iceberg.

- Amazon S3 es el mejor lugar para crear lagos de datos debido a sus capacidades de durabilidad, disponibilidad, escalabilidad, seguridad, conformidad y auditoría. Iceberg se diseñó y creó para interactuar con Amazon S3 sin problemas y proporciona soporte para muchas de las funciones de Amazon S3, tal como se indica en la documentación de Iceberg. Además, Amazon S3 Tables ofrece el primer almacén de objetos en la nube con soporte Iceberg integrado y agiliza el almacenamiento de datos tabulares a escala. Gracias a la compatibilidad con S3 Tables para Iceberg, puede consultar fácilmente sus datos tabulares mediante motores de consulta populares AWS y de terceros.
- La próxima generación de SageMaker se basa en una arquitectura abierta que unifica el acceso a los datos en los lagos de datos de Amazon S3, los almacenes de datos de Amazon Redshift y las fuentes de datos federadas y de terceros. Estas capacidades le ayudan a crear AI/ML aplicaciones y análisis potentes en una sola copia de los datos. Lakehouse es totalmente compatible con Iceberg, por lo que tiene la flexibilidad de acceder a los datos y consultarlos in situ mediante la API REST de Iceberg.
- Amazon EMR es una solución de big data para el procesamiento de datos a escala de petabytes, el análisis interactivo y el aprendizaje automático que utiliza marcos de código abierto como

Apache Spark, Flink, Trino y Hive. Amazon EMR puede ejecutarse en clústeres personalizados de Amazon Elastic Compute Cloud EC2 (Amazon), Amazon Elastic Kubernetes Service (Amazon EKS) AWS Outposts o Amazon EMR Serverless.

- Amazon Athena es un servicio de análisis interactivo sin servidor que se basa en marcos de código abierto. Admite formatos de archivos y tablas abiertas y proporciona una forma simplificada y flexible de analizar petabytes de datos allí donde se encuentran. Athena proporciona soporte nativo para consultas de lectura, viaje en el tiempo, escritura y DDL para Iceberg y utiliza el AWS Glue Data Catalog metaalmacén de Iceberg.
- Amazon Redshift es un almacén de datos en la nube a escala de petabytes que admite opciones de implementación basadas en clústeres y sin servidor. Amazon Redshift Spectrum puede consultar tablas externas que estén registradas y almacenadas en Amazon S3. AWS Glue Data Catalog Redshift Spectrum también admite el formato de almacenamiento Iceberg.
- AWS Gluees un servicio de integración de datos sin servidor que facilita el descubrimiento, la preparación, el traslado y la integración de datos de múltiples fuentes para el análisis, el aprendizaje automático (ML) y el desarrollo de aplicaciones. Está totalmente integrado con Iceberg. En concreto, puede realizar operaciones de lectura y escritura en las tablas de Iceberg mediante AWS Glue tareas, gestionar las tablas a través de ellas AWS Glue Data Catalog(compatibles con la metatienda de Hive), detectar y registrar las tablas automáticamente mediante AWS Glue rastreadores y evaluar la calidad de los datos de las tablas de Iceberg mediante la función de calidad de los datos. AWS Glue AWS Glue Data Catalog También permite recopilar estadísticas de columnas, calcular y actualizar el número de valores distintos (NDVs) para cada columna de las tablas Iceberg y optimizaciones automáticas de las tablas (compactación, retención de instantáneas, eliminación de archivos huérfanos). AWS Glue también admite integraciones sin ETL a partir de listas y aplicaciones de terceros en tablas de Iceberg Servicios de AWS .
- Amazon Data Firehose es un servicio totalmente gestionado para entregar datos de streaming en tiempo real a destinos como Amazon S3, Amazon Redshift, Amazon Service, OpenSearch Amazon Serverless, OpenSearch Splunk, tablas de Apache Iceberg y cualquier punto de enlace HTTP o HTTP personalizado propiedad de proveedores de servicios externos compatibles, incluidos Datadog, Dynatrace LogicMonitor, MongoDB, New Relic, Coralogix y Elastic. Con Firehose, no necesitas escribir aplicaciones ni administrar recursos. Configure los productores de datos para que envíen datos a Firehose y este entrega inmediatamente los datos al destino que usted especificó. También puede configurar Firehose para transformar los datos antes de entregarlos.

- Amazon Managed Service for Apache Flink es un servicio de Amazon totalmente gestionado que te permite utilizar una aplicación Apache Flink para procesar datos de streaming. Soporta tanto la lectura como la escritura en tablas de Iceberg, y permite procesar y analizar datos en tiempo real.
- Amazon SageMaker AI admite el almacenamiento de conjuntos de funciones en Amazon SageMaker AI Feature Store mediante el formato Iceberg.
- AWS Lake Formationproporciona permisos de control de acceso básicos y detallados para acceder a los datos, incluidas las tablas Iceberg utilizadas por Athena o Amazon Redshift. Para obtener más información sobre el soporte de permisos para las tablas Iceberg, consulte la documentación de Lake Formation.

AWS cuenta con una amplia gama de servicios compatibles con Iceberg, pero cubrir todos estos servicios va más allá del alcance de esta guía. Las siguientes secciones tratan sobre Spark (streaming estructurado y por lotes) en Amazon EMR y AWS Glue Athena SQL. En la siguiente sección se ofrece un vistazo rápido al soporte de Iceberg en Athena SQL.

Introducción a las tablas Iceberg en Amazon Athena SQL

Amazon Athena ofrece soporte integrado para Iceberg. Puede utilizar Iceberg sin ningún paso o configuración adicionales, excepto para configurar los requisitos previos del servicio que se detallan en la sección Primeros pasos de la documentación de Athena. En esta sección se ofrece una breve introducción a la creación de tablas en Athena. Para obtener más información, consulte Trabajar con tablas Iceberg mediante Athena SQL más adelante en esta guía.

Puede crear tablas Iceberg AWS utilizando diferentes motores. Esas tablas funcionan a la perfección en todas Servicios de AWS ellas. Para crear sus primeras tablas Iceberg con Athena SQL, puede utilizar el siguiente código repetitivo.

```
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'
)
```

En las siguientes secciones se proporcionan ejemplos de cómo crear tablas Iceberg particionadas y no particionadas en Athena. Para obtener más información, consulte la sintaxis de Iceberg detallada en la documentación de Athena.

Crear una tabla sin particiones

La siguiente instrucción de ejemplo personaliza el código SQL estándar para crear una tabla Iceberg sin particiones en Athena. Puede añadir esta declaración al editor de consultas de la consola de Athena para crear la tabla.

```
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'
)
```

Para step-by-step obtener instrucciones sobre cómo usar el editor de consultas, consulte Primeros pasos en la documentación de Athena.

Creación de una tabla particionada

La siguiente declaración crea una tabla particionada basada en la fecha utilizando el concepto de partición oculta de Iceberg. Utiliza la `day()` transformación para derivar particiones diarias, utilizando el `dd-mm-yyyy` formato, a partir de una columna de fecha y hora. Iceberg no almacena este valor como una columna nueva en el conjunto de datos. En cambio, el valor se obtiene sobre la marcha cuando se escriben o consultan datos.

```
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'
)
```

Crear una tabla y cargar datos con una sola sentencia CTAS

En los ejemplos particionados y no particionados de las secciones anteriores, las tablas Iceberg se crean como tablas vacías. Puede cargar datos en las tablas mediante la sentencia o. `INSERT MERGE` Como alternativa, puede usar una `CREATE TABLE AS SELECT (CTAS)` sentencia para crear y cargar datos en una tabla Iceberg en un solo paso.

El CTAS es la mejor forma en Athena de crear una tabla y cargar datos en una sola declaración. El siguiente ejemplo ilustra cómo usar CTAS para crear una tabla Iceberg (`iceberg_ctas_table`) a partir de una Hive/Parquet tabla existente (`hive_table`) en Athena.

```
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
```

Para obtener más información sobre las CTAS, consulte la documentación de las CTAS de Athena.

Inserción, actualización y eliminación de datos

Athena admite diferentes formas de escribir datos en una tabla Iceberg mediante las sentenciasINSERT INTO, UPDATEMERGE INTO, y M. DELETE FRO

> **Note**
>
> Athena SQL no admite actualmente este enfoque. copy-on-write UPDATEMERGE INTO, y DELETE FROM las operaciones siempre utilizan el merge-on-read enfoque con eliminaciones posicionales, independientemente de las propiedades de la tabla especificadas. Si configuras propiedades de tabla como `write.update.modewrite.merge.mode`, y `write.delete.mode` para usarlas copy-on-write, tus consultas no fallarán, pero Athena las ignorará y seguirá usándolas. merge-on-read

La siguiente sentencia se utiliza INSERT INTO para añadir datos a una tabla de 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;
```

Código de salida de ejemplo:

Para obtener más información, consulte la documentación de Athena.

Consultando tablas de Iceberg

Puede ejecutar consultas SQL normales en sus tablas de Iceberg mediante Athena SQL, como se ilustra en el ejemplo anterior.

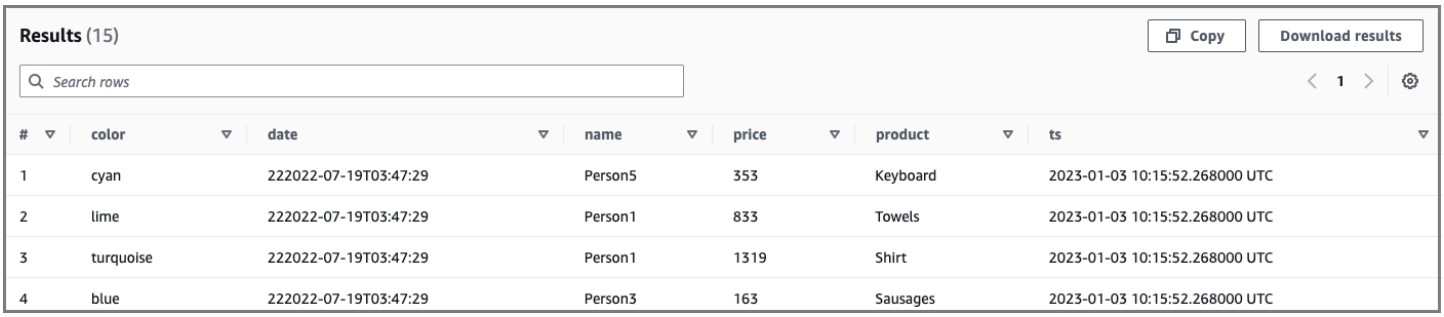
Además de las consultas habituales, Athena también admite consultas de viajes en el tiempo para tablas Iceberg. Como se ha mencionado anteriormente, puedes cambiar los registros existentes actualizándolos o eliminándolos en una tabla de Iceberg, por lo que resulta práctico utilizar las consultas de viajes en el tiempo para buscar versiones anteriores de la tabla en función de una marca de tiempo o un identificador instantáneo.

Por ejemplo, la siguiente declaración actualiza un valor de color y`Person5`, a continuación, muestra un valor anterior del 4 de enero de 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'
```

Código de salida de ejemplo:



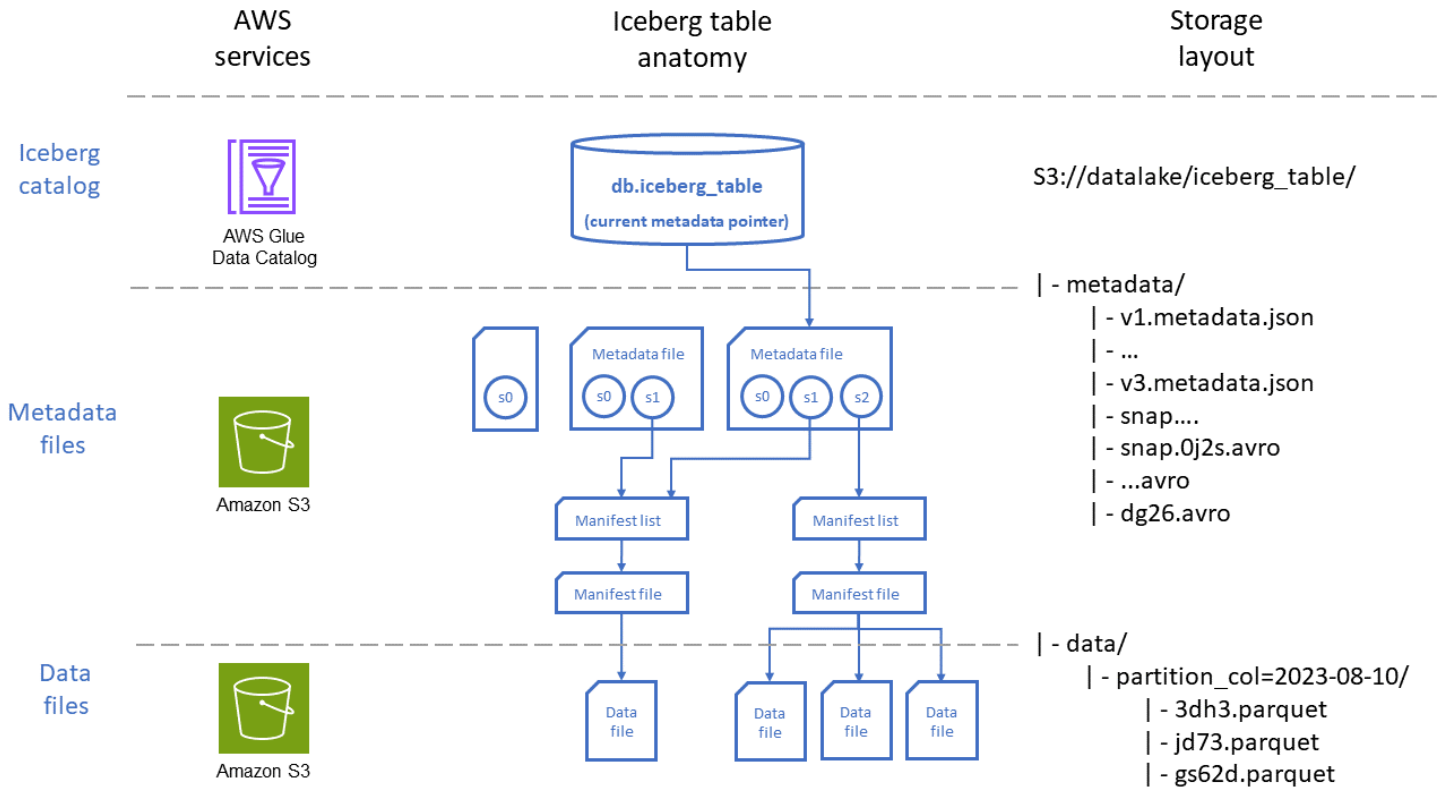
Para ver la sintaxis y ejemplos adicionales de consultas sobre viajes en el tiempo, consulte la documentación de Athena.

Anatomía de la tabla de iceberg

Ahora que hemos explicado los pasos básicos para trabajar con mesas Iceberg, profundicemos en los intrincados detalles y el diseño de una mesa Iceberg.

Para habilitar las funciones descritas anteriormente en esta guía, Iceberg está diseñado con capas jerárquicas de archivos de datos y metadatos. Estas capas administran los metadatos de forma inteligente para optimizar la planificación y la ejecución de las consultas.

El siguiente diagrama muestra la organización de una tabla Iceberg desde dos perspectivas: la Servicios de AWS utilizada para almacenar la tabla y la ubicación de los archivos en Amazon S3.



Como se muestra en el diagrama, una tabla de iceberg consta de tres capas principales:

- Catálogo Iceberg: AWS Glue Data Catalog se integra de forma nativa con Iceberg y, en la mayoría de los casos de uso, es la mejor opción para las cargas de trabajo que se ejecutan en ellas. AWS Los servicios que interactúan con las tablas Iceberg (por ejemplo, Athena) utilizan el catálogo para buscar la versión instantánea actual de la tabla, ya sea para leer o escribir datos.
- Capa de metadatos: los archivos de metadatos, es decir, los archivos de manifiesto y los archivos de listas de manifiestos, realizan un seguimiento de información como el esquema de las tablas, la estrategia de partición y la ubicación de los archivos de datos, así como de las estadísticas a nivel

de columna, como los rangos mínimo y máximo de los registros que se almacenan en cada archivo de datos. Estos archivos de metadatos se almacenan en Amazon S3 dentro de la ruta de la tabla.

  - Los archivos de manifiesto contienen un registro para cada archivo de datos, que incluye su ubicación, formato, tamaño, suma de verificación y otra información relevante.
  - Las listas de manifiestos proporcionan un índice de los archivos de manifiesto. A medida que aumenta el número de archivos de manifiesto en una tabla, dividir esa información en subsecciones más pequeñas ayuda a reducir el número de archivos de manifiesto que las consultas deben analizar.
  - Los archivos de metadatos contienen información sobre toda la tabla Iceberg, incluidas las listas de manifiestos, los esquemas, los metadatos de las particiones, los archivos de instantáneas y otros archivos que se utilizan para administrar los metadatos de la tabla.
- Capa de datos: esta capa contiene los archivos que contienen los registros de datos con los que se ejecutarán las consultas. Estos archivos se pueden almacenar en diferentes formatos, incluidos Apache Parquet, Apache Avro y Apache ORC.
  - Los archivos de datos contienen los registros de datos de una tabla.
  - Los archivos de eliminación codifican las operaciones de eliminación y actualización a nivel de fila en una tabla Iceberg. Iceberg tiene dos tipos de archivos de eliminación, tal y como se describe en la documentación de Iceberg. Estos archivos se crean mediante operaciones que utilizan el merge-on-read modo.

Trabajar con Iceberg en Amazon EMR

Amazon EMR proporciona procesamiento de datos a escala de petabytes, análisis interactivos y aprendizaje automático en la nube mediante marcos de código abierto como Apache Spark, Apache Hive, Flink y Trino.

> **Note**
>
> En esta guía se utilizan Apache Spark como ejemplos.

Amazon EMR admite varias opciones de implementación: Amazon EMR activado, Amazon EMR en EKS, EC2 Amazon EMR Serverless y Amazon EMR activado. AWS Outposts Para elegir una opción de despliegue para su carga de trabajo, consulte las preguntas frecuentes de Amazon EMR.

Compatibilidad de versiones y funciones

La versión 6.5.0 de Amazon EMR y las versiones posteriores admiten Apache Iceberg de forma nativa. Para obtener una lista de las versiones de Iceberg compatibles con cada versión de Amazon EMR, consulte el historial de versiones de Iceberg en la documentación de Amazon EMR. Consulte también las secciones de Uso de un clúster con Iceberg para ver qué funciones de Iceberg son compatibles con Amazon EMR en diferentes marcos.

Le recomendamos que utilice la última versión de Amazon EMR para beneficiarse de la última versión compatible de Iceberg. En los ejemplos de código y las configuraciones de esta sección se supone que está utilizando la versión emr-7.8.0 de Amazon EMR.

Creación de un clúster de Amazon EMR con Iceberg

Para crear un clúster de Amazon EMR en Amazon EC2 con Iceberg instalado, siga las instrucciones de la documentación de Amazon EMR.

En concreto, el clúster debe configurarse con la siguiente clasificación:

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

```
}]
```

También puede optar por utilizar Amazon EMR Serverless o Amazon EMR en EKS como opciones de implementación para sus cargas de trabajo de Iceberg, a partir de Amazon EMR 6.6.0.

Desarrollo de aplicaciones Iceberg en Amazon EMR

Para desarrollar el código Spark para sus aplicaciones Iceberg, puede utilizar Amazon EMR Studio, que es un entorno de desarrollo integrado (IDE) basado en la web para cuadernos Jupyter totalmente gestionados que se ejecutan en clústeres de Amazon EMR.

Uso de las libretas Amazon EMR Studio

Puede desarrollar aplicaciones Spark de forma interactiva en las libretas de Amazon EMR Studio Workspace y conectar esas libretas a Amazon EMR en los EC2 clústeres o a Amazon EMR en los puntos de enlace gestionados por EKS. Consulte Servicio de AWS la documentación para obtener instrucciones sobre cómo configurar un EMR Studio para Amazon EMR en EKS y EC2Amazon EMR en EKS.

Para usar Iceberg en EMR Studio, siga estos pasos:

1. Lance un clúster de Amazon EMR con Iceberg activado, tal y como se indica en Uso de un clúster con Iceberg instalado.
2. Configure un estudio de EMR. Para obtener instrucciones, consulte Configurar un Amazon EMR Studio.
3. Abre un cuaderno EMR Studio Workspace y ejecuta el siguiente código como primera celda del cuaderno para configurar tu sesión de Spark para usar 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>.type": "glue",
        "spark.sql.extensions":
 "org.apache.iceberg.spark.extensions.IcebergSparkSessionExtensions"
    }
}
```

donde:

- `<catalog_name>`es el nombre del catálogo de sesiones de Iceberg Spark. Sustitúyalo por el nombre que prefieras y recuerda cambiar las referencias en todas las configuraciones asociadas a este catálogo. En tu código, puedes hacer referencia a tus tablas de Iceberg utilizando el nombre completo de la tabla, incluido el nombre del catálogo de sesiones de Spark, de la siguiente manera:

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

  Como alternativa, puedes cambiar el catálogo predeterminado por el catálogo de Iceberg que has definido configurando el nombre `spark.sql.defaultCatalog` de tu catálogo. Este segundo enfoque le permite hacer referencia a las tablas sin el prefijo del catálogo, lo que puede simplificar las consultas.
- `<catalog_name>.warehouse`apunta a la ruta de Amazon S3 en la que desea almacenar sus datos y metadatos.
- Para convertir el catálogo en un AWS Glue Data Catalog, `spark.sql.catalog.<catalog_name>.type` configúrelo en`glue`. Esta clave es necesaria para apuntar a una clase de implementación para cualquier implementación de catálogo personalizada. La sección de mejores prácticas generales que aparece más adelante en esta guía describe los diferentes catálogos compatibles con Iceberg.

4. Ahora puedes empezar a desarrollar de forma interactiva tu aplicación de Spark para Iceberg en el portátil, como lo harías con cualquier otra aplicación de Spark.

Para obtener más información sobre cómo configurar Spark para Apache Iceberg mediante Amazon EMR Studio, consulte la entrada del blog Cree un lago de datos evolutivo, compatible con ACID y de alto rendimiento con Apache Iceberg en Amazon EMR.

Ejecutando trabajos de Iceberg en Amazon EMR

Tras desarrollar el código de aplicación de Spark para la carga de trabajo de Iceberg, puede ejecutarlo en cualquier opción de despliegue de Amazon EMR compatible con Iceberg (consulte las preguntas frecuentes sobre Amazon EMR).

Al igual que con otros trabajos de Spark, puede enviar trabajos a un Amazon EMR en un EC2 clúster añadiendo pasos o enviando trabajos de Spark de forma interactiva al nodo principal. Para ejecutar un trabajo de Spark, consulta las siguientes páginas de documentación de Amazon EMR:

- Para obtener una descripción general de las diferentes opciones para enviar trabajos a un Amazon EMR en un EC2 clúster e instrucciones detalladas para cada opción, consulte Enviar trabajo a un clúster.
- Para Amazon EMR en EKS, consulte Ejecutar trabajos de Spark con. StartJobRun
- Para EMR Serverless, consulte Ejecución de trabajos.

En las siguientes secciones se proporciona un ejemplo de cada opción de implementación de Amazon EMR.

Amazon EMR en EC2

Puedes seguir estos pasos para enviar el trabajo de Iceberg Spark:

1. Cree el archivo `emr_step_iceberg.json` con el siguiente contenido en su estación de trabajo:

```
[{
    "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>.type=glue",
        "--conf",
        "spark.sql.catalog.<catalog_name>.warehouse=s3://YOUR-BUCKET-NAME/YOUR-
FOLDER-NAME/",
        "s3://YOUR-BUCKET-NAME/code/iceberg-job.py"
    ]
}]
```

2. Modifica el archivo de configuración para tu trabajo específico de Spark personalizando las opciones de configuración de Iceberg que aparecen resaltadas en negrita.
3. Envía el paso mediante AWS Command Line Interface ()AWS CLI. Ejecute el comando en el directorio donde se encuentra el `emr_step_iceberg.json` archivo.

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

Amazon EMR sin servidor

Para enviar un trabajo de Iceberg Spark a EMR Serverless mediante: AWS CLI

1. Cree el archivo e`mr_serverless_iceberg.jso`n con el siguiente contenido en su estación de trabajo:

```
{
    "applicationId": "<APPLICATION_ID>",
    "executionRoleArn": "<ROLE_ARN>",
    "name": "iceberg-test-job",
    "jobDriver": {
        "sparkSubmit": {
            "entryPoint": "s3://YOUR-BUCKET-NAME/code/iceberg-job.py",
            "entryPointArguments": []
        }
    },
    "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>.type": "glue",
                "spark.sql.catalog.<catalog_name>.warehouse": "s3://YOUR-BUCKET-NAME/
YOUR-FOLDER-NAME/",
                "spark.jars":"/usr/share/aws/iceberg/lib/iceberg-spark3-runtime.jar",

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

```
}
```

2. Modifica el archivo de configuración para tu trabajo específico de Spark personalizando las opciones de configuración de Iceberg que aparecen resaltadas en negrita.
3. Envía el trabajo mediante. AWS CLI Ejecute el comando en el directorio donde se encuentra el `emr_serverless_iceberg.json` archivo:

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

Para enviar un trabajo de Iceberg Spark a EMR Serverless mediante la consola EMR Studio:

1. Siga las instrucciones de la documentación de EMR Serverless.
2. Para la configuración de Job, utilice la configuración de Iceberg para Spark proporcionada para Iceberg AWS CLI y personalice los campos resaltados para Iceberg. Para obtener instrucciones detalladas, consulte Uso de Apache Iceberg con EMR Serverless en la documentación de Amazon EMR.

Amazon EMR en EKS

Para enviar un trabajo de Iceberg Spark a Amazon EMR en EKS mediante: AWS CLI

1. Cree el archivo `emr_eks_iceberg.json` con el siguiente contenido en su estación de trabajo:

```
{
    "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>.type": "glue",
                "spark.sql.catalog.<catalog_name>.warehouse": "s3://YOUR-BUCKET-NAME/
YOUR-FOLDER-NAME/",
                "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 el archivo de configuración de tu trabajo de Spark personalizando las opciones de configuración de Iceberg que aparecen resaltadas en negrita.
3. Envía el trabajo mediante. AWS CLI Ejecute el siguiente comando en el directorio donde se encuentra el `emr_eks_iceberg.json` archivo:

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

Para obtener instrucciones detalladas, consulte Uso de Apache Iceberg con Amazon EMR en EKS en la documentación de Amazon EMR en EKS.

Prácticas recomendadas para Amazon EMR

En esta sección se proporcionan pautas generales para ajustar los trabajos de Spark en Amazon EMR a fin de optimizar la lectura y la escritura de datos en las tablas de Iceberg. Para conocer las mejores prácticas específicas de Iceberg, consulte la sección de mejores prácticas más adelante en esta guía.

- Utilice la última versión de Amazon EMR: Amazon EMR proporciona optimizaciones de Spark listas para usar con el tiempo de ejecución de Amazon EMR Spark. AWS mejora el rendimiento del motor de ejecución de Spark con cada nueva versión.
- Determina la infraestructura óptima para tus cargas de trabajo de Spark: las cargas de trabajo de Spark pueden requerir diferentes tipos de hardware para diferentes características de trabajo a fin de garantizar un rendimiento óptimo. Amazon EMR admite varios tipos de instancias (como optimizadas para cómputo, optimizadas para memoria, de uso general y optimizadas para almacenamiento) para cubrir todos los tipos de requisitos de procesamiento. Cuando incorpore nuevas cargas de trabajo, le recomendamos que utilice tipos de instancias generales como M5 o M6g. Supervise el sistema operativo (SO) y las métricas de YARN de Ganglia y Amazon CloudWatch para determinar los cuellos de botella del sistema (CPU, memoria, almacenamiento y E/S) en momentos de máxima carga y elija el hardware adecuado.
- **`spark.sql.shuffle.partitions`**Ajustar: establezca la `spark.sql.shuffle.partitions` propiedad en el número total de núcleos virtuales (núcleos virtuales) del clúster o en un múltiplo de ese valor (normalmente, de 1 a 2 veces el número total de núcleos virtuales). Esta configuración afecta al paralelismo de Spark cuando se utilizan particiones por hash y rango como modo de distribución de escritura. Solicita una reproducción aleatoria antes de escribir para organizar los datos, lo que garantiza la alineación de las particiones.
- Habilite el escalado administrado: para casi todos los casos de uso, le recomendamos que habilite el escalado administrado y la asignación dinámica. Sin embargo, si tiene una carga de trabajo que sigue un patrón predecible, le sugerimos que desactive el escalado automático y la asignación dinámica. Cuando el escalado gestionado esté activado, le recomendamos que utilice instancias puntuales para reducir los costes. Utilice instancias puntuales para los nodos de tareas en lugar de los nodos principales o maestros. Cuando utilice instancias puntuales, utilice flotas de instancias con varios tipos de instancias por flota para garantizar la disponibilidad puntual.
- Utilice la unión por transmisión siempre que sea posible: la unión por transmisión (del lado del mapa) es la combinación más óptima, siempre que una de sus tablas sea lo suficientemente pequeña como para caber en la memoria del nodo más pequeño (en el orden de MBs) y esté realizando una unión equi (=). Se admiten todos los tipos de uniones, excepto las uniones externas completas. Una unión de difusión difunde la tabla más pequeña como una tabla hash en todos los nodos de trabajo de la memoria. Una vez difundida la tabla pequeña, no podrá realizar cambios en ella. Como la tabla hash se encuentra localmente en la máquina virtual Java (JVM), se puede combinar fácilmente con la tabla grande en función de la condición de unión mediante una combinación hash. Las uniones por transmisión ofrecen un alto rendimiento debido a una sobrecarga mínima de reproducción aleatoria.

- Ajuste el recolector de basura: si los ciclos de recolección de basura (GC) son lentos, considere cambiar del recolector de basura paralelo predeterminado al G1GC para obtener un mejor rendimiento. Para optimizar el rendimiento del GC, puede ajustar los parámetros del GC. Para hacer un seguimiento del rendimiento del GC, puedes monitorizarlo mediante la interfaz de usuario de Spark. Lo ideal es que el tiempo de GC sea inferior o igual al 1 por ciento del tiempo total de ejecución de la tarea.

Trabajando con Iceberg en AWS Glue

AWS Gluees un servicio de integración de datos sin servidor que facilita el descubrimiento, la preparación, el traslado y la integración de datos de múltiples fuentes para el análisis, el aprendizaje automático (ML) y el desarrollo de aplicaciones. Una de las principales capacidades AWS Glue es su capacidad para realizar operaciones de extracción, transformación y carga (ETL) de forma sencilla y rentable. Esto ayuda a clasificar los datos, limpiarlos, enriquecerlos y moverlos de forma fiable entre varios almacenes de datos y flujos de datos.

AWS Glue Los trabajos encapsulan scripts que definen la lógica de transformación mediante un tiempo de ejecución de Apache Spark o Python. AWS Glue los trabajos se pueden ejecutar tanto en modo por lotes como en modo streaming.

Al crear trabajos de Iceberg AWS Glue, según la versión de AWS Glue, puede utilizar la integración nativa de Iceberg o una versión personalizada de Iceberg para adjuntar las dependencias de Iceberg al trabajo.

Uso de la integración nativa de Iceberg

AWS Glue Las versiones 3.0, 4.0 y 5.0 admiten de forma nativa formatos de lagos de datos transaccionales como Apache Iceberg, Apache Hudi y Linux Foundation Delta Lake para Spark. AWS Glue Esta función de integración simplifica los pasos de configuración necesarios para empezar a utilizar estos marcos en. AWS Glue

Para habilitar el soporte de Iceberg para su AWS Glue trabajo, configure el trabajo: elija la pestaña Detalles del trabajo para su AWS Glue trabajo, desplácese hasta Parámetros del trabajo en Propiedades avanzadas y establezca la clave `--datalake-formats` y su valor en. `iceberg`

Si va a crear un trabajo con una libreta, puede configurar el parámetro en la primera celda de la libreta utilizando la `%%configure` magia de la siguiente manera:

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

La `iceberg` configuración de `--datalake-formats` in AWS Glue corresponde a versiones específicas de Iceberg en función de su AWS Glue versión:

AWS Glue versión	Versión predeterminada de Iceberg
5.0	1.7.1
4.0	1.0.0
3.0	0.13.1

Uso de una versión de Iceberg personalizada

En algunas situaciones, es posible que desee conservar el control sobre la versión Iceberg para el trabajo y actualizarla a su propio ritmo. Por ejemplo, la actualización a una versión posterior puede desbloquear el acceso a nuevas funciones y mejoras de rendimiento. Para usar una versión específica de Iceberg AWS Glue, puede proporcionar sus propios archivos JAR.

Antes de implementar una versión de Iceberg personalizada, compruebe la compatibilidad con su AWS Glue entorno consultando la sección de AWS Glue versiones de la AWS Glue documentación. Por ejemplo, la AWS Glue versión 5.0 requiere compatibilidad con Spark 3.5.4.

Por ejemplo, para ejecutar AWS Glue trabajos que usen la versión 1.9.1 de Iceberg, sigue estos pasos:

1. Adquiera y cargue los archivos JAR necesarios en Amazon S3:
   a. Descargue iceberg-spark-runtime-3.5_2.12-1.9.1.jar y -1.9.1.jar del repositorio Maven de Apache. iceberg-aws-bundle
   b. Cargue estos archivos en la ubicación del bucket de S3 que haya designado (por ejemplo,). `s3://your-bucket-name/jars/`
2. Configure los parámetros de trabajo para su AWS Glue trabajo de la siguiente manera:
   a. Especifique la ruta S3 completa a ambos archivos JAR en el `--extra-jars` parámetro, separándolos con una coma (por ejemplo,`s3://your-bucket-name/jars/iceberg-spark-runtime-3.5_2.12-1.9.1.jar,s3://your-bucket-name/jars/iceberg-aws-bundle-1.9.1.jar`).
   b. No incluya `iceberg` como valor para el parámetro `--datalake-formats`.
   c. Si usa AWS Glue 5.0, debe establecer el `--user-jars-first` parámetro en. `true`

Configuraciones de Spark para Iceberg en AWS Glue

En esta sección se analizan las configuraciones de Spark necesarias para crear un trabajo de AWS Glue ETL para un conjunto de datos de Iceberg. Puedes establecer estas configuraciones usando la clave de `--conf` Spark con una lista separada por comas de todas las claves y valores de configuración de Spark. Puedes usar la `%%configure` magia en un cuaderno o en la sección de parámetros de Job de la AWS Glue Studio consola.

```
%glue_version 5.0

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

Configura la sesión de Spark con las siguientes propiedades:

- `<catalog_name>`es el nombre del catálogo de sesiones de Iceberg Spark. Sustitúyalo por el nombre que prefieras y recuerda cambiar las referencias en todas las configuraciones asociadas a este catálogo. En tu código, puedes hacer referencia a tus tablas de Iceberg utilizando el nombre completo de la tabla, incluido el nombre del catálogo de sesiones de Spark, de la siguiente manera:

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

  Como alternativa, puedes cambiar el catálogo predeterminado por el catálogo de Iceberg que has definido configurando el nombre `spark.sql.defaultCatalog` de tu catálogo. Puede utilizar este segundo enfoque para hacer referencia a las tablas sin el prefijo del catálogo, lo que puede simplificar las consultas.
- `<catalog_name>.<warehouse>`apunta a la ruta de Amazon S3 en la que desea almacenar sus datos y metadatos.
- Para convertir el catálogo en un AWS Glue Data Catalog, `spark.sql.catalog.<catalog_name>.type` configúrelo en`glue`. Esta clave es necesaria para apuntar a una clase de implementación para cualquier implementación de catálogo personalizada. Para ver los catálogos compatibles con Iceberg, consulte la sección de mejores prácticas generales más adelante en esta guía.

Por ejemplo, si tiene un catálogo llamado`glue_iceberg`, puede configurar su trabajo mediante varias `--conf` claves de la siguiente manera:

```
%%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.type=glue"
}
```

Como alternativa, puedes usar el código para añadir las configuraciones anteriores a tu script de Spark de la siguiente manera:

```
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.type", "glue") \
                    .getOrCreate()
```

Mejores prácticas para los AWS Glue trabajos

Esta sección proporciona pautas generales para ajustar los trabajos de Spark AWS Glue a fin de optimizar la lectura y la escritura de datos en las tablas de Iceberg. Para conocer las mejores prácticas específicas de Iceberg, consulta la sección de mejores prácticas más adelante en esta guía.

- Utilice la última versión AWS Glue y actualícela siempre que sea posible: las nuevas versiones AWS Glue ofrecen mejoras en el rendimiento, reducen los tiempos de inicio y ofrecen nuevas funciones. También son compatibles con las versiones más recientes de Spark que podrían ser necesarias para las versiones más recientes de Iceberg. Para ver una lista de AWS Glue las versiones disponibles y las versiones de Spark compatibles con ellas, consulta la AWS Glue documentación.

- Optimiza la memoria de AWS Glue trabajo: sigue las recomendaciones de la entrada del AWS blog Optimiza la gestión de la memoria en AWS Glue.
- Use AWS Glue Auto Scaling: cuando habilita Auto Scaling, ajusta AWS Glue automáticamente la cantidad de AWS Glue trabajadores de forma dinámica en función de su carga de trabajo. Esto ayuda a reducir el costo de su AWS Glue trabajo durante los picos de carga, ya AWS Glue que reduce el número de trabajadores cuando la carga de trabajo es pequeña y los trabajadores están inactivos. Para usar AWS Glue Auto Scaling, debe especificar un número máximo de trabajadores al que se puede escalar su AWS Glue trabajo. Para obtener más información, consulte Uso del escalado automáticoAWS Glue en la AWS Glue documentación.
- Utilice la versión de Iceberg que desee: la integración AWS Glue nativa de Iceberg es lo mejor para empezar a utilizar Iceberg. Sin embargo, para las cargas de trabajo de producción, le recomendamos que añada dependencias de biblioteca (tal y como se ha explicado anteriormente en esta guía) para tener un control total sobre la versión de Iceberg. Este enfoque le ayuda a beneficiarse de las últimas funciones de Iceberg y de las mejoras de rendimiento en sus trabajos. AWS Glue
- Habilita la interfaz de usuario de Spark para monitorizar y depurar. También puedes usar la interfaz de usuario de Spark AWS Glue para inspeccionar tu trabajo de Iceberg visualizando las diferentes etapas de un trabajo de Spark en un gráfico acíclico dirigido (DAG) y supervisando los trabajos en detalle. La interfaz de usuario de Spark proporciona una forma eficaz de solucionar problemas y optimizar los trabajos de Iceberg. Por ejemplo, puedes identificar las etapas más complicadas en las que se produce mucha confusión o se pierde mucho espacio en el disco para identificar las oportunidades de ajuste. Para obtener más información, consulte Supervisión de trabajos mediante la interfaz de usuario web de Apache Spark en la documentación. AWS Glue

Trabajar con tablas Iceberg mediante Apache Spark

En esta sección se proporciona una descripción general del uso de Apache Spark para interactuar con las tablas Iceberg. Los ejemplos son código repetitivo que se puede ejecutar en Amazon EMR o. AWS Glue

Nota: La interfaz principal para interactuar con las tablas de Iceberg es SQL, por lo que la mayoría de los ejemplos combinarán Spark SQL con la API. DataFrames

Crear y escribir tablas Iceberg

Puedes usar Spark SQL y Spark DataFrames para crear y añadir datos a las tablas de Iceberg.

Uso de Spark SQL

Para escribir un conjunto de datos de Iceberg, usa sentencias SQL estándar de Spark, como `CREATE TABLE` y`INSERT INTO`.

Tablas sin particionar

Este es un ejemplo de cómo crear una tabla Iceberg sin particiones 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')
""")
```

Para insertar datos en una tabla sin particiones, usa una declaración estándar: `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
""")
```

Tablas particionadas

Este es un ejemplo de cómo crear una tabla Iceberg particionada 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')
""")
```

Para insertar datos en una tabla Iceberg particionada con Spark SQL, usa una declaración estándar: `INSERT INTO`

```
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
""")
```

> **Note**
>
> A partir de Iceberg 1.5.0, el modo de distribución de `hash` escritura es el predeterminado al insertar datos en tablas particionadas. Para obtener más información, consulta Cómo escribir modos de distribución en la documentación de Iceberg.

Uso de la API DataFrames

Para escribir un conjunto de datos de Iceberg, puedes usar la `DataFrameWriterV2` API.

Para crear una tabla de iceberg y escribir datos en ella, usa la función `df.writeTo( t)`. Si la tabla existe, utilice la `.append()` función. Si no es así, usa `.create().` Los siguientes ejemplos usan`.createOrReplace()`, que es una variación de lo `.create()` que equivale a`CREATE OR REPLACE TABLE AS SELECT`.

Tablas sin particionar

Para crear y rellenar una tabla Iceberg sin particiones mediante la API: `DataFrameWriterV2`

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

Para insertar datos en una tabla de Iceberg sin particiones existente mediante la API: `DataFrameWriterV2`

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

Tablas particionadas

Para crear y rellenar una tabla de Iceberg particionada mediante la API: `DataFrameWriterV2`

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

Para insertar datos en una tabla de Iceberg particionada mediante la API: `DataFrameWriterV2`

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

Actualización de datos en tablas de iceberg

El siguiente ejemplo muestra cómo actualizar los datos de una tabla de iceberg. En este ejemplo se modifican todas las filas que tienen un número par en la `c_customer_sk` columna.

```
spark.sql(f"""
```

```
UPDATE {CATALOG_NAME}.{db.name}.{table.name}
SET c_email_address = 'even_row'
WHERE c_customer_sk % 2 == 0
""")
```

Esta operación utiliza la copy-on-write estrategia predeterminada, por lo que reescribe todos los archivos de datos afectados.

Modificación de datos en tablas de Iceberg

La alteración de los datos consiste en insertar nuevos registros de datos y actualizar los registros de datos existentes en una sola transacción. Para descomponer los datos en una tabla de iceberg, se utiliza la declaración. `SQL MERGE INTO`

El siguiente ejemplo altera el contenido de la tabla} dentro de la tabla`{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 *
""")
```

- Si un registro de cliente que está `{UPSERT_TABLE_NAME}` ya existe `{TABLE_NAME}` con el mismo`c_customer_id`, el valor del `{UPSERT_TABLE_NAME}` registro anula el `c_email_address` valor existente (operación de actualización).
- Si un registro de cliente incluido `{UPSERT_TABLE_NAME}` no existe en`{TABLE_NAME}`, el `{UPSERT_TABLE_NAME}` registro se agrega a `{TABLE_NAME}` (operación de inserción).

Eliminar datos de las tablas de Iceberg

Para eliminar datos de una tabla de iceberg, utilice la `DELETE FROM` expresión y especifique un filtro que coincida con las filas que desee eliminar.

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

```
""")
```

Si el filtro coincide con una partición completa, Iceberg elimina solo los metadatos y deja los archivos de datos en su lugar. De lo contrario, solo reescribe los archivos de datos afectados.

El método de eliminación toma los archivos de datos afectados por la `WHERE` cláusula y crea una copia de los mismos sin los registros eliminados. A continuación, crea una nueva instantánea de la tabla que apunta a los nuevos archivos de datos. Por lo tanto, los registros eliminados siguen presentes en las instantáneas anteriores de la tabla. Por ejemplo, si recupera la instantánea anterior de la tabla, verá los datos que acaba de eliminar. Para obtener información sobre cómo eliminar instantáneas antiguas innecesarias con los archivos de datos relacionados con fines de limpieza, consulte la sección Mantenimiento de archivos mediante la compactación, que aparece más adelante en esta guía.

Lectura de datos

Puedes leer el estado más reciente de tus tablas de Iceberg en Spark tanto con Spark SQL como con. DataFrames

Ejemplo de uso de Spark SQL:

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

Ejemplo de uso de la DataFrames API:

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

Uso del viaje en el tiempo

Cada operación de escritura (insertar, actualizar, modificar, eliminar) en una tabla de Iceberg crea una nueva instantánea. A continuación, puede utilizar estas instantáneas para viajar en el tiempo, es decir, para retroceder en el tiempo y comprobar el estado de una tabla en el pasado.

Para obtener información sobre cómo recuperar el historial de las instantáneas de las tablas mediante valores de uso `snapshot-id` y temporización, consulte la sección Acceso a los metadatos que aparece más adelante en esta guía.

La siguiente consulta de viaje en el tiempo muestra el estado de una tabla en función de un dato específico`snapshot-id`.

Con Spark SQL:

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

Uso de la 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 siguiente consulta de viaje en el tiempo muestra el estado de una tabla en función de la última instantánea que se creó antes de una marca de tiempo específica, en milisegundos ()`as-of-timestamp`.

Con Spark SQL:

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

Uso de la API DataFrames :

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

Uso de consultas incrementales

También puede utilizar las instantáneas de Iceberg para leer los datos adjuntos de forma incremental.

Nota: Actualmente, esta operación admite la lectura de datos de instantáneas. `append` No admite la obtención de datos de operaciones como `replaceoverwrite`, o. `delete` Además, las operaciones de lectura incremental no se admiten en la sintaxis SQL de Spark.

En el siguiente ejemplo, se recuperan todos los registros adjuntos a una tabla de Iceberg entre la instantánea `start-snapshot-id` (exclusiva) y la `end-snapshot-id` (incluida).

```
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}")
)
```

Acceder a los metadatos

Iceberg proporciona acceso a sus metadatos a través de SQL. Puede acceder a los metadatos de cualquier tabla (`<table_name>`) consultando el espacio de nombres. `<table_name>.<metadata_table>` Para obtener una lista completa de las tablas de metadatos, consulte Inspección de tablas en la documentación de Iceberg.

El siguiente ejemplo muestra cómo acceder a la tabla de metadatos del historial de Iceberg, que muestra el historial de confirmaciones (cambios) de una tabla de Iceberg.

Uso de Spark SQL (con la `%%sql` magia) desde una libreta Amazon EMR Studio:

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

Uso de la DataFrames API:

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

Código de salida de ejemplo:

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

Trabajar con tablas Iceberg mediante Trino

En esta sección se describe cómo configurar y operar tablas Iceberg mediante Trino en Amazon EMR. Los ejemplos son código repetitivo que puede ejecutar en un clúster de Amazon EMR en EC2. En los ejemplos de código y las configuraciones de esta sección se supone que está utilizando la versión emr-7.9.0 de Amazon EMR.

Configuración de Amazon EMR en EC2

1. Cree un archivo `iceberg.properties` con el siguiente contenido. La `iceberg.file-format=parquet` configuración determina el formato de almacenamiento predeterminado para las tablas nuevas si el formato no se especifica explícitamente en la `CREATE TABLE` declaración.

```
connector.name=iceberg
iceberg.catalog.type=glue
iceberg.file-format=parquet
fs.native-s3.enabled=true
```

2. Cargue el archivo `iceberg.properties` en el bucket de S3.
3. Cree una acción de arranque que copie el `iceberg.properties` archivo de su bucket de S3 y lo almacene como un archivo de configuración de Trino en el clúster de Amazon EMR que va a crear. Asegúrese de reemplazarlo por el nombre `<S3-bucket-name>` de su bucket de S3.

```
#!/bin/bash
set -ex
sudo aws s3 cp s3://<S3-bucket-name>/iceberg.properties /etc/trino/conf/catalog/
iceberg.properties
```

4. Cree un clúster de Amazon EMR con Trino instalado y especifique la ejecución del script anterior como una acción de arranque. Este es un ejemplo de comando AWS Command Line Interface (AWS CLI) para crear el clúster:

```
aws emr create-cluster --release-label emr-7.9.0 \
--applications Name=Trino \
--region <region> \
--name Trino_Iceberg_Cluster \
--bootstrap-actions '[{"Path":"s3://<S3-bucket-name>/bootstrap.sh","Name":"Add
 iceberg.properties"}]' \
```

```
--instance-groups
 '[{"InstanceGroupType":"MASTER","InstanceCount":1,"InstanceType":"m5.xlarge"},
{"InstanceGroupType":"CORE","InstanceCount":3,"InstanceType":"m5.xlarge"}]' \
--service-role "<IAM-service-role>" \
--ec2-attributes '{"KeyName":"<key-name>","InstanceProfile":"<EMR-EC2-instance-
profile>"}'
```

donde sustituyes:

- `<S3-bucket-name>`con el nombre de su bucket de S3
- `<region>`con tu específico Región de AWS
- `<key-name>`con tu key pair. Si el par de claves no existe, se creará.
- `<IAM-service-role>`con su función de servicio de Amazon EMR que siga el principio del privilegio mínimo.
- `<EMR-EC2-instance-profile>`con su perfil de instancia.

5. Cuando se haya inicializado el clúster de Amazon EMR, puede inicializar una sesión de Trino ejecutando el siguiente comando:

```
trino-cli
```

6. En la CLI de Trino, puede ver los catálogos ejecutando:

```
SHOW CATALOGS;
```

Creación de tablas de Iceberg

Para crear una tabla de iceberg, puede usar la sentencia. `CREATE TABLE` Este es un ejemplo de cómo crear una tabla particionada que usa la partición oculta de Iceberg:

```
CREATE TABLE iceberg.iceberg_db.iceberg_table (
             userid int,
             firstname varchar,
             city varchar)
        WITH (
             format = 'PARQUET',
             partitioning = ARRAY['city', 'bucket(userid, 16)'],
             location = 's3://<S3-bucket>/<prefix>');
```

> **Note**
>
> Si no especificas el formato, se usará el `iceberg.file-format` valor que configuraste en la sección anterior.

Para insertar datos, utilice el `INSERT INTO` comando. A continuación se muestra un ejemplo:

```
INSERT INTO iceberg.iceberg_db.iceberg_table (userid, firstname, city)
VALUES
    (1001, 'John', 'New York'),
    (1002, 'Mary', 'Los Angeles'),
    (1003, 'Mateo', 'Chicago'),
    (1004, 'Shirley', 'Houston'),
    (1005, 'Diego', 'Miami'),
    (1006, 'Nikki', 'Seattle'),
    (1007, 'Pat', 'Boston'),
    (1008, 'Terry', 'San Francisco'),
    (1009, 'Richard', 'Denver'),
    (1010, 'Pat', 'Phoenix');
```

Lectura de tablas de iceberg

Puede leer el estado más reciente de su tabla de iceberg mediante una `SELECT` declaración, de la siguiente manera:

```
SELECT * FROM iceberg.iceberg_db.iceberg_table;
```

Dividir los datos en tablas de iceberg

Puede realizar una operación de alteración (insertar simultáneamente nuevos registros y actualizar los existentes) mediante la instrucción. `MERGE INTO` A continuación se muestra un ejemplo:

```
MERGE INTO iceberg.iceberg_db.iceberg_table target
USING (
      VALUES
        (1001, 'John Updated', 'Boston'),       -- Update existing user
        (1002, 'Mary Updated', 'Seattle'),      -- Update existing user
        (1011, 'Martha', 'Portland'),           -- Insert new user
```

```
        (1012, 'Paulo', 'Austin')               -- Insert new user
    ) AS source (userid, firstname, city)
  ON target.userid = source.userid
WHEN MATCHED THEN
     UPDATE SET
       firstname = source.firstname,
       city = source.city
WHEN NOT MATCHED THEN
     INSERT (userid, firstname, city)
     VALUES (source.userid, source.firstname, source.city);
```

Eliminar registros de las tablas de Iceberg

Para eliminar datos de una tabla de iceberg, utilice la `DELETE FROM` expresión y especifique un filtro que coincida con las filas que desee eliminar. A continuación se muestra un ejemplo:

```
DELETE FROM iceberg.iceberg_db.iceberg_table WHERE userid IN (1003, 1004);
```

Consulta de los metadatos de la tabla de Iceberg

Iceberg proporciona acceso a sus metadatos a través de SQL. Puede acceder a los metadatos de cualquier tabla (`<table_name>`) consultando el espacio de nombres. `"<table_name>.$<metadata_table>"` Para obtener una lista completa de las tablas de metadatos, consulte Inspección de tablas en la documentación de Iceberg.

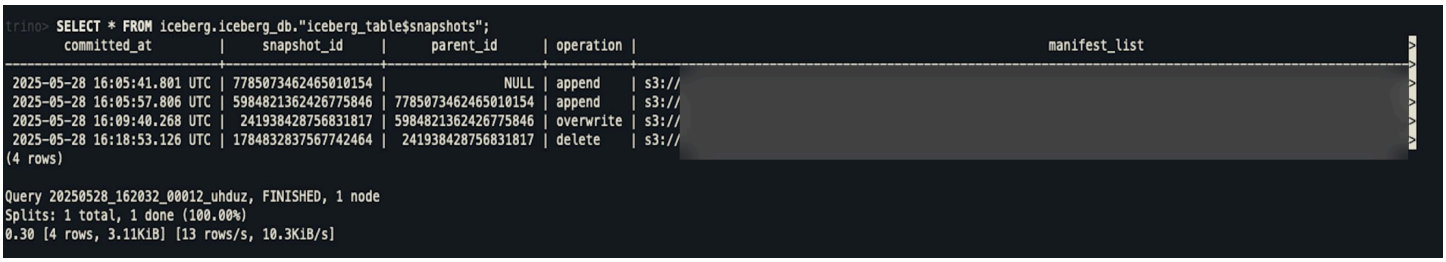
A continuación, se muestra un ejemplo de una lista de consultas para inspeccionar los metadatos de Iceberg:

```
SELECT  FROM iceberg.iceberg_db."iceberg_table$snapshots";
SELECT  FROM iceberg.iceberg_db."iceberg_table$history";
SELECT  FROM iceberg.iceberg_db."iceberg_table$partitions";
SELECT  FROM iceberg.iceberg_db."iceberg_table$files";
SELECT  FROM iceberg.iceberg_db."iceberg_table$manifests";
SELECT  FROM iceberg.iceberg_db."iceberg_table$refs";
SELECT * FROM iceberg.iceberg_db."iceberg_table$metadata_log_entries";
```

Por ejemplo, esta consulta:

```
SELECT * FROM iceberg.iceberg_db."iceberg_table$snapshots";
```

proporciona el resultado:



Uso del viaje en el tiempo

Cada operación de escritura (insertar, actualizar, modificar o eliminar) en una tabla de Iceberg crea una nueva instantánea. A continuación, puede utilizar estas instantáneas para viajar en el tiempo, es decir, para retroceder en el tiempo y comprobar el estado de una tabla en el pasado.

La siguiente consulta de viaje en el tiempo muestra el estado de una tabla en función de un dato específico: `snapshot_id`

```
SELECT *
FROM iceberg.iceberg_db.iceberg_table FOR VERSION AS OF 241938428756831817;
```

La siguiente consulta de viaje en el tiempo muestra el estado de una tabla en función de una marca de tiempo específica:

```
SELECT *
FROM iceberg.iceberg_db.iceberg_table FOR TIMESTAMP AS OF TIMESTAMP '2025-05-28
 16:09:40.268 UTC'
```

Consideraciones a la hora de utilizar Iceberg con Trino

Las operaciones de escritura de Trino en las tablas de Iceberg siguen el merge-on-readdiseño, por lo que crean archivos de eliminación posicional en lugar de reescribir archivos de datos completos que se ven afectados por las actualizaciones o eliminaciones. Si quieres usar copy-on-write este enfoque, considera usar Spark para las operaciones de escritura.

Trabajar con tablas Iceberg mediante Amazon Data Firehose

Amazon Data Firehose es un servicio sin servidor y sin código que permite enviar flujos de datos de más de 20 fuentes, como logs AWS WAF , Amazon Logs, Amazon CloudWatch Kinesis Data Streams y Amazon Managed Streaming for Apache Kafka (Amazon MSK) a destinos como Amazon S3, Amazon Redshift, Snowflake y Splunk. AWS IoT

Puede usar Firehose para entregar datos de streaming directamente a las tablas de Apache Iceberg en Amazon S3. Con Firehose, puede enrutar los registros de un único flujo a diferentes tablas de Apache Iceberg y aplicar automáticamente operaciones de inserción, actualización y eliminación a los registros de las tablas. Firehose garantiza la entrega de una sola vez en las mesas Iceberg. Esta característica requiere el uso de AWS Glue Data Catalog.

Firehose también puede entregar datos de streaming directamente a las tablas de Amazon S3. Estas tablas proporcionan un almacenamiento optimizado para cargas de trabajo de análisis a gran escala e incluyen funciones que mejoran continuamente el rendimiento de las consultas y reducen los costos de almacenamiento de los datos tabulares.

Para obtener información sobre cómo configurar una transmisión Firehose para entregar datos a las tablas de Apache Iceberg, consulte Configurar la transmisión Firehose en la documentación de Firehose o la entrada del blog Transmita datos en tiempo real a tablas de Apache Iceberg en Amazon S3 con Amazon Data Firehose.

Trabajar con tablas Iceberg mediante Athena SQL

Amazon Athena ofrece soporte integrado para Apache Iceberg y no requiere pasos ni configuración adicionales. Esta sección proporciona una descripción detallada de las funciones compatibles y una guía de alto nivel para usar Athena para interactuar con las tablas de Iceberg.

Compatibilidad de versiones y funciones

Soporte para especificaciones de tablas Iceberg

La especificación de la tabla Iceberg de Apache especifica cómo deben comportarse las tablas Iceberg. Athena admite la versión 2 del formato de tabla, por lo que cualquier tabla Iceberg que cree con la consola, la CLI o el SDK utiliza esa versión de forma inherente.

Si usa una tabla Iceberg que se creó con otro motor, como Apache Spark en Amazon EMR AWS Glue, asegúrese de configurar la versión del formato de la tabla mediante las propiedades de la tabla. Como referencia, consulte la sección Creación y escritura de tablas de iceberg, que aparece anteriormente en esta guía.

Compatibilidad con las funciones de Iceberg

Puedes usar Athena para leer y escribir en las tablas de Iceberg. Al cambiar los datos mediante las `DELETE FROM` sentencias `UPDATEMERGE INTO`, y, Athena solo admite el merge-on-read modo. Esta propiedad no se puede cambiar. Para actualizar o eliminar datos con copy-on-write, debe utilizar otros motores como Apache Spark en Amazon EMR o. AWS Glue En la siguiente tabla se resume la compatibilidad con las funciones de Iceberg en Athena.

		Soporte DDL		Soporte DML		AWS Lake Formation por motivos de seguridad (opcional)
	Formato de tabla	Crear tablas	Evolución del esquema	Lectura de datos	Escritura de datos	Control de acceso por

		Soporte DDL		Soporte DML		AWS Lake Formation por motivos de seguridad (opcional)
						filas/columnas
Amazon Athena	Versión 2	✓	✓	✓	X Copy-on-write	✓
					✓ Merge-on-read	✓

> **Note**
>
> - Athena no admite consultas incrementales.
> - En Athena, las operaciones de actualización, eliminación y fusión siempre se utilizan de forma predeterminada para fusionar en lectura (MoR), independientemente de cualquier configuración de copia al escribir (CoW) en las propiedades de la tabla, ya que CoW no es compatible.

Trabajar con tablas Iceberg

Para empezar rápidamente a utilizar Iceberg en Athena, consulte la sección Introducción a las tablas de Iceberg en Athena SQL, que aparece anteriormente en esta guía.

En la siguiente tabla se enumeran las limitaciones y recomendaciones.

Escenario	Limitación	Recomendación
Tabla: generación de DDL	Las tablas de iceberg creadas con otros motores pueden tener propiedades que no	Usa la sentencia equivalente en el motor que creó la

Escenario	Limitación	Recomendación
	están expuestas en Athena. Para estas tablas, no es posible generar el DDL.	tabla (por ejemplo, la `SHOW CREATE TABLE` sentencia de Spark).
Prefijos aleatorios de Amazon S3 en objetos que se escriben en una tabla de Iceberg	De forma predeterminada, las tablas Iceberg que se crean con Athena tienen `write.object-storage.enabled` la propiedad habilitada.	Para deshabilitar este comportamiento y obtener el control total sobre las propiedades de la tabla Iceberg, cree una tabla Iceberg con otro motor, como Spark en Amazon EMR o. AWS Glue
Consultas incrementales	Actualmente no es compatible con Athena.	Para usar consultas incrementales para habilitar canalizaciones de ingesta de datos incrementales, usa Spark en Amazon EMR o. AWS Glue

Trabajar con tablas Iceberg mediante PyIceberg

En esta sección se explica cómo puede interactuar con las tablas Iceberg mediante el uso de. PyIceberg Los ejemplos proporcionados son código repetitivo que puede ejecutar en EC2 instancias, AWS Lambdafunciones o cualquier entorno de Python de Amazon Linux 2023 con las credenciales configuradas correctamente.AWS

Requisitos previos

> **Note**
>
> En estos ejemplos se utiliza la versión 1.9.1. PyIceberg

Para trabajar con PyIceberg él, necesita una PyIceberg AWS SDK para Python (Boto3) instalación. A continuación, se muestra un ejemplo de cómo puede configurar un entorno virtual de Python para trabajar con él PyIceberg y AWS Glue Data Catalog:

1. Descárguelo PyIcebergmediante el instalador del paquete pip python. También necesitas Boto3 para interactuar con él. Servicios de AWS Puede configurar un entorno virtual Python local para realizar pruebas mediante estos comandos:

```
python3 -m venv my_env
cd my_env/bin/
source activate
pip install "pyiceberg[pyarrow,pandas,glue]"
pip install boto3
```

2. Ejecute `python` para abrir el shell de Python y probar los comandos.

Conexión al catálogo de datos

Para empezar a trabajar con las tablas Iceberg AWS Glue, primero tiene que conectarse al AWS Glue Data Catalog.

La `load_catalog` función inicializa una conexión al catálogo de datos mediante la creación de un objeto de catálogo que sirve como interfaz principal para todas las operaciones de Iceberg:

```
from pyiceberg.catalog import load_catalog
region = "us-east-1"

glue_catalog = load_catalog(
    'default',
    **{
        'client.region': region
    },
    type='glue'
)
```

Listar y crear bases de datos

Para enumerar las bases de datos existentes, utilice la `list_namespaces` función:

```
databases = glue_catalog.list_namespaces()
print(databases)
```

Para crear una base de datos nueva, utilice la `create_namespace` función:

```
database_name="mydb"
s3_db_path=f"s3://amzn-s3-demo-bucket/{database_name}"

glue_catalog.create_namespace(database_name, properties={"location": s3_db_path})
```

Crear y escribir tablas Iceberg

Tablas no particionadas

Este es un ejemplo de cómo crear una tabla Iceberg sin particiones mediante la función: `create_table`

```
from pyiceberg.schema import Schema
from pyiceberg.types import NestedField, StringType, DoubleType

database_name="mydb"
table_name="pyiceberg_table"
s3_table_path=f"s3://amzn-s3-demo-bucket/{database_name}/{table_name}"
```

```
schema = Schema(
    NestedField(1, "city", StringType(), required=False),
    NestedField(2, "lat", DoubleType(), required=False),
    NestedField(3, "long", DoubleType(), required=False),
)

glue_catalog.create_table(f"{database_name}.{table_name}", schema=schema,
 location=s3_table_path)
```

Puedes usar la `list_tables` función para comprobar la lista de tablas de una base de datos:

```
tables = glue_catalog.list_tables(namespace=database_name)
print(tables)
```

Puede usar la append función e PyArrow insertar datos dentro de una tabla Iceberg:

```
import pyarrow as pa
df = pa.Table.from_pylist(
    [
        {"city": "Amsterdam", "lat": 52.371807, "long": 4.896029},
        {"city": "San Francisco", "lat": 37.773972, "long": -122.431297},
        {"city": "Drachten", "lat": 53.11254, "long": 6.0989},
        {"city": "Paris", "lat": 48.864716, "long": 2.349014},
    ],
)

table = glue_catalog.load_table(f"{database_name}.{table_name}")
table.append(df)
```

Tablas particionadas

Este es un ejemplo de cómo crear una tabla Iceberg particionada con particiones ocultas mediante la función y: `create_table` `PartitionSpec`

```
from pyiceberg.schema import Schema
from pyiceberg.types import (
    NestedField,
    StringType,
    FloatType,
    DoubleType,
    TimestampType,
```

```
)

# Define the schema
schema = Schema(
    NestedField(field_id=1, name="datetime", field_type=TimestampType(),
 required=True),
    NestedField(field_id=2, name="drone_id", field_type=StringType(), required=True),
    NestedField(field_id=3, name="lat", field_type=DoubleType(), required=False),
    NestedField(field_id=4, name="lon", field_type=DoubleType(), required=False),
    NestedField(field_id=5, name="height", field_type=FloatType(), required=False),
)

from pyiceberg.partitioning import PartitionSpec, PartitionField
from pyiceberg.transforms import DayTransform

partition_spec = PartitionSpec(
    PartitionField(
        source_id=1,  # Refers to "datetime"
        field_id=1000,
        transform=DayTransform(),
        name="datetime_day"
    )
)

database_name="mydb"
partitioned_table_name="pyiceberg_table_partitioned"
s3_table_path=f"s3://amzn-s3-demo-bucket/{database_name}/{partitioned_table_name}"

glue_catalog.create_table(
    identifier=f"{database_name}.{partitioned_table_name}",
    schema=schema,
    location=s3_table_path,
    partition_spec=partition_spec
)
```

Puedes insertar datos en una tabla particionada del mismo modo que en una tabla sin particiones. La partición se gestiona automáticamente.

```
from datetime import datetime
arrow_schema = pa.schema([
    pa.field("datetime", pa.timestamp("us"), nullable=False),
    pa.field("drone_id", pa.string(), nullable=False),
    pa.field("lat", pa.float64()),
```

```
    pa.field("lon", pa.float64()),
    pa.field("height", pa.float32()),
])

data = [
    {
        "datetime": datetime(2024, 6, 1, 12, 0, 0),
        "drone_id": "drone_001",
        "lat": 52.371807,
        "lon": 4.896029,
        "height": 120.5,
    },
    {
        "datetime": datetime(2024, 6, 1, 12, 5, 0),
        "drone_id": "drone_002",
        "lat": 37.773972,
        "lon": -122.431297,
        "height": 150.0,
    },
    {
        "datetime": datetime(2024, 6, 2, 9, 0, 0),
        "drone_id": "drone_001",
        "lat": 53.11254,
        "lon": 6.0989,
        "height": 110.2,
    },
    {
        "datetime": datetime(2024, 6, 2, 9, 30, 0),
        "drone_id": "drone_003",
        "lat": 48.864716,
        "lon": 2.349014,
        "height": 145.7,
    },
]

df = pa.Table.from_pylist(data, schema=arrow_schema)

table = glue_catalog.load_table(f"{database_name}.{partitioned_table_name}")
table.append(df)
```

Lectura de datos

Puede utilizar la PyIceberg `scan` función para leer datos de sus tablas Iceberg. Puede filtrar filas, seleccionar columnas específicas y limitar el número de registros devueltos.

```
table= glue_catalog.load_table(f"{database_name}.{table_name}")
scan_df = table.scan(
    row_filter=(
        f"city = 'Amsterdam'"
    ),
    selected_fields=("city", "lat"),
    limit=100,
).to_pandas()

print(scan_df)
```

Eliminación de datos

La PyIceberg `delete` función le permite eliminar registros de la tabla mediante`delete_filter`:

```
table = glue_catalog.load_table(f"{database_name}.{table_name}")
table.delete(delete_filter="city == 'Paris'")
```

Acceder a los metadatos

PyIceberg proporciona varias funciones para acceder a los metadatos de las tablas. A continuación, le indicamos cómo puede ver la información sobre las instantáneas de las tablas:

```
#List of snapshots
table.snapshots()

#Current snapshot
table.current_snapshot()

#Take a previous snapshot
second_last_snapshot_id=table.snapshots()[-2].snapshot_id
print(f"Second last SnapshotID: {second_last_snapshot_id}")
```

Para obtener una lista detallada de los metadatos disponibles, consulta la sección de referencia de códigos de metadatos de la PyIceberg documentación.

Uso del viaje en el tiempo

Puede utilizar instantáneas de tablas para viajar en el tiempo para acceder a estados anteriores de la tabla. A continuación, te explicamos cómo ver el estado de la tabla antes de la última operación:

```
second_last_snapshot_id=table.snapshots()[-2].snapshot_id

time_travel_df = table.scan(
    limit=100,
    snapshot_id=second_last_snapshot_id
).to_pandas()

print(time_travel_df)
```

Para obtener una lista completa de las funciones disponibles, consulta la documentación de la API de PyIceberg Python.

Trabajando con la especificación de formato de tabla Iceberg versión 3

La versión más reciente de la especificación de formato de tabla Iceberg de Apache es la versión 3. Esta versión presenta capacidades avanzadas para crear lagos de datos a escala de petabytes con un rendimiento mejorado y una reducción de los gastos operativos. Aborda los cuellos de botella de rendimiento más comunes que se presentan en la versión 2, especialmente en lo que respecta a las actualizaciones por lotes y las operaciones de eliminación de conformidad.

AWS proporciona soporte para los vectores de deleción y el linaje de filas, tal como se definen en la especificación de la versión 3 de Iceberg. Estas funciones están disponibles con Apache Spark en las siguientes ubicaciones. Servicios de AWS

Servicio de AWS	Soporte para la versión 3
Amazon EMR para Apache Spark	Amazon EMR versión 7.12 y versiones posteriores
AWS Glue	Sí
AWS Glue: API REST Iceberg, mantenimiento de tablas	Sí
Cuadernos Amazon SageMaker Unified Studio	Sí
Tablas Amazon S3: API REST de Iceberg, mantenimiento de tablas	Sí
Amazon Athena (Trino)	No

Características principales de la versión 3

Los vectores de eliminación sustituyen a los archivos de eliminación posicional que se utilizaban en la versión 2 por un formato binario eficiente almacenado como archivos Puffin. De este modo, se elimina la amplificación de escritura que se produce al realizar actualizaciones aleatorias por lotes, y se eliminan las eliminaciones en cumplimiento del Reglamento General de Protección de Datos

(GDPR), y se reduce considerablemente la sobrecarga que supone mantener los datos actualizados. Organizations que procesan actualizaciones de alta frecuencia verán mejoras inmediatas en el rendimiento de escritura y reducirán los costos de almacenamiento debido a la menor cantidad de archivos pequeños.

El linaje de filas permite un seguimiento preciso de los cambios a nivel de fila. Los sistemas posteriores pueden procesar los cambios de forma gradual, lo que acelera las canalizaciones de datos y reduce los costos informáticos de los flujos de trabajo de captura de datos sobre cambios (CDC). Esta función integrada elimina la necesidad de implementaciones de seguimiento de cambios personalizadas.

Compatibilidad de versiones

La versión 3 mantiene la compatibilidad con versiones anteriores de las tablas de la versión 2. Los servicios de AWS admiten las tablas de la versión 2 y la versión 3 de forma simultánea, por lo que puede:

- Ejecute consultas en las tablas de la versión 2 y la versión 3.
- Actualice las tablas de la versión 2 existentes a la versión 3 sin tener que volver a escribir los datos.
- Ejecute consultas de viaje en el tiempo que abarquen las instantáneas de las versiones 2 y 3.
- Utilice la evolución del esquema y la partición oculta en todas las versiones de la tabla.

Cómo empezar con la versión 3

Requisitos previos

Antes de trabajar con las tablas de la versión 3, asegúrese de tener:

- Y Cuenta de AWS con los permisos AWS Identity and Access Management (IAM) adecuados.
- Acceso a uno o más servicios de AWS análisis (Amazon EMR, cuadernos AWS Glue Amazon SageMaker Unified Studio o Amazon S3 Tables).
- Un depósito de S3 para almacenar los datos y metadatos de las tablas.
- Un bucket de mesa para empezar a utilizar Amazon S3 Tables o un bucket S3 de uso general si está creando su propia infraestructura Iceberg.
- Un catálogo configurado. AWS Glue

Creación de tablas de la versión 3

Creación de nuevas tablas

Para crear una nueva tabla de la versión 3 de Iceberg, establezca la propiedad de la `format-version` tabla en 3.

Con Spark SQL:

```
CREATE TABLE IF NOT EXISTS myns.orders_v3 (
    order_id bigint,
    customer_id string,
    order_date date,
    total_amount decimal(10,2),
    status string,
    created_at timestamp
)
USING iceberg
TBLPROPERTIES (
    'format-version' = '3'
)
```

Actualización de las tablas de la versión 2 a la versión 3

Puede actualizar las tablas de la versión 2 existentes a la versión 3 de forma atómica sin tener que volver a escribir los datos.

Con Spark SQL:

```
ALTER TABLE myns.existing_table
SET TBLPROPERTIES ('format-version' = '3')
```

> **Important**
>
> La versión 3 es una actualización unidireccional. Después de actualizar una tabla de la versión 2 a la versión 3, no se puede volver a degradar a la versión 2 mediante las operaciones estándar.

Qué sucede durante la actualización:

- Se crea una nueva instantánea de metadatos de forma atómica.
- Los archivos de datos de Parquet existentes se reutilizan.
- Los campos de linaje de filas se añaden a los metadatos de la tabla.

Después de la actualización

- La siguiente compactación eliminará los archivos de eliminación de la versión 2 anterior.
- Las nuevas modificaciones utilizarán los archivos vectoriales de eliminación de la versión 3.

La actualización no rellena de forma histórica los registros de seguimiento de los cambios de linaje de las filas.

Habilitar los vectores de deleción

Para aprovechar los vectores de eliminación para actualizar, eliminar y fusionar, configure el modo de escritura.

Con Spark SQL:

```
ALTER TABLE myns.orders_v3
SET TBLPROPERTIES ('format-version' = '3',
                   'write.delete.mode' = 'merge-on-read',
                   'write.update.mode' = 'merge-on-read',
                   'write.merge.mode' = 'merge-on-read'
                  )
```

Esta configuración garantiza que las operaciones de actualización, eliminación y fusión creen archivos vectoriales de eliminación en lugar de reescribir archivos de datos completos.

Uso del linaje de filas para el seguimiento de cambios

La versión 3 agrega automáticamente campos de metadatos de linaje de filas para realizar un seguimiento de los cambios.

Con Spark SQL:

```
# Query with parameter value provided
last_processed_sequence = 47
```

```
SELECT
    id,
    data,
    _row_id,
    _last_updated_sequence_number
FROM myns.orders_v3
WHERE _last_updated_sequence_number > :last_processed_sequence
```

El `_row_id` campo identifica de forma única cada fila y `_last_updated_sequence_number` registra cuándo se modificó la fila por última vez. Utilice estos campos para:

- Identifique las filas modificadas para su procesamiento incremental.
- Realice un seguimiento del linaje de datos para comprobar el cumplimiento.
- Optimice las canalizaciones de los CDC.
- Reduzca los costos de cómputo procesando solo los cambios.

Prácticas recomendadas para la versión 3

Cuándo usar la versión 3

Considere la posibilidad de actualizar a la versión 3 o comenzar con ella cuando:

- Realiza actualizaciones o eliminaciones por lotes con frecuencia.
- Debes cumplir con el GDPR o con los requisitos de conformidad con las eliminaciones.
- Sus cargas de trabajo implican interrupciones de alta frecuencia.
- Necesita flujos de trabajo de los CDC eficientes.
- Desea reducir los costos de almacenamiento de los archivos pequeños.
- Necesita mejores capacidades de seguimiento de cambios.

Optimizar el rendimiento de escritura

- Habilite los vectores de eliminación para cargas de trabajo con un uso intensivo de actualizaciones:

```
SET TBLPROPERTIES (
'write.delete.mode' = 'merge-on-read',
'write.update.mode' = 'merge-on-read',
```

```
'write.merge.mode' = 'merge-on-read'
)
```

- Configure los tamaños de archivo adecuados:

```
SET TBLPROPERTIES (
'write.target-file-size-bytes' = '536870912'  — 512 MB
)
```

Optimización del rendimiento de lectura

- Utilice el linaje de filas para el procesamiento incremental.
- Utilice el viaje en el tiempo para acceder a los datos históricos sin necesidad de copiarlos.
- Habilite la recopilación de estadísticas para planificar mejor las consultas.

Estrategia de migración

Al migrar de la versión 2 a la versión 3, siga estas prácticas recomendadas:

- Realice primero la prueba en un entorno que no sea de producción para validar el proceso de actualización y el rendimiento.
- Actualice durante los períodos de baja actividad para minimizar el impacto en las operaciones simultáneas.
- Supervise el rendimiento inicial y realice un seguimiento de las métricas después de la actualización.
- Ejecute la compactación para consolidar los archivos eliminados después de la actualización.
- Actualiza la documentación de tu equipo para que refleje las características de la versión 3.

Consideraciones sobre compatibilidad

- Versiones del motor: asegúrate de que todos los motores que acceden a la tabla sean compatibles con la versión 3.
- Herramientas de terceros: compruebe la compatibilidad de la herramienta con la versión 3 antes de realizar la actualización.
- Estrategia de Backup: pruebe los procedimientos de recuperación basados en instantáneas.

- Supervisión: actualice los paneles de supervisión para las métricas específicas de la versión 3.

Resolución de problemas

Problemas comunes

Error: «No se admite el formato de la versión 3»

- Compruebe que la versión de su motor sea compatible con la versión 3. Para obtener información específica, consulte la tabla al principio de esta sección.
- Compruebe la compatibilidad del catálogo.
- Asegúrese de utilizar las versiones más recientes de Servicios de AWS.

Degradación del rendimiento tras la actualización

- Compruebe que no haya errores de compactación. Para obtener más información, consulte Registro y supervisión de tablas S3 en la documentación de Amazon S3.
- Confirme que los vectores de eliminación estén habilitados. Se deben establecer las siguientes propiedades:

```
SET TBLPROPERTIES (
'write.delete.mode' = 'merge-on-read',
'write.update.mode' = 'merge-on-read',
'write.merge.mode' = 'merge-on-read'
)
```

  Puede verificar las propiedades de la tabla con el siguiente código:

```
DESCRIBE FORMATTED myns.orders_v3
```

- Revise su estrategia de particiones. El exceso de particiones puede provocar archivos pequeños. Ejecute la siguiente consulta para obtener el tamaño medio de los archivos de la tabla:

```
SELECT avg(file_size_in_bytes) as avg_file_size_bytes
FROM myns.orders_v3.files
```

Incompatibilidad con herramientas de terceros

- Compruebe que la herramienta es compatible con la especificación de la versión 3.
- Considere la posibilidad de mantener las tablas de la versión 2 para las herramientas no compatibles.
- Póngase en contacto con el proveedor de la herramienta para conocer el cronograma de soporte de la versión 3.

Obtener ayuda

- Para problemas Servicio de AWS específicos, póngase en contacto con AWS Support.
- Para obtener ayuda de la comunidad de Iceberg, usa el canal de Slack de Iceberg.
- Para obtener información sobre cómo Servicios de AWS gestionar tus cargas de trabajo de análisis, consulta Analytics en. AWS

Precios

- Precios de cómputo y almacenamiento de Amazon EMR
- SageMakerPrecios de Amazon
- AWS Glue precios del catálogo de datos y ejecución de tareas
- S3 mide el almacenamiento y solicita los precios

Disponibilidad.

La compatibilidad con la versión 3 de la especificación de formato de tabla Iceberg está disponible en todos los Regiones de AWS lugares en los que funcionan las tablas Amazon EMR AWS Glue AWS Glue Data Catalog,, y S3. Para obtener información sobre la disponibilidad en regiones, consulte Servicios de AWS by Region.

Recursos adicionales

- Documentación de Apache Iceberg
- Especificaciones de la mesa Apache Iceberg
- Guía para migrar datos tabulares de Amazon S3 a S3 Tables
- Tutorial: Cómo empezar con las tablas de S3

Migración de tablas existentes a Iceberg

Esta sección se centra en la migración de las tablas de estilo Hive existentes al formato Iceberg. Se aplica a las tablas que utilizan formatos tradicionales compatibles con Hive, como Apache Parquet o Apache ORC. Esta información no se aplica a las tablas que ya utilizan formatos de tabla modernos, como Linux Foundation, Delta Lake o Apache Hudi.

Para migrar sus tablas actuales de estilo Hive al formato Iceberg, puede utilizar la migración de datos local o completa:

- La migración in situ es el proceso de generar los archivos de metadatos de Iceberg sobre los archivos de datos existentes.
- La migración completa de datos crea la capa de metadatos de Iceberg y también reescribe los archivos de datos existentes de la tabla original a la nueva tabla de Iceberg.

En las siguientes secciones se proporciona una descripción detallada de cada método de migración, incluidas step-by-step las instrucciones y consideraciones para su implementación. Para obtener más información sobre estas estrategias de migración, consulte la sección sobre migración de tablas de la documentación de Iceberg.

Tras revisar los detalles de los métodos de migración de datos internos y completos, consulta las dos secciones clave siguientes para ayudarte en el proceso de toma de decisiones:

- La elección de una estrategia de migración proporciona orientación a través de una serie de preguntas y escenarios que le ayudarán a determinar el enfoque de migración más adecuado en función de sus requisitos y casos de uso específicos.
- El resumen de las opciones de migración proporciona una tabla completa en la que se comparan las características y consideraciones clave de las diferentes opciones de migración. Esta tabla sirve como guía de referencia rápida y ofrece una comparación de características para ayudarle a comprender las desventajas técnicas entre los métodos.

Migración in situ

La migración in situ elimina la necesidad de volver a escribir todos los archivos de datos. En su lugar, los archivos de metadatos de Iceberg se generan y se vinculan a los archivos de datos existentes.

Este método suele ser más rápido y rentable, especialmente para conjuntos de datos o tablas de gran tamaño que tienen formatos de archivo compatibles, como Parquet, Avro y ORC.

> **Note**
>
> La migración in situ no se puede utilizar al migrar a Amazon S3 Tables.

Iceberg ofrece dos opciones principales para implementar la migración in situ:

- Se utiliza el procedimiento de captura de pantalla para crear una nueva tabla de Iceberg y, al mismo tiempo, mantener la tabla de origen sin cambios. Para obtener más información, consulte la tabla de instantáneas en la documentación de Iceberg.
- Uso del procedimiento de migración para crear una nueva tabla de Iceberg como sustitución de la tabla de origen. Para obtener más información, consulte Migrate Table en la documentación de Iceberg. Aunque este procedimiento funciona con Hive Metastore (HMS), actualmente no es compatible con. AWS Glue Data Catalog El procedimiento de replicación de la tabla, que se describe más AWS Glue Data Catalog adelante en esta guía, proporciona una solución alternativa para lograr un resultado similar con el catálogo de datos.

Después de realizar la migración in situ mediante uno `snapshot` o varios archivos de datos`migrate`, es posible que algunos archivos de datos permanezcan sin migrar. Esto suele ocurrir cuando los escritores siguen escribiendo en la tabla de origen durante o después de la migración. Para incorporar los archivos restantes a la tabla Iceberg, puede utilizar el procedimiento add_files. Para obtener más información, consulte Añadir archivos en la documentación de Iceberg.

Supongamos que tiene una `products` tabla basada en Parquet que se creó y rellenó en Athena de la siguiente manera:

```
CREATE EXTERNAL TABLE mydb.products (
    product_id INT,
    product_name STRING
)
PARTITIONED BY (category STRING)
STORED AS PARQUET
LOCATION 's3://amzn-s3-demo-bucket/products/';

INSERT INTO mydb.products
VALUES
```

```
    (1001, 'Smartphone', 'electronics'),
    (1002, 'Laptop', 'electronics'),
    (2001, 'T-Shirt', 'clothing'),
    (2002, 'Jeans', 'clothing');
```

En las siguientes secciones se explica cómo puede utilizar los `migrate` procedimientos `snapshot` y con esta tabla.

Opción 1: procedimiento de captura de pantalla

El `snapshot` procedimiento crea una nueva tabla Iceberg con un nombre diferente, pero que replica el esquema y la partición de la tabla de origen. Esta operación deja la tabla de origen completamente inalterada durante y después de la acción. De forma eficaz, crea una copia ligera de la tabla, que resulta especialmente útil para probar escenarios o explorar datos sin correr el riesgo de modificar la fuente de datos original. Este enfoque permite un período de transición en el que tanto la tabla original como la tabla Iceberg permanecen disponibles (consulte las notas al final de esta sección). Una vez finalizadas las pruebas, puede pasar la nueva mesa Iceberg a la de producción haciendo la transición de todos los grabadores y lectores a la nueva tabla.

Puede ejecutar el `snapshot` procedimiento mediante Spark en cualquier modelo de implementación de Amazon EMR (por ejemplo, Amazon EMR en EC2, Amazon EMR en EKS, EMR Serverless) y. AWS Glue

Para probar la migración in situ con el procedimiento de Spark, siga estos pasos`snapshot`:

1. Abre una aplicación de Spark y configura la sesión de Spark con los siguientes ajustes:
   - `"spark.sql.extensions":"org.apache.iceberg.spark.extensions.IcebergSparkSess`
   - `"spark.sql.catalog.spark_catalog":"org.apache.iceberg.spark.SparkSessionCata`
   - `"spark.sql.catalog.spark_catalog.type":"glue"`
   - `"spark.hadoop.hive.metastore.client.factory.class":"com.amazonaws.glue.catal`
2. Ejecute el `snapshot` procedimiento para crear una nueva tabla de Iceberg que apunte a los archivos de datos de la tabla original:

```
spark.sql(f"""
CALL system.snapshot(
source_table => 'mydb.products',
table => 'mydb.products_iceberg',
location => 's3://amzn-s3-demo-bucket/products_iceberg/'
)
```

```
"""
).show(truncate=False)
```

El marco de datos de salida contiene `imported_files_count` (el número de archivos que se agregaron).

3. Valide la nueva tabla consultándola:

```
spark.sql(f"""
SELECT * FROM mydb.products_iceberg LIMIT 10
"""
).show(truncate=False)
```

Notas

- Tras ejecutar el procedimiento, cualquier modificación del archivo de datos en la tabla de origen desincronizará la tabla generada. Los archivos nuevos que añada no estarán visibles en la tabla Iceberg y los archivos que elimine afectarán a las capacidades de consulta de la tabla Iceberg. Para evitar los problemas de sincronización:
  - Si la nueva tabla Iceberg está destinada a ser utilizada en producción, detenga todos los procesos que escriban en la tabla original y redirija los procesos a la nueva tabla.
  - Si necesita un período de transición o si la nueva tabla de Iceberg es para realizar pruebas, consulte Mantener sincronizadas las tablas de Iceberg tras una migración local, más adelante en esta sección, para obtener instrucciones sobre cómo mantener la sincronización de las tablas.
- Al utilizar `snapshot` este procedimiento, la `gc.enabled` propiedad se establece `false` en las propiedades de la tabla Iceberg creada. Esta configuración prohíbe acciones como `expire_snapshotsremove_orphan_files`, o `DROP TABLE` con la `PURGE` opción, que eliminarían físicamente los archivos de datos. Las operaciones de eliminación o fusión tipo iceberg, que no afectan directamente a los archivos fuente, siguen estando permitidas.
- Para que su nueva tabla Iceberg sea completamente funcional, sin límites en cuanto a las acciones que eliminan físicamente los archivos de datos, puede cambiar la propiedad de la `gc.enabled` tabla a. `true` Sin embargo, esta configuración permitirá realizar acciones que afecten a los archivos de datos de origen, lo que podría dañar el acceso a la tabla original. Por lo tanto, cambie la `gc.enabled` propiedad solo si ya no necesita mantener la funcionalidad de la tabla original. Por ejemplo:

```
spark.sql(f"""
ALTER TABLE mydb.products_iceberg
SET TBLPROPERTIES ('gc.enabled' = 'true');
""")
```

Opción 2: procedimiento de migración

El `migrate` procedimiento crea una nueva tabla Iceberg que tiene el mismo nombre, esquema y partición que la tabla de origen. Cuando se ejecuta este procedimiento, bloquea la tabla de origen y le cambia el nombre a `<table_name>_BACKUP_` (o a un nombre personalizado especificado en el parámetro del `backup_table_name` procedimiento).

> **Note**
>
> Si establece el parámetro de `drop_backup` procedimiento en`true`, la tabla original no se conservará como copia de seguridad.

En consecuencia, el procedimiento de la `migrate` tabla requiere que se detengan todas las modificaciones que afecten a la tabla de origen antes de realizar la acción. Antes de ejecutar el `migrate` procedimiento:

- Detenga todos los grabadores que interactúen con la tabla fuente.
- Modifique los lectores y escritores que no sean compatibles de forma nativa con Iceberg para habilitar la compatibilidad con Iceberg.

Por ejemplo:

- Athena sigue trabajando sin modificaciones.
- Spark requiere:
  - Archivos Iceberg Java Archive (JAR) que se incluirán en la ruta de clases (consulte las secciones Cómo trabajar con Iceberg en Amazon EMR y Trabajar con Iceberg AWS Glue en las secciones anteriores de esta guía).

- Las siguientes configuraciones del catálogo de sesiones de Spark (se utilizan `SparkSessionCatalog` para añadir compatibilidad con Iceberg y, al mismo tiempo, mantener las funcionalidades de catálogo integradas para tablas que no son de Iceberg):
  - `"spark.sql.extensions":"org.apache.iceberg.spark.extensions.IcebergSparkSes`
  - `"spark.sql.catalog.spark_catalog":"org.apache.iceberg.spark.SparkSessionCat`
  - `"spark.sql.catalog.spark_catalog.type":"glue"`
  - `"spark.hadoop.hive.metastore.client.factory.class":"com.amazonaws.glue.cata`

Tras ejecutar el procedimiento, puedes reiniciar las grabadoras con su nueva configuración de Iceberg.

Actualmente, el `migrate` procedimiento no es compatible con el AWS Glue Data Catalog, porque el catálogo de datos no admite la `RENAME` operación. Por lo tanto, le recomendamos que utilice este procedimiento únicamente cuando trabaje con Hive Metastore. Si utiliza el catálogo de datos, consulte la siguiente sección para ver un enfoque alternativo.

Puede ejecutar el `migrate` procedimiento en todos los modelos de implementación de Amazon EMR (Amazon EMR en EC2, Amazon EMR en EKS, EMR Serverless) y AWS Glue, pero requiere una conexión configurada a Hive Metastore. Amazon EMR en EC2 es la opción recomendada porque proporciona una configuración integrada de Hive Metastore, que minimiza la complejidad de la configuración.

Para probar la migración in situ con el procedimiento `migrate` Spark desde un clúster de Amazon EMR en EC2 configurado con Hive Metastore, siga estos pasos:

1. Inicie una aplicación de Spark y configure la sesión de Spark para usar la implementación del catálogo de Iceberg Hive. Por ejemplo, si utiliza la `pyspark` CLI:

```
pyspark --conf
 spark.sql.extensions=org.apache.iceberg.spark.extensions.IcebergSparkSessionExtensions
 --conf spark.sql.catalog.spark_catalog=org.apache.iceberg.spark.SparkSessionCatalog
 --conf spark.sql.catalog.spark_catalog.type=hive
```

2. Cree una `products` tabla en Hive Metastore. Esta es la tabla de origen, que ya existe en una migración típica.
   a. Cree la tabla Hive `products` externa en Hive Metastore para que apunte a los datos existentes en Amazon S3:

```
spark.sql(f"""
CREATE EXTERNAL TABLE products (
    product_id INT,
    product_name STRING
)
PARTITIONED BY (category STRING)
STORED AS PARQUET
LOCATION 's3://amzn-s3-demo-bucket/products/';
"""
)
```

b. Agregue las particiones existentes mediante el comando: `MSCK REPAIR TABLE`

```
spark.sql(f"""
MSCK REPAIR TABLE products
"""
)
```

c. Confirme que la tabla contiene datos ejecutando una `SELECT` consulta:

```
spark.sql(f"""
SELECT * FROM products
"""
).show(truncate=False)
```

Código de salida de ejemplo:

```
>>> spark.sql(f"""
... SELECT * FROM products
... """
... ).show(truncate=False)
+----------+------------+-----------+
|product_id|product_name|category   |
+----------+------------+-----------+
|1001      |Smartphone  |electronics|
|1002      |Laptop      |electronics|
|2001      |T-Shirt     |clothing   |
|2002      |Jeans       |clothing   |
+----------+------------+-----------+
```

3. Utilice el procedimiento de Iceberg: `migrate`

```
df_res=spark.sql(f"""
```

```
CALL system.migrate(
table => 'default.products'
)
"""
)

df_res.show()
```

El marco de datos de salida contiene `migrated_files_count` (el número de archivos que se agregaron a la tabla Iceberg):

```
>>> df_res.show()
+-------------------+
|migrated_files_count|
+-------------------+
|                  2|
+-------------------+
```

4. Confirme que se creó la tabla de respaldo:

```
spark.sql("show tables").show()
```

Código de salida de ejemplo:

```
>>> spark.sql("show tables").show()
+---------+---------------+-----------+
|namespace|      tableName|isTemporary|
+---------+---------------+-----------+
|  default|       products|      false|
|  default|products_backup_|      false|
+---------+---------------+-----------+
```

5. Valide la operación consultando la tabla de Iceberg:

```
spark.sql(f"""
SELECT * FROM products
"""
).show(truncate=False)
```

Notas

- Tras ejecutar el procedimiento, todos los procesos actuales que consultan o escriben en la tabla de origen se verán afectados si no están configurados correctamente con el soporte de Iceberg. Por lo tanto, le recomendamos que siga estos pasos:
  1. Detenga todos los procesos utilizando la tabla de origen antes de la migración.
  2. Realice la migración.
  3. Reactive los procesos utilizando la configuración de Iceberg adecuada.
- Si se producen modificaciones en los archivos de datos durante el proceso de migración (se añaden archivos nuevos o se eliminan archivos), la tabla generada no estará sincronizada. Para ver las opciones de sincronización, consulte Mantener sincronizadas las tablas de Iceberg tras una migración local más adelante en esta sección.

Replicar el procedimiento de migración de tablas en AWS Glue Data Catalog

Puede replicar el resultado del procedimiento de migración AWS Glue Data Catalog (haciendo una copia de seguridad de la tabla original y sustituyéndola por una tabla Iceberg) siguiendo estos pasos:

1. Utilice el procedimiento de captura de pantalla para crear una nueva tabla de iceberg que apunte a los archivos de datos de la tabla original.
2. Haga una copia de seguridad de los metadatos de la tabla original en el catálogo de datos:
   a. Utilice la GetTableAPI para recuperar la definición de la tabla de origen.
   b. Usa la GetPartitionsAPI para recuperar la definición de partición de la tabla fuente.
   c. Utilice la CreateTableAPI para crear una tabla de respaldo en el catálogo de datos.
   d. Utilice la BatchCreatePartitionAPI CreatePartitiono para registrar las particiones en la tabla de respaldo del catálogo de datos.
3. Cambie la propiedad de la tabla `gc.enabled` Iceberg `false` a para habilitar todas las operaciones de la tabla.
4. Elimine la tabla original.
5. Localice el archivo JSON de metadatos de la tabla Iceberg en la carpeta de metadatos de la ubicación raíz de la tabla.

6. Registre la nueva tabla en el catálogo de datos mediante el procedimiento register_table con el nombre de la tabla original y la ubicación del `metadata.json` archivo que se creó mediante el procedimiento: `snapshot`

```
spark.sql(f"""
CALL system.register_table(
    table => 'mydb.products',
    metadata_file => '{iceberg_metadata_file}'
)
"""
).show(truncate=False)
```

Mantener sincronizadas las tablas de Iceberg tras la migración in situ

El `add_files` procedimiento proporciona una forma flexible de incorporar los datos existentes en las tablas de Iceberg. En concreto, registra los archivos de datos existentes (como los archivos Parquet) haciendo referencia a sus rutas absolutas en la capa de metadatos de Iceberg. De forma predeterminada, el procedimiento agrega archivos de todas las particiones de tabla a una tabla de Iceberg, pero puede agregar archivos de particiones específicas de forma selectiva. Este enfoque selectivo es particularmente útil en varios escenarios:

- Cuando se agregan nuevas particiones a la tabla de origen después de la migración inicial.
- Cuando se agregan o eliminan archivos de datos de las particiones existentes después de la migración inicial. Sin embargo, si se vuelven a añadir particiones modificadas, primero se debe eliminar la partición. Más adelante en esta sección se proporciona más información al respecto.

Estas son algunas consideraciones para utilizar el `add_file` procedimiento después de realizar (`snapshotomigrate`) la migración in situ, a fin de mantener la nueva tabla de Iceberg sincronizada con los archivos de datos de origen:

- Cuando se agreguen nuevos datos a las nuevas particiones de la tabla de origen, utilice el `add_files` procedimiento con la `partition_filter` opción de incorporar estas adiciones de forma selectiva en la tabla de Iceberg:

```
spark.sql(f"""
CALL system.add_files(
source_table => 'mydb.products',
table => 'mydb.products_iceberg',
```

```
partition_filter => map('category', 'electronics')
).show(truncate=False)
```

o bien:

```
spark.sql(f"""
CALL system.add_files(
source_table => '`parquet`.`s3://amzn-s3-demo-bucket/products/`',
table => 'mydb.products_iceberg',
partition_filter => map('category', 'electronics')
).show(truncate=False)
```

- El `add_files` procedimiento busca archivos en toda la tabla de origen o en particiones específicas cuando se especifica la `partition_filter` opción e intenta añadir todos los archivos que encuentre a la tabla de Iceberg. De forma predeterminada, la propiedad del `check_duplicate_files` procedimiento está establecida en`true`, lo que impide que el procedimiento se ejecute si los archivos ya existen en la tabla Iceberg. Esto es importante porque no hay una opción integrada para omitir los archivos agregados anteriormente y, al deshabilitarlos, los archivos se `check_duplicate_files` añadirán dos veces, lo que generará duplicados. Cuando se agreguen nuevos archivos a la tabla de origen, siga estos pasos:

  1. Para particiones nuevas, `add_files` utilícelo con `partition_filter` a para importar solo los archivos de la nueva partición.
  2. En el caso de las particiones existentes, elimine primero la partición de la tabla de Iceberg y, a continuación, vuelva a ejecutarla `add_files` para esa partición, especificando la. `partition_filter` Por ejemplo:

     ```
     # We initially perform in-place migration with snapshot
     spark.sql(f"""
     CALL system.snapshot(
     source_table => 'mydb.products',
     table => 'mydb.products_iceberg',
     location => 's3://amzn-s3-demo-bucket/products_iceberg/'
     )
     """
     ).show(truncate=False)

     # Then on the source table, some new files were generated under the
      category='electronics' partition. Example:
     spark.sql("""
     INSERT INTO mydb.products
     ```

```
VALUES (1003, 'Tablet', 'electronics')
""")

# We delete the modified partition from the Iceberg table. Note this is a metadata
 operation only
spark.sql("""
DELETE FROM mydb.products_iceberg WHERE category = 'electronics'
""")

# We add_files from the modified partition
spark.sql("""
CALL system.add_files(
  source_table => 'mydb.products',
  table => 'mydb.products_iceberg',
  partition_filter => map('category', 'electronics')
)
""").show(truncate=False)
```

> **Note**
>
> Cada `add_files` operación genera una nueva instantánea de la tabla de Iceberg con datos adjuntos.

Elegir la estrategia de migración local adecuada

Para elegir la mejor estrategia de migración local, tenga en cuenta las preguntas de la siguiente tabla.

Pregunta	Recomendación	Explicación
¿Desea realizar una migración rápida sin tener que volver a escribir los datos y, al mismo tiempo, mantener accesibles las tablas de Hive e Iceberg para realizar pruebas o realizar una transición gradual?	`snapshot`procedimiento seguido de procedimiento `add_files`	Utilice el `snapshot` procedimiento para crear una nueva tabla Iceberg clonando el esquema y haciendo referencia a los archivos de datos, sin modificar la tabla de origen. Utilice el `add_files` procedimiento para incorpora

Pregunta	Recomendación	Explicación
		r las particiones que se agregaron o modificaron después de la migración, teniendo en cuenta que volver a agregar las particiones modificadas requiere primero eliminarlas.
¿Utiliza Hive Metastore y desea reemplazar la tabla de Hive por una tabla de Iceberg inmediatamente, sin tener que volver a escribir los datos?	`migrate` `add_files` procedimiento seguido de procedimiento	Utilice `migrate` este procedimiento para crear una tabla Iceberg, hacer una copia de seguridad de la tabla de origen y sustituir la tabla original por la versión Iceberg. Nota: Esta opción es compatible con Hive Metastore, pero no con. AWS Glue Data Catalog Utilice el `add_files` procedimiento para incorpora r las particiones que se agregaron o modificaron después de la migración , teniendo en cuenta que volver a añadir las particiones modificadas requiere primero eliminarlas.

Pregunta	Recomendación	Explicación
¿Está utilizando la tabla Hive por una tabla Iceberg AWS Glue Data Catalog y desea sustituirla inmediatamente, sin tener que volver a escribir los datos?	Adaptación del `migrate` procedimiento, seguida del procedimiento `add_files`	Replicar `migrate` el comportamiento del procedimiento: 1. Se utiliza `snapshot` para crear una tabla de iceberg. 2. Realice una copia de seguridad de los metadatos de la tabla original mediante AWS Glue APIs. 3. Actívela `gc.enabled` en las propiedades de la tabla Iceberg. 4. Elimine la tabla original. 5. Se utiliza `register_table` para crear una nueva entrada de tabla con el nombre original. Nota: Esta opción requiere la gestión manual de las llamadas a la AWS Glue API para la copia de seguridad de los metadatos. Utilice el `add_files` procedimiento para incorporar las particiones que se agregaron o modificaron después de la migración. Tenga en cuenta que, para volver a agregar las particiones modificadas, primero es necesario eliminarlas.

Migración completa de datos

La migración completa de datos recrea los archivos de datos y los metadatos. Este enfoque lleva más tiempo y requiere recursos informáticos adicionales en comparación con la migración in situ. Sin embargo, la migración completa de los datos ofrece importantes oportunidades para mejorar la calidad de las tablas y optimizar los patrones de acceso y almacenamiento de datos.

Durante la migración completa de los datos, puede realizar varias operaciones beneficiosas, como la validación de los datos para garantizar su integridad y corrección, las modificaciones del esquema para cumplir mejor los requisitos actuales y los ajustes de la estrategia de partición para mejorar el rendimiento de las consultas. También puede reordenar los datos para optimizar los patrones de acceso comunes, implementar la partición oculta de Iceberg para mejorar la eficacia de las consultas y realizar la conversión del formato de archivo (por ejemplo, de CSV a Parquet) si lo desea.

Estas capacidades hacen que la migración completa de datos sea ideal para la transición al formato Iceberg y para refinar y optimizar de manera integral su estrategia de almacenamiento de datos. Si bien la migración completa de los datos requiere más tiempo y recursos por adelantado, las mejoras resultantes en la calidad de los datos, la organización y el rendimiento de las consultas pueden proporcionar beneficios a largo plazo. Para implementar una migración de datos completa, utilice una de las siguientes opciones:

- Usa la declaración `CREATE TABLE ... AS SELECT` (CTAS) en Spark (en Amazon EMR AWS Glue o) o en Athena. Puede establecer la especificación de partición y las propiedades de la tabla de la nueva tabla Iceberg mediante las cláusulas y. `PARTITIONED BY` `TBLPROPERTIES` Puede cambiar el esquema y las particiones de la nueva tabla según sus necesidades en lugar de heredarlos de la tabla de origen.
- Lee la tabla de origen y escribe los datos como una nueva tabla Iceberg mediante Spark en Amazon AWS Glue EMR o. Para obtener más información, consulta Cómo crear una tabla en la documentación de Iceberg.

Elección de una estrategia de migración

Al realizar la transición al formato Iceberg, es crucial elegir entre la migración local o la migración completa. Para determinar el enfoque más adecuado para sus necesidades específicas, tenga en cuenta las siguientes preguntas y recomendaciones:

Pregunta	Recomendación
¿Qué formato tiene el archivo de datos (por ejemplo, CSV o Apache Parquet)?	• Considere la posibilidad de migrar in situ si el formato de su archivo de tabla es Parquet, ORC o Avro. • Para otros formatos, como CSV, JSON, etc., utilice la migración de datos completa.
¿Desea actualizar o consolidar el esquema de la tabla?	• Si desea evolucionar el esquema de la tabla mediante las capacidades nativas de Iceberg, considere la posibilidad de migrar in situ. Por ejemplo, puede cambiar el nombre de las columnas después de la migración. (El esquema se puede cambiar en la capa de metadatos de Iceberg). • Si desea eliminar columnas enteras porque ya no son necesarias, le recomendamos que utilice la migración de datos completa.
¿Se beneficiaría la tabla de cambiar la estrategia de partición?	• Si el enfoque de particionamiento de Iceberg cumple con sus requisitos (por ejemplo, los datos nuevos se almacenan utilizando el nuevo diseño de particiones mientras que las particiones existentes permanecen como están), considere la posibilidad de migrar in situ. • Si desea utilizar particiones ocultas en la tabla, considere la posibilidad de migrar todos los datos. Para obtener más información sobre las particiones ocultas, consulte la sección de prácticas recomendadas.
¿Sería beneficioso para la tabla añadir o cambiar la estrategia de ordenación?	• Para añadir o cambiar el orden de clasificación de los datos, es necesario volver a escribir el conjunto de datos. En este caso, considere la posibilidad de utilizar la migración de datos completa. • En el caso de tablas grandes en las que resulta prohibitivamente caro reescribir todas las particiones de la tabla, considere la posibilidad de migrar in situ y ejecutar la compactación (con la clasificación habilitada) para las particiones a las que se accede con más frecuencia.

Pregunta	Recomendación
¿La tabla tiene muchos archivos pequeños?	• La combinación de archivos pequeños en archivos más grandes requiere volver a escribir el conjunto de datos. En este caso, considere la posibilidad de utilizar la migración de datos completa. • En el caso de tablas grandes en las que resulta prohibiti vamente caro reescribir todas las particiones de la tabla, considere la posibilidad de migrar in situ y ejecutar la compactación (con la clasificación habilitada) para las particiones a las que se accede con más frecuencia.

Resumen de las opciones de migración

En esta tabla se resumen las principales características y consideraciones de cada opción de migración.

Caracter stica	Migración in situ instantánea	Migración in situ migrate	Migración completa de datos CTAS o (CREAR TABLA + INSERTAR)
Mejoras en el diseño de los datos como parte del proceso de migració			

Caracter stica	Migración in situ instantánea	Migración in situ migrate	Migración completa de datos CTAS o (CREAR TABLA + INSERTAR)
• Reord los datos	⊗No	⊗No	⊘Sí
• Cambi la partici (por ejempl para usar la partici oculta de Iceber	⊗No	⊗No	⊘Sí
• Cambi el esque de la tabla	⊗No	⊗No	⊘Sí
• Optimi el tamañ del archiv	⊗No	⊗No	⊘Sí

Caracter stica	Migración in situ instantánea	Migración in situ migrate	Migración completa de datos CTAS o (CREAR TABLA + INSERTAR)
• Valide el esquer de los datos exister s antes de añadirl	⊗No	⊗No	⊘Sí
Formato: de archivo compatib es	Parquet, Avro, ORC	Parquet, Avro, ORC	Parquet, Avro, ORC, JSON, CSV
Sustituci ón de la tabla fuente por una tabla Iceberg	⊗No (crea una tabla nueva, pero con pasos adicional es puede reemplazar la tabla de origen)	⊘Sí (crea una tabla de respaldo y sustituye la tabla de origen por una tabla Iceberg)	⊗No (crea una tabla nueva)

Caracter stica	Migración in situ instantánea	Migración in situ migrate	Migración completa de datos CTAS o (CREAR TABLA + INSERTAR)
Impacto en la tabla de origen			
• Operaes de elimina ón de archivo en la tabla Iceber (expi: apsho operac es, elimina ón de una tabla con purga)	Corrompe la tabla de fuentes	Daña la tabla de respaldo	Seguro, la fuente no se ve afectada

Caracter stica	Migración in situ instantánea	Migración in situ migrate	Migración completa de datos CTAS o (CREAR TABLA + INSERTAR)
Impacto en una mesa de iceberg			
• Impact si se elimina los archivo de la tabla fuente	Daña la tabla Iceberg	Corrompe la mesa Iceberg	No afecta a la mesa Iceberg
• Impact si se añade nuevos archivo en la ubicac de la tabla de origen	No está visible en la tabla nueva (es necesario incorporar una partición con`add_files`)	No está visible en la nueva tabla (es necesario incorpora r una partición con`add_files`)	No está visible en la nueva tabla (necesito `INSERT INTO` la nueva mesa)
Costo	Bajo	Bajo	Más alto (reescritura completa de los datos)
Velocida de migració	Fast	Fast	Más lento

Característica	Migración in situ instantánea	Migración in situ migrate	Migración completa de datos CTAS o (CREAR TABLA + INSERTAR)
Se puede usar para migrar a Amazon S3 Tables	⊗No	⊗No	⊙Sí
Requiere un DDL manual	⊗No (el esquema y las particiones se copian de la tabla fuente)	⊗No (el esquema y las particiones se copian de la tabla fuente)	Si usa CTAS, solo requiere especificar la partición
El mejor uso	Migración rápida sin reescribir los datos, lo que permite side-by-side utilizar Hive e Iceberg para realizar pruebas o realizar una transición gradual.	Sustituir una tabla de Hive sin tener que volver a escribir los datos cuando sea aceptable un cambio inmediato.	Optimización completa de Iceberg con reescritura de datos. Ideal para rediseñar particiones o esquemas, o para mejorar el diseño y el rendimiento. Se recomienda siempre que sea posible.

Mejores prácticas para optimizar las cargas de trabajo de Apache Iceberg

Iceberg es un formato de tabla diseñado para simplificar la administración de los lagos de datos y mejorar el rendimiento de las cargas de trabajo. Los diferentes casos de uso pueden priorizar diferentes aspectos, como el costo, el rendimiento de lectura, el rendimiento de escritura o la retención de datos, por lo que Iceberg ofrece opciones de configuración para gestionar estas desventajas. En esta sección, se proporciona información para optimizar y ajustar las cargas de trabajo de Iceberg a fin de adaptarlas a sus necesidades.

Temas

Prácticas recomendadas generales

Sea cual sea su caso de uso, cuando utilice Apache Iceberg en una AWS aplicación, le recomendamos que siga estas prácticas recomendadas generales.

- Utilice la versión 2 del formato Iceberg.

  Athena usa el formato Iceberg versión 2 de forma predeterminada.

  Cuando utilices Spark en Amazon EMR o AWS Glue para crear tablas de Iceberg, especifica la versión del formato tal y como se describe en la documentación de Iceberg.
- Utilícela AWS Glue Data Catalog como catálogo de datos.

  Athena usa el de forma AWS Glue Data Catalog predeterminada.

  Cuando utilices Spark en Amazon EMR o AWS Glue para trabajar con Iceberg, añade la siguiente configuración a tu sesión de Spark para usar el. AWS Glue Data Catalog Para obtener más

información, consulta la sección Configuraciones de Spark para Iceberg que aparece AWS Glue anteriormente en esta guía.

```
"spark.sql.catalog.<your_catalog_name>.type": "glue"
```

- Utilízalo AWS Glue Data Catalog como gestor de bloqueos.

  Athena usa el AWS Glue Data Catalog como administrador de bloqueos de forma predeterminada para las tablas Iceberg.

  Cuando utilices Spark en Amazon EMR o AWS Glue para trabajar con Iceberg, asegúrate de configurar la configuración de tu sesión de Spark para utilizarla AWS Glue Data Catalog como administrador de bloqueos. Para obtener más información, consulta Optimistic Locking en la documentación de Iceberg.

- Utilice la compresión Zstandard (ZSTD).

  El códec de compresión predeterminado de Iceberg es gzip, que se puede modificar mediante la propiedad table. `write.<file_type>.compression-codec` Athena ya usa ZSTD como códec de compresión predeterminado para las tablas Iceberg.

  En general, recomendamos usar el códec de compresión ZSTD porque logra un equilibrio entre GZIP y Snappy y ofrece un buen rendimiento sin comprometer la relación de compresión. read/write Además, los niveles de compresión se pueden ajustar para adaptarlos a tus necesidades. Para obtener más información, consulte los niveles de compresión ZSTD en Athena en la documentación de Athena.

  Puede que Snappy ofrezca el mejor rendimiento general de lectura y escritura, pero tiene una relación de compresión inferior a la de GZIP y ZSTD. Si prioriza el rendimiento, incluso si eso implica almacenar grandes volúmenes de datos en Amazon S3, Snappy podría ser la mejor opción.

Optimizar el rendimiento de lectura

En esta sección se describen las propiedades de la tabla que se pueden ajustar para optimizar el rendimiento de lectura, independientemente del motor.

Particiones

Al igual que con las tablas de Hive, Iceberg utiliza las particiones como capa principal de indexación para evitar leer archivos de metadatos y datos innecesarios. Las estadísticas de las columnas también se tienen en cuenta como una capa secundaria de indexación para mejorar aún más la planificación de las consultas, lo que se traduce en un mejor tiempo de ejecución general.

Partición de datos

Para reducir la cantidad de datos que se escanean al consultar las tablas de Iceberg, elige una estrategia de partición equilibrada que se ajuste a los patrones de lectura esperados:

- Identifica las columnas que se utilizan con frecuencia en las consultas. Estas son las candidatas ideales para particionar. Por ejemplo, si normalmente consulta datos de un día determinado, un ejemplo natural de columna de partición sería una columna de fecha.
- Elija una columna de partición de baja cardinalidad para evitar crear un número excesivo de particiones. Demasiadas particiones pueden aumentar el número de archivos de la tabla, lo que puede afectar negativamente al rendimiento de las consultas. Como regla general, «demasiadas particiones» se puede definir como un escenario en el que el tamaño de los datos de la mayoría de las particiones es inferior a 2 a 5 veces el valor establecido por`target-file-size-bytes`.

> **Note**
>
> Si normalmente realizas consultas con filtros en una columna de cardinalidad alta (por ejemplo, una `id` columna que puede tener miles de valores), utiliza la función de partición oculta de Iceberg con transformaciones de cubos, como se explica en la siguiente sección.

Usa particiones ocultas

Si sus consultas suelen filtrar por un derivado de una columna de la tabla, utilice particiones ocultas en lugar de crear nuevas columnas de forma explícita para que funcionen como particiones. Para obtener más información sobre esta función, consulte la documentación de Iceberg.

Por ejemplo, en un conjunto de datos que tiene una columna de marca de tiempo (por ejemplo,`2023-01-01 09:00:00`), en lugar de crear una nueva columna con la fecha analizada (por ejemplo,`2023-01-01`), utilice transformaciones de partición para extraer la parte de fecha de la marca de tiempo y crear estas particiones sobre la marcha.

Los casos de uso más comunes de la partición oculta son:

- Particionar por fecha u hora, cuando los datos tienen una columna de marca de tiempo. Iceberg ofrece múltiples transformaciones para extraer las partes de fecha u hora de una marca de tiempo.
- El particionamiento en una función hash de una columna, cuando la columna de partición tiene una cardinalidad alta y daría como resultado demasiadas particiones. La transformación de cubos de Iceberg agrupa varios valores de partición en menos particiones ocultas (de cubos) mediante el uso de funciones de hash en la columna de particiones.

Consulte las transformaciones de partición en la documentación de Iceberg para obtener una descripción general de todas las transformaciones de partición disponibles.

Las columnas que se utilizan para crear particiones ocultas pueden formar parte de los predicados de las consultas mediante el uso de funciones SQL habituales, como y. `year()` `month()` Los predicados también se pueden combinar con operadores como y. `BETWEEN` `AND`

> **Note**
>
> Iceberg no puede realizar una depuración de particiones para funciones que producen un tipo de datos diferente; por ejemplo,. `substring(event_time, 1, 10) = '2022-01-01'`

Utilice la evolución de particiones

Utilice la evolución de particiones de Iceberg cuando la estrategia de partición existente no sea óptima. Por ejemplo, si eliges particiones horarias que resultan ser demasiado pequeñas (de solo unos pocos megabytes cada una), considera cambiarlas a particiones diarias o mensuales.

Puede utilizar este enfoque cuando al principio no esté clara cuál es la mejor estrategia de partición para una tabla y desee afinar su estrategia de partición a medida que obtenga más información. Otro uso eficaz de la evolución de las particiones es cuando los volúmenes de datos cambian y la estrategia de particionamiento actual pierde eficacia con el paso del tiempo.

Para obtener instrucciones sobre cómo hacer evolucionar las particiones, consulte las extensiones SQL de ALTER TABLE en la documentación de Iceberg.

Ajustar el tamaño de los archivos

La optimización del rendimiento de las consultas implica minimizar la cantidad de archivos pequeños en las tablas. Para obtener un buen rendimiento de las consultas, generalmente recomendamos mantener los archivos Parquet y ORC de más de 100 MB.

El tamaño del archivo también afecta a la planificación de las consultas en las tablas Iceberg. A medida que aumenta el número de archivos de una tabla, también lo hace el tamaño de los archivos de metadatos. Los archivos de metadatos más grandes pueden ralentizar la planificación de las consultas. Por lo tanto, cuando el tamaño de la tabla aumente, aumente el tamaño del archivo para reducir la expansión exponencial de los metadatos.

Utilice las siguientes prácticas recomendadas para crear archivos del tamaño adecuado en las tablas Iceberg.

Establezca el tamaño del archivo de destino y del grupo de filas

Iceberg ofrece los siguientes parámetros de configuración clave para ajustar el diseño del archivo de datos. Le recomendamos que utilice estos parámetros para establecer el tamaño del archivo de destino y el tamaño del grupo de filas o del campo.

Parámetro	Valor predeterminado	Comentario
`write.target-file-size-bytes`	512 MB	Este parámetro especific a el tamaño máximo de archivo que creará Iceberg. Sin embargo, es posible que algunos archivos se escriban con un tamaño inferior a este límite.
`write.parquet.row-group-size-bytes`	128 MB	Tanto Parquet como ORC almacenan los datos en fragmentos para que los motores puedan evitar leer todo el archivo en algunas operaciones.

Parámetro	Valor predeterminado	Comentario
`write.orc.stripe-size-bytes`	64 MB	
`write.distribution-mode`	Ninguno, para la versión 1.1 y anteriores de Iceberg Hash, a partir de la versión 1.2 de Iceberg	Iceberg solicita a Spark que clasifique los datos entre sus tareas antes de guardarlos en el almacenamiento.

- En función del tamaño esperado de la tabla, sigue estas pautas generales:
  - Tablas pequeñas (hasta unos pocos gigabytes): reduzca el tamaño del archivo de destino a 128 MB. Reduzca también el tamaño del grupo de filas o de las franjas (por ejemplo, a 8 o 16 MB).
  - Tablas de tamaño mediano a grande (desde unos pocos gigabytes hasta cientos de gigabytes): los valores predeterminados son un buen punto de partida para estas tablas. Si las consultas son muy selectivas, ajuste el tamaño del grupo de filas o de las franjas (por ejemplo, a 16 MB).
  - Tablas muy grandes (cientos de gigabytes o terabytes): aumente el tamaño del archivo de destino a 1024 MB o más y considere la posibilidad de aumentar el tamaño del grupo de filas o de las franjas si las consultas suelen extraer grandes conjuntos de datos.

- Para garantizar que las aplicaciones de Spark que escriben en tablas de Iceberg creen archivos del tamaño adecuado, defina la `write.distribution-mode` propiedad en o. `hash` `range` Para obtener una explicación detallada de la diferencia entre estos modos, consulta Cómo escribir modos de distribución en la documentación de Iceberg.

Estas son pautas generales. Le recomendamos que realice pruebas para identificar los valores más adecuados para sus tablas y cargas de trabajo específicas.

Realice una compactación regular

Las configuraciones de la tabla anterior establecen un tamaño de archivo máximo que pueden crear las tareas de escritura, pero no garantizan que los archivos tengan ese tamaño. Para garantizar un tamaño de archivo adecuado, ejecute la compactación con regularidad para combinar archivos pequeños en archivos más grandes. Para obtener una guía detallada sobre cómo ejecutar la compactación, consulte la sección Compactación de iceberg más adelante en esta guía.

Optimice las estadísticas de las columnas

Iceberg utiliza las estadísticas de columnas para depurar los archivos, lo que mejora el rendimiento de las consultas al reducir la cantidad de datos que escanean las consultas. Para aprovechar las estadísticas de columnas, asegúrate de que Iceberg recopile las estadísticas de todas las columnas que se utilizan con frecuencia en los filtros de consultas.

De forma predeterminada, Iceberg recopila las estadísticas solo de las 100 primeras columnas de cada tabla, tal y como se define en la propiedad de la tabla. `write.metadata.metrics.max-inferred-column-defaults` Si la tabla tiene más de 100 columnas y las consultas suelen hacer referencia a columnas fuera de las 100 primeras columnas (por ejemplo, puede que las consultas se filtren por la columna 132), asegúrese de que Iceberg recopile las estadísticas de esas columnas. Existen dos opciones para lograrlo:

- Al crear la tabla Iceberg, reordene las columnas para que las columnas que necesite para las consultas se encuentren dentro del rango de columnas establecido por `write.metadata.metrics.max-inferred-column-defaults` (el valor predeterminado es 100).

  Nota: Si no necesita estadísticas sobre 100 columnas, puede ajustar la `write.metadata.metrics.max-inferred-column-defaults` configuración al valor que desee (por ejemplo, 20) y reordenar las columnas para que las columnas que necesite para leer y escribir consultas estén dentro de las 20 primeras columnas de la parte izquierda del conjunto de datos.
- Si utilizas solo unas pocas columnas en los filtros de consulta, puedes deshabilitar la propiedad general para la recopilación de métricas y elegir de forma selectiva columnas individuales para recopilar estadísticas, como se muestra en este ejemplo:

```
.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: Las estadísticas de columnas son más eficaces cuando los datos se ordenan en esas columnas. Para obtener más información, consulte la sección Establecer el orden de clasificación más adelante en esta guía.

Elija la estrategia de actualización adecuada

Utilice una copy-on-write estrategia para optimizar el rendimiento de lectura, cuando las operaciones de escritura más lentas sean aceptables para su caso de uso. Esta es la estrategia por defecto que utiliza Iceberg.

Copy-on-write da como resultado un mejor rendimiento de lectura, ya que los archivos se escriben directamente en el almacenamiento de forma optimizada para la lectura. Sin embargo, en comparación con merge-on-read, cada operación de escritura lleva más tiempo y consume más recursos informáticos. Esto supone un equilibrio clásico entre la latencia de lectura y la de escritura. Por lo general, copy-on-write es ideal para casos de uso en los que la mayoría de las actualizaciones se colocan en las mismas particiones de tabla (por ejemplo, para cargas de lotes diarias).

Copy-on-write las configuraciones (`write.update.modewrite.delete.mode`, y`write.merge.mode`) se pueden configurar a nivel de tabla o de forma independiente en el lado de la aplicación.

Utilice la compresión ZSTD

Puede modificar el códec de compresión utilizado por Iceberg mediante la propiedad table. `write.<file_type>.compression-codec` Se recomienda utilizar el códec de compresión ZSTD para mejorar el rendimiento general de las tablas.

De forma predeterminada, las versiones 1.3 y anteriores de Iceberg utilizan la compresión GZIP, que proporciona un read/write rendimiento más lento en comparación con el ZSTD.

Nota: Es posible que algunos motores utilicen valores predeterminados diferentes. Este es el caso de las tablas Iceberg que se crean con Athena o Amazon EMR versión 7.x.

Establezca el orden de clasificación

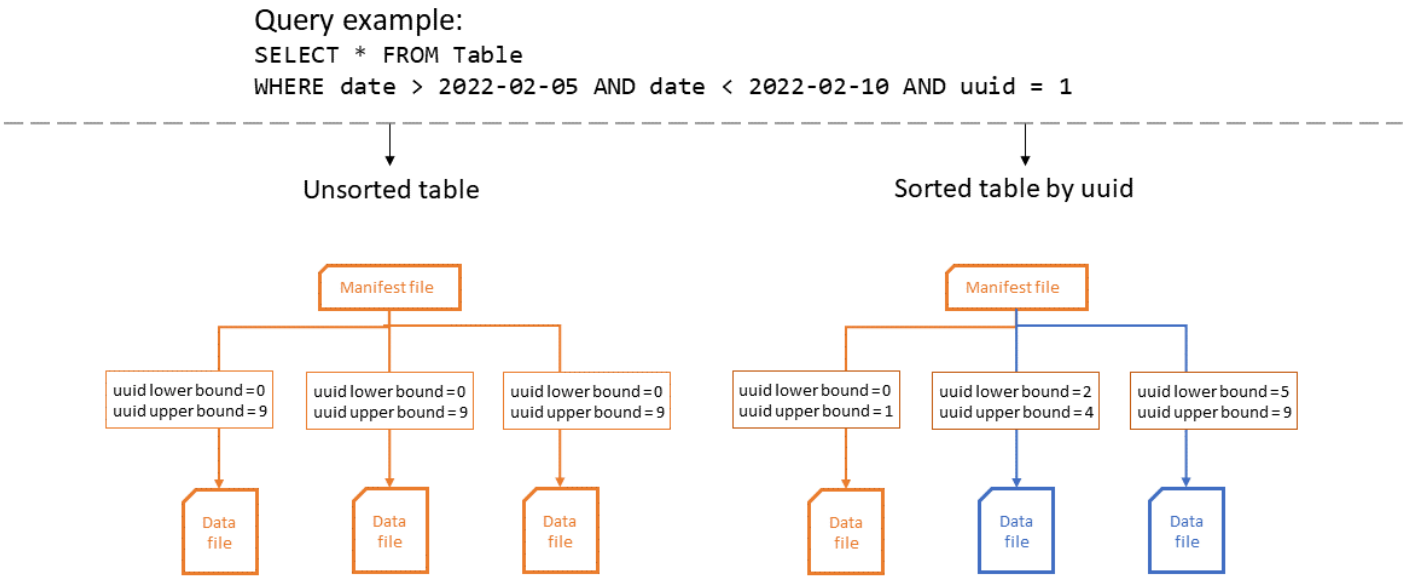
Para mejorar el rendimiento de lectura en las tablas Iceberg, se recomienda ordenar la tabla en función de una o más columnas que se utilizan con frecuencia en los filtros de consulta. La clasificación, combinada con las estadísticas de columnas de Iceberg, puede hacer que la depuración de archivos sea considerablemente más eficiente, lo que se traduce en operaciones de lectura más rápidas. La clasificación también reduce el número de solicitudes de Amazon S3 para consultas que utilizan las columnas de ordenación de los filtros de consultas.

Puedes establecer un orden de clasificación jerárquico a nivel de tabla ejecutando una sentencia de lenguaje de definición de datos (DDL) con Spark. Para ver las opciones disponibles, consulta la

documentación de Iceberg. Una vez establecido el orden de clasificación, los grabadores aplicarán este orden a las operaciones de escritura de datos posteriores en la tabla Iceberg.

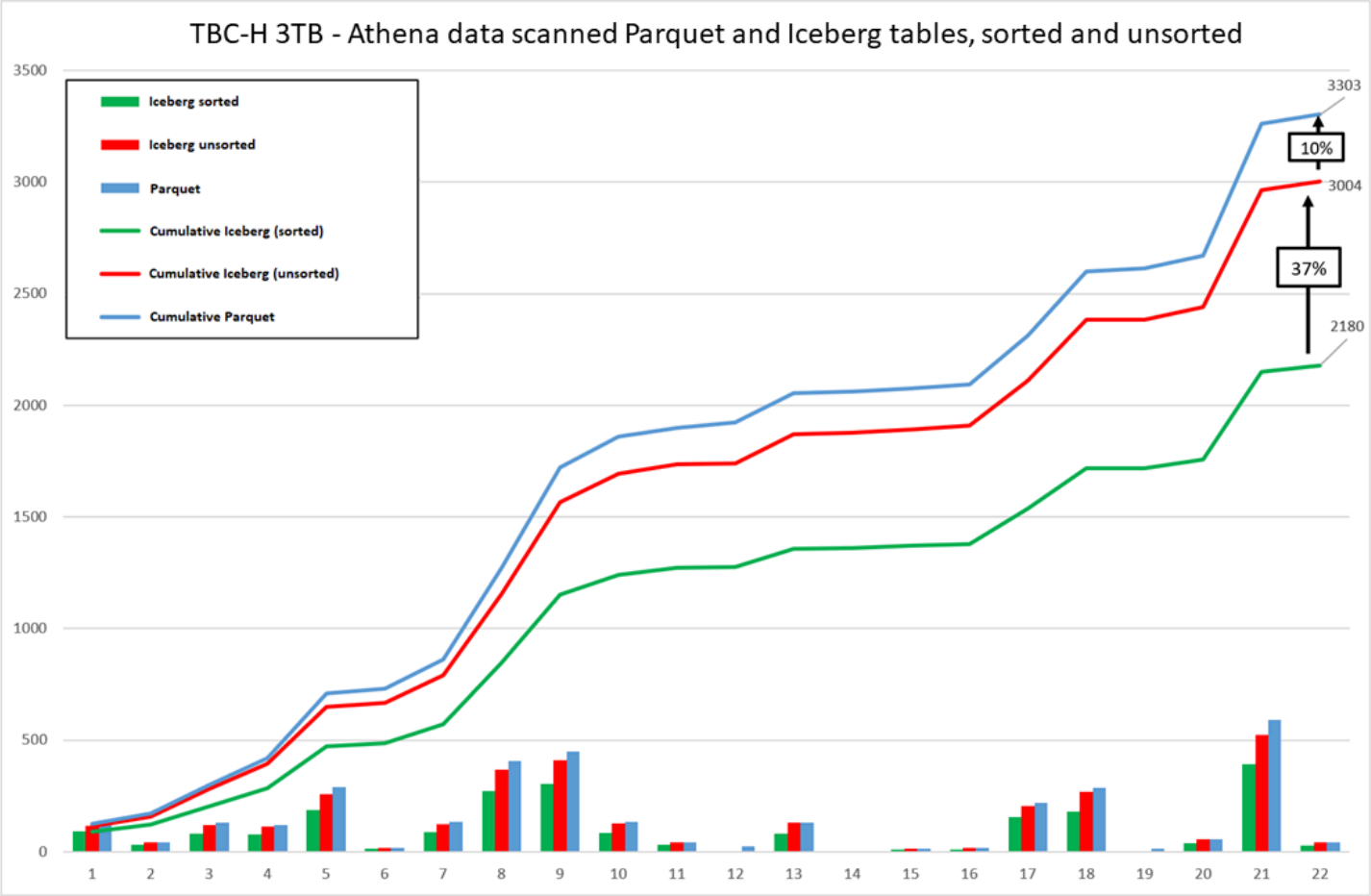
Por ejemplo, en las tablas divididas por fecha (`yyyy-mm-dd`), donde la mayoría de las consultas filtran por`uuid`, puedes usar la opción DDL `Write Distributed By Partition Locally Ordered` para asegurarte de que Spark escriba archivos con rangos que no se superpongan.

El siguiente diagrama ilustra cómo mejora la eficiencia de las estadísticas de columnas cuando se ordenan las tablas. En el ejemplo, la tabla ordenada solo necesita abrir un archivo y se beneficia al máximo de la partición y el archivo de Iceberg. En la tabla sin ordenar, `uuid` puede existir alguno en cualquier archivo de datos, por lo que la consulta debe abrir todos los archivos de datos.

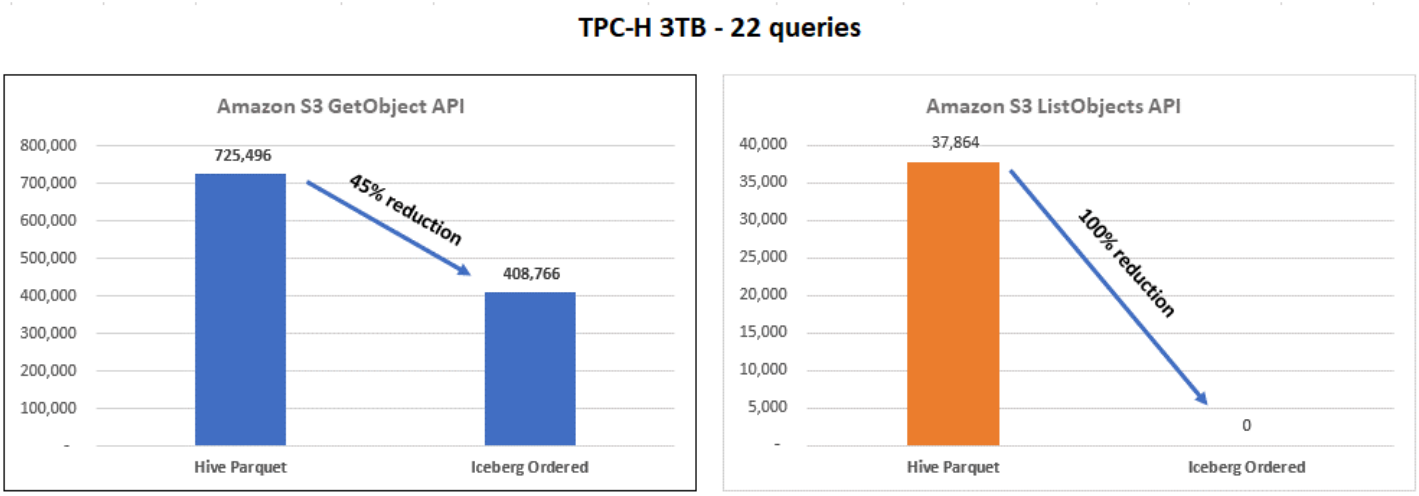


Cambiar el orden de clasificación no afecta a los archivos de datos existentes. Puede utilizar la compactación de iceberg para aplicar el orden de clasificación a los mismos.

El uso de tablas ordenadas por Iceberg puede reducir los costes de la carga de trabajo, como se muestra en el siguiente gráfico.



Estos gráficos resumen los resultados de utilizar el índice de referencia TPC-H para las tablas Hive (Parquet) en comparación con las tablas ordenadas de Iceberg. Sin embargo, los resultados pueden ser diferentes para otros conjuntos de datos o cargas de trabajo.



Optimización del rendimiento de escritura

En esta sección se describen las propiedades de las tablas que se pueden ajustar para optimizar el rendimiento de escritura en las tablas Iceberg, independientemente del motor.

Configure el modo de distribución de la tabla

Iceberg ofrece varios modos de distribución de escritura que definen cómo se distribuyen los datos de escritura en las tareas de Spark. Para obtener una descripción general de los modos disponibles, consulta Cómo escribir los modos de distribución en la documentación de Iceberg.

Para los casos de uso que dan prioridad a la velocidad de escritura, especialmente en cargas de trabajo de streaming, `write.distribution-mode` establézcalo en. `none` Esto garantiza que Iceberg no solicite una reorganización adicional de Spark y que los datos se escriban a medida que estén disponibles en las tareas de Spark. Este modo es especialmente adecuado para las aplicaciones de streaming estructurado de Spark.

> **Note**
>
> Si se configura el modo de distribución de escritura, se `none` suelen producir numerosos archivos pequeños, lo que reduce el rendimiento de lectura. Se recomienda compactarlos con regularidad para consolidar estos archivos pequeños en archivos del tamaño adecuado para el rendimiento de las consultas.

Elija la estrategia de actualización adecuada

Utilice una merge-on-read estrategia para optimizar el rendimiento de escritura cuando las operaciones de lectura más lentas de los datos más recientes sean aceptables para su caso de uso.

Cuando lo usas merge-on-read, Iceberg graba las actualizaciones y las elimina en el almacenamiento como pequeños archivos independientes. Cuando se lee la tabla, el lector debe combinar estos cambios con los archivos base para obtener la última vista de los datos. Esto se traduce en una penalización del rendimiento de las operaciones de lectura, pero acelera la escritura de actualizaciones y eliminaciones. Por lo general, merge-on-read es ideal para transmitir cargas de trabajo con actualizaciones o trabajos con pocas actualizaciones distribuidas en muchas particiones de tablas.

Puede configurar merge-on-read las configuraciones (`write.update.modewrite.delete.mode`, y`write.merge.mode`) a nivel de tabla o de forma independiente desde el punto de vista de la aplicación.

Su uso merge-on-read requiere una compactación regular para evitar que el rendimiento de lectura se degrade con el tiempo. La compactación reconcilia las actualizaciones y las eliminaciones con los archivos de datos existentes para crear un nuevo conjunto de archivos de datos, lo que elimina la penalización de rendimiento en la parte de lectura. De forma predeterminada, la compactación de Iceberg no fusiona ni elimina los archivos a menos que se cambie el valor predeterminado de la `delete-file-threshold` propiedad por un valor menor (consulte la documentación de Iceberg). Para obtener más información sobre la compactación, consulte la sección Compactación de iceberg más adelante en esta guía.

Elija el formato de archivo correcto

Iceberg admite la escritura de datos en los formatos Parquet, ORC y Avro. El formato predeterminado es Parquet. Parquet y ORC son formatos en columnas que ofrecen un rendimiento de lectura superior, pero generalmente son más lentos de escribir. Esto representa el equilibrio típico entre el rendimiento de lectura y el de escritura.

Si la velocidad de escritura es importante para su caso de uso, como en el caso de cargas de trabajo en streaming, considere la posibilidad de escribir en formato Avro configurándolo `Avro` en `write-format` las opciones de grabación. Como Avro es un formato basado en filas, proporciona tiempos de escritura más rápidos a costa de un rendimiento de lectura más lento.

Para mejorar el rendimiento de lectura, ejecute una compactación regular para combinar y transformar archivos Avro pequeños en archivos Parquet más grandes. El resultado del proceso de compactación depende de la configuración de la `write.format.default` tabla. El formato predeterminado de Iceberg es Parquet, por lo que si escribe en Avro y, a continuación, ejecuta la compactación, Iceberg transformará los archivos de Avro en archivos de Parquet. A continuación se muestra un ejemplo:

```
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))
```

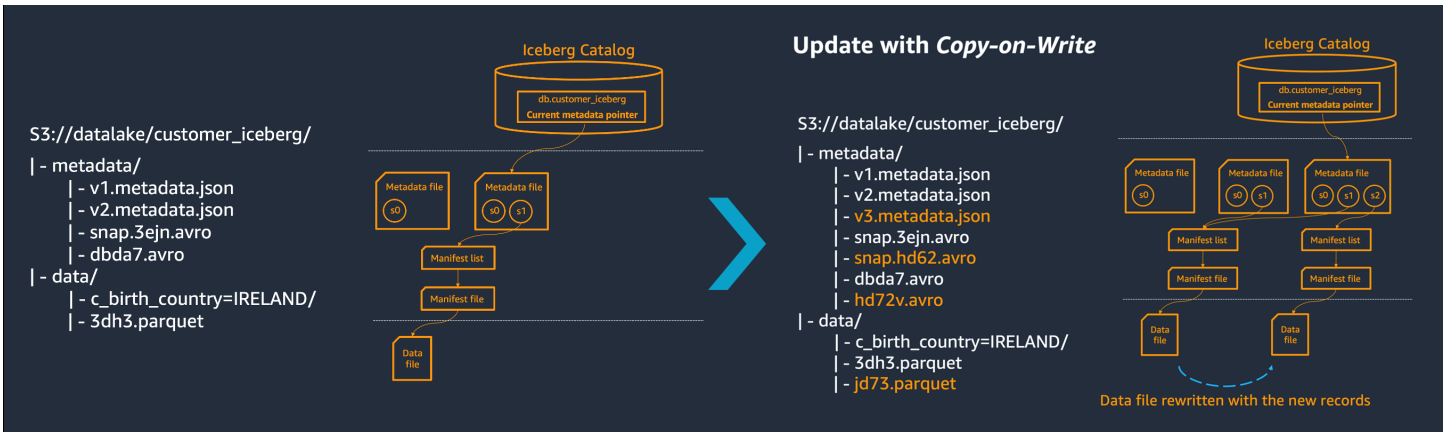
```
    OPTIONS (
      'format-version'='2',
      write.format.default'=parquet)
""")

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()
```

Optimizar el almacenamiento

Al actualizar o eliminar los datos de una tabla Iceberg, se aumenta el número de copias de los datos, como se muestra en el siguiente diagrama. Lo mismo ocurre con la compactación en ejecución: aumenta el número de copias de datos en Amazon S3. Esto se debe a que Iceberg trata los archivos subyacentes a todas las tablas como inmutables.



Siga las prácticas recomendadas de esta sección para gestionar los costes de almacenamiento.

Habilite S3 Intelligent-Tiering

Utilice la clase de almacenamiento Amazon S3 Intelligent-Tiering para mover automáticamente los datos al nivel de acceso más rentable cuando cambien los patrones de acceso. Esta opción no supone una sobrecarga operativa ni afecta al rendimiento.

Nota: No utilice los niveles opcionales (como Archive Access y Deep Archive Access) de S3 Intelligent-Tiering with Iceberg tables. Para archivar datos, consulte las directrices de la siguiente sección.

También puede usar las reglas del ciclo de vida de Amazon S3 para establecer sus propias reglas para mover objetos a otra clase de almacenamiento de Amazon S3, como S3 Standard-IA o S3 One Zone-IA (consulte Transiciones compatibles y restricciones relacionadas en la documentación de Amazon S3).

Archive o elimine las instantáneas históricas

Por cada transacción confirmada (inserción, actualización, combinación, compactación) en una tabla de Iceberg, se crea una nueva versión o instantánea de la tabla. Con el tiempo, el número de versiones y el número de archivos de metadatos de Amazon S3 se acumulan.

Es necesario conservar las instantáneas de una tabla para funciones como el aislamiento de las instantáneas, la reversión de tablas y las consultas sobre viajes en el tiempo. Sin embargo, los costes de almacenamiento aumentan con la cantidad de versiones que se conservan.

En la siguiente tabla se describen los patrones de diseño que puede implementar para administrar los costos en función de sus requisitos de retención de datos.

Patrón de diseño	Solución	Casos de uso
Eliminar instantáneas antiguas	• Utilice la sentencia VACUUM de Athena para eliminar las instantáneas antiguas. Esta operación no implica ningún coste informático. • Como alternativa, puedes usar Spark en Amazon EMR o AWS Glue para eliminar instantáneas. Para obtener más información, consulta expire_snapshots en la documentación de Iceberg.	Este enfoque elimina las instantáneas que ya no se necesitan para reducir los costos de almacenamiento. Puede configurar cuántas instantáneas deben conservar se o durante cuánto tiempo, en función de sus requisitos de retención de datos. Esta opción realiza una eliminación definitiva de las instantáneas. No puede revertir ni viajar en el tiempo a instantáneas caducadas.

Patrón de diseño	Solución	Casos de uso
Establezca políticas de retención para instantáneas específicas	1. Utilice etiquetas para marcar instantáneas específicas y definir una política de retención en Iceberg. Para obtener más información, consulte las etiquetas históricas en la documentación de Iceberg. Por ejemplo, puedes conservar una instantán ea al mes durante un año mediante la siguiente sentencia SQL en Spark on Amazon EMR: `ALTER TABLE glue_cata log.db.table CREATE TAG 'EOM-01' AS OF VERSION 30 RETAIN 365 DAYS` 2. Usa Spark en Amazon EMR o AWS Glue para eliminar las instantáneas intermedi as restantes sin etiquetar.	Este patrón es útil para cumplir con los requisitos empresariales o legales que requieren que muestres el estado de una tabla en un momento dado del pasado. Al colocar políticas de retención en instantáneas etiquetadas específicas, puede eliminar otras instantáneas (sin etiquetar) que se hayan creado. De esta forma, puede cumplir con los requisitos de retención de datos sin conservar todas las instantán eas creadas.

Patrón de diseño	Solución	Casos de uso
Archive las instantáneas antiguas	1. Usa las etiquetas de Amazon S3 para marcar objetos con Spark. (Las etiquetas Amazon S3 son diferentes de las etiquetas Iceberg; para obtener más información, consulte la documentación de Iceberg). Por ejemplo: `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 en Amazon EMR o AWS Glue para eliminar instantáneas. Al utilizar la configuración del ejemplo, este procedimiento etiqueta los objetos y los separa de los metadatos de la tabla Iceberg en lugar de eliminarlos de Amazon S3. 3. Utilice las reglas del ciclo de vida de S3 para transferir los objetos etiquetados `to_archive` a una de las clases de almacenamiento de S3 Glacier. 4. Para consultar datos archivados:	Este patrón le permite conservar todas las versiones e instantáneas de las tablas a un costo menor. No puede viajar en el tiempo ni volver a las instantáneas archivadas sin restaurar primero esas versiones como tablas nuevas. Esto suele ser aceptable para fines de auditoría. Puede combinar este enfoque con el patrón de diseño anterior y establecer políticas de retención para instantáneas específicas.

Patrón de diseño	Solución	Casos de uso
	• Restaure los objetos archivados (este paso no es necesario si los objetos se han transferido a la clase de almacenamiento Amazon Glacier Instant Retrieval). • Utilice el procedimiento register_table de Iceberg para registrar la instantánea como una tabla en el catálogo. Para obtener instrucciones detalladas, consulte la entrada del AWS blog Mejore la eficiencia operativa de las tablas Apache Iceberg basadas en los lagos de datos de Amazon S3.	

Elimine los archivos huérfanos

En determinadas situaciones, las aplicaciones de Iceberg pueden fallar antes de que usted confirme sus transacciones. Esto deja los archivos de datos en Amazon S3. Como no se ha realizado ninguna confirmación, estos archivos no se asociarán a ninguna tabla, por lo que puede que tenga que limpiarlos de forma asíncrona.

Para gestionar estas eliminaciones, puede utilizar la declaración VACUUM en Amazon Athena. Esta declaración elimina las instantáneas y también elimina los archivos huérfanos. Esto resulta muy rentable, ya que Athena no cobra por el coste informático de esta operación. Además, no es necesario programar ninguna operación adicional al utilizar el `VACUUM` estado de cuenta.

Como alternativa, puedes usar Spark en Amazon EMR o AWS Glue para ejecutar el `remove_orphan_files` procedimiento. Esta operación tiene un coste informático y debe programarse de forma independiente. Para obtener más información, consulte la documentación de Iceberg.

Mantenimiento de tablas mediante compactación

Iceberg incluye funciones que le permiten llevar a cabo operaciones de mantenimiento de la tabla después de escribir los datos en la tabla. Algunas operaciones de mantenimiento se centran en simplificar los archivos de metadatos, mientras que otras mejoran la forma en que se agrupan los datos en los archivos para que los motores de consultas puedan localizar de forma eficiente la información necesaria para responder a las solicitudes de los usuarios. Esta sección se centra en las optimizaciones relacionadas con la compactación.

Compactación de iceberg

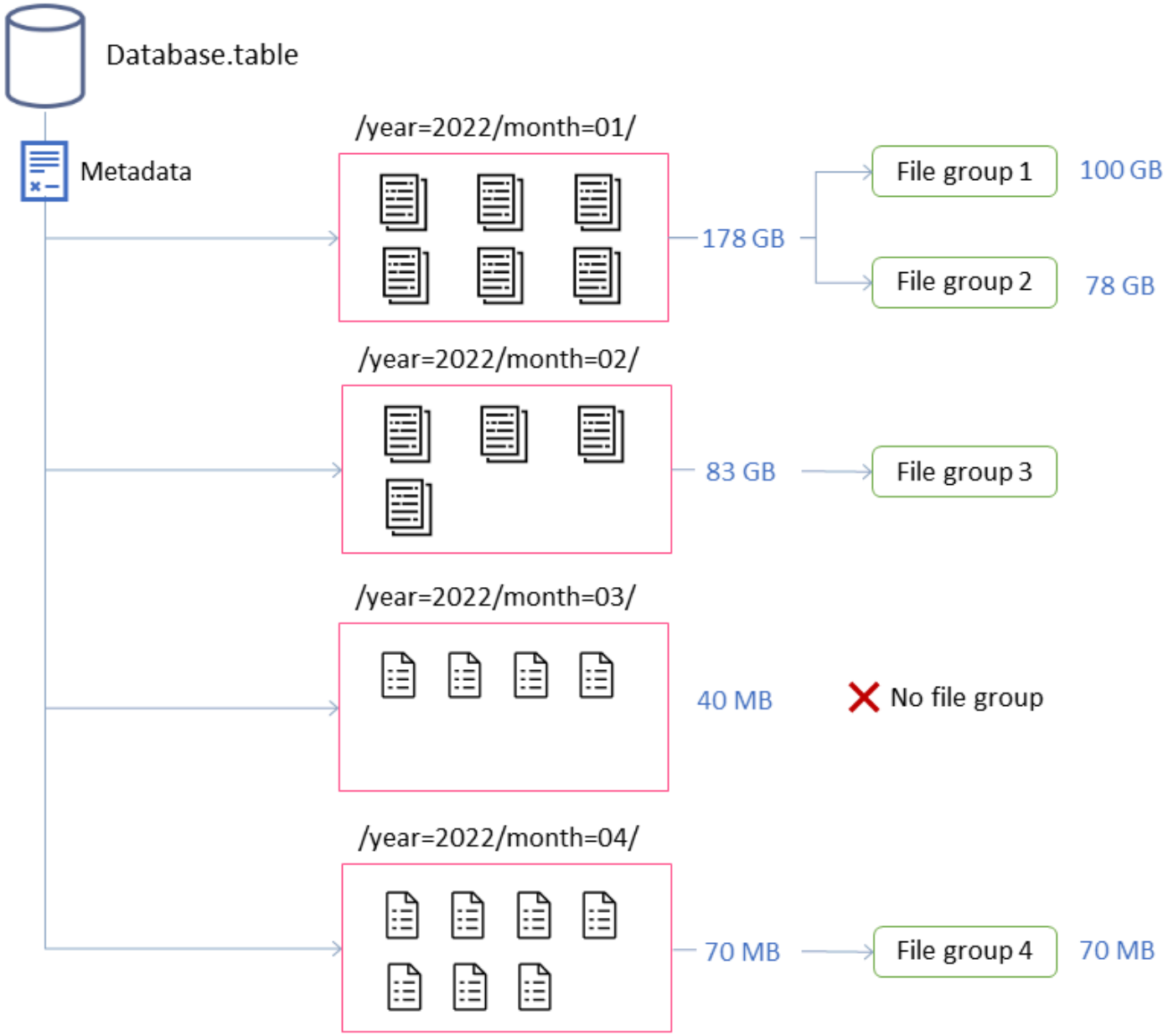
En Iceberg, puede utilizar la compactación para realizar cuatro tareas:

- Combinar archivos pequeños en archivos más grandes que, por lo general, tienen un tamaño superior a 100 MB. Esta técnica se conoce como embalaje en papelera.
- Combinación de archivos eliminados con archivos de datos. Los archivos de eliminación se generan mediante actualizaciones o eliminaciones que utilizan este enfoque. merge-on-read
- (Re) ordenar los datos de acuerdo con los patrones de consulta. Los datos se pueden escribir sin ningún orden de clasificación o con un orden de clasificación adecuado para escrituras y actualizaciones.
- Agrupar los datos mediante curvas que rellenan espacios para optimizarlos para distintos patrones de consulta, especialmente la clasificación en orden z.

Activado AWS, puede ejecutar operaciones de compactación y mantenimiento de mesas para Iceberg a través de Amazon Athena o mediante Spark en Amazon EMR o. AWS Glue

Al ejecutar la compactación mediante el procedimiento rewrite_data_files, puede ajustar varios botones para controlar el comportamiento de compactación. En el siguiente diagrama se muestra el comportamiento predeterminado del empaquetado de contenedores. Comprender la compactación del embalaje de contenedores es clave para entender las implementaciones de clasificación jerárquica y clasificación en orden Z, ya que son extensiones de la interfaz de embalaje de

contenedores y funcionan de manera similar. La principal diferencia es el paso adicional necesario para ordenar o agrupar los datos.



En este ejemplo, la tabla Iceberg consta de cuatro particiones. Cada partición tiene un tamaño y un número de archivos diferentes. Si inicias una aplicación Spark para ejecutar la compactación, la aplicación crea un total de cuatro grupos de archivos para procesarlos. Un grupo de archivos es una abstracción de Iceberg que representa un conjunto de archivos que se procesarán en una sola tarea de Spark. Es decir, la aplicación de Spark que ejecuta la compactación creará cuatro trabajos de Spark para procesar los datos.

Ajustar el comportamiento de compactación

Las siguientes propiedades clave controlan la forma en que se seleccionan los archivos de datos para la compactación:

- MAX_FILE_GROUP_SIZE_BYTES establece el límite de datos para un solo grupo de archivos (trabajo de Spark) en 100 GB de forma predeterminada. Esta propiedad es especialmente importante para las tablas sin particiones o las tablas con particiones que ocupan cientos de gigabytes. Al establecer este límite, puede desglosar las operaciones para planificar el trabajo y avanzar, al tiempo que evita que se agoten los recursos del clúster.

  Nota: Cada grupo de archivos se ordena por separado. Por lo tanto, si desea realizar una ordenación a nivel de partición, debe ajustar este límite para que coincida con el tamaño de la partición.
- El valor predeterminado de MIN_FILE_SIZE_BYTES o MIN_FILE_SIZE_DEFAULT_RATIO es el 75 por ciento del tamaño del archivo de destino establecido a nivel de tabla. Por ejemplo, si una tabla tiene un tamaño objetivo de 512 MB, cualquier archivo que sea inferior a 384 MB se incluirá en el conjunto de archivos que se compactarán.
- El valor predeterminado de MAX_FILE_SIZE_BYTES o MAX_FILE_SIZE_DEFAULT_RATIO es el 180 por ciento del tamaño del archivo de destino. Al igual que ocurre con las dos propiedades que establecen los tamaños mínimos de los archivos, estas propiedades se utilizan para identificar los archivos candidatos para el trabajo de compactación.
- MIN_INPUT_FILES especifica el número mínimo de archivos que se deben compactar si el tamaño de la partición de una tabla es menor que el tamaño del archivo de destino. El valor de esta propiedad se utiliza para determinar si vale la pena compactar los archivos en función del número de archivos (el valor predeterminado es 5).
- DELETE_FILE_THRESHOLD especifica el número mínimo de operaciones de eliminación de un archivo antes de incluirlo en la compactación. A menos que especifique lo contrario, la compactación no combina los archivos de eliminación con los archivos de datos. Para habilitar esta funcionalidad, debe establecer un valor umbral mediante esta propiedad. Este umbral es específico de los archivos de datos individuales, por lo que si lo establece en 3, un archivo de datos solo se reescribirá si hay tres o más archivos de eliminación que hagan referencia a él.

Estas propiedades proporcionan información sobre la formación de los grupos de archivos del diagrama anterior.

Por ejemplo, la partición etiquetada `month=01` incluye dos grupos de archivos porque supera el límite de tamaño máximo de 100 GB. Por el contrario, la `month=02` partición contiene un solo grupo de archivos porque tiene menos de 100 GB. La `month=03` partición no cumple con el requisito mínimo predeterminado de cinco archivos de entrada. Como resultado, no se compactará. Por último, aunque la `month=04` partición no contenga datos suficientes para formar un único archivo del tamaño deseado, los archivos se compactarán porque la partición incluye más de cinco archivos pequeños.

Puede configurar estos parámetros para que Spark se ejecute en Amazon EMR o. AWS Glue En el caso de Amazon Athena, puede administrar propiedades similares mediante las propiedades de la tabla (que comienzan con el prefijo`optimize_`).

Ejecutar la compactación con Spark en Amazon EMR o AWS Glue

En esta sección se describe cómo dimensionar correctamente un clúster de Spark para ejecutar la utilidad de compactación de Iceberg. El siguiente ejemplo utiliza Amazon EMR Serverless, pero puede utilizar la misma metodología en Amazon EMR en EC2 o EKS, o en. AWS Glue

Puede aprovechar la correlación entre los grupos de archivos y las tareas de Spark para planificar los recursos del clúster. Para procesar los grupos de archivos de forma secuencial, teniendo en cuenta el tamaño máximo de 100 GB por grupo de archivos, puedes configurar las siguientes propiedades de Spark:

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

Si desea acelerar la compactación, puede escalar horizontalmente aumentando el número de grupos de archivos que se compactan en paralelo. También puede escalar Amazon EMR mediante un escalado manual o dinámico.

- Escalado manual (por ejemplo, en un factor de 4)
  - `MAX_CONCURRENT_FILE_GROUP_REWRITES`= `4` (nuestro factor)
  - `spark.executor.instances`= `5` (valor utilizado en el ejemplo) x `4` (nuestro factor) = `20`
  - `spark.dynamicAllocation.enabled` = `FALSE`
- Escalado dinámico

- `spark.dynamicAllocation.enabled= TRUE` (predeterminado, no es necesario realizar ninguna acción)
- MAX_CONCURRENT_FILE_GROUP_REWRITES = `N` (alinee este valor `conspark.dynamicAllocation.maxExecutors`, que es 100 de forma predeterminada; según las configuraciones del ejecutor del ejemplo, puede establecerlo en 20) `N`

Estas son pautas para ayudar a dimensionar el clúster. Sin embargo, también debes monitorear el rendimiento de tus trabajos de Spark para encontrar la mejor configuración para tus cargas de trabajo.

Ejecutar la compactación con Amazon Athena

Athena ofrece una implementación de la utilidad de compactación de Iceberg como una función gestionada a través de la declaración OPTIMIZE. Puede utilizar esta afirmación para ejecutar la compactación sin tener que evaluar la infraestructura.

Esta instrucción agrupa los archivos pequeños en archivos más grandes mediante el algoritmo de empaquetado en contenedores y combina los archivos de eliminación con los archivos de datos existentes. Para agrupar los datos mediante una ordenación jerárquica o una ordenación en orden z, utiliza Spark en Amazon EMR o. AWS Glue

Puedes cambiar el comportamiento predeterminado de la `OPTIMIZE` declaración al crear la tabla pasando las propiedades de la tabla en la `CREATE TABLE` declaración, o después de crearla usando la declaración. `ALTER TABLE` Para ver los valores predeterminados, consulte la documentación de Athena.

Recomendaciones para ejecutar la compactación

Caso de uso	Recomendación
Realizar la compactación del embalaje de contenedores según un cronograma	• Usa la `OPTIMIZE` sentencia de Athena si no sabes cuántos archivos pequeños contiene tu tabla. El modelo de precios de Athena se basa en los datos escaneados, por lo que si no hay archivos que compactar, estas operaciones no conllevan ningún coste. Para evitar que se agoten los tiempos de espera

Caso de uso	Recomendación
	en las mesas de Athena, ejecute `OPTIMIZE` de forma gradual. per-table-partition • Utilice Amazon EMR o el AWS Glue escalado dinámico cuando espere compactar grandes volúmenes de archivos pequeños.
Ejecute la compactación del embalaje de contenedores en función de eventos	• Utilice Amazon EMR o el AWS Glue escalado dinámico cuando espere compactar grandes volúmenes de archivos pequeños.
Ejecutar la compactación para ordenar los datos	• Utilice Amazon EMR o AWS Glue, porque la clasificación es una operación cara y es posible que necesite transferir datos al disco.
Ejecute la compactación para agrupar los datos mediante la clasificación en orden z	• Utilice Amazon EMR o AWS Glue, porque la clasificación en orden z es una operación muy cara y es posible que necesite transferir datos al disco.
Ejecuta la compactación en particiones que otras aplicaciones podrían actualizar debido a la llegada tardía de los datos	• Utilice Amazon EMR o. AWS Glue Habilite la propiedad Iceberg PARTIAL_PROGRESS_ENABLED. Al utilizar esta opción, Iceberg divide el resultado de la compactación en varias confirmaciones. Si se produce una colisión (es decir, si el archivo de datos se actualiza mientras se está compactan do), esta configuración reduce el coste del reintento al limitarlo a la confirmación que incluye el archivo afectado. De lo contrario , puede que tenga que volver a compactar todos los archivos.

Caso de uso	Recomendación
Ejecutar la compactación en particiones frías (particiones de datos que ya no reciben escrituras activas)	• Utilice Amazon EMR o. AWS Glue En el `rewrite_data_files` procedimiento, especifique un `where` predicado que excluya las particiones escritas activamente. Esta estrategia evita los conflictos de datos entre los grabadores y los trabajos de compactac ión, y deja solo los conflictos de metadatos que Iceberg puede resolver automátic amente.

Uso de cargas de trabajo de Iceberg en Amazon S3

En esta sección se describen las propiedades de Iceberg que puede utilizar para optimizar la interacción de Iceberg con Amazon S3.

Evite el particionamiento en caliente (errores HTTP 503)

Algunas aplicaciones de lagos de datos que se ejecutan en Amazon S3 gestionan millones o miles de millones de objetos y procesan petabytes de datos. Esto puede provocar que los prefijos reciban un gran volumen de tráfico, lo que normalmente se detecta a través de errores HTTP 503 (servicio no disponible). Para evitar este problema, utilice las siguientes propiedades de Iceberg:

- `write.distribution-mode`Configúrelo `hash` `range` para que Iceberg escriba archivos de gran tamaño, lo que se traduce en menos solicitudes de Amazon S3. Esta es la configuración preferida y debería abordar la mayoría de los casos.
- Si sigue experimentando 503 errores debido al inmenso volumen de datos en sus cargas de trabajo, puede hacerlo `true` en Iceberg. `write.object-storage.enabled` Esto le indica a Iceberg que codifique los nombres de los objetos y distribuya la carga entre varios prefijos aleatorios de Amazon S3.

Para obtener más información sobre estas propiedades, consulte Escribir propiedades en la documentación de Iceberg.

Utilice las operaciones de mantenimiento de Iceberg para liberar los datos no utilizados

Para gestionar las tablas de Iceberg, puede utilizar la API principal de Iceberg, los clientes de Iceberg (como Spark) o los servicios gestionados como Amazon Athena. Para eliminar archivos antiguos o no utilizados de Amazon S3, le recomendamos que utilice únicamente Iceberg native APIs para eliminar instantáneas, eliminar archivos de metadatos antiguos y eliminar archivos huérfanos.

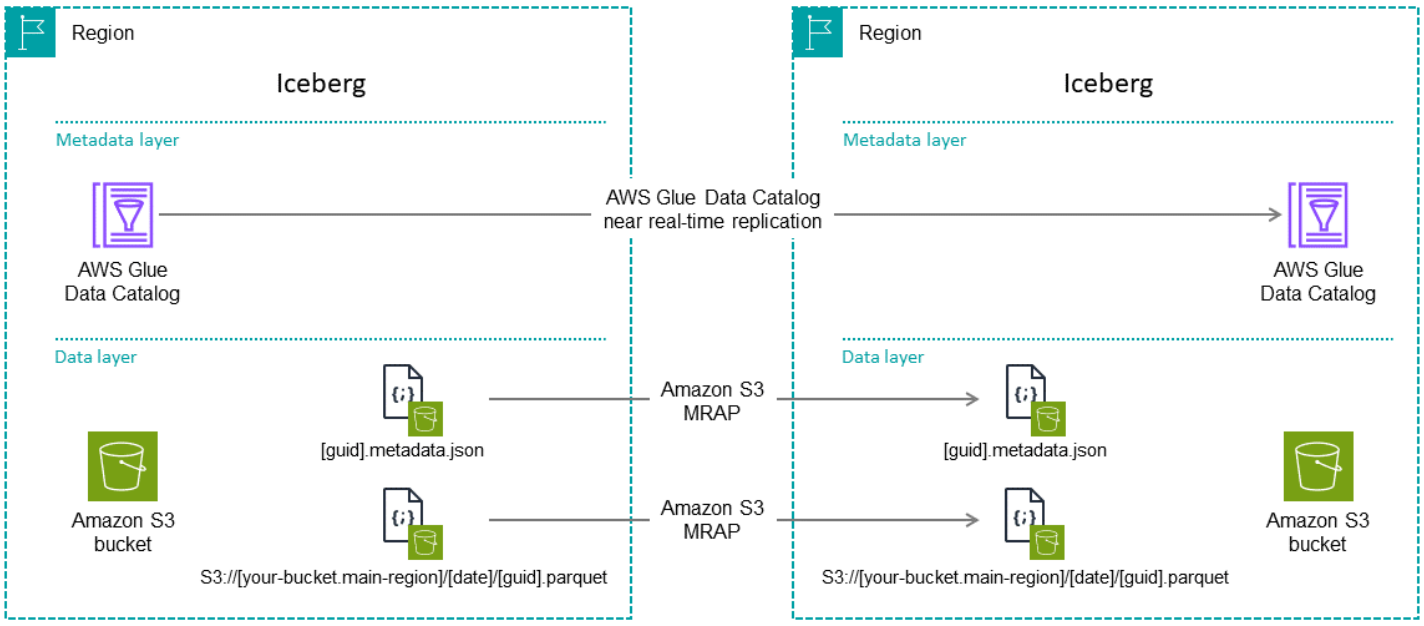
El uso de Amazon S3 APIs a través de Boto3, el SDK de Amazon S3 o AWS Command Line Interface (AWS CLI), o el uso de cualquier otro método que no sea de Iceberg para sobrescribir o eliminar archivos de Amazon S3 de una tabla de Iceberg, provoca que la tabla se dañe y se produzcan errores en las consultas.

Replique los datos en todas partes Regiones de AWS

Al almacenar tablas Iceberg en Amazon S3, puede utilizar las funciones integradas en Amazon S3, como la replicación entre regiones (CRR) y los puntos de acceso multiregión (MRAP), para replicar datos en varias. Regiones de AWS El MRAP proporciona un punto de enlace global para que las aplicaciones accedan a los depósitos de S3 que se encuentran en varios. Regiones de AWS Iceberg no admite rutas relativas, pero puede usar MRAP para realizar operaciones de Amazon S3 mediante el mapeo de cubos a puntos de acceso. El MRAP también se integra perfectamente con el proceso de replicación entre regiones de Amazon S3, que introduce un retraso de hasta 15 minutos. Debe replicar los archivos de datos y metadatos.

> **Important**
>
> Actualmente, la integración de Iceberg con el MRAP solo funciona con Apache Spark. Si necesitas realizar una conmutación por error a la secundaria Región de AWS, tienes que planificar la redirección de las consultas de los usuarios a un entorno de Spark SQL (como Amazon EMR) en la región de conmutación por error.

Las funciones CRR y MRAP te ayudan a crear una solución de replicación entre regiones para las tablas de Iceberg, como se muestra en el siguiente diagrama.



Para configurar esta arquitectura de replicación entre regiones:

1. Cree tablas mediante la ubicación del MRAP. Esto garantiza que los archivos de metadatos de Iceberg apunten a la ubicación del MRAP en lugar de a la ubicación física del depósito.
2. Replique archivos Iceberg mediante Amazon S3 MRAP. El MRAP admite la replicación de datos con un acuerdo de nivel de servicio (SLA) de 15 minutos. Iceberg evita que las operaciones de lectura introduzcan inconsistencias durante la replicación.
3. Haga que las tablas estén disponibles AWS Glue Data Catalog en la región secundaria. Puede elegir entre dos opciones.
   - Configure una canalización para replicar los metadatos de la tabla Iceberg mediante AWS Glue Data Catalog la replicación. Esta utilidad está disponible en el repositorio de replicación de GitHub Glue Catalog y Lake Formation Permissions. Este mecanismo basado en eventos replica las tablas de la región de destino en función de los registros de eventos.
   - Registre las tablas en la región secundaria cuando necesite realizar una conmutación por error. Para esta opción, puede utilizar la utilidad anterior o el procedimiento Iceberg register_table y dirigirlo al archivo más reciente. `metadata.json`

Supervisión de las cargas de trabajo de Apache Iceberg

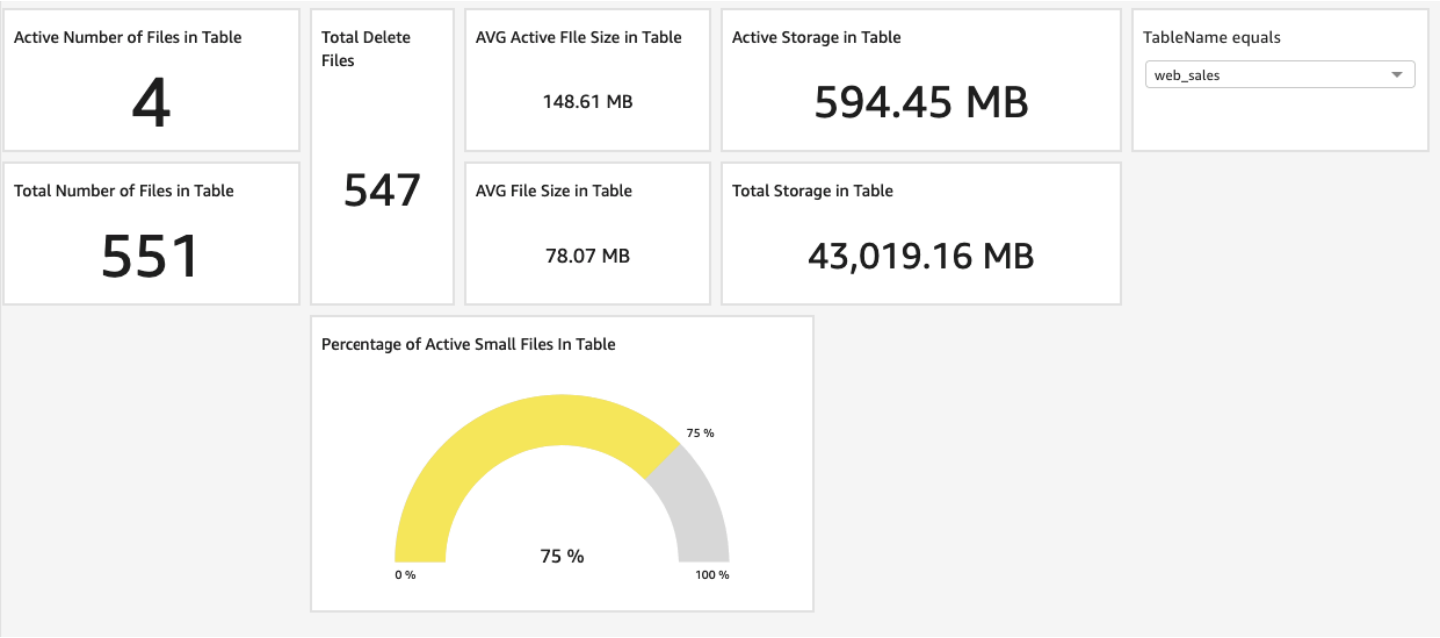
Para supervisar las cargas de trabajo de Iceberg, tiene dos opciones: analizar las tablas de metadatos o utilizar indicadores de métricas. Los indicadores de métricas se introdujeron en la versión 1.2 de Iceberg y solo están disponibles para los catálogos REST y JDBC.

Si las utiliza AWS Glue Data Catalog, puede obtener información sobre el estado de sus tablas de Iceberg configurando la supervisión sobre las tablas de metadatos que expone Iceberg.

La supervisión es crucial para la gestión del rendimiento y la solución de problemas. Por ejemplo, cuando una partición de una tabla Iceberg alcanza un porcentaje determinado de archivos pequeños, la carga de trabajo puede iniciar un trabajo de compactación para consolidar los archivos en archivos más grandes. Esto evita que las consultas se ralenticen más allá de un nivel aceptable.

Supervisión a nivel de tabla

La siguiente pantalla muestra un panel de monitoreo de tablas que se creó en Amazon Quick Sight. Este panel consulta las tablas de metadatos de Iceberg mediante Spark SQL y captura métricas detalladas, como la cantidad de archivos activos y el almacenamiento total. Luego, esta información se almacena en AWS Glue tablas con fines operativos. Por último, se crea un panel de Quick Sight, como se muestra en la siguiente ilustración, mediante Amazon Athena. Esta información le ayuda a identificar y abordar problemas específicos en sus sistemas.



El ejemplo de panel de Quick Sight recopila los siguientes indicadores clave de rendimiento (KPIs) para una tabla Iceberg:

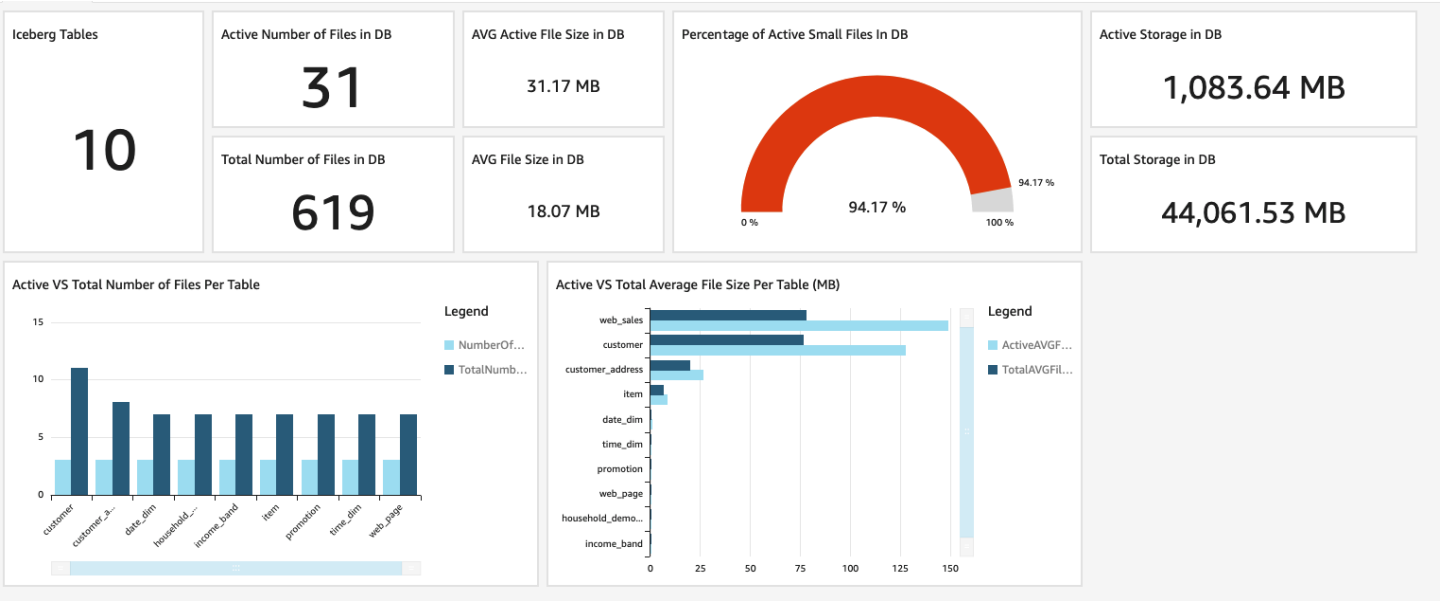
KPI	Descripción	Query
Número de archivos	El número de archivos de la tabla Iceberg (para todas las instantáneas)	`select count(*) from <catalog.database.table_name>.all_files`
Número de archivos activos	El número de archivos activos en la última instantánea de la tabla Iceberg	`select count(*) from <catalog.database.table_name>.files`
Tamaño medio de los archivos	El tamaño medio de los archivos, en megabytes, de todos los archivos de la tabla Iceberg	`select avg(file_size_in_bytes)/1000000 from <catalog.database.table_name>.all_files`
Tamaño medio de los archivos activos	El tamaño medio de los archivos activos de la tabla Iceberg, en megabytes	`select avg(file_size_in_bytes)/1000000 from <catalog.database.table_name>.files`
Porcentaje de archivos pequeños	El porcentaje de archivos activos que ocupan menos de 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`
Tamaño total de almacenamiento	El tamaño total de todos los archivos de la tabla, excluidos los archivos huérfanos y las versiones de objetos de	`select sum(file_size_in_bytes)/1000000`

KPI	Descripción	Query
	Amazon S3 (si están habilitadas)	`from <catalog.database.table_name>.all_files`
Tamaño total del almacenamiento activo	El tamaño total de todos los archivos de las instantáneas actuales de una tabla determinada	`select sum(file_size_in_bytes)/1000000 from <catalog.database.table_name>.files`

Para obtener más información sobre la creación de paneles, consulte la documentación de Quick Sight.

Supervisión a nivel de base de datos

El siguiente ejemplo muestra un panel de supervisión que se creó en Quick Sight para proporcionar una visión general del nivel de base de datos de un conjunto de tablas KPIs Iceberg.



Este panel recopila lo siguiente: KPIs

KPI	Descripción	Query
Número de archivos	El número de archivos de la base de datos Iceberg (para todas las instantáneas)	Este panel utiliza las consultas a nivel de tabla proporcionadas en la sección anterior y consolida los resultados.
Número de archivos activos	El número de archivos activos en la base de datos de Iceberg (según las últimas instantáneas de las tablas de Iceberg)	
Tamaño medio de los archivos	El tamaño medio de los archivos, en megabytes, de todos los archivos de la base de datos Iceberg	
Tamaño medio de los archivos activos	El tamaño medio de los archivos, en megabytes, de todos los archivos activos de la base de datos Iceberg	
Porcentaje de archivos pequeños	El porcentaje de archivos activos que ocupan menos de 100 MB en la base de datos Iceberg	
Tamaño total de almacenamiento	El tamaño total de todos los archivos de la base de datos, excepto los archivos huérfanos y las versiones de objetos de Amazon S3 (si están habilitadas)	
Tamaño total del almacenamiento activo	El tamaño total de todos los archivos de las instantáneas actuales de todas las tablas de la base de datos	

Mantenimiento preventivo

Al configurar las capacidades de monitoreo descritas en las secciones anteriores, puede abordar el mantenimiento de la mesa desde un ángulo preventivo en lugar de reactivo. Por ejemplo, puede usar las métricas a nivel de tabla y base de datos para programar acciones como las siguientes:

- Utilice la compactación por compartimentos para agrupar archivos pequeños cuando una tabla llegue a N archivos pequeños.
- Utilice la compactación por empaquetado de contenedores para fusionar y eliminar archivos cuando una tabla llegue a N archivos eliminados en una partición determinada.
- Elimine los archivos pequeños que ya estaban compactados eliminando las instantáneas cuando el almacenamiento total sea X veces mayor que el almacenamiento activo.

Gobernanza y control de acceso para Apache Iceberg en AWS

Apache Iceberg se integra AWS Lake Formation para simplificar la gobernanza de los datos. Esta integración permite a los administradores de lagos de datos asignar permisos de acceso a nivel de celda a las tablas de Iceberg. Para ver un ejemplo de consulta de tablas Iceberg mediante Amazon Athena AWS Lake Formation y, consulte AWS la entrada del blog Interactúe con tablas Iceberg de Apache con Amazon Athena y cruce cuentas de permisos detallados mediante. AWS Lake Formation

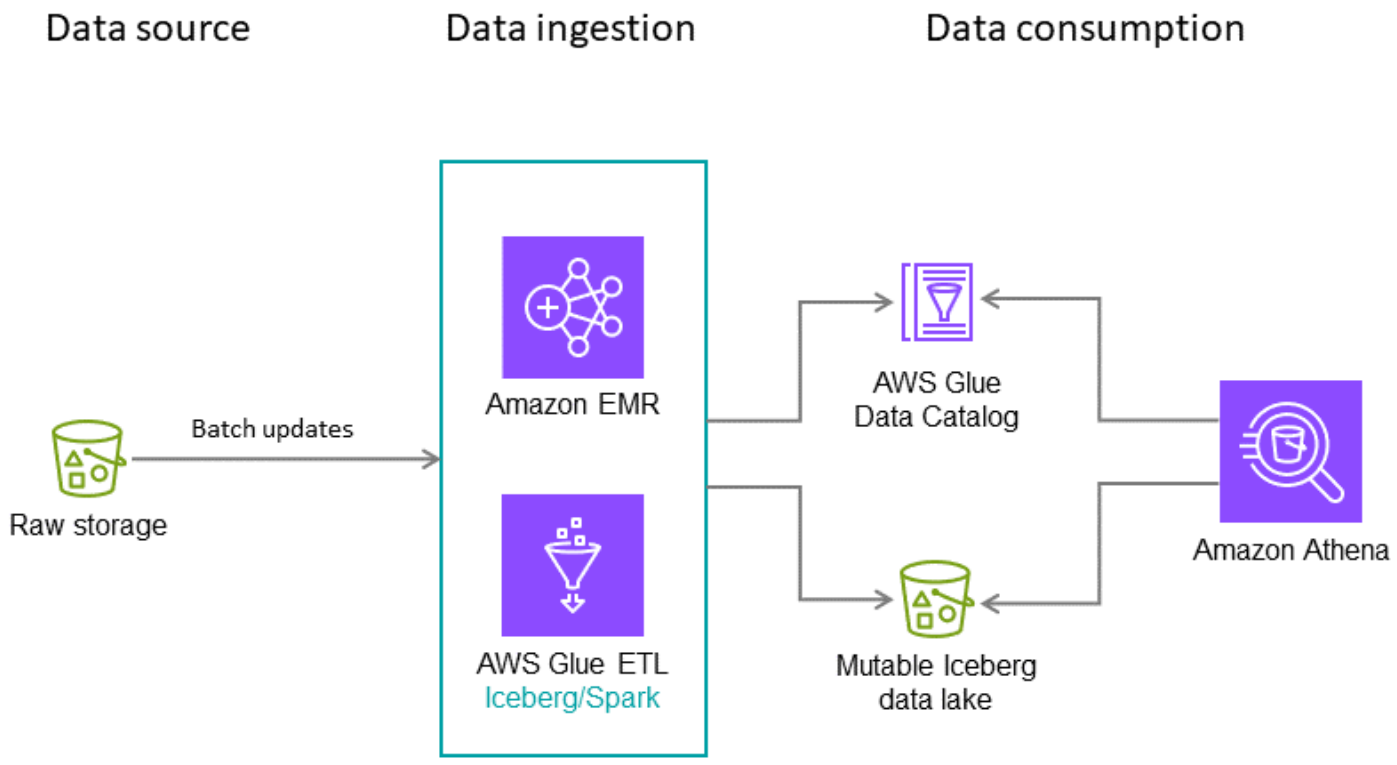
Arquitecturas de referencia para Apache Iceberg en AWS

En esta sección se proporcionan ejemplos de cómo aplicar las mejores prácticas en diferentes casos de uso, como la ingesta por lotes y un lago de datos que combina la ingesta de datos por lotes y en streaming.

Ingestión nocturna por lotes

Para este hipotético caso de uso, supongamos que su mesa Iceberg ingiere transacciones con tarjetas de crédito todas las noches. Cada lote contiene solo actualizaciones incrementales, que deben combinarse en la tabla de destino. Varias veces al año, se reciben datos históricos completos. Para este escenario, recomendamos la arquitectura y las configuraciones siguientes.

Nota: Esto es solo un ejemplo. La configuración óptima depende de sus datos y requisitos.



Recomendaciones:

- Tamaño del archivo: 128 MB, porque las tareas de Apache Spark procesan los datos en fragmentos de 128 MB.

- Tipo de escritura: copy-on-write. Como se detalló anteriormente en esta guía, este enfoque ayuda a garantizar que los datos se escriban de forma optimizada para la lectura.
- Variables de partición:. year/month/day En nuestro caso de uso hipotético, consultamos datos recientes con mayor frecuencia, aunque ocasionalmente realizamos escaneos completos de tablas de datos de los últimos dos años. El objetivo del particionamiento es impulsar las operaciones de lectura rápida en función de los requisitos del caso de uso.
- Orden de clasificación: marca de tiempo
- Catálogo de datos: AWS Glue Data Catalog

Lago de datos que combina la ingesta por lotes y la ingesta casi en tiempo real

Puede aprovisionar un lago de datos en Amazon S3 que comparta datos por lotes y en streaming entre cuentas y regiones. Para ver un diagrama de arquitectura y detalles, consulte la entrada del AWS blog Cree un lago de datos transaccional con Apache Iceberg y comparta datos entre cuentas con Amazon AWS Lake Formation Athena. AWS Glue

Recursos

- Uso del marco Iceberg en AWS Glue(documentación)AWS Glue
- Iceberg (documentación de Amazon EMR)
- Uso de tablas Apache Iceberg (documentación de Amazon Athena)
- Documentación de Amazon S3
- Documentación rápida
- Replicación de los permisos de Glue Catalog y Lake Formation (GitHub repositorio)
- Documentación de Apache Iceberg
- Documentación de Apache Spark

Colaboradores

Las siguientes personas son autores, AWS coautores y revisores de esta guía.

Colaboradores

- Stefano Sandona, arquitecto de soluciones de Big Data
- Imtiaz (Taz) Sayed, arquitecto de soluciones y líder tecnológico de análisis
- Shana Schipers, arquitecta de soluciones de big data
- Prashant Singh, ingeniero de desarrollo de software, Amazon EMR
- Arun A K, arquitecto de soluciones, big data y ETL
- Francisco Morillo, arquitecto de soluciones de streaming
- Suthan Phillips, arquitecto de análisis, Amazon EMR
- Sercan Karaoglu, arquitecto de soluciones
- Yonatan Dolan, especialista en análisis
- Guy Bachar, arquitecto de soluciones
- Sofia Zilberman, arquitecta de soluciones, Streaming
- Dan Stair, arquitecto de soluciones especializado
- Sakti Mishra, arquitecta de soluciones
- Ron Ortloff, director principal de productos de Amazon S3
- David Zhang, arquitecto de soluciones de análisis

Revisores

- Rick Sears, director general de Amazon EMR
- Linda OConnor, especialista en EMR de Amazon
- Ian Meyers, director de Amazon EMR
- Vinita Ananth, directora de gestión de productos de Amazon EMR
- Jason Berkowitz, gerente de producto, AWS Lake Formation
- Mahesh Mishra, director de producto de Amazon Redshift
- Vladimir Zlatkin, gerente de arquitectura de soluciones de big data
- Karthik Prabhakar, arquitecto de análisis, Amazon EMR

- Vijay Jain, gerente de producto
- Anupriti Warade, directora de productos de Amazon S3
- Ajit Tandale, arquitecto de soluciones de datos
- Gwen Chen, directora de marketing de productos
- Noritaka Sekiyama, arquitecta principal de Big Data

Historial de documentos

En la siguiente tabla, se describen cambios significativos de esta guía. Si quiere recibir notificaciones de futuras actualizaciones, puede suscribirse a las notificaciones RSS.

Cambio	Descripción	Fecha
Sección nueva	Se agregó información sobre cómo trabajar con la versión 3 de la especificación de formato de tabla Iceberg.	26 de noviembre de 2025
Corrección	Se corrigió la información sobre la `gc.enabled` configuración en la sección del procedimiento de captura de imágenes.	24 de octubre de 2025
Adiciones	Se agregaron nuevas secciones sobre cómo trabajar con tablas de Iceberg mediante Trino y PyIceberg se amplió la sección sobre la migración de tablas a Iceberg.	12 de agosto de 2025
Actualizaciones	Información mejorada y aclarada en toda la guía para reflejar las versiones más recientes de AWS Glue Amazon EMR y Apache Iceberg.	14 de julio de 2025
Adiciones	Se ha añadido una nueva sección sobre cómo trabajar con tablas Iceberg mediante Amazon Data Firehose.	20 de febrero de 2025

Publicación inicial	—	30 de abril de 2024

AWS Glosario de orientación prescriptiva

Los siguientes son términos de uso común en las estrategias, guías y patrones proporcionados por la Guía AWS prescriptiva. Para sugerir entradas, utilice el enlace Enviar comentarios al final del glosario.

Números

Las 7 R

Siete estrategias de migración comunes para trasladar aplicaciones a la nube. Estas estrategias se basan en las 5 R que Gartner identificó en 2011 y consisten en lo siguiente:

- Refactorizar/rediseñar: traslade una aplicación y modifique su arquitectura mediante el máximo aprovechamiento de las características nativas en la nube para mejorar la agilidad, el rendimiento y la escalabilidad. Por lo general, esto implica trasladar el sistema operativo y la base de datos. Ejemplo: Migrar la base de datos de Oracle en las instalaciones a Amazon Aurora PostgreSQL-Compatible Edition.
- Redefinir la plataforma (transportar y redefinir): traslade una aplicación a la nube e introduzca algún nivel de optimización para aprovechar las capacidades de la nube. Ejemplo: Migrar la base de datos Oracle en las instalaciones a Amazon Relational Database Service (Amazon RDS) para Oracle en la nube de Nube de AWS.
- Recomprar (readquirir): cambie a un producto diferente, lo cual se suele llevar a cabo al pasar de una licencia tradicional a un modelo SaaS. Ejemplo: Migrar el sistema de administración de las relaciones con los clientes (CRM) a Salesforce.com.
- Volver a alojar (migrar mediante lift-and-shift): traslade una aplicación a la nube sin realizar cambios para aprovechar las capacidades de la nube. Ejemplo: Migrar la base de datos de Oracle en las instalaciones a Oracle en una instancia de EC2 en la Nube de AWS.
- Reubicar: (migrar el hipervisor mediante lift and shift): traslade la infraestructura a la nube sin comprar equipo nuevo, reescribir aplicaciones o modificar las operaciones actuales. Los servidores se migran de una plataforma en las instalaciones a un servicio en la nube para la misma plataforma. Ejemplo: migrar una Microsoft Hyper-V aplicación a AWS.
- Retener (revisitar): conserve las aplicaciones en el entorno de origen. Estas pueden incluir las aplicaciones que requieren una refactorización importante, que desee posponer para más adelante, y las aplicaciones heredadas que desee retener, ya que no hay ninguna justificación empresarial para migrarlas.

- Retirar: retire o elimine las aplicaciones que ya no sean necesarias en un entorno de origen.

A

ABAC

Consulte control de acceso basado en atributos.

servicios abstractos

Consulte servicios administrados.

ACID

Consulte atomicidad, consistencia, aislamiento, durabilidad.

migración activa-activa

Método de migración de bases de datos en el que las bases de datos de origen y destino se mantienen sincronizadas (mediante una herramienta de replicación bidireccional o mediante operaciones de escritura doble) y ambas bases de datos gestionan las transacciones de las aplicaciones conectadas durante la migración. Este método permite la migración en lotes pequeños y controlados, en lugar de requerir una transición única. Es más flexible, pero requiere más trabajo que una migración activa-pasiva.

migración activa-pasiva

Método de migración de bases de datos en el que las bases de datos de origen y destino se mantienen sincronizadas, pero solo la de origen gestiona las transacciones de las aplicaciones conectadas, mientras los datos se replican en la de destino. La base de datos de destino no acepta ninguna transacción durante la migración.

función de agregación

Función SQL que actúa en un grupo de filas y calcula un único valor de devolución para el grupo. Entre los ejemplos de funciones de agregación se incluyen `SUM` y `MAX`.

IA

Consulte inteligencia artificial.

AIOps

Consulte operaciones de inteligencia artificial

anonimización

El proceso de eliminar permanentemente la información personal de un conjunto de datos. La anonimización puede ayudar a proteger la privacidad personal. Los datos anonimizados ya no se consideran datos personales.

antipatrones

Una solución que se utiliza con frecuencia para un problema recurrente en el que la solución es contraproducente, ineficaz o menos eficaz que una alternativa.

control de aplicaciones

Enfoque de seguridad que permite usar de manera exclusiva aplicaciones aprobadas para ayudar a proteger un sistema contra el malware.

cartera de aplicaciones

Recopilación de información detallada sobre cada aplicación que utiliza una organización, incluido el costo de creación y mantenimiento de la aplicación y su valor empresarial. Esta información es clave para el proceso de detección y análisis de la cartera y ayuda a identificar y priorizar las aplicaciones que se van a migrar, modernizar y optimizar.

inteligencia artificial (IA)

El campo de la informática que se dedica al uso de tecnologías informáticas para realizar funciones cognitivas que suelen estar asociadas a los seres humanos, como el aprendizaje, la resolución de problemas y el reconocimiento de patrones. Para más información, consulte ¿Qué es la inteligencia artificial?

operaciones de inteligencia artificial (AIOps)

El proceso de utilizar técnicas de machine learning para resolver problemas operativos, reducir los incidentes operativos y la intervención humana, y mejorar la calidad del servicio. Para obtener más información sobre cómo AIOps se utiliza en la estrategia de AWS migración, consulte la guía de integración de operaciones.

cifrado asimétrico

Algoritmo de cifrado que utiliza un par de claves, una clave pública para el cifrado y una clave privada para el descifrado. Puede compartir la clave pública porque no se utiliza para el descifrado, pero el acceso a la clave privada debe estar sumamente restringido.

atomicidad, consistencia, aislamiento, durabilidad (ACID)

Conjunto de propiedades de software que garantizan la validez de los datos y la fiabilidad operativa de una base de datos, incluso en caso de errores, cortes de energía u otros problemas.

control de acceso basado en atributos (ABAC)

La práctica de crear permisos detallados basados en los atributos del usuario, como el departamento, el puesto de trabajo y el nombre del equipo. Para obtener más información, consulte ABAC AWS en la documentación AWS Identity and Access Management (IAM).

origen de datos fidedigno

Ubicación en la que se almacena la versión principal de los datos, que se considera la fuente de información más fiable. Puede copiar los datos del origen de datos autorizado a otras ubicaciones con el fin de procesarlos o modificarlos, por ejemplo, anonimizarlos, redactarlos o seudonimizarlos.

Zona de disponibilidad

Una ubicación distinta dentro de una Región de AWS que está aislada de los fallos en otras zonas de disponibilidad y que proporciona una conectividad de red económica y de baja latencia a otras zonas de disponibilidad de la misma región.

AWS Marco de adopción de la nube (AWS CAF)

Un marco de directrices y mejores prácticas AWS para ayudar a las organizaciones a desarrollar un plan eficiente y eficaz para migrar con éxito a la nube. AWS CAF organiza la orientación en seis áreas de enfoque denominadas perspectivas: negocios, personas, gobierno, plataforma, seguridad y operaciones. Las perspectivas empresariales, humanas y de gobernanza se centran en las habilidades y los procesos empresariales; las perspectivas de plataforma, seguridad y operaciones se centran en las habilidades y los procesos técnicos. Por ejemplo, la perspectiva humana se dirige a las partes interesadas que se ocupan de los Recursos Humanos (RR. HH.), las funciones del personal y la administración de las personas. Desde esta perspectiva, AWS CAF proporciona orientación para el desarrollo, la formación y la comunicación de las personas a fin de preparar a la organización para una adopción exitosa de la nube. Para obtener más información, consulte la Página web de AWS CAF y el Documento técnico de AWS CAF.

AWS Marco de calificación de la carga de trabajo (AWS WQF)

Herramienta que evalúa las cargas de trabajo de migración de bases de datos, recomienda estrategias de migración y proporciona estimaciones de trabajo. AWS WQF se incluye con AWS

Schema Conversion Tool ().AWS SCT Analiza los esquemas de bases de datos y los objetos de código, el código de las aplicaciones, las dependencias y las características de rendimiento y proporciona informes de evaluación.

B

bot malicioso

Bot destinado a causar interrupciones o daños a personas u organizaciones.

BCP

Consulte planificación de la continuidad del negocio.

gráfico de comportamiento

Una vista unificada e interactiva del comportamiento de los recursos y de las interacciones a lo largo del tiempo. Puede utilizar un gráfico de comportamiento con Amazon Detective para examinar los intentos de inicio de sesión fallidos, las llamadas sospechosas a la API y acciones similares. Para obtener más información, consulte Datos en un gráfico de comportamiento en la documentación de Detective.

sistema big-endian

Un sistema que almacena primero el byte más significativo. Consulte también endianidad.

clasificación binaria

Un proceso que predice un resultado binario (una de las dos clases posibles). Por ejemplo, es posible que su modelo de ML necesite predecir problemas como “¿Este correo electrónico es spam o no es spam?” o “¿Este producto es un libro o un automóvil?”.

filtro de floración

Estructura de datos probabilística y eficiente en términos de memoria que se utiliza para comprobar si un elemento es miembro de un conjunto.

implementación azul/verde

Estrategia de implementación en la que se crean dos entornos separados, pero idénticos. La versión actual de la aplicación se ejecuta en un entorno (azul) y la nueva versión de la aplicación se ejecuta en el otro entorno (verde). Esta estrategia lo ayuda a hacer reversiones rápidas con un impacto mínimo.

bot

Aplicación de software que ejecuta tareas automatizadas a través de Internet y simula la actividad o interacción humana. Algunos bots son útiles o beneficiosos, como los rastreadores web que indexan la información de Internet. Otros bots, conocidos como bots maliciosos, tienen como objetivo causar interrupciones o daños a personas u organizaciones.

botnet

Redes de bots infectadas por malware y que están bajo el control de una sola parte, conocida como pastor de bots u operador de bots. Las botnets son el mecanismo más conocido para escalar los bots y su impacto.

branch

Área contenida de un repositorio de código. La primera rama que se crea en un repositorio es la rama principal. Puede crear una rama nueva a partir de una rama existente y, a continuación, desarrollar características o corregir errores en la rama nueva. Una rama que se genera para crear una característica se denomina comúnmente rama de característica. Cuando la característica se encuentra lista para su lanzamiento, se vuelve a combinar la rama de característica con la rama principal. Para obtener más información, consulte Acerca de las sucursales (GitHub documentación).

acceso de emergencia

En circunstancias excepcionales y mediante un proceso aprobado, es una forma rápida de que un usuario pueda acceder a un Cuenta de AWS sitio al que normalmente no tiene permisos de acceso. Para más información, consulte el indicador Implement break-glass procedures en la guía de AWS Well-Architected.

estrategia de implementación sobre infraestructura existente

La infraestructura existente en su entorno. Al adoptar una estrategia de implementación sobre infraestructura existente para una arquitectura de sistemas, se diseña la arquitectura en función de las limitaciones de los sistemas y la infraestructura actuales. Si está ampliando la infraestructura existente, puede combinar las estrategias de implementación sobre infraestructuras existentes y de implementación desde cero.

caché de búfer

El área de memoria donde se almacenan los datos a los que se accede con más frecuencia.

capacidad empresarial

Lo que hace una empresa para generar valor (por ejemplo, ventas, servicio al cliente o marketing). Las arquitecturas de microservicios y las decisiones de desarrollo pueden estar impulsadas por las capacidades empresariales. Para obtener más información, consulte la sección Organizado en torno a las capacidades empresariales del documento técnico Ejecutar microservicios en contenedores en AWS.

planificación de la continuidad del negocio (BCP)

Plan que aborda el posible impacto de un evento disruptivo, como una migración a gran escala en las operaciones y permite a la empresa reanudar las operaciones rápidamente.

C

CAF

Consulte AWS Cloud Adoption Framework.

implementación canario

Lanzamiento lento e incremental de una versión para los usuarios finales. Cuando tenga mayor confianza en la nueva versión, la implementa y reemplaza la versión actual en su totalidad.

CCoE

Consulte Centro de excelencia en la nube.

CDC

Consulte captura de datos de cambios.

captura de datos de cambio (CDC)

Proceso de seguimiento de los cambios en un origen de datos, como una tabla de base de datos, y registro de los metadatos relacionados con el cambio. Puede utilizar los CDC para diversos fines, como auditar o replicar los cambios en un sistema de destino para mantener la sincronización.

ingeniería del caos

Introducción intencionada de fallos o eventos disruptivos para poner a prueba la resiliencia de un sistema. Puedes usar AWS Fault Injection Service (AWS FIS) para realizar experimentos que estresen tus AWS cargas de trabajo y evalúen su respuesta.

CI/CD

Consulte integración continua y entrega continua.

clasificación

Un proceso de categorización que permite generar predicciones. Los modelos de ML para problemas de clasificación predicen un valor discreto. Los valores discretos siempre son distintos entre sí. Por ejemplo, es posible que un modelo necesite evaluar si hay o no un automóvil en una imagen.

cifrado del cliente

Cifrado de datos localmente, antes de que el objetivo los Servicio de AWS reciba.

Centro de excelencia en la nube (CCoE)

Equipo multidisciplinario que impulsa los esfuerzos de adopción de la nube en toda la organización, incluido el desarrollo de las prácticas recomendadas en la nube, la movilización de recursos, el establecimiento de plazos de migración y la dirección de la organización durante las transformaciones a gran escala. Para obtener más información, consulte las publicaciones de CCo E en el blog de estrategia Nube de AWS empresarial.

computación en la nube

La tecnología en la nube que se utiliza normalmente para la administración de dispositivos de IoT y el almacenamiento de datos de forma remota. La computación en la nube suele estar relacionada con la tecnología de computación de periferia.

modelo operativo en la nube

En una organización de TI, el modelo operativo que se utiliza para crear, madurar y optimizar uno o más entornos de nube. Para obtener más información, consulte Creación de su modelo operativo de nube.

etapas de adopción de la nube

Las siguientes son las cuatro fases por las que suelen pasar las empresas cuando migran a la Nube de AWS:

- Proyecto: ejecución de algunos proyectos relacionados con la nube con fines de prueba de concepto y aprendizaje
- Fundamento: realizar inversiones fundamentales para escalar su adopción de la nube (p. ej., crear una landing zone, definir una CCo E, establecer un modelo de operaciones)

- Migración: migración de aplicaciones individuales
- Reinvención: optimización de productos y servicios e innovación en la nube

Stephen Orban definió estas etapas en la entrada del blog The Journey Toward Cloud-First & the Stages of Adoption en el blog Nube de AWS Enterprise Strategy. Para obtener información sobre su relación con la estrategia de AWS migración, consulte la guía de preparación para la migración.

CMDB

Consulte base de datos de administración de configuración.

repositorio de código

Una ubicación donde el código fuente y otros activos, como documentación, muestras y scripts, se almacenan y actualizan mediante procesos de control de versiones. Algunos repositorios en la nube comunes son GitHub o Bitbucket Cloud. Cada versión del código se denomina rama. En una estructura de microservicios, cada repositorio se encuentra dedicado a una única funcionalidad. Una sola canalización de CI/CD puede utilizar varios repositorios.

caché en frío

Una caché de búfer que está vacía no está bien poblada o contiene datos obsoletos o irrelevantes. Esto afecta al rendimiento, ya que la instancia de la base de datos debe leer desde la memoria principal o el disco, lo que es más lento que leer desde la memoria caché del búfer.

datos fríos

Datos a los que se accede con poca frecuencia y que suelen ser históricos. Al consultar este tipo de datos, normalmente se aceptan consultas lentas. Trasladar estos datos a niveles o clases de almacenamiento de menor rendimiento y menos costosos puede reducir los costos.

visión artificial (CV)

Campo de la IA que utiliza el machine learning para analizar y extraer información de formatos visuales, como imágenes y videos digitales. Por ejemplo, Amazon SageMaker AI proporciona algoritmos de procesamiento de imágenes para CV.

deriva de configuración

En el caso de una carga de trabajo, un cambio en la configuración con respecto al estado esperado. Podría provocar que la carga de trabajo deje de cumplir las normas y, por lo general, es gradual e involuntaria.

base de datos de administración de configuración (CMDB)

Repositorio que almacena y administra información sobre una base de datos y su entorno de TI, incluidos los componentes de hardware y software y sus configuraciones. Por lo general, los datos de una CMDB se utilizan en la etapa de detección y análisis de la cartera de productos durante la migración.

paquete de conformidad

Un conjunto de AWS Config reglas y medidas correctivas que puede reunir para personalizar sus controles de conformidad y seguridad. Puede implementar un paquete de conformidad como una entidad única en una región Cuenta de AWS y, o en una organización, mediante una plantilla YAML. Para obtener más información, consulta los paquetes de conformidad en la documentación. AWS Config

integración y entrega continuas (CI/CD)

El proceso de automatización de las etapas de origen, compilación, prueba, puesta en escena y producción del proceso de publicación del software. CI/CD se describe comúnmente como una canalización. CI/CD puede ayudarlo a automatizar los procesos, mejorar la productividad, mejorar la calidad del código y entregar más rápido. Para obtener más información, consulte Beneficios de la entrega continua. CD también puede significar implementación continua. Para obtener más información, consulte Entrega continua frente a implementación continua.

CV

Consulte visión artificial.

D

datos en reposo

Datos que están estacionarios en la red, como los datos que se encuentran almacenados.

clasificación de datos

Un proceso para identificar y clasificar los datos de su red en función de su importancia y sensibilidad. Es un componente fundamental de cualquier estrategia de administración de riesgos de ciberseguridad porque lo ayuda a determinar los controles de protección y retención adecuados para los datos. La clasificación de datos es un componente del pilar de seguridad del AWS Well-Architected Framework. Para obtener más información, consulte Clasificación de datos.

deriva de datos

Una variación significativa entre los datos de producción y los datos que se utilizaron para entrenar un modelo de machine learning, o un cambio significativo en los datos de entrada a lo largo del tiempo. La deriva de datos puede reducir la calidad, la precisión y la imparcialidad generales de las predicciones de los modelos de machine learning.

datos en tránsito

Datos que se mueven de forma activa por la red, por ejemplo, entre los recursos de la red.

malla de datos

Marco de arquitectura que proporciona una propiedad de datos distribuida y descentralizada con una administración y una gobernanza centralizadas.

minimización de datos

El principio de recopilar y procesar solo los datos estrictamente necesarios. Practicar la minimización de los datos Nube de AWS puede reducir los riesgos de privacidad, los costos y la huella de carbono de la analítica.

perímetro de datos

Un conjunto de barreras preventivas en su AWS entorno que ayudan a garantizar que solo las identidades confiables accedan a los recursos confiables desde las redes esperadas. Para obtener más información, consulte Crear un perímetro de datos sobre. AWS

preprocesamiento de datos

Transformar los datos sin procesar en un formato que su modelo de ML pueda analizar fácilmente. El preprocesamiento de datos puede implicar eliminar determinadas columnas o filas y corregir los valores faltantes, incoherentes o duplicados.

procedencia de los datos

El proceso de rastrear el origen y el historial de los datos a lo largo de su ciclo de vida, por ejemplo, la forma en que se generaron, transmitieron y almacenaron los datos.

titular de los datos

Persona cuyos datos se recopilan y procesan.

almacenamiento de datos

Sistema de administración de datos que respalda la inteligencia empresarial, como los análisis. Los almacenes de datos suelen contener grandes cantidades de datos históricos y, por lo general, se utilizan para las consultas y los análisis.

lenguaje de definición de datos (DDL)

Instrucciones o comandos para crear o modificar la estructura de tablas y objetos de una base de datos.

lenguaje de manipulación de datos (DML)

Instrucciones o comandos para modificar (insertar, actualizar y eliminar) la información de una base de datos.

DDL

Consulte lenguaje de definición de bases de datos.

conjunto profundo

Combinar varios modelos de aprendizaje profundo para la predicción. Puede utilizar conjuntos profundos para obtener una predicción más precisa o para estimar la incertidumbre de las predicciones.

aprendizaje profundo

Un subcampo del ML que utiliza múltiples capas de redes neuronales artificiales para identificar el mapeo entre los datos de entrada y las variables objetivo de interés.

defense-in-depth

Un enfoque de seguridad de la información en el que se distribuyen cuidadosamente una serie de mecanismos y controles de seguridad en una red informática para proteger la confidencialidad, la integridad y la disponibilidad de la red y de los datos que contiene. Al adoptar esta estrategia AWS, se añaden varios controles en diferentes capas de la AWS Organizations estructura para ayudar a proteger los recursos. Por ejemplo, un defense-in-depth enfoque podría combinar la autenticación multifactorial, la segmentación de la red y el cifrado.

administrador delegado

En AWS Organizations, un servicio compatible puede registrar una cuenta de AWS miembro para administrar las cuentas de la organización y gestionar los permisos de ese servicio. Esta

cuenta se denomina administrador delegado para ese servicio. Para obtener más información y una lista de servicios compatibles, consulte Servicios que funcionan con AWS Organizations en la documentación de AWS Organizations .

Implementación

El proceso de hacer que una aplicación, características nuevas o correcciones de código se encuentren disponibles en el entorno de destino. La implementación abarca implementar cambios en una base de código y, a continuación, crear y ejecutar esa base en los entornos de la aplicación.

entorno de desarrollo

Consulte entorno.

control de detección

Un control de seguridad que se ha diseñado para detectar, registrar y alertar después de que se produzca un evento. Estos controles son una segunda línea de defensa, ya que lo advierten sobre los eventos de seguridad que han eludido los controles preventivos establecidos. Para obtener más información, consulte Controles de detección en Implementación de controles de seguridad en AWS.

asignación de flujos de valor para el desarrollo (DVSM)

Proceso que se utiliza para identificar y priorizar las restricciones que afectan negativamente a la velocidad y la calidad en el ciclo de vida del desarrollo de software. DVSM amplía el proceso de asignación del flujo de valor diseñado originalmente para las prácticas de fabricación ajustada. Se centra en los pasos y los equipos necesarios para crear y transferir valor a través del proceso de desarrollo de software.

gemelo digital

Representación virtual de un sistema del mundo real, como un edificio, una fábrica, un equipo industrial o una línea de producción. Los gemelos digitales son compatibles con el mantenimiento predictivo, la supervisión remota y la optimización de la producción.

tabla de dimensiones

En un esquema en estrella, tabla más pequeña que contiene los atributos de datos sobre los datos cuantitativos en una tabla de hechos. Los atributos de la tabla de dimensiones suelen ser campos de texto o números discretos que se comportan como texto. Estos atributos se suelen utilizar para restringir consultas, filtrarlas y etiquetar los conjuntos de resultados.

desastre

Un evento que impide que una carga de trabajo o un sistema cumplan sus objetivos empresariales en su ubicación principal de implementación. Estos eventos pueden ser desastres naturales, fallos técnicos o el resultado de acciones humanas, como una configuración incorrecta involuntaria o un ataque de malware.

recuperación de desastres (DR)

Estrategia y proceso que utiliza para minimizar el tiempo de inactividad y la pérdida de datos a causa de un desastre. Para obtener más información, consulte Recuperación ante desastres de cargas de trabajo en AWS: Recovery in the Cloud in the AWS Well-Architected Framework.

DML

Consulte lenguaje de manipulación de bases de datos.

diseño basado en el dominio

Un enfoque para desarrollar un sistema de software complejo mediante la conexión de sus componentes a dominios en evolución, o a los objetivos empresariales principales, a los que sirve cada componente. Este concepto lo introdujo Eric Evans en su libro, Diseño impulsado por el dominio: abordando la complejidad en el corazón del software (Boston: Addison-Wesley Professional, 2003). Para obtener información sobre cómo utilizar el diseño basado en dominios con el patrón de higos estranguladores, consulte Modernización gradual de los servicios web antiguos de Microsoft ASP.NET (ASMX) mediante contenedores y Amazon API Gateway.

DR

Consulte recuperación ante desastres.

Detección de desviaciones

Seguimiento de las desviaciones con respecto a una configuración con línea de base. Por ejemplo, puedes usarlo AWS CloudFormation para detectar desviaciones en los recursos del sistema o puedes usarlo AWS Control Tower para detectar cambios en tu landing zone que puedan afectar al cumplimiento de los requisitos de gobierno.

DVSM

Consulte asignación de flujos de valor para el desarrollo.

E

EDA

Consulte análisis de datos de tipo exploratorio.

EDI

Consulte intercambio electrónico de datos.

computación en la periferia

La tecnología que aumenta la potencia de cálculo de los dispositivos inteligentes en la periferia de una red de IoT. En comparación con la computación en la nube, la computación de periferia puede reducir la latencia de la comunicación y mejorar el tiempo de respuesta.

intercambio electrónico de datos (EDI)

Intercambio automatizado de documentos comerciales entre organizaciones. Para más información, consulte ¿Qué es el intercambio electrónico de datos?

cifrado

Proceso de computación que transforma datos de texto plano, que son legibles por humanos, en texto cifrado.

clave de cifrado

Cadena criptográfica de bits aleatorios que se genera mediante un algoritmo de cifrado. Las claves pueden variar en longitud y cada una se ha diseñado para ser impredecible y única.

endianidad

El orden en el que se almacenan los bytes en la memoria del ordenador. Los sistemas big-endianos almacenan primero el byte más significativo. Los sistemas Little-Endian almacenan primero el byte menos significativo.

punto de conexión

Consulte punto de conexión de servicio.

servicio de punto de conexión

Servicio que puede alojar en una nube privada virtual (VPC) para compartir con otros usuarios. Puede crear un servicio de punto final AWS PrivateLink y conceder permisos a otros Cuentas de AWS o a responsables AWS Identity and Access Management (de IAM). Estas cuentas o

entidades principales pueden conectarse a su servicio de punto de conexión de forma privada mediante la creación de puntos de conexión de VPC de interfaz. Para obtener más información, consulte Creación de un servicio de punto de conexión en la documentación de Amazon Virtual Private Cloud (Amazon VPC).

planificación de recursos empresariales (ERP)

Sistema que automatiza y administra los procesos empresariales clave (como la contabilidad, MES y la administración de proyectos) de una empresa.

cifrado de sobre

El proceso de cifrar una clave de cifrado con otra clave de cifrado. Para obtener más información, consulte el cifrado de sobres en la documentación de AWS Key Management Service (AWS KMS).

entorno

Una instancia de una aplicación en ejecución. Los siguientes son los tipos de entornos más comunes en la computación en la nube:

- entorno de desarrollo: instancia de una aplicación en ejecución que solo se encuentra disponible para el equipo principal responsable del mantenimiento de la aplicación. Los entornos de desarrollo se utilizan para probar los cambios antes de promocionarlos a los entornos superiores. Este tipo de entorno a veces se denomina entorno de prueba.
- entornos inferiores: todos los entornos de desarrollo de una aplicación, como los que se utilizan para las compilaciones y pruebas iniciales.
- entorno de producción: instancia de una aplicación en ejecución a la que pueden acceder los usuarios finales. En un CI/CD proceso, el entorno de producción es el último entorno de implementación.
- entornos superiores: todos los entornos a los que pueden acceder usuarios que no sean del equipo de desarrollo principal. Esto puede incluir un entorno de producción, entornos de preproducción y entornos para las pruebas de aceptación por parte de los usuarios.

epopeya

En las metodologías ágiles, son categorías funcionales que ayudan a organizar y priorizar el trabajo. Las epopeyas brindan una descripción detallada de los requisitos y las tareas de implementación. Por ejemplo, las epopeyas AWS de seguridad de CAF incluyen la gestión de identidades y accesos, los controles de detección, la seguridad de la infraestructura, la protección de datos y la respuesta a incidentes. Para obtener más información sobre las epopeyas en la estrategia de migración de AWS , consulte la Guía de implementación del programa.

ERP

Consulte planificación de recursos empresariales.

análisis de datos de tipo exploratorio (EDA)

El proceso de analizar un conjunto de datos para comprender sus características principales. Se recopilan o agregan datos y, a continuación, se realizan las investigaciones iniciales para encontrar patrones, detectar anomalías y comprobar las suposiciones. El EDA se realiza mediante el cálculo de estadísticas resumidas y la creación de visualizaciones de datos.

F

tabla de hechos

Tabla central de un esquema en estrella. Almacena datos cuantitativos sobre operaciones empresariales. Por lo general, una tabla de hechos contiene dos tipos de columnas: las que contienen medidas y las que contienen una clave externa para una tabla de dimensiones.

Fail Fast

Filosofía que utiliza pruebas frecuentes e incrementales para reducir el ciclo de vida del desarrollo. Es una parte fundamental de los enfoques ágiles.

límite de aislamiento de errores

En el Nube de AWS, un límite, como una zona de disponibilidad Región de AWS, un plano de control o un plano de datos, que limita el efecto de una falla y ayuda a mejorar la resiliencia de las cargas de trabajo. Para más información, consulte AWS Fault Isolation Boundaries.

rama de característica

Consulte rama.

características

Los datos de entrada que se utilizan para hacer una predicción. Por ejemplo, en un contexto de fabricación, las características pueden ser imágenes que se capturan periódicamente desde la línea de fabricación.

importancia de las características

La importancia que tiene una característica para las predicciones de un modelo. Por lo general, esto se expresa como una puntuación numérica que se puede calcular mediante diversas

técnicas, como las explicaciones aditivas de Shapley (SHAP) y los gradientes integrados. Para obtener más información, consulte Interpretabilidad del modelo de aprendizaje automático con. AWS

transformación de funciones

Optimizar los datos para el proceso de ML, lo que incluye enriquecer los datos con fuentes adicionales, escalar los valores o extraer varios conjuntos de información de un solo campo de datos. Esto permite que el modelo de ML se beneficie de los datos. Por ejemplo, si divide la fecha del “27 de mayo de 2021 00:15:37” en “jueves”, “mayo”, “2021” y “15”, puede ayudar al algoritmo de aprendizaje a aprender patrones matizados asociados a los diferentes componentes de los datos.

peticiones con pocos pasos

Proporcionar a un LLM una pequeña cantidad de ejemplos que demuestren la tarea y el resultado deseado antes de pedirle que lleve a cabo una tarea similar. Esta técnica es una aplicación del aprendizaje contextual, mediante el que los modelos aprenden a partir de ejemplos (pasos) incrustados en las peticiones. La técnica de peticiones con pocos pasos puede ser eficaz para las tareas que requieren un formato, un razonamiento o un conocimiento del dominio específicos. Consulte también peticiones desde cero.

FGAC

Consulte control de acceso detallado.

control de acceso preciso (FGAC)

El uso de varias condiciones que tienen por objetivo permitir o denegar una solicitud de acceso.

migración relámpago

Método de migración de bases de datos que utiliza la replicación continua de datos mediante la captura de datos de cambio para migrar los datos en el menor tiempo posible, en lugar de utilizar un enfoque gradual. El objetivo es reducir al mínimo el tiempo de inactividad.

FM

Consulte modelo fundacional.

Modelo fundacional (FM)

Una gran red neuronal de aprendizaje profundo que se ha estado entrenando con conjuntos de datos masivos de datos generalizados y sin etiquetar. FMs son capaces de realizar una amplia variedad de tareas generales, como comprender el lenguaje, generar texto e imágenes

y conversar en lenguaje natural. Para más información, consulte ¿Qué son los modelos fundacionales?

G

IA generativa

Subconjunto de modelos de IA que se entrenaron con grandes cantidades de datos y que pueden utilizar una simple petición de texto para crear contenido y artefactos nuevos, como imágenes, videos, texto y audio. Para más información, consulte ¿Qué es la IA generativa?

bloqueo geográfico

Consulte restricciones geográficas.

restricciones geográficas (bloqueo geográfico)

En Amazon CloudFront, una opción para impedir que los usuarios de países específicos accedan a las distribuciones de contenido. Puede utilizar una lista de permitidos o bloqueados para especificar los países aprobados y prohibidos. Para obtener más información, consulta la sección Restringir la distribución geográfica del contenido en la CloudFront documentación.

Flujo de trabajo de Gitflow

Un enfoque en el que los entornos inferiores y superiores utilizan diferentes ramas en un repositorio de código fuente. El flujo de trabajo de Gitflow se considera heredado, mientras que el flujo de trabajo basado en enlaces troncales es el enfoque moderno preferido.

imagen dorada

Instantánea de un sistema o software que se usa como plantilla para implementar nuevas instancias de ese sistema o software. Por ejemplo, en la fabricación, una imagen dorada se puede utilizar para aprovisionar software en varios dispositivos y ayuda a mejorar la velocidad, la escalabilidad y la productividad de las operaciones de fabricación de dispositivos.

estrategia de implementación desde cero

La ausencia de infraestructura existente en un entorno nuevo. Al adoptar una estrategia de implementación desde cero para una arquitectura de sistemas, puede seleccionar todas las tecnologías nuevas sin que estas deban ser compatibles con una infraestructura existente, lo que también se conoce como implementación sobre infraestructura existente. Si está ampliando la infraestructura existente, puede combinar las estrategias de implementación sobre infraestructuras existentes y de implementación desde cero.

barrera de protección

Una regla de alto nivel que ayuda a regular los recursos, las políticas y el cumplimiento en todas las unidades organizativas (OUs). Las barreras de protección preventivas aplican políticas para garantizar la alineación con los estándares de conformidad. Se implementan mediante políticas de control de servicios y límites de permisos de IAM. Las barreras de protección de detección detectan las vulneraciones de las políticas y los problemas de conformidad, y generan alertas para su corrección. Se implementan mediante Amazon AWS Config AWS Security Hub CSPM GuardDuty AWS Trusted Advisor, Amazon Inspector y AWS Lambda cheques personalizados.

H

HA

Consulte alta disponibilidad.

migración heterogénea de bases de datos

Migración de la base de datos de origen a una base de datos de destino que utilice un motor de base de datos diferente (por ejemplo, de Oracle a Amazon Aurora). La migración heterogénea suele ser parte de un esfuerzo de rediseño de la arquitectura y convertir el esquema puede ser una tarea compleja. AWS ofrece AWS SCT, lo cual ayuda con las conversiones de esquemas.

alta disponibilidad (HA)

La capacidad de una carga de trabajo para funcionar de forma continua, sin intervención, en caso de desafíos o desastres. Los sistemas de alta disponibilidad están diseñados para realizar una conmutación por error automática, ofrecer un rendimiento de alta calidad de forma constante y gestionar diferentes cargas y fallos con un impacto mínimo en el rendimiento.

modernización histórica

Un enfoque utilizado para modernizar y actualizar los sistemas de tecnología operativa (TO) a fin de satisfacer mejor las necesidades de la industria manufacturera. Un histórico es un tipo de base de datos que se utiliza para recopilar y almacenar datos de diversas fuentes en una fábrica.

datos de reserva

Parte de los datos históricos etiquetados que se ocultan de un conjunto de datos que se utiliza para entrenar un modelo de machine learning. Puede utilizar los datos de reserva para evaluar el rendimiento del modelo mediante la comparación de las predicciones del modelo con los datos de reserva.

migración homogénea de bases de datos

Migración de la base de datos de origen a una base de datos de destino que comparte el mismo motor de base de datos (por ejemplo, Microsoft SQL Server a Amazon RDS para SQL Server). La migración homogénea suele formar parte de un esfuerzo para volver a alojar o redefinir la plataforma. Puede utilizar las utilidades de bases de datos nativas para migrar el esquema.

datos recientes

Datos a los que se accede con frecuencia, como datos en tiempo real o datos traslacionales recientes. Por lo general, estos datos requieren un nivel o una clase de almacenamiento de alto rendimiento para proporcionar respuestas rápidas a las consultas.

hotfix

Una solución urgente para un problema crítico en un entorno de producción. Debido a su urgencia, una revisión suele realizarse fuera del flujo de trabajo de DevOps publicación típico.

periodo de hiperatención

Periodo, inmediatamente después de la transición, durante el cual un equipo de migración administra y monitorea las aplicaciones migradas en la nube para solucionar cualquier problema. Por lo general, este periodo dura de 1 a 4 días. Al final del periodo de hiperatención, el equipo de migración suele transferir la responsabilidad de las aplicaciones al equipo de operaciones en la nube.

I

IaC

Consulte infraestructura como código.

políticas basadas en identidades

Política asociada a uno o más directores de IAM que define sus permisos en el entorno. Nube de AWS

aplicación inactiva

Aplicación que utiliza un promedio de CPU y memoria de entre 5 y 20 por ciento durante un periodo de 90 días. En un proyecto de migración, es habitual retirar estas aplicaciones o mantenerlas en las instalaciones.

IIoT

Consulte Internet de las cosas industrial.

infraestructura inmutable

Modelo que implementa una nueva infraestructura para las cargas de trabajo de producción en lugar de actualizar o modificar la infraestructura existente o aplicarle revisiones. Las infraestructuras inmutables son de manera intrínseca más coherentes, fiables y predecibles que las infraestructuras mutables. Para más información, consulte la práctica recomendada Implementación mediante una infraestructura inmutable en el Marco de AWS Well-Architected.

VPC entrante (de entrada)

En una arquitectura de AWS cuentas múltiples, una VPC que acepta, inspecciona y enruta las conexiones de red desde fuera de una aplicación. La arquitectura AWS de referencia de seguridad recomienda configurar la cuenta de red con entradas, salidas e inspección VPCs para proteger la interfaz bidireccional entre la aplicación y el resto de Internet.

migración gradual

Estrategia de transición en la que se migra la aplicación en partes pequeñas en lugar de realizar una transición única y completa. Por ejemplo, puede trasladar inicialmente solo unos pocos microservicios o usuarios al nuevo sistema. Tras comprobar que todo funciona correctamente, puede trasladar microservicios o usuarios adicionales de forma gradual hasta que pueda retirar su sistema heredado. Esta estrategia reduce los riesgos asociados a las grandes migraciones.

Industria 4.0

Término que introdujo Klaus Schwab en 2016 para referirse a la modernización de los procesos de fabricación mediante los avances en la conectividad, los datos en tiempo real, la automatización, el análisis, la IA y el ML.

infraestructura

Todos los recursos y activos que se encuentran en el entorno de una aplicación.

infraestructura como código (IaC)

Proceso de aprovisionamiento y administración de la infraestructura de una aplicación mediante un conjunto de archivos de configuración. La IaC se ha diseñado para ayudarlo a centralizar la administración de la infraestructura, estandarizar los recursos y escalar con rapidez a fin de que los entornos nuevos sean repetibles, fiables y consistentes.

Internet de las cosas industrial (T) IIo

El uso de sensores y dispositivos conectados a Internet en los sectores industriales, como el productivo, el eléctrico, el automotriz, el sanitario, el de las ciencias de la vida y el de la agricultura. Para obtener más información, consulte Creación de una estrategia de transformación digital de la Internet de las cosas (IIoT) industrial.

VPC de inspección

En una arquitectura de AWS cuentas múltiples, una VPC centralizada que gestiona las inspecciones del tráfico de red VPCs entre Internet y las redes locales (en una misma o Regiones de AWS diferente). La arquitectura AWS de referencia de seguridad recomienda configurar su cuenta de red con entrada, salida e inspección VPCs para proteger la interfaz bidireccional entre la aplicación e Internet en general.

Internet de las cosas (IoT)

Red de objetos físicos conectados con sensores o procesadores integrados que se comunican con otros dispositivos y sistemas a través de Internet o de una red de comunicación local. Para obtener más información, consulte ¿Qué es IoT?.

interpretabilidad

Característica de un modelo de machine learning que describe el grado en que un ser humano puede entender cómo las predicciones del modelo dependen de sus entradas. Para obtener más información, consulte Interpretabilidad del modelo de aprendizaje automático con. AWS

IoT

Consulte Internet de las cosas.

biblioteca de información de TI (ITIL)

Conjunto de prácticas recomendadas para ofrecer servicios de TI y alinearlos con los requisitos empresariales. La ITIL proporciona la base para la ITSM.

administración de servicios de TI (ITSM)

Actividades asociadas con el diseño, la implementación, la administración y el soporte de los servicios de TI para una organización. Para obtener información sobre la integración de las operaciones en la nube con las herramientas de ITSM, consulte la Guía de integración de operaciones.

ITIL

Consulte biblioteca de información de TI.

ITSM

Consulte administración de servicios de TI.

L

control de acceso basado en etiquetas (LBAC)

Una implementación del control de acceso obligatorio (MAC) en la que a los usuarios y a los propios datos se les asigna explícitamente un valor de etiqueta de seguridad. La intersección entre la etiqueta de seguridad del usuario y la etiqueta de seguridad de los datos determina qué filas y columnas puede ver el usuario.

zona de aterrizaje

Una landing zone es un AWS entorno multicuenta bien diseñado, escalable y seguro. Este es un punto de partida desde el cual las empresas pueden lanzar e implementar rápidamente cargas de trabajo y aplicaciones con confianza en su entorno de seguridad e infraestructura. Para obtener más información sobre las zonas de aterrizaje, consulte Configuración de un entorno de AWS seguro y escalable con varias cuentas.

modelo de lenguaje de gran tamaño (LLM)

Modelo de IA de aprendizaje profundo que se entrenó previamente con una gran cantidad de datos. Un LLM puede llevar a cabo varias tareas, como responder preguntas, resumir documentos, traducir textos a otros idiomas y completar oraciones. Para obtener más información, consulte Qué son. LLMs

migración grande

Migración de 300 servidores o más.

LBAC

Consulte control de acceso basado en etiquetas.

privilegio mínimo

La práctica recomendada de seguridad que consiste en conceder los permisos mínimos necesarios para realizar una tarea. Para obtener más información, consulte Aplicar permisos de privilegio mínimo en la documentación de IAM.

migrar mediante lift-and-shift

Consulte Las 7 R.

sistema little-endian

Un sistema que almacena primero el byte menos significativo. Consulte también endianidad.

LLM

Consulte modelo de lenguaje de gran tamaño.

entornos inferiores

Consulte entorno.

M

machine learning (ML)

Un tipo de inteligencia artificial que utiliza algoritmos y técnicas para el reconocimiento y el aprendizaje de patrones. El ML analiza y aprende de los datos registrados, como los datos del Internet de las cosas (IoT), para generar un modelo estadístico basado en patrones. Para más información, consulte Machine learning.

rama principal

Consulte rama.

malware

Software diseñado para comprometer la seguridad o la privacidad de la computadora. El malware podría interrumpir los sistemas informáticos, filtrar información confidencial u obtener acceso no autorizado. Algunos ejemplos de malware son los virus, los gusanos, el ransomware, los troyanos, el spyware y los registradores de pulsaciones de teclas.

Servicios administrados

Servicios de AWS para lo cual AWS opera la capa de infraestructura, el sistema operativo y las plataformas, y se accede a los puntos finales para almacenar y recuperar datos. Amazon Simple Storage Service (Amazon S3) y Amazon DynamoDB son ejemplos de servicios administrados. También se conocen como servicios abstractos.

sistema de ejecución de fabricación (MES)

Sistema de software para seguir, supervisar, documentar y controlar los procesos de producción que convierten las materias primas en productos acabados en la zona de producción.

MAP

Consulte Programa de aceleración de la migración.

mecanismo

Proceso completo mediante el que se crea una herramienta, se impulsa su adopción y, a continuación, se inspeccionan los resultados para hacer ajustes. Un mecanismo es un ciclo que se refuerza y mejora por sí mismo a medida que funciona. Para obtener más información, consulte Creación de mecanismos en el AWS Well-Architected Framework.

cuenta de miembro

Todas las Cuentas de AWS demás cuentas, excepto la de administración, que forman parte de una organización. AWS Organizations Una cuenta no puede pertenecer a más de una organización a la vez.

MES

Consulte sistema de ejecución de fabricación.

Message Queuing Telemetry Transport (MQTT)

Un protocolo de comunicación ligero machine-to-machine (M2M), basado en el patrón de publicación/suscripción, para dispositivos de IoT con recursos limitados.

microservicio

Un servicio pequeño e independiente que se comunica a través de una red bien definida APIs y que, por lo general, es propiedad de equipos pequeños e independientes. Por ejemplo, un sistema de seguros puede incluir microservicios que se adapten a las capacidades empresariales, como las de ventas o marketing, o a subdominios, como las de compras, reclamaciones o análisis. Los beneficios de los microservicios incluyen la agilidad, la escalabilidad flexible, la facilidad de implementación, el código reutilizable y la resiliencia. Para obtener más información, consulte Integrar microservicios mediante AWS servicios sin servidor.

arquitectura de microservicios

Un enfoque para crear una aplicación con componentes independientes que ejecutan cada proceso de la aplicación como un microservicio. Estos microservicios se comunican a través de una interfaz bien definida mediante un uso ligero. APIs Cada microservicio de esta arquitectura se puede actualizar, implementar y escalar para satisfacer la demanda de funciones específicas de una aplicación. Para obtener más información, consulte Implementación de microservicios en. AWS

Programa de aceleración de la migración (MAP)

Un AWS programa que proporciona soporte de consultoría, formación y servicios para ayudar a las organizaciones a crear una base operativa sólida para migrar a la nube y para ayudar a compensar el costo inicial de las migraciones. El MAP incluye una metodología de migración para ejecutar las migraciones antiguas de forma metódica y un conjunto de herramientas para automatizar y acelerar los escenarios de migración más comunes.

migración a escala

Proceso de transferencia de la mayoría de la cartera de aplicaciones a la nube en oleadas, con más aplicaciones desplazadas a un ritmo más rápido en cada oleada. En esta fase, se utilizan las prácticas recomendadas y las lecciones aprendidas en las fases anteriores para implementar una fábrica de migración de equipos, herramientas y procesos con el fin de agilizar la migración de las cargas de trabajo mediante la automatización y la entrega ágil. Esta es la tercera fase de la estrategia de migración de AWS.

fábrica de migración

Equipos multifuncionales que agilizan la migración de las cargas de trabajo mediante enfoques automatizados y ágiles. Los equipos de las fábricas de migración suelen incluir a analistas y propietarios de operaciones, empresas, ingenieros de migración, desarrolladores y DevOps profesionales que trabajan a pasos agigantados. Entre el 20 y el 50 por ciento de la cartera de aplicaciones empresariales se compone de patrones repetidos que pueden optimizarse mediante un enfoque de fábrica. Para obtener más información, consulte la discusión sobre las fábricas de migración y la Guía de fábricas de migración a la nube en este contenido.

metadatos de migración

Información sobre la aplicación y el servidor que se necesita para completar la migración. Cada patrón de migración requiere un conjunto diferente de metadatos de migración. Algunos ejemplos de metadatos de migración son la subred de destino, el grupo de seguridad y AWS la cuenta.

patrón de migración

Tarea de migración repetible que detalla la estrategia de migración, el destino de la migración y la aplicación o el servicio de migración utilizados. Ejemplo: rehospede la migración a Amazon EC2 AWS con Application Migration Service.

Migration Portfolio Assessment (MPA)

Herramienta en línea que proporciona información a fin de validar los argumentos comerciales necesarios para migrar a la Nube de AWS. La MPA ofrece una evaluación detallada de la cartera

(adecuación del tamaño de los servidores, precios, comparaciones del costo total de propiedad, análisis de los costos de migración), así como una planificación de la migración (análisis y recopilación de datos de aplicaciones, agrupación de aplicaciones, priorización de la migración y planificación de oleadas). La herramienta MPA (requiere iniciar sesión) está disponible de forma gratuita para todos los AWS consultores y consultores de los socios de APN.

Evaluación de la preparación para la migración (MRA)

Proceso que consiste en obtener información sobre el estado de preparación de una organización para la nube, identificar sus puntos fuertes y débiles y elaborar un plan de acción para cerrar las brechas identificadas mediante el AWS CAF. Para obtener más información, consulte la Guía de preparación para la migración. La MRA es la primera fase de la estrategia de migración de AWS.

estrategia de migración

Enfoque utilizado para migrar una carga de trabajo a la Nube de AWS. Para más información, consulte la entrada Las 7 R de este glosario y también Mobilize your organization to accelerate large-scale migrations.

ML

Consulte machine learning.

modernización

Transformar una aplicación obsoleta (antigua o monolítica) y su infraestructura en un sistema ágil, elástico y de alta disponibilidad en la nube para reducir los gastos, aumentar la eficiencia y aprovechar las innovaciones. Para más información, consulte Strategy for modernizing applications in the Nube de AWS.

evaluación de la preparación para la modernización

Evaluación que ayuda a determinar la preparación para la modernización de las aplicaciones de una organización; identifica los beneficios, los riesgos y las dependencias; y determina qué tan bien la organización puede soportar el estado futuro de esas aplicaciones. El resultado de la evaluación es un esquema de la arquitectura objetivo, una hoja de ruta que detalla las fases de desarrollo y los hitos del proceso de modernización y un plan de acción para abordar las brechas identificadas. Para más información, consulte Evaluating modernization readiness for applications in the Nube de AWS.

aplicaciones monolíticas (monolitos)

Aplicaciones que se ejecutan como un único servicio con procesos estrechamente acoplados. Las aplicaciones monolíticas presentan varios inconvenientes. Si una característica de la

aplicación experimenta un aumento en la demanda, se debe escalar toda la arquitectura. Agregar o mejorar las características de una aplicación monolítica también se vuelve más complejo a medida que crece la base de código. Para solucionar problemas con la aplicación, puede utilizar una arquitectura de microservicios. Para obtener más información, consulte Descomposición de monolitos en microservicios.

MPA

Consulte Migration Portfolio Assessment.

MQTT

Consulte Message Queuing Telemetry Transport.

clasificación multiclase

Un proceso que ayuda a generar predicciones para varias clases (predice uno de más de dos resultados). Por ejemplo, un modelo de ML podría preguntar "¿Este producto es un libro, un automóvil o un teléfono?" o "¿Qué categoría de productos es más interesante para este cliente?".

infraestructura mutable

Modelo que actualiza y modifica la infraestructura actual para las cargas de trabajo de producción. Para mejorar la coherencia, la fiabilidad y la previsibilidad, el AWS Well-Architected Framework recomienda el uso de una infraestructura inmutable como práctica recomendada.

O

OAC

Consulte control de acceso de origen.

OAI

Consulte identidad de acceso de origen.

OCM

Consulte administración del cambio organizacional.

migración fuera de línea

Método de migración en el que la carga de trabajo de origen se elimina durante el proceso de migración. Este método implica un tiempo de inactividad prolongado y, por lo general, se utiliza para cargas de trabajo pequeñas y no críticas.

OI

Consulte integración de operaciones.

OLA

Consulte acuerdo de nivel operativo.

migración en línea

Método de migración en el que la carga de trabajo de origen se copia al sistema de destino sin que se desconecte. Las aplicaciones que están conectadas a la carga de trabajo pueden seguir funcionando durante la migración. Este método implica un tiempo de inactividad nulo o mínimo y, por lo general, se utiliza para cargas de trabajo de producción críticas.

OPC-UA

Consulte Open Process Communications: arquitectura unificada.

Open Process Communications: arquitectura unificada (OPC-UA)

Un protocolo de machine-to-machine comunicación (M2M) para la automatización industrial. OPC-UA establece un estándar de interoperabilidad con esquemas de autenticación, autorización y cifrado de datos.

acuerdo de nivel operativo (OLA)

Acuerdo que aclara lo que los grupos de TI operativos se comprometen a ofrecerse entre sí, para respaldar un acuerdo de nivel de servicio (SLA).

revisión de la preparación operativa (ORR)

Lista de comprobación de preguntas y prácticas recomendadas asociadas que son útiles para comprender, evaluar, prevenir o reducir el alcance de los incidentes y posibles errores. Para más información, consulte Operational Readiness Reviews (ORR) en el Marco de AWS Well-Architected.

tecnología operativa (TO)

Sistemas de hardware y software que funcionan con el entorno físico para controlar las operaciones, los equipos y la infraestructura industriales. En el sector de la fabricación, la integración de los sistemas de TO y tecnología de la información (TI) es un enfoque clave para las transformaciones de la industria 4.0.

integración de operaciones (OI)

Proceso de modernización de las operaciones en la nube, que implica la planificación de la preparación, la automatización y la integración. Para obtener más información, consulte la Guía de integración de las operaciones.

registro de seguimiento organizativo

Un registro creado por y AWS CloudTrail que registra todos los eventos para todos los miembros Cuentas de AWS de una organización. AWS Organizations Este registro de seguimiento se crea en cada Cuenta de AWS que forma parte de la organización y realiza un seguimiento de la actividad en cada cuenta. Para obtener más información, consulte Crear un registro para una organización en la CloudTrail documentación.

administración del cambio organizacional (OCM)

Marco para administrar las transformaciones empresariales importantes y disruptivas desde la perspectiva de las personas, la cultura y el liderazgo. La OCM ayuda a las empresas a prepararse para nuevos sistemas y estrategias y a realizar la transición a ellos, al acelerar la adopción de cambios, abordar los problemas de transición e impulsar cambios culturales y organizacionales. En la estrategia de AWS migración, este marco se denomina aceleración de personal, debido a la velocidad de cambio que requieren los proyectos de adopción de la nube. Para obtener más información, consulte la Guía de OCM.

control de acceso de origen (OAC)

En CloudFront, una opción mejorada para restringir el acceso y proteger el contenido del Amazon Simple Storage Service (Amazon S3). El OAC admite todos los buckets de S3 Regiones de AWS, el cifrado del lado del servidor AWS KMS (SSE-KMS) y las solicitudes dinámicas `PUT` y `DELETE` dirigidas al bucket de S3.

identidad de acceso de origen (OAI)

En CloudFront, una opción para restringir el acceso y proteger el contenido de Amazon S3. Cuando utiliza OAI, CloudFront crea un principal con el que Amazon S3 puede autenticarse. Los directores autenticados solo pueden acceder al contenido de un bucket de S3 a través de una distribución específica. CloudFront Consulte también el OAC, que proporciona un control de acceso más detallado y mejorado.

ORR

Consulte revisión de la preparación operativa.

OT

Consulte tecnología operativa.

VPC saliente (de salida)

En una arquitectura de AWS cuentas múltiples, una VPC que gestiona las conexiones de red que se inician desde una aplicación. La arquitectura AWS de referencia de seguridad recomienda configurar la cuenta de red con entradas, salidas e inspección VPCs para proteger la interfaz bidireccional entre la aplicación e Internet en general.

P

límite de permisos

Una política de administración de IAM que se adjunta a las entidades principales de IAM para establecer los permisos máximos que puede tener el usuario o el rol. Para obtener más información, consulte Límites de permisos en la documentación de IAM.

información de identificación personal (PII)

Información que, vista directamente o combinada con otros datos relacionados, puede utilizarse para deducir de manera razonable la identidad de una persona. Algunos ejemplos de información de identificación personal son los nombres, las direcciones y la información de contacto.

PII

Consulte información de identificación personal.

manual de estrategias

Conjunto de pasos predefinidos que capturan el trabajo asociado a las migraciones, como la entrega de las funciones de operaciones principales en la nube. Un manual puede adoptar la forma de scripts, manuales de procedimientos automatizados o resúmenes de los procesos o pasos necesarios para operar un entorno modernizado.

PLC

Consulte controlador lógico programable.

PLM

Consulte administración del ciclo de vida del producto.

policy

Objeto que puede definir permisos (consulte política basada en identidad), especificar las condiciones de acceso (consulte política basada en recursos) o definir los permisos máximos para todas las cuentas de una organización de AWS Organizations (consulte política de control de servicio).

persistencia políglota

Elegir de forma independiente la tecnología de almacenamiento de datos de un microservicio en función de los patrones de acceso a los datos y otros requisitos. Si sus microservicios tienen la misma tecnología de almacenamiento de datos, pueden enfrentarse a desafíos de implementación o experimentar un rendimiento deficiente. Los microservicios se implementan más fácilmente y logran un mejor rendimiento y escalabilidad si utilizan el almacén de datos que mejor se adapte a sus necesidades.

evaluación de cartera

Proceso de detección, análisis y priorización de la cartera de aplicaciones para planificar la migración. Para obtener más información, consulte la Evaluación de la preparación para la migración.

predicate

Condición de consulta que devuelve `true` o `false`. En general, se encuentra en una cláusula `WHERE`.

inserción de predicados

Técnica de optimización de consultas en bases de datos que filtra los datos de la consulta antes de transferirlos. Esta técnica reduce la cantidad de datos de la base de datos relacional que se tienen que recuperar y procesar. Además, mejora el rendimiento de las consultas.

control preventivo

Un control de seguridad diseñado para evitar que ocurra un evento. Estos controles son la primera línea de defensa para evitar el acceso no autorizado o los cambios no deseados en la red. Para obtener más información, consulte Controles preventivos en Implementación de controles de seguridad en AWS.

entidad principal

Una entidad AWS que puede realizar acciones y acceder a los recursos. Esta entidad suele ser un usuario raíz para un Cuenta de AWS rol de IAM o un usuario. Para obtener más información, consulte Entidad principal en Términos y conceptos de roles en la documentación de IAM.

Privacidad desde el diseño

Enfoque de ingeniería de sistemas que tiene en cuenta la privacidad durante todo el proceso de desarrollo.

zonas alojadas privadas

Un contenedor que contiene información sobre cómo desea que Amazon Route 53 responda a las consultas de DNS de un dominio y sus subdominios dentro de uno o más VPCs. Para obtener más información, consulte Uso de zonas alojadas privadas en la documentación de Route 53.

control proactivo

Control de seguridad que se diseñó para evitar la implementación de recursos que no cumplan con la normativa. Estos controles analizan los recursos antes de aprovisionarlos. Si el recurso no cumple con los requisitos del control, no se aprovisiona. Para obtener más información, consulte la guía de referencia de controles en la AWS Control Tower documentación y consulte Controles proactivos en la sección Implementación de controles de seguridad en AWS.

administración del ciclo de vida del producto (PLM)

Administración de los datos y los procesos de un producto a lo largo de todo su ciclo de vida, desde el diseño, el desarrollo y el lanzamiento, pasando por el crecimiento y la madurez, hasta la reducción de su uso y su retirada.

entorno de producción

Consulte entorno.

controlador lógico programable (PLC)

En el sector de la fabricación, computadora adaptable y altamente fiable que supervisa las máquinas y automatiza los procesos de fabricación.

encadenamiento de peticiones

Uso de la salida de una petición de LLM como entrada para la siguiente petición a fin de generar mejores respuestas. Esta técnica se utiliza para dividir una tarea compleja en tareas secundarias o para refinar o ampliar de forma iterativa una respuesta preliminar. Ayuda a mejorar la precisión y la relevancia de las respuestas de un modelo y permite obtener resultados más detallados y personalizados.

seudonimización

El proceso de reemplazar los identificadores personales de un conjunto de datos por valores de marcadores de posición. La seudonimización puede ayudar a proteger la privacidad personal. Los datos seudonimizados siguen considerándose datos personales.

publish/subscribe (pub/sub)

Patrón que permite establecer comunicaciones asíncronas entre microservicios para mejorar la escalabilidad y la capacidad de respuesta. Por ejemplo, en un MES basado en microservicios, un microservicio puede publicar mensajes de eventos en un canal al que se pueden suscribir otros microservicios. El sistema puede agregar nuevos microservicios sin cambiar el servicio de publicación.

Q

plan de consulta

Serie de pasos, como instrucciones, que se utilizan para acceder a los datos de un sistema de base de datos relacional SQL.

regresión del plan de consulta

El optimizador de servicios de la base de datos elige un plan menos óptimo que antes de un cambio determinado en el entorno de la base de datos. Los cambios en estadísticas, restricciones, configuración del entorno, enlaces de parámetros de consultas y actualizaciones del motor de base de datos PostgreSQL pueden provocar una regresión del plan.

R

Matriz RACI

Consulte responsable, fiable, consultada e informada (RACI).

RAG

Consulte generación aumentada por recuperación.

ransomware

Software malicioso que se ha diseñado para bloquear el acceso a un sistema informático o a los datos hasta que se efectúe un pago.

Matriz RASCI

Consulte responsable, fiable, consultada e informada (RACI).

RCAC

Consulte control de acceso por filas y columnas.

réplica de lectura

Una copia de una base de datos que se utiliza con fines de solo lectura. Puede enrutar las consultas a la réplica de lectura para reducir la carga en la base de datos principal.

rediseñar

Consulte Las 7 R.

objetivo de punto de recuperación (RPO)

La cantidad de tiempo máximo aceptable desde el último punto de recuperación de datos. Esto determina qué se considera una pérdida de datos aceptable entre el último punto de recuperación y la interrupción del servicio.

objetivo de tiempo de recuperación (RTO)

La demora máxima aceptable entre la interrupción del servicio y el restablecimiento del servicio.

refactorizar

Consulte Las 7 R.

Region

Conjunto de AWS recursos en un área geográfica. Cada uno Región de AWS está aislado e independiente de los demás para proporcionar tolerancia a las fallas, estabilidad y resiliencia. Para más información, consulte Specify which Regiones de AWS your account can use.

regresión

Una técnica de ML que predice un valor numérico. Por ejemplo, para resolver el problema de "¿A qué precio se venderá esta casa?", un modelo de ML podría utilizar un modelo de regresión lineal para predecir el precio de venta de una vivienda en función de datos conocidos sobre ella (por ejemplo, los metros cuadrados).

volver a alojar

Consulte Las 7 R.

versión

En un proceso de implementación, el acto de promover cambios en un entorno de producción.

reubicar

Consulte Las 7 R.

redefinir la plataforma

Consulte Las 7 R.

recomprar

Consulte Las 7 R.

resiliencia

Capacidad de una aplicación para resistir interrupciones o recuperarse de ellas. Al planificar la resiliencia en la Nube de AWS, la alta disponibilidad y la recuperación ante desastres son consideraciones comunes. Para más información, consulte Resiliencia en la Nube de AWS.

política basada en recursos

Una política asociada a un recurso, como un bucket de Amazon S3, un punto de conexión o una clave de cifrado. Este tipo de política especifica a qué entidades principales se les permite el acceso, las acciones compatibles y cualquier otra condición que deba cumplirse.

matriz responsable, confiable, consultada e informada (RACI)

Una matriz que define las funciones y responsabilidades de todas las partes involucradas en las actividades de migración y las operaciones de la nube. El nombre de la matriz se deriva de los tipos de responsabilidad definidos en la matriz: responsable (R), contable (A), consultado (C) e informado (I). El tipo de soporte (S) es opcional. Si incluye el soporte, la matriz se denomina matriz RASCI y, si la excluye, se denomina matriz RACI.

control receptivo

Un control de seguridad que se ha diseñado para corregir los eventos adversos o las desviaciones con respecto a su base de seguridad. Para obtener más información, consulte Controles receptivos en Implementación de controles de seguridad en AWS.

retain

Consulte Las 7 R.

retirar

Consulte Las 7 R.

Generación aumentada de recuperación (RAG)

Tecnología de IA generativa mediante la que un LLM hace referencia a un origen de datos autorizado que se encuentra fuera de sus orígenes de datos de entrenamiento antes de generar una respuesta. Por ejemplo, un modelo de RAG podría hacer una búsqueda semántica en la base de conocimientos o en los datos personalizados de una organización. Para más información, consulte ¿Qué es RAG (generación aumentada por recuperación)?

rotación

Proceso mediante el que periódicamente se actualiza un secreto para que resulte más difícil que un atacante pueda acceder a las credenciales.

control de acceso por filas y columnas (RCAC)

El uso de expresiones SQL básicas y flexibles que tienen reglas de acceso definidas. El RCAC consta de permisos de fila y máscaras de columnas.

RPO

Consulte objetivo de punto de recuperación.

RTO

Consulte objetivo de tiempo de recuperación.

manual de procedimientos

Conjunto de procedimientos manuales o automatizados necesarios para realizar una tarea específica. Por lo general, se diseñan para agilizar las operaciones o los procedimientos repetitivos con altas tasas de error.

S

SAML 2.0

Un estándar abierto que utilizan muchos proveedores de identidad (IdPs). Esta función permite el inicio de sesión único (SSO) federado, de modo que los usuarios pueden iniciar sesión Consola de administración de AWS o llamar a las operaciones de la AWS API sin tener que crear un

usuario en IAM para todos los miembros de la organización. Para obtener más información sobre la federación basada en SAML 2.0, consulte Acerca de la federación basada en SAML 2.0 en la documentación de IAM.

SCADA

Consulte control de supervisión y adquisición de datos.

SCP

Consulte política de control de servicio.

secreta

En AWS Secrets Manager, información confidencial o restringida, como una contraseña o credenciales de usuario, que se almacena de forma cifrada. Se compone del valor del secreto y de sus metadatos. El valor del secreto puede ser binario, una sola cadena o varias cadenas. Para más información, consulte What's in a Secrets Manager secret? en la documentación de Secrets Manager.

seguridad desde el diseño

Enfoque de ingeniería de sistemas que tiene en cuenta la seguridad durante todo el proceso de desarrollo.

control de seguridad

Barrera de protección técnica o administrativa que impide, detecta o reduce la capacidad de un agente de amenazas para aprovechar una vulnerabilidad de seguridad. Existen cuatro tipos de controles de seguridad principales: preventivos, de detección, de respuesta y proactivos.

refuerzo de la seguridad

Proceso de reducir la superficie expuesta a ataques para hacerla más resistente a los ataques. Esto puede incluir acciones, como la eliminación de los recursos que ya no se necesitan, la implementación de prácticas recomendadas de seguridad consistente en conceder privilegios mínimos o la desactivación de características innecesarias en los archivos de configuración.

sistema de información sobre seguridad y administración de eventos (SIEM)

Herramientas y servicios que combinan sistemas de administración de información sobre seguridad (SIM) y de administración de eventos de seguridad (SEM). Un sistema de SIEM recopila, monitorea y analiza los datos de servidores, redes, dispositivos y otras fuentes para detectar amenazas y brechas de seguridad y generar alertas.

automatización de la respuesta de seguridad

Acción predefinida y programada que está diseñada para responder automáticamente a un evento de seguridad o corregirlo. Estas automatizaciones sirven como controles de seguridad preventivos o adaptables que le ayudan a implementar las mejores prácticas AWS de seguridad. La modificación de un grupo de seguridad de VPC, la aplicación de revisiones a una instancia de Amazon EC2 o la rotación de credenciales son algunos ejemplos de acciones de respuesta automatizadas.

cifrado del servidor

Cifrado de los datos en su destino, por parte de Servicio de AWS quien los recibe.

política de control de servicio (SCP)

Política que proporciona un control centralizado de los permisos de todas las cuentas de una organización en AWS Organizations. SCPs defina barreras o establezca límites a las acciones que un administrador puede delegar en usuarios o roles. Puede utilizarlas SCPs como listas de permitidos o rechazados para especificar qué servicios o acciones están permitidos o prohibidos. Para obtener más información, consulte las políticas de control de servicios en la AWS Organizations documentación.

punto de enlace de servicio

La URL del punto de entrada de un Servicio de AWS. Para conectarse mediante programación a un servicio de destino, puede utilizar un punto de conexión. Para obtener más información, consulte Puntos de conexión de Servicio de AWS en Referencia general de AWS.

acuerdo de nivel de servicio (SLA)

Acuerdo que aclara lo que un equipo de TI se compromete a ofrecer a los clientes, como el tiempo de actividad y el rendimiento del servicio.

indicador de nivel de servicio (SLI)

Medición de un aspecto del rendimiento de un servicio, como la tasa de errores, la disponibilidad o el rendimiento.

objetivo de nivel de servicio (SLO)

Métrica objetivo que representa el estado de un servicio medido mediante un indicador de nivel de servicio.

modelo de responsabilidad compartida

Un modelo que describe la responsabilidad con AWS la que compartes la seguridad y el cumplimiento de la nube. AWS es responsable de la seguridad de la nube, mientras que usted es responsable de la seguridad en la nube. Para obtener más información, consulte el Modelo de responsabilidad compartida.

SIEM

Consulte sistema de administración de eventos e información de seguridad.

único punto de error (SPOF)

Error en un único componente crítico de una aplicación que puede interrumpir el sistema.

SLA

Consulte acuerdo de nivel de servicio.

SLI

Consulte indicador de nivel de servicio.

SLO

Consulte objetivo de nivel de servicio.

split-and-seed modelo

Un patrón para escalar y acelerar los proyectos de modernización. A medida que se definen las nuevas funciones y los lanzamientos de los productos, el equipo principal se divide para crear nuevos equipos de productos. Esto ayuda a ampliar las capacidades y los servicios de su organización, mejora la productividad de los desarrolladores y apoya la innovación rápida. Para más información, consulte Phased approach to modernizing applications in the Nube de AWS.

SPOF

Consulte único punto de error.

esquema en estrella

Estructura organizativa de una base de datos que utiliza una tabla de hechos de gran tamaño para almacenar datos transaccionales o medidos y una o varias tablas dimensionales más pequeñas para almacenar los atributos de los datos. Esta estructura está diseñada para utilizarse en un almacén de datos o con fines de inteligencia empresarial.

patrón de higo estrangulador

Un enfoque para modernizar los sistemas monolíticos mediante la reescritura y el reemplazo gradual de las funciones del sistema hasta que se pueda desmantelar el sistema heredado. Este patrón utiliza la analogía de una higuera que crece hasta convertirse en un árbol estable y, finalmente, se apodera y reemplaza a su host. El patrón fue presentado por Martin Fowler como una forma de gestionar el riesgo al reescribir sistemas monolíticos. Para ver un ejemplo con la aplicación de este patrón, consulte Modernización gradual de los servicios web antiguos de Microsoft ASP.NET (ASMX) mediante contenedores y Amazon API Gateway.

subred

Un intervalo de direcciones IP en la VPC. Una subred debe residir en una sola zona de disponibilidad.

control de supervisión y adquisición de datos (SCADA)

En el sector de la fabricación, sistema que utiliza hardware y software para supervisar los activos físicos y las operaciones de producción.

cifrado simétrico

Un algoritmo de cifrado que utiliza la misma clave para cifrar y descifrar los datos.

pruebas sintéticas

Prueba de un sistema de manera que simule las interacciones de los usuarios para detectar posibles problemas o supervisar el rendimiento. Puede usar Amazon CloudWatch Synthetics para crear estas pruebas.

petición del sistema

Técnica para proporcionar contexto, instrucciones o pautas a un LLM para dirigir su comportamiento. Las peticiones del sistema ayudan a establecer el contexto y las reglas para las interacciones con los usuarios.

T

etiquetas

Pares clave-valor que actúan como metadatos para organizar los recursos. AWS Las etiquetas pueden ayudar a administrar, identificar, organizar, buscar y filtrar recursos de . Para obtener más información, consulte Etiquetado de los recursos de AWS.

variable de destino

El valor que intenta predecir en el ML supervisado. Esto también se conoce como variable de resultado. Por ejemplo, en un entorno de fabricación, la variable objetivo podría ser un defecto del producto.

lista de tareas

Herramienta que se utiliza para hacer un seguimiento del progreso mediante un manual de procedimientos. La lista de tareas contiene una descripción general del manual de procedimientos y una lista de las tareas generales que deben completarse. Para cada tarea general, se incluye la cantidad estimada de tiempo necesario, el propietario y el progreso.

entorno de prueba

Consulte entorno.

entrenamiento

Proporcionar datos de los que pueda aprender su modelo de ML. Los datos de entrenamiento deben contener la respuesta correcta. El algoritmo de aprendizaje encuentra patrones en los datos de entrenamiento que asignan los atributos de los datos de entrada al destino (la respuesta que desea predecir). Genera un modelo de ML que captura estos patrones. Luego, el modelo de ML se puede utilizar para obtener predicciones sobre datos nuevos para los que no se conoce el destino.

puerta de enlace de tránsito

Un centro de tránsito de red que puede usar para interconectar sus redes con VPCs las locales. Para obtener más información, consulte Qué es una pasarela de tránsito en la AWS Transit Gateway documentación.

flujo de trabajo basado en enlaces troncales

Un enfoque en el que los desarrolladores crean y prueban características de forma local en una rama de característica y, a continuación, combinan esos cambios en la rama principal. Luego, la rama principal se adapta a los entornos de desarrollo, preproducción y producción, de forma secuencial.

acceso de confianza

Otorgar permisos a un servicio que especifique para realizar tareas en su organización AWS Organizations y en sus cuentas en su nombre. El servicio de confianza crea un rol vinculado al servicio en cada cuenta, cuando ese rol es necesario, para realizar las tareas de administración

por usted. Para obtener más información, consulte AWS Organizations Utilización con otros AWS servicios en la AWS Organizations documentación.

ajuste

Cambiar aspectos de su proceso de formación a fin de mejorar la precisión del modelo de ML. Por ejemplo, puede entrenar el modelo de ML al generar un conjunto de etiquetas, incorporar etiquetas y, luego, repetir estos pasos varias veces con diferentes ajustes para optimizar el modelo.

equipo de dos pizzas

Un DevOps equipo pequeño al que puedes alimentar con dos pizzas. Un equipo formado por dos integrantes garantiza la mejor oportunidad posible de colaboración en el desarrollo de software.

U

incertidumbre

Un concepto que hace referencia a información imprecisa, incompleta o desconocida que puede socavar la fiabilidad de los modelos predictivos de ML. Hay dos tipos de incertidumbre: la incertidumbre epistémica se debe a datos limitados e incompletos, mientras que la incertidumbre aleatoria se debe al ruido y la aleatoriedad inherentes a los datos. Para más información, consulte la guía Cuantificación de la incertidumbre en los sistemas de aprendizaje profundo.

tareas indiferenciadas

También conocido como tareas arduas, es el trabajo que es necesario para crear y operar una aplicación, pero que no proporciona un valor directo al usuario final ni proporciona una ventaja competitiva. Algunos ejemplos de tareas indiferenciadas son la adquisición, el mantenimiento y la planificación de la capacidad.

entornos superiores

Consulte entorno.

V

succión

Una operación de mantenimiento de bases de datos que implica limpiar después de las actualizaciones incrementales para recuperar espacio de almacenamiento y mejorar el rendimiento.

control de versión

Procesos y herramientas que realizan un seguimiento de los cambios, como los cambios en el código fuente de un repositorio.

Emparejamiento de VPC

Una conexión entre dos VPCs que le permite enrutar el tráfico mediante direcciones IP privadas. Para obtener más información, consulte ¿Qué es una interconexión de VPC? en la documentación de Amazon VPC.

vulnerabilidad

Defecto de software o hardware que pone en peligro la seguridad del sistema.

W

caché caliente

Un búfer caché que contiene datos actuales y relevantes a los que se accede con frecuencia. La instancia de base de datos puede leer desde la caché del búfer, lo que es más rápido que leer desde la memoria principal o el disco.

datos templados

Datos a los que el acceso es infrecuente. Al consultar este tipo de datos, normalmente se aceptan consultas moderadamente lentas.

función de ventana

Función SQL que hace un cálculo en un grupo de filas que se relacionan de alguna manera con el registro actual. Las funciones de ventana son útiles para las tareas de procesamiento, como calcular una media móvil o acceder al valor de las filas en función de la posición relativa de la fila actual.

carga de trabajo

Conjunto de recursos y código que ofrece valor comercial, como una aplicación orientada al cliente o un proceso de backend.

flujo de trabajo

Grupos funcionales de un proyecto de migración que son responsables de un conjunto específico de tareas. Cada flujo de trabajo es independiente, pero respalda a los demás flujos de trabajo del proyecto. Por ejemplo, el flujo de trabajo de la cartera es responsable de priorizar las aplicaciones, planificar las oleadas y recopilar los metadatos de migración. El flujo de trabajo de la cartera entrega estos recursos al flujo de trabajo de migración, que luego migra los servidores y las aplicaciones.

WORM

Consulte escritura única y lectura múltiple.

WQF

Consulte AWS Workload Qualification Framework.

escritura única y lectura múltiple (WORM)

Modelo de almacenamiento que escribe los datos una sola vez y evita que se eliminen o modifiquen. Los usuarios autorizados pueden leer los datos tantas veces como sea necesario, pero no los pueden cambiar. Esta infraestructura de almacenamiento de datos se considera inmutable.

Z

ataque de día cero

Ataque, normalmente de malware, que se aprovecha de una vulnerabilidad de día cero.

vulnerabilidad de día cero

Un defecto o una vulnerabilidad sin mitigación en un sistema de producción. Los agentes de amenazas pueden usar este tipo de vulnerabilidad para atacar el sistema. Los desarrolladores suelen darse cuenta de la vulnerabilidad a raíz del ataque.

peticiones desde cero

Proporcionar a un LLM instrucciones para llevar a cabo una tarea, pero sin ejemplos (pasos) que puedan ayudar a guiarlo. El LLM debe usar los conocimientos del entrenamiento previo para

llevar a cabo la tarea. La eficacia de la petición desde cero depende de la complejidad de la tarea y de la calidad de la petición. Consulte también peticiones con pocos pasos.

aplicación zombi

Aplicación que utiliza un promedio de CPU y memoria menor al 5 por ciento. En un proyecto de migración, es habitual retirar estas aplicaciones.

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la version original de inglés, prevalecerá la version en inglés.